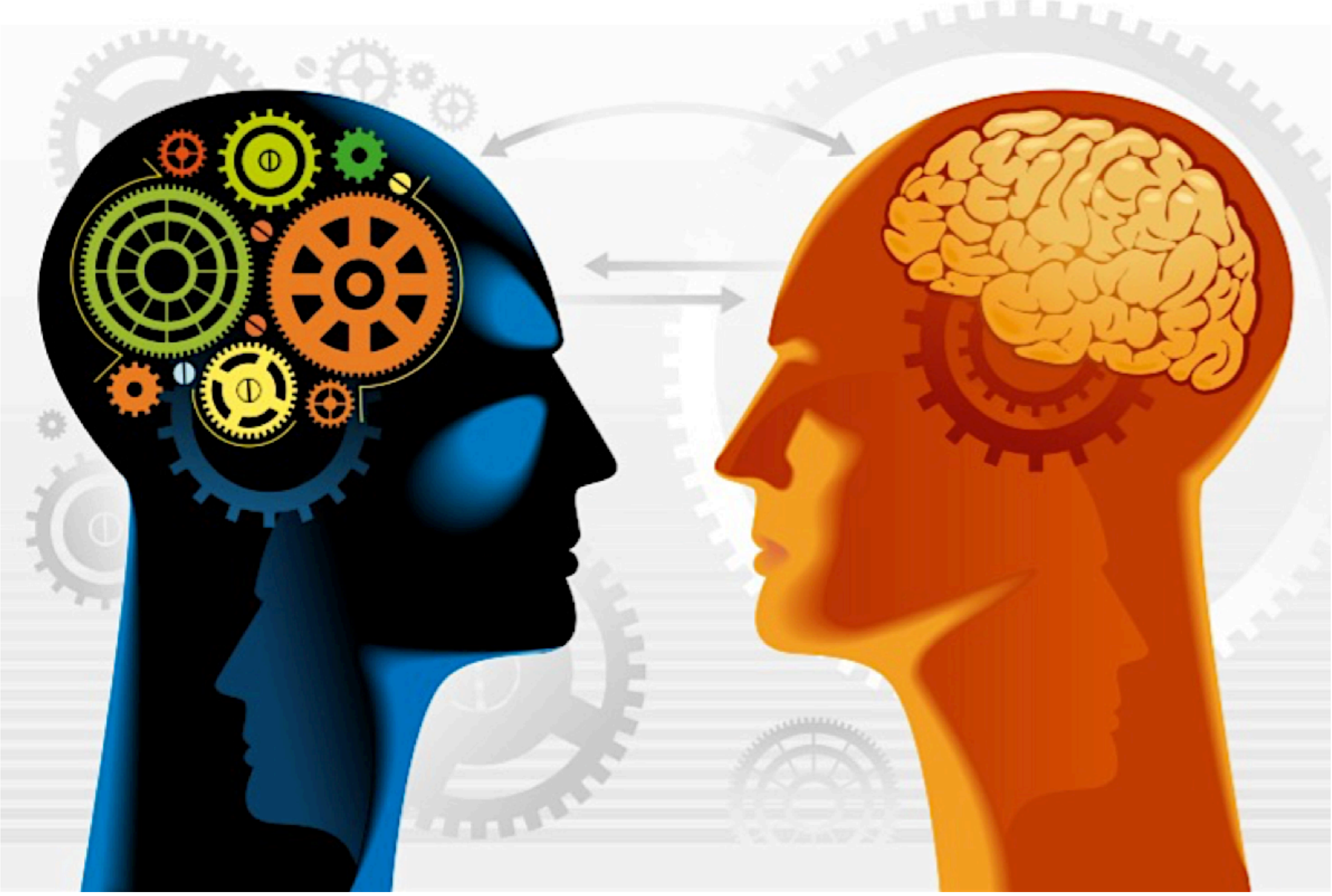
ENHANCING ANALYST FORECAST ACCURACY WITH AUTOMATED TEXTUAL ANALYSIS

Authors:
Mette Louise Duus Kühnel (81538)
Mathias Lund Nielsen (41105)

Supervisors:
Thomas Plenborg &
Thomas Riise Johansen

May 15, 2018
Characters: 217,528
Pages: 119

Master's Thesis
MSc (Finance and Accounting)
Copenhagen Business School



Abstract

This study sets out to examine to what extent models, based on automated textual analysis of the content of 10-K and 10-Q filings, can be used to enhance the accuracy of analysts' earnings per share forecasts. The past decades have witnessed significant increases in computational power and an explosion of digitally available text. Researchers within the fields of accounting and finance have exploited these developments in attempts to predict events, such as bankruptcies and fraud, by using various textual sources, such as corporate disclosures and online news. However, the research of analysts' forecasts seems to have focused more on behavioral aspects, e.g. analyst biases and the abilities of analysts to incorporate information, than on testing whether the significant increase in available information and the development in tools for utilizing such information, can be applied to enhance the forecast accuracy of analysts.

By applying DataRobot, a state of the art machine learning platform, and the textual content of 10-K and 10-Q filings, directional models for each of the six included sector-specific subsamples are constructed. These models are set to predict whether consensus will over- or underestimate EPS in the following quarter, and they are built on data containing the textual content and submission dates of such filings of S&P 500 companies from years 2012-2017, as well as corresponding historic earnings surprise data.

It is found that analysts, who are not data scientists, are able to significantly enhance the accuracy of their earnings per share forecasts by implementing automated textual analysis as suggested in this study. The findings indicate that 10-K and 10-Q filings contain information that analysts fail to fully incorporate in their forecasts, and the suggested tool is able to identify patterns in regards to such forecasting difficulties. Thus, besides providing analysts with predictions in regards to the directional shifts, they should make, in order to enhance their forecast accuracy, the output of the models can be beneficial to analysts by possibly identifying information that they seem to fail to comprehend. These findings are concluded to be robust to differences in market capitalization, analyst coverage, document length and the number of quarters used for training data in the modeling. Lastly, through conducting an extensive literature review and through analyses of model performance, the findings illustrate that no one type of model is consistently superior across or within different predictive tasks.

Table of Content

1 Introduction and Motivation

Analysts' forecasts have been the subject of research for many years. As early as the 1980s, Hassell (1988) established a significant relationship between analyst forecasts and the information disclosed by corporations. However, due to the limited data available and the technical abilities at the time, Hassell's (1988) research was based on a sample consisting of only 120 observations of management's forecasts of earnings. In the recent decades, computational power has increased significantly, and an explosion of text that is digitally available has been witnessed (Ittoo et al., 2015; Loughran and McDonald, 2016). Many researchers have attempted to utilize the information available in corporate disclosures in order to predict events such as fraud and bankruptcies. Similarly focusing on the content of corporate disclosures, researchers have examined the relationship between such content and analyst behavior. Li (2010a) emphasizes that the textual content in corporate disclosures can be a way for analysts to understand firm behavior. Lehavy et al. (2011) find that analyst forecast accuracy declines and that analyst earnings forecast dispersion increases when the readability of corporate disclosures decreases. The results of Lobo et al. (1998) indicate that the implementation of legislation requiring an increase in company information disclosure did in fact result in a considerable increase in analyst forecast accuracy. Such findings cause the authors to argue that analysts are able to incorporate much of the information made available to market participants. On the other hand, Clement (1999) finds that analysts are constrained by factors such as limited resources and portfolio complexity. Thus, the author argues that not all of the information available is fully grasped by analysts.

While researchers have been and are exploiting the increase in available textual data and the development in technology, practitioners are just starting to recognize the potential economic value that lies in such sources (Ittoo et al., 2015). As they will most likely not, like some researchers, be data science specialists, ease of use is a criterion that textual analysis has to fulfill in order to be successfully implemented (Ittoo et al., 2015). "*Thus, they should be presented with applications that are simple and easy to use, without requiring them to tamper with the applications' internal mechanics*" (Ibid., p. 105). Market leading financial institutions, such as Goldman Sachs Asset

Management (2016) and Deutsche Bank (2018), predict that automated textual analysis will be a critical tool for tomorrow's investors, as it will enable them to use non-numerical, language-based data to gauge how management is thinking about the future of a company. However, as argued by Chan and Franklin (2011, p. 189), the current level of available information has "*created a pressing need for better knowledge discovery and the construction of applications for managing the knowledge that is extracted*". Several of these applications, such as DataRobot, are available to both researchers and companies, and it is argued that analysts (and anyone else for that matter) no longer need to master programming skills to apply them. However, as the literature suggests that practical exploitation is yet to happen, it could be examined whether practitioners could benefit from applying such applications on available textual sources, and furthermore, whether exploitation of such requires extensive data science knowledge. This leads to the research question framing this study:

To what extent can models based on automated textual analysis of the content in form 10-K and form 10-Q filings be used to enhance the accuracy of analysts' earnings per share forecasts?

The aim is to provide predictions that can be used by analysts as a guide to the direction in which their estimates should be shifted in order to enhance their accuracy. In order to answer the research question, directional models, based on the textual content of 10-K and 10-Q filings of S&P 500 companies from 2012-2017 and past earnings surprise data, are built in DataRobot. This is illustrated in Figure 1.1. By doing so, it is tested whether analysts, by using automated textual analysis, are able to determine the direction in which EPS estimates should be shifted in order to enhance forecast accuracy.

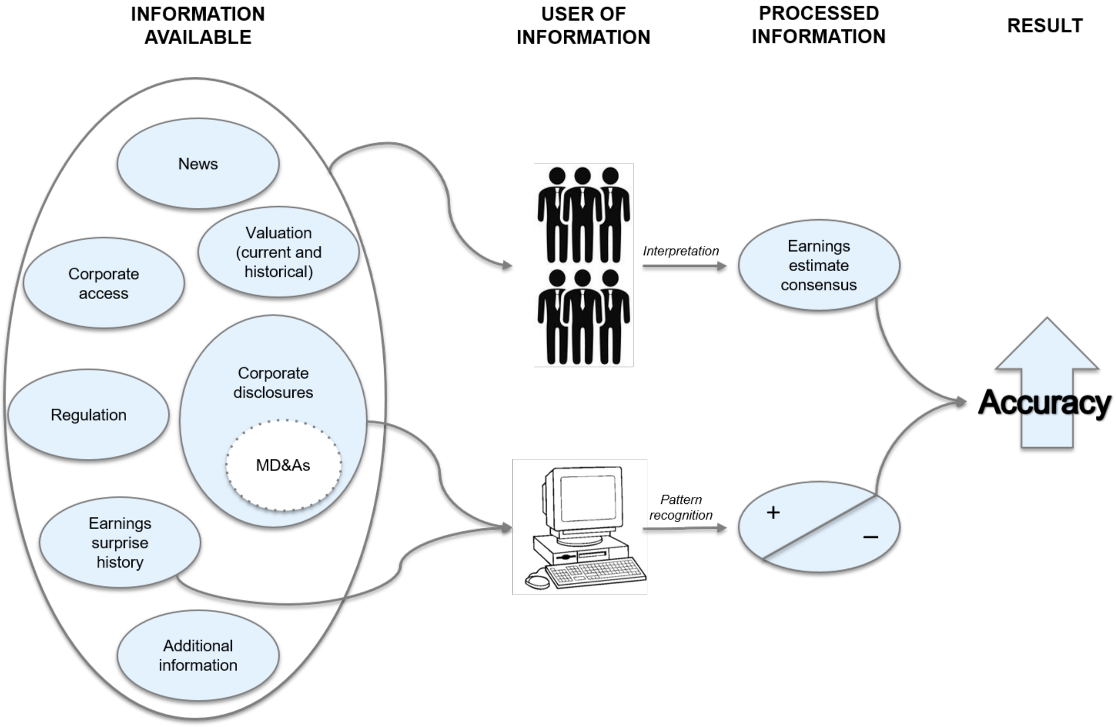


Figure 1.1: Research visualization

Earnings surprise data is used as a proxy for over- and underestimations of EPS, and a variable, indicating whether consensus was above or below actual EPS, is used as the target variable in the modeling. The textual content of the filings and the date of submission are the only variables used as input for the models. Each model construction[1] will be based on sector-specific subsamples[2], and the constructed models will be evaluated and compared based on various metrics such as Logarithmic Loss (Log Loss) and Area Under the Curve (AUC). By analyzing the output of the models, it is attempted to identify whether certain textual characteristics, such as the use of specific words, are associated with analysts' overestimation or underestimation of EPS. Furthermore, it is examined whether such pattern recognition can lead to enhancements of analysts' forecast accuracy.

The study will contribute to the existing literature by examining whether analysts are able to increase the accuracy of their EPS forecasts by applying automated textual analysis on the textual content of corporate filings. Furthermore, the study contributes by analyzing whether there are

[1] A model construction refers to the process of constructing several models based on a single data sample uploaded in DataRobot

[2] Companies are divided into 11 sectors according to The Global Industry Classification Standards (GICS)

certain concepts in the filings that are indicative of analysts' overestimation (underestimation) of EPS in the following quarter and whether analysts should be particularly aware of certain characteristics when applying the tool. It is argued that such knowledge is valuable to multiple stakeholders, such as researchers, analysts, and investors.

The remainder of this paper is organized as follows. Below is a description of the scope of this study. Section 2 provides a review of relevant literature within the fields of machine learning, accounting, and finance. Section 3 provides descriptions of the methodologies applied for data acquisition, data preparation, and modeling, as well as descriptions of the analysis framework and the methodology applied for robustness tests. In section 4, results are presented and analyzed in sector-specific sections. The sector-specific findings are subsequently applied in an overall analysis identifying general trends across sectors, prediction targets, and model types. Furthermore, this section includes tests of the robustness of the findings. Section 6 presents a discussion addressing findings in relation to previous research, limitations, implications, and suggestions for further research, and in section 7, the conclusion and contributions of this study will be presented.

1.1 Scope

US listed companies in the S&P 500 index from 2012 to 2017 are examined. Three main reasons lead to the choice of using S&P 500 companies: 1) All reports of the companies are in English, and this is the language for which textual analysis tools are by far most developed at the time of conducting this study, 2) All of the companies are subject to the same accounting framework (US GAAP) and regulatory environment, and 3) The constituents of the index represent approximately 80% of available market capitalization in the US (Standard & Poor's, 2018) and thus, it is argued that these companies are the ones of highest interest to analysts and investors. The scope is limited to six years due to the sample size cap of 500 MB for academic licenses in DataRobot. As noted, sector-specific models will be constructed. As certain sector subsamples (Energy, Materials, Real Estate, Telecommunication Services, and Utilities) consisted of a limited number of observations (less than 30 observations in one or more quarters), they were excluded. Thus, model constructions were performed for the six remaining sector subsamples: Consumer Discretionary, Consumer Staples, Financials, Health Care, Industrials, and Information Technology.

2 Literature Review and Background

The purposes of this literature review are to map relevant previous literature and to provide the reader with an overview of key focus areas. After reading the following review, the reader will have been introduced to the different aspects of analyst forecast accuracy that have been focused on by researchers. Furthermore, the reader will have obtained knowledge in regards to what events within finance and accounting, researchers have attempted to predict, and the models that are commonly used. Providing such a review will enable an assessment of how this paper contributes to the existing literature. Furthermore, it will provide a foundation for a discussion of the findings that are presented.

2.1 Methodology

By using a systematic, seven step-methodology, similar to the one applied by Amani and Fadlalla (2017), a limited number of papers that together constitute a representative sample of the relevant literature are identified, and the key findings of the literature are mapped. Where it is deemed appropriate, the approach of Amani and Fadlalla (2017) has been altered to better support the objective of this study. In such cases, it will be explicitly addressed. The applied methodology is illustrated in Figure 2.1.

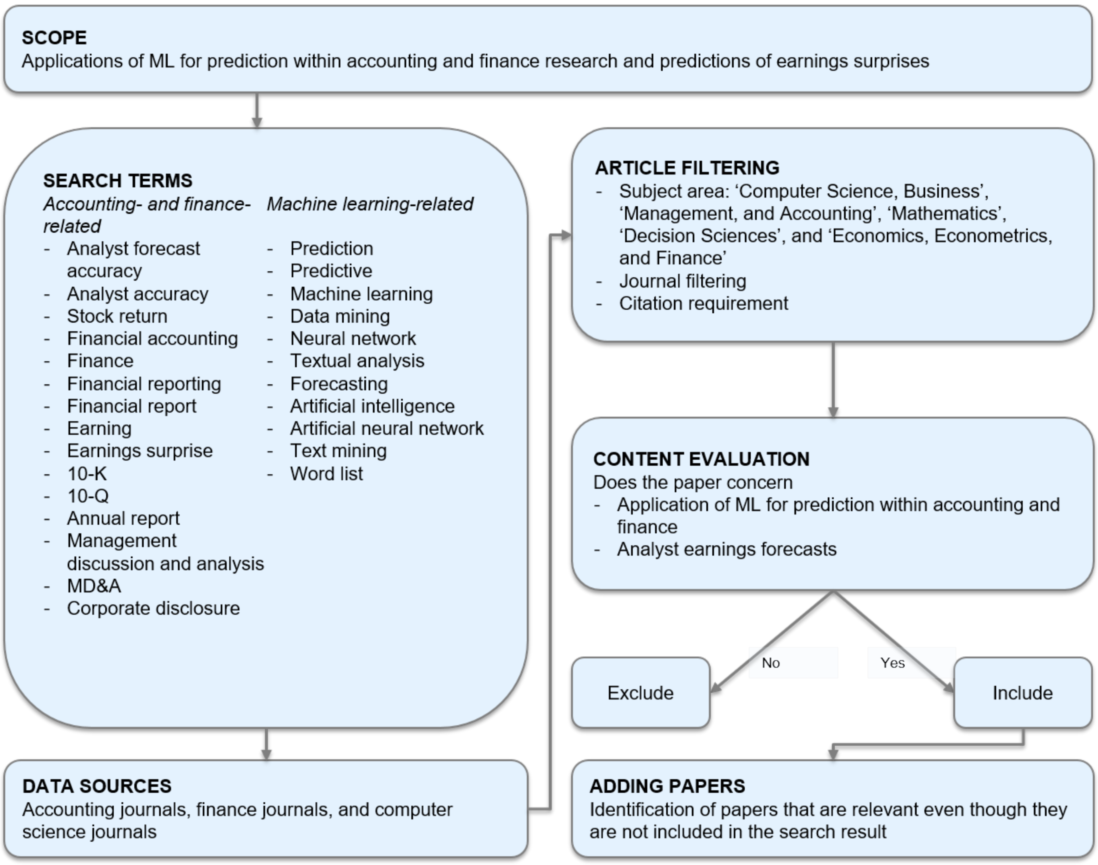

SCOPE
Applications of ML for prediction within accounting and finance research and predictions of earnings surprises
SEARCH TERMS
Accounting- and finance-related
- Analyst forecast accuracy
- Analyst accuracy
- Stock return
- Financial accounting
- Finance
- Financial reporting
- Financial report
- Earning
- Earnings surprise
- 10-K
- 10-Q
- Annual report
- Management discussion and analysis
- MD&A
- Corporate disclosure
Machine learning-related
- Prediction
- Predictive
- Machine learning
- Data mining
- Neural network
- Textual analysis
- Forecasting
- Artificial intelligence
- Artificial neural network
- Text mining
- Word list
ARTICLE FILTERING
- Subject area: 'Computer Science, Business', 'Management, and Accounting', 'Mathematics', 'Decision Sciences', and 'Economics, Econometrics, and Finance'
- Journal filtering
- Citation requirement
CONTENT EVALUATION
Does the paper concern
- Application of ML for prediction within accounting and finance
- Analyst earnings forecasts
No
Yes
Exclude
Include
DATA SOURCES
Accounting journals, finance journals, and computer science journals
ADDING PAPERS
Identification of papers that are relevant even though they are not included in the search result

Figure 2.1: Methodology for literature selection

Step 1: Scoping of the literature review. The focus of the literature review is on applications of machine learning for predictions within the fields of accounting and finance and papers that concern analysts' forecasts or the accuracy thereof (disregarding whether machine learning is used or not).

Step 2: Identification of search items. The papers included in this review were found using Scopus, which is an abstract and citation database covering more than 69 million records from more than 5,000 international publishers (Elsevier, 2017). Keywords that would capture the relevant literature when conducting a Scopus-search were identified, and as the scope was to include literature in the cross section of accounting and finance and machine learning, two lists of keywords, representing each of the fields (one of the lists combining finance and accounting keywords), were constructed.

The keywords are listed in Figure 2.1 above[3]. The title, abstract, or keywords of each paper had to contain one or more words from both lists. It is argued that relevant papers would include a combination of words from each list, and that such search criteria would yield a reasonably broad set of papers.

Step 3: Identification of data sources. To ensure the relation to the subject, the search was limited to papers published in accounting journals, finance journals, and computer science journals. The quality of the chosen journals was not further assessed, which poses the risk of inclusion of low quality papers. However, it is argued that such risk was mitigated by the subsequent article filtering (see Step 5) and by the subjective selection of papers (see Step 6).

Step 4: Article collection. Based on the selected keywords, a search in the keywords, abstracts, and titles of the papers was conducted on Scopus. This search resulted in 12,410 papers[4].

Step 5: Article filtering. To limit the inclusion of papers, all papers that were not assigned one of the following subject areas were excluded: 'Computer Science', 'Business, Management, and Accounting', 'Mathematics', 'Decision Sciences', and 'Economics, Econometrics, and Finance'. This resulted in a reduction from 12,410 papers to 9,142.

Furthermore, only papers published in journals were included. This reduced the number of papers to 5,809. Additionally, specific journals were selected, as discussed in Step 3, and this limited the search from 2,223 papers to 2,541 papers.

Lastly, at least one citing per year was required for papers published prior to 2017. Papers published in or after 2017 that had never been cited were included no matter their number of citings, as they might be relevant to the research, even though they, due to their recent publishing, have not had a fair chance of being cited. These criteria reduced the number of papers from 2,541 to 1,482.

[3] 'Accounting' was excluded as keyword, as conducting a search for 'Accounting' in abstracts resulted in too many irrelevant papers due to the proverb-version of the word

[4] The final search string is available in 8

Step 6: Content evaluation. All papers fulfilling the criteria above were reviewed, and papers not deemed relevant based on their titles and abstracts were excluded. In cases where relevance could not be determined based on title or abstract, the introduction and the conclusion of the paper were read. This subjective selection resulted in a reduction in the number of papers from 1,482 to 233.

Step 7: Adding papers. This step is an adjustment to the methodology of Amani and Fadlalla (2017). While going through the included literature, it was found that some papers, in spite of their relevance, had failed to fulfill the criteria, and these papers were subsequently included. Such papers were identified by e.g. references in included papers. This step increased the scope from 233 papers to 254 papers.

2.2 Review

In order to provide the reader with a preliminary overview of the existing literature, included papers have been mapped as visualized in Figure 2.2. The litereature review is structured accordingly.

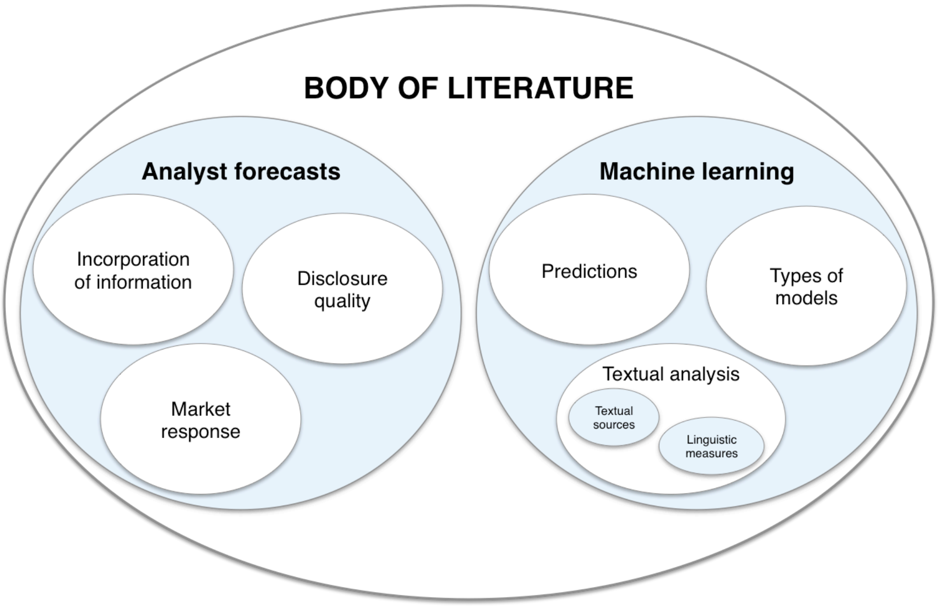


Figure 2.2: Mapping of the literature

Papers have not been limited to one section, as the subject areas are not exclusionary. In the following, the findings and focus areas of each section will be described.

2.2.1 Analyst Forecasts

The review reveals that, when examining analyst forecasts and the accuracy thereof, researchers focus on three main areas: How and if analysts incorporate different types of information, how the quality of corporate disclosures affects analyst accuracy, and how the market responds to analyst forecasts.

2.2.1.1 Incorporation of Information

Several researchers focus on analysts' ability to incorporate the informational value in management's earnings forecasts (e.g. Hassell et al., 1988; Kim and Song, 2015; Michel, 2017). The research done by Hassell et al. (1988) is motivated by the notion that managers effectively make valuable inside information available to analysts when making public announcements concerning their beliefs about future earnings. The authors argue that the way analysts use management's earnings forecasts depends on the perceived accuracy of the projections. If perceived as precise, analysts will tend to revise their estimate to align it with the forecast provided by management, whereas a projection perceived as less precise might only be included as part of the calculations done by the analyst. Hassell et al. (1988) find that analyst forecast errors decline in the four weeks following management announcements at a statistically significantly greater rate than the forecast errors of the earnings of similar firms without management projections. Furthermore, their results indicate that analysts are able to identify situations in which their own initial forecasts are more accurate than management's and thus avoid being misled. Lastly, Hassell et al. (1988) find that analysts are able to respond with restraint to situations where management's projections indicate the correct direction but are exaggerations in terms of magnitude, but not when management underestimates that magnitude. Supporting the findings of Hassell et al. (1988), Michel (2017) finds in more recent research that more than one third of analyst forecasts are equal to the minimum or maximum company-issued guidance range within a week of its issuance. Thus, his overall results indicate that analysts use the guidance range as a reference. Kim and Song (2015) similarly argue that analysts rely heavily on management forecasts when calculating their estimates. The authors even state that evidence suggests that analysts revise their forecasts within five days of the issuance of management forecasts 60% of the time and that the relative accuracy of analyst forecasts is greater for firms with management forecasts than for firms without. Moreover, Michel (2017) finds that less experienced analysts are more inclined to exceed the guidance range and issue bold forecasts than more experienced analysts, as they have no reputation to speak of. To a certain extent, such findings are contradictory to the findings of Clement and Tse (2005). They argue that less experienced analysts act with herding behavior, revising their estimates closer to consensus, as the likelihood of getting fired for bold inaccurate estimates is larger, when you have a short track-record.

Other researchers focus on analysts' abilities to comprehend specific accounting items such as tax carry forwards (Amir and Sougiannis, 1999), pension information (Picconi, 2006), and effective tax rates (Plumlee, 2003). Amir and Sougiannis (1999) argue that management providing "*accounting measurement of tax carry forwards is another way of providing a management earnings forecasts*" (Ibid., p. 1), as estimating carry forwards requires use of private information. Their results indicate that analysts consider the earnings of firms with carry forwards less persistent, and that there is a tendency of more optimism and less preciseness in analyst earnings forecasts when firms have carry forwards. Thus, their findings suggest that analysts fail to fully comprehend the implication of carry forwards on future earnings. Picconi (2006) finds that, even though analysts deal with the pension information available in 10-Ks on a repeated basis, they fail to incorporate the quantifiable earnings effects in their estimates. Plumlee (2003) examines how tax-law changes affect analysts' effective tax rate forecasts. The author finds that analysts "*assimilate less complex information to a greater extent than they assimilate more complex information*" (Ibid., p. 275). Thus, he argues that the quality of forecasts that analysts base on more complex information is lower. Plumlee (2003) argues that his findings demonstrate that complexity of information should be considered when determining the reason for inaccurate forecasts.

Ayres et al. (2017) show that firms with higher fair value intensity have more accurate analyst earnings forecasts. The authors emphasize though, that the relationship does not hold for financial companies, indicating that "*qualitative differences concerning the fair value assets themselves may be driving the real impact*" (Ibid., p. 68).

Lobo et al. (1998) find that the implementation of SFAS No. 14[5] significantly increased analyst earnings forecast accuracy, indicating that analysts are able to incorporate the additional information available to them. This is consistent with Hassell et al. (1988) who find that analysts are able to integrate additional information provided to them in management forecasts. In line with such findings, the results of Lang and Lundholm (1996) indicate that the amount of disclosure available is positively correlated with both analyst coverage and analyst forecast accuracy. Thus, their findings similarly suggest that analysts use the information available. Furthermore, Bhandari et al. (2017) find that analyst forecast accuracy increases with the magnitude of inside debt (debt-like compensation). The authors argue that this holds because inside debt is associated with increased propensity of firms to provide voluntary disclosures, thereby making more information available to analysts.

Other researchers have focused on how tone affects analyst forecasts. The findings of Hribar and McInnis (2012) suggest that analysts have relatively more optimistic forecasts for uncertain firms[6] when investor sentiment is high and relatively less optimistic forecasts for such firms when sentiment is low. Mayew and Venkatachalam (2012) measure sentiment in earnings conference calls. The authors find that analysts do not incorporate the fact that the positive and negative affects displayed by managers are informative of future performance in their near-term earnings forecasts and that analysts incorporate only positive affects when making stock recommendation changes.

Researchers have also examined how biases that are not related to the reporting company impact analysts' forecast accuracy (e.g. Clement, 1999; Clement and Tse, 2005; Lim, 2001; Mikhail et al., 1999). According to Clement (1999), there are positive relationships between analyst forecast accuracy and experience and analyst accuracy and the resources available, respectively. Furthermore, Clement (1999) finds that there is a negative relationship between analyst forecast

[5] Legislation requiring multisegment companies to disclose additional information about segments beyond the segment revenue and income, such as information in regards to identifiable assets, asset valuation allowances, capital expenditures, and effects of accounting principle changes on segment income in addition to the segment revenue and income reporting requirements under the extant SEC rule. Furthermore, the legislation changed *"the definition of a reportable segment and, in many cases, the number of segments on which information is to be disclosed"* (Lobo et al., 1998, p. 969). One of the objectives of the legislation was *"to permit better assessment of the enterprise's past performance and future prospects"*

[6] The authors define uncertain firms as small, young, and volatile, and certain firms as big, old, and smooth

accuracy and portfolio complexity. Clement and Tse (2005) find that lower accuracy for inexperienced analysts, to some extent, is led by herding behavior, as bold, inaccurate estimates significantly increase the likelihood of an analyst getting fired if the analyst has limited experience and track record. Mikhail et al. (1999) find that absolute forecast accuracy is of less importance to analysts than relative accuracy (to closest peers), as the probability of employee turnover is uncorrelated with absolute accuracy. Their findings on the other hand indicate that relative accuracy is in fact negatively correlated with employee turnover. Lim (2001) finds that analysts, on average, are positively biased, as being so leads to better management access, which is useful to analysts when forecasting earnings in uncertain information environments. Thus, it is argued that accuracy will, in some cases, be of secondary priority.

Other information related areas are addressed by e.g. Lobo and Nair (1990), who find that forecasts generated from a combination of statistical and judgmental forecasts[7] have lower errors than even the most accurate individual forecasts (either statistical and judgmental).

2.2.1.2 Disclosure Quality

While the research above focuses on analysts' ability to comprehend and integrate information, other researchers have focused on the quality and readability of corporate disclosures and on whether these characteristics affect analysts' forecasts (Behn et al., 2008; Lehavy et al., 2011).

When examining the effect, disclosure readability has on analyst forecasts, Behn et al. (2008) use audit quality as a proxy for readability, while Lehavy et al. (2011) measure readability using the Fog Index (see Section 2.2.3.2 on Linguistic Measures). Behn et al. (2008) find that analyst accuracy is higher for firms that are audited by a Big 5 auditor[8] (even after controlling for firm size). This suggests that higher reporting quality results in more accurate analyst earnings forecasts. Lehavy et al. (2011) find that less readable 10-Ks are associated with lower accuracy in analyst earnings forecasts.

[7] Defined as forecasts performed by security analysts

[8] Until 2002 The Big 4 were known as The Big 5 consisting of Deloitte, PwC, Ernst & Young, KPMG, and Arthur Andersen. The latter company was excluded after the Enron scandal (The Big 4 Accounting Firms, 2018).

2.2.1.3 Market Response

As opposed to focusing on analyst behavior, some researchers focus on how the market reacts to analyst forecasts (Kerl and Ohlert, 2015; Kim and Song, 2015; Michel, 2017).

Michel (2017) investigates investor reactions and finds that investors tend to overreact to analyst forecasts that are exactly equal to the minimum or maximum of the company-issued guidance range[9]. Similarly focusing on investor reactions to analyst forecasts, Kerl and Ohlert (2015) find that, despite the fact that star-analysts[10] outperform their peers in forecasting earnings, investors do not react differently to forecasts depending on the issuer. Thus, they argue that investors are not aware of the quality difference of analysts. Kim and Song (2015) examine stock-price reactions to analyst forecast revisions around earnings announcements, finding that price reactions to analyst forecast revisions are highly related to management forecasts.

2.2.2 Machine Learning

2.2.2.1 Predictions

Just as this study applies machine learning for predicting the direction of analyst over-/underestimations of EPS, researchers have used machine learning for the prediction of other events within the fields of finance and accounting. Among the most popular areas within the corpus of papers included in this review are stock returns (e.g. Balakrishnan et al., 2010; Bollen et al., 2011; Li et al., 2014; Zhong and Enke, 2017), financial distress and bankruptcy (e.g. Cecchini et al., 2010; Chen, 2014; du Jardin, 2015), and fraud (e.g. Cecchini et al., 2010a; Glancy and Yadav, 2011; Loughran and McDonald, 2011).

Applying automated textual analysis for the prediction of stock returns, Balakrishnan et al. (2010) exploit the narrative content of 10-K filings for market performance prediction, Li et al. (2014) utilize the sentiment of web media, and Bollen et al. (2011) use Twitter sentiment. Both Bollen et al. (2011) and O'Connor and Madden (2006) focus on predicting the direction of the Dow Jones Industrial Average (DJIA), whereas Zhong and Enke (2017) focus on forecasting the daily direction

[9] According to Michel's results, investors *"bid down (up) the prices of stocks for which analysts have provided a forecast exactly equal to the low (high) endpoint of the CIG range more so than for other comparable forecasts"* (Michel, 2017, p. 340)

[10] Star-analysts are identified based on Thomson Reuters' StarMine rankings ("Top Earnings Estimators" and/or "Top Stock Pickers") (Kerl and Ohlert, 2015, p. 98)

of the S&P 500 Index ETF return. When basing their investment decisions on the predictions of their neural network model, Bollen et al. (2011) able to earn an annual return of 23.5% (DJIA grew by 13.03%). The use of models varies among researchers (for additional information, see Section 2.2.2.2 on Types of Models). Some construct neural network models (e.g. O'Connor and Madden, 2006; Patel et al., 2015; Zhong and Enke, 2017) and some apply support vector regression (SVR) (Lu, 2013; Lu et al., 2009; Patel et al., 2015), whereas others construct ensemble models (Patel et al., 2015). Focusing on the Indian stock market, Patel et al. (2015) proposes a two-stage fusion approach in which SVR is used in the first stage and combined with artificial neural network (ANN), random forest (RF), and SVR in the second stage. They find that the ensemble models outperform the single models and that the SVR-ANN model performs best overall. Other researchers that attempt to predict stock returns include e.g. Kim (2003), Yeh et al. (2011), and Araújo (2011).

The use of models similarly varies in research predicting financial distress and bankruptcy. Examples include e.g. neural networks (Chen and Du, 2009; Huang et al., 2010; Tsai and Wu, 2008; Zhang et al., 1999) and support vector machines (SVM) (Chen, 2014; Shie et al., 2012). Some researchers compare the performance of different models. Geng et al. (2015) compare neural networks, decision trees, and an SVM, whereas Jo et al. (1997) compare neural networks to multivariate discriminant analysis and case-based forecasting. In both papers, neural networks are found to have superior performance, when predicting bankruptcies. As with stock return predictions, researchers have also applied automated textual analysis in attempts to predict financial distress and bankruptcy. Both Cecchini et al. (2010) and Mayew et al. (2015) focus on the Management Discussion and Analysis (MD&A) section. Whereas Cecchini et al. (2010) create wordlists that are indicative of bankruptcy, Mayew et al. (2015) construct two measures: one that measures management's explicit mention of the possibility that the firm may be unable to continue as a going concern, and one that measures the overall sentiment. Both studies find that the content of the MD&A is incrementally predictive of bankruptcy. Furthermore, Mayew et al. (2015) find that it has virtually the same explanatory power as financial variables. Rönnqvist and Sarlin (2017) similarly focus on textual content but instead of using MD&As, they examine news for signaling the level of bank-stress-related reporting. Both Boyacioglu et al. (2009) and De Andrés et al. (2011) focus on countries outside the US. The sample of Boyacioglu et al. (2009) consists of banks in Turkey, and De Andrés et al. (2011) attempt to predict bankruptcy for Spanish companies. Du Jardin (2017, 2015) focuses on the time horizon for bankruptcy prediction, providing models with improved prediction accuracy for a three-year horizon and a five-year horizon, respectively. Other focus areas within bankruptcy prediction include the use of information regarding the company's directors and managers (Tobback et al., 2017), the incorporation of explicit bankruptcy domain knowledge (du Jardin, 2016), and focus on efficiency as a proxy for poor management (Xu and Wang, 2009).

Within fraud prediction, many researchers apply textual analysis (e.g. Cecchini et al., 2010; Glancy and Yadav, 2011; Humpherys et al., 2011; Loughran and McDonald, 2011a). Focusing on 10-K filings, Loughran and McDonald (2011a) find that certain phrases are red flags indicating questionable company behavior. Hajek and Henriques (2017) similarly use the textual content in 10-Ks, but they combine it with financial variables, finding that both can be useful in detecting non-fraudulent firms, but that non-annual report data (such as analysts' forecasts of earnings and revenues) are necessary to detect fraudulent firms. Both Cecchini et al. (2010) and Glancy and Yadav (2011) use MD&As as the textual source for their research, and both use machine learning methods to construct lists of words that are indicative of fraud. In spite of such equalities, none of the words appearing on their lists are the same[11]. Possible reasons might be that Cecchini et al. (2010) include both one-grams, bi-grams, and tri-grams[12], whereas Glancy and Yadav (2011) only include one-grams, or the fact that their data samples are based on observations from different time periods (1993-2002 and 2006-2008, respectively). The choice of models similarly varies. Examples include e.g. SVM (Cecchini et al., 2010; Öğüt et al., 2009) ensemble models (Whiting et al., 2012). Other researchers focusing on fraud include e.g. Kirkos et al. (2007), Zhou and Kapoor (2011), and Ravisankar et al. (2011).

Researchers have focused on several different areas such as credit scoring of lenders (AghaeiRad et al., 2017; Florez-Lopez and Ramon-Jeronimo, 2015; Schebesch and Stecking, 2005; Wang et al., 2011), forecasting of earnings and earnings growth (Li, 2010; Nekrasov and Ogneva, 2011),

[11] Both lists are included in Appendix II

[12] One-grams, bi-grams, and tri-grams are types of n-grams. N-grams of text are a set of co-occurring words within a given window (in these cases, the windows are one, two, and three words, respectively)

forecasting of firm performance proxied by ROE and ROA (Delen et al., 2013), volatility forecasting (Hajizadeh et al., 2012; Liu and Maheu, 2009; Taylor, 2005), interest rate forecasting (Kim and Noh, 1997), foreign exchange rate forecasting (Kodogiannis and Lolis, 2002; Nag and Mitra, 2002; Ni and Yin, 2009; Yao and Tan, 2000) and financial risk forecasting (Tsai and Wang, 2017).

2.2.2.2 Types of Models

As can be seen in the previous section, researchers vary in their use of models, both across and within similar predictive tasks. Thus, based on the above, it seems as if no one type of model is generally preferred to others. However, looking through the body of papers, it is clear that some are more prevalent. Among these are neural networks (NNs), support vector machines (SVMs), support vector regression (SVR), decision trees (DTs), ensemble models, and Naïve Bayes. In the following sections, the focus will not be on explaining the technicalities of the different types of models, but rather on exemplifying their use in research and determining the advantages and disadvantages they are subject to.

NNs are used for various research, such as the forecasting of bankruptcy (Tung et al., 2004), earnings per share (Zhang et al., 2004), fraud (Kirkos et al., 2007), and foreign exchange rates (Shen et al., 2015). Just like researchers apply NNs within many fields, they apply several types of them. Examples include a Self-Organizing Fuzzy NN (Bollen et al., 2011), a back-propagation NN (BPNN) (Moshiri et al., 1999), and a stochastic time effective function NN (STNN) (Wang and Wang, 2015). When comparing the performance of different types of NNs, Moshiri et al. (1999) find that a recurrent NN outperforms a BPNN when forecasting longer horizons, whereas it underperforms for one-month ahead forecasts. Wang and Wang (2015) find that an STNN paired with principal component analysis for feature selection performs better than a traditional BPNN (also when BPNN is used in combination with principal component analysis). Several researchers find that the predictive performance of NNs is superior to that of other models. Geng et al. (2015) e.g. find that NNs perform better DTs and SVMs when predicting financial distress of Chinese companies and Zhang et al. (2004) find that NNs perform better than univariate and multivariate linear models when forecasting EPS. Both Kaastra and Boyd (1996) and Chen et al. (2013) highlight the advantages of NNs, arguing that they are flexible function approximators that are able to map any nonlinear functions, and that these features make them appropriate for pattern recognition, classification, and forecasting. In spite of their suggested superiority to other models, NNs are also subject to criticism. Mainly, this criticism addresses their black box nature, but their excessive training times, software requirements, and the necessity of experimentally selecting a large number of parameters are also emphasized as disadvantages (Kaastra and Boyd, 1996). In addition, not all researchers find that the predictive accuracy of NNs outperforms that of other models: Kirkos et al. (2007) e.g. find that a Bayesian Belief Network performs better in fraud detection.

As most real-life problems are not linear, the commonly used support vector classifier was extended to an SVM. Besides from being applicable for the approximation of non-linear relationships, this model can be used beyond binary classification problems (James et al., 2013, p. 337). SVMs are used for several prediction purposes. Both Shie et al. (2012) and Chen (2014) combine SVM with particle swarm optimization (PSO) for feature selection to predict financial distress. Cecchini et al. (2010) use SVM when predicting fraud and bankruptcy based on the use of specific words in corporate disclosures. Schebesch and Sleeking (2005) apply SVM in credit scoring and create a classification model to help banks decide whether an applicant should be approved for a loan or not. Ravisankar et al. (2011) use several models, among these, SVMs, to identify companies that exercise financial statement fraud. SVR is an extension of SVM that is similarly used in various prediction scenarios, such as the prediction of financial time series (Lu et al., 2009) and stock returns (Lu, 2013; Yeh et al., 2011).

Other widely used types of models are DTs, which can be applied to both regression and classification problems (James et al., 2013, p. 303). Within the corpus of papers included in this review, examples of applications include e.g. fraud detection (Kirkos et al., 2007), bankruptcy prediction (Gepp et al., 2010), and event prediction based on financial documents (Chan and Franklin, 2011). Both Kirkos et al. (2007) and Gepp et al. (2010) find that a DT outperforms other models (a Bayesian Belief network, an NN, and a multiple discriminant approach, respectively). In spite of these findings, the main advantages of DTs do not lie in their accuracy, but in their interpretability and mirroring of human decision making (James et al., 2013, p. 315). Especially in business applications, users choose to apply DTs due to such interpretability, well aware of the costs that such a model choice may have in accuracy (Florez-Lopez and Ramon-Jeronimo, 2015). Furthermore, researchers should be aware of the non-robustness of DTs when deciding which models to use, as this lack of robustness entails that small changes in the data can have large effects on the function approximation (the estimated tree). This might compromise the replicability of research. A possible way of increasing the accuracy of DTs might be to use random forests or boosting. These models, known as ensemble models, use trees as building blocks for more powerful prediction models (James et al., 2013, p. 316). The characteristics and use of such models are elaborated upon later in this section.

The Naïve Bayes Classifier is similarly widely applied. Researchers have e.g. used it in textual analysis. Li (2010) manually categorizes 30,000 forward-looking sentences in 10-K MD&As and uses them as training data for a Naïve Bayes learning algorithm in order to enable it to categorize the tone (positive, neutral, or negative) and content of other statements. Humpherys et al. (2011) similarly focus on the content of MD&As in order to detect fraud by using measures constructed by Zhou et al. (2004) (such as average sentence length and the number of modal verbs divided by the total number of verbs). The criticism of the classifier mainly addresses its "naïve" part, which is the part of the technique that assumes independence. It is assumed that the probability of the appearance of a word is unaffected by the presence (or absence) of each other word (Li, 2010), even though this, most likely, never holds. Nevertheless, it simplifies the computation and avoids the "curse of dimensionality"-problem[13] (Ibid., p. 1060).

Several other models such as logistic regression, vector distance classifier, multivariate discriminant analysis, singular value decomposition model, Leave-One-Out-Incremental Extreme Learning Machine, Genetic Algorithm (GA), and multi-criteria decision aid exist, and as it can be inferred, the number of models applied in the cross field of machine learning and accounting is almost endless. However, one more technique, that has proven to be among the most powerful prediction models (Whiting et al., 2012), will be elaborated. The notion that no single method is always the best, has led to the creation of ensemble models (Yu et al., 2014). Examples of applications of ensemble models include fraud prediction (Whiting et al., 2012), stock return prediction (Patel et

[13] The curse of dimensionality refers to the decrease in performance that can happen as dimensions increase. The phenomenon is a problem for e.g. the Naïve Bayes classifier and other local approaches (James et al., 2013, p. 168)

al., 2015), financial crisis prediction (Tsai and Wu, 2008; Wang and Wu, 2017), and credit scoring (Tsai and Wu, 2008). Not all researchers find that ensemble models are superior, though. In their bankruptcy and credit scoring research, Tsai and Wu (2008) examine the suggested superiority of ensemble methods compared to single classifiers, and find that the multiple classifiers only outperform the single classifier in one of three datasets. Similarly, Geng et al. (2015) find that the performance of an NN for financial distress prediction is better than that of both a DT and an ensemble of multiple classifiers. However, both Whiting et al. (2012) and Patel et al. (2015) find that ensemble models outperform single models.

2.2.3 Textual Analysis

Just as researchers use different models when applying machine learning, they focus on different sources and use different measures when applying textual analysis. The amount of electronic financial texts available to researchers is increasing, and software for textual preprocessing and analysis is continuously developing. These factors have caused a great increase in the research of different financial texts. Furthermore, the possibilities when examining the predictive value of these textual sources are endless, as the number of measures, researchers can apply, is indefinite. If no suitable measure for a given study exists, they are oftentimes able to construct their own. While this sometimes requires extensive work (as e.g. classifying thousands of words or sentences), it can also be straightforward and simple (e.g. when using domain knowledge to manually define a list of words indicative of a specific event). The following sections seek to map out the differences in the literature in regards to the textual sources and linguistic measures applied.

2.2.3.1 Textual Sources

Textual sources can be created by outside stakeholders, such as journalists and investors, or within companies. While the first includes sources such as micro blogging, stock message boards, and news, the latter includes e.g. annual and quarterly reports and earnings press releases. Each of the above are examined by researchers for their predictive value on a variety of outcomes, such as bankruptcy and stock returns, and the following section will elaborate on how they have been applied by researchers.

Bollen et al. (2011) attempt to analyze the moods of more than 9 million business-related tweets on Twitter, and in doing so, they are able to predict the daily direction of DJIA with 87.6% accuracy. The literature indicates that the use of online micro blogging data has since become popular: Oliveira et al. (2017) use tweets just as Bollen et al. (2011), but combine this source with survey indices[14] in order to forecast stock market variables such as returns, volatility, and trading volume. They find that both sentiment and posting volume are relevant variables for the forecasting of returns of the S&P 500 index. Oliveira et al. (2016), Checkley et al. (2017), and Houlihan and Creamer (2017) use data from StockTwits. Checkley et al. (2017) examine the time horizon of the predictive ability of microblogging sentiment, finding that it is minutes, rather than hours or days, whereas Houlihan and Creamer (2017) merely focus on predicting stock returns based on the sentiment of the StockTwits. Yang et al. (2017) argue that even though tweets are faster than news in revealing market information, they are not considered equally reliable. Thus, the authors include both news and tweets. Eliacik and Erdogan (2018) propose a sentiment analysis that takes the social network information into account as well (emphasizing that previous research has not considered the social status of the microbloggers and the impact thereof).

Antweiler and Frank (2004) and Das and Chen (2007) examine whether the content on stock message boards is noisy, finding that, while some of it might be, this is not the case for all of it. Several researchers find that stock message boards carry valuable information, and Das and Chen (2007) argue that aggregated stock messages have strong predictive power. The content is applied in e.g. prediction of stock returns (Antweiler and Frank, 2004; Das and Chen, 2007; Zhang et al., 2012) and in an examination of the initial high returns and long-run underperformance of IPOs (Tsukioka et al., 2017).

Many researchers use news articles in their research. Several are alike in that they measure sentiment (Kräussl and Mirgorodskaya, 2017; Mangee, 2018; Tetlock et al., 2008; Uhl, 2014), but they differ in e.g. choice of news source and choice of wordlist. Both Tetlock et al. (2008) and Mangee (2018) use the Harvard General Inquirer (Harvard GI) for classifications of words into categories. However, whereas Tetlock et al. (2008) only uses the Harvard GI, Mangee (2018)

[14] Such as the American Association of Individual Investors survey and the Investors Intelligence survey

moreover uses the LM word list[15], which was published in 2011, finding a stronger relationship for the context-specific LM dictionary-measure than for the Harvard GI-based measure. The news sources used include the Wall Street Journal (Mangee, 2018; Tetlock et al., 2008), Reuters (Uhl, 2014), the Bloomberg News Terminal (Mangee, 2018), and Dow Jones News Services (Tetlock et al., 2008). Tetlock et al. (2008), Uhl (2014), Kräussl and Mirgorodskaya (2017), and Mangee (2018) all focus on predicting stock returns, whereas Huang et al. (2010) construct a news headline agent to assist investors in deciding when to buy and sell stocks. Their agent disseminates new headlines based on their significance degree on the fluctuation of the price index on the next trading day.

The textual sources above have in common that they are not created or controlled by the companies but rather by external stakeholders. Corporate disclosures, such as annual and quarterly reports and earnings press releases, are, on the other hand, content created by the companies themselves. In regards to these, Li (2010a) argues that their textual information is important for financial accounting research, as it provides a context for the data generating function of the numeric financial data. Hence, several researchers e.g. focus on the entire content of annual and quarterly company reports, while others focus on the content of the MD&A section only, or on earnings conference calls.

Using the content of the entire 10-K and 10-Q filings, Loughran and McDonald (2011) find that specific phrases are linked with significantly lower excess stock returns on filing dates, higher volatility, and greater analyst earnings forecast dispersion. Similarly focusing on specific expressions, Chen et al. (2013) find that subjective expressions of managers that are negative, neutral, or positive can be indicative of the performance of companies. They e.g. state that there is a tendency to a usage of more optimistic words when managers are trying to obfuscate negative financial performance and to draw attention to the positive financial performance. Examining 10-Ks, Li (2008) finds a positive relationship between readability and earnings persistence. The focus of Dyer et al. (2017) varies from that of the above, as their study concerns identification of trends in 10-K disclosures from 1996-2013 (such as increased length and decreased readability) instead of predictions of specific events.

[15] The LM word list was constructed by Loughran and McDonald (2011). Based on the content of 10-K reports, the word list divides more than 80,000 words among six categories (negative, positive, uncertainty, litigious, strong model, and weak modal) (for further elaboration, see Section 2.2.3.2)

The MD&A section is argued to be the most important textual section of the 10-K and 10-Q disclosure (Hajek, 2017), assessing both liquidity, capital resources, and operations in a manner that makes it understandable to investors (Li, 2010a). According to Goel and Uzuner (2016), the section contains nonfactual content that is perceived with greater interest by users of the reports than other sections. Due to the regarded importance of MD&As, they are widely used by researchers as textual sources. The literature includes research using the content of MD&As for the prediction of e.g. future performance measured by earnings (Li, 2010a), fraud (Cecchini et al., 2010a; Glancy and Yadav, 2011; Goel and Uzuner, 2016; Hajek and Henriques, 2017), bankruptcy (Cecchini et al., 2010; Mayew et al., 2015), and stock returns (Hajek, 2017). When examining MD&As, researchers use different measures such as sentiment (Goel and Uzuner, 2016; Hajek, 2017; Hajek and Henriques, 2017; Li, 2010), readability (Hajek, 2017), and context-specific measures (e.g. concept scores measuring the use of specific bankruptcy or fraud indicative terms as constructed by Cecchini et al. (2010)). The measures applied vary, and no measure is generally preferable to others. Thus, while Hajek (2017) finds that using a bag-of-words approach provides more accurate predictions than using sentiment, this might not be the case for other studies. By using sentiment, Goel and Uzuner (2016) e.g. find evidence that fraudulent MD&As have more positive and negative sentiment, i.e. larger polarity, and a greater proportion of subjective content than objective content. Davis and Tama-Sweet (2012) compare the language used by managers in MD&As and earnings press releases. They find that the pessimistic language contained in MD&As provides incremental information to that disclosed in press releases. In order to measure sentiment, researchers must use either pre classified lists of words or classify words/sentences in categories themselves (for elaboration on sentiment measures, see Section 2.2.3.2). The pre classified lists used by researchers focusing on MD&As include e.g. Harvard GI (Li, 2010), Diction (Li, 2010) and the LM word list (Feldman et al., 2010; Goel and Uzuner, 2016; Hajek and Henriques, 2017; Loughran and McDonald, 2011; Mayew et al., 2015). Murphy et al. (2018) question the ability of linguistics in MD&As as indicators of fraud because MD&As are written by many individuals (some of whom are unaware of the financial misrepresentation), but they find that naive and innocent participants actually unwittingly write MD&As associated with fraudulent financial statements with relatively little suspicion. Feldman et al. (2010) similarly question the informational content of MD&As and measure tone change in order to explore whether their content has incremental information beyond financial measures. Other researchers similarly combine financial variables (such as P/E ratio, P/B ratio, and ROE) and linguistic variables (Cecchini et al., 2010; Hajek, 2017; Hajek and Henriques, 2017; Mayew et al., 2015). In doing so, Mayew et al. (2015) find that adding linguistic measures to their model improves the predictive power with almost 50% when compared to the predictive power of using financial variables alone. Cecchini et al. (2010) are similarly able to improve their prediction accuracy from 80% to 83.87% and from 75% to 81.97% (bankruptcy and fraud predictions, respectively) by combining linguistic measures and financial measures, but whereas Mayew et al. (2015) add linguistic measures to a model based on financial measures, Cecchini et al. (2010) do it in the opposite order.

Earnings conference calls differ from the above in their format, as they are originally verbal sources that have since been transcribed to enable textual analysis of their content. Nevertheless, their content stems from the company, and researchers have e.g. found that the market more efficiently processes information released through this channel than information released in 10-K and 10-Q filings (Davis and Tama-Sweet, 2012). Several researchers focus on earnings conference calls (Davis and Tama-Sweet, 2012; Milian and Smith, 2017; Price et al., 2012), and their purposes vary. Milian and Smith (2017) find that there is a strong association between the amount of praise managers get from analysts and earnings surprises and the earnings announcement stock returns, respectively. Davis and Tama-Sweet (2012) argue that managers will act strategically and report less pessimistic language and more optimistic language in earnings conference calls relative to MD&As, as the market is more efficient in processing conference call content than MD&A content. Price et al. (2012) measure sentiment and find that firms with positive tone in the question-and-answer session of the call experience significantly higher stock returns, both in a three-day and a two-month window.

All of the above sources are prominent in the literature, but multiple other sources, that have not been as widely used, exist. These include e.g. IPO documents (Brau et al., 2016) and disparate online sources, such as the number of page visits to pertinent Wikipedia pages and the amount of online content produced on a particular day about a company (Weng et al., 2017).

In their review of the textual sentiment research in finance, Kearney and Liu (2014) compare the advantages and disadvantages of the various textual sources. They emphasize that corporate disclosures are suitable as sources when "*studying the role of qualitative information in individual firm performance and stock pricing*" (Ibid., p. 174), whereas news stories can be suitable in both market-level and firm-level research. However, they note that, since news stories come from outsiders, one might not capture insiders' views and perspectives when analyzing them. In terms of timing, Kearney and Liu (2014) argue that news generally concern past events, whereas e.g. the MD&As of 10-K and 10-Q filings and earnings conference calls contain forward-looking statements, which gives the latter more value when predicting future outcomes. Shortcomings of these as opposed to news stories might lie in their infrequent release, which is normally quarterly or even annual. In regards to micro blogging, its content can be both forward-looking and it is indeed timely, but, due to its unregulated form, it is likely to contain a lot of noise, making analysis more difficult and time-consuming. These reasons cause Kearney and Liu (2014) to argue that such Internet postings are not an ideal source of information. Instead they advise that researchers employ as many information sources as possible.

2.2.3.2 Linguistic Measures

Among the papers included in this study that focus on textual analysis, certain measures are more predominant than others. These are measures of sentiment, context-specific measures (such as creating concept scores based on terms that are indicative of certain events), and readability measures. The following section elaborate on different variations of each measure by providing examples of applications as well as the advantages and disadvantages emphasized by researchers.

Measuring sentiment has been popular in financial research for many years and the research in behavioral finance on how sentiment impacts decision-makers, institutions, and markets has especially been intensified during the past decade (Kearney and Liu, 2014). The papers using sentiment measures concern e.g. prediction of stock returns (Davis and Tama-Sweet, 2012; Feldman et al., 2010; Li, 2010; Zhang et al., 2012) and earnings (Li, 2010), and researchers extract sentiment from sources such as stock message boards (Zhang et al., 2012), MD&As (Davis and Tama-Sweet, 2012; Feldman et al., 2010), 10-Ks and 10-Qs (Li, 2010), earnings press releases (Davis and Tama-Sweet, 2012; Price et al., 2012), news (Tetlock, 2007; Tetlock et al., 2008), and microblogging (Antweiler and Frank, 2004; Bollen et al., 2011; Das and Chen, 2007). One of the most common approaches when measuring sentiment is the dictionary-based approach, in which the software reads and classifies words (or phrases or sentences) into specific groups based on a set of predefined categories in a dictionary or wordlist (Kearney and Liu, 2014). When using such an approach, the main difference between applications lies in the choice of dictionary. Most researchers initially used the Harvard GI in finance and accounting research as it was one of the first available word lists (Loughran and McDonald, 2016). Examples of researchers using this dictionary include Tetlock (2007), Tetlock et al. (2008), and Das and Chen (2007). In spite of the extensive use of the Harvard GI in past years, its appropriateness for being used to analyze texts in finance and accounting has been questioned, as it was not developed in a suitable context.

Researchers find that using general dictionaries is inappropriate for analysis of financial texts, as doing so might result in wrongful classifications that will add noise or errors to the measurement of tone (Feldman et al., 2010; Goel and Uzuner, 2016; Hajek, 2017; Loughran and McDonald, 2016, 2011). This is argued based on the fact that some words, construed to be negative in the Harvard GI (such as cancer, cost, capital, and depreciation), are in fact not negative in a financial context. Loughran and McDonald (2011) provide evidence based on 50,115 firm-year 10-Ks between 1994 and 2008 that the Harvard GI list misclassifies words when gauging the tone in financial texts. To overcome such problems, Loughran and McDonald (2011) constructed the LM word list, which now contains more than 80,000 words divided on six categories (negative, positive, uncertainty, litigious, strong model, and weak modal). Its foundation in 10-K filings makes it suitable for analysis of financial texts, and, according to Kearney and Liu (2014), the LM word list has become predominant in more recent studies. Examples of researchers applying the LM word list include e.g. Feldman et al. (2010), Mangee (2018), Goel and Uzuner (2016), Hajek and Henriques (2017), and Mayew et al. (2015). In their study, Feldman et al. (2010) even state that they use the LM word list instead of the previously used Harvard GI in order to improve their previous research.

Other examples of dictionaries used by researchers include Diction (Davis and Tama-Sweet, 2012; Hajek, 2017; Price et al., 2012) and the Henry Word List (Davis and Tama-Sweet, 2012). As Diction was originally developed for analyzing political discourse, it has been subject to criticism similar to that of the Harvard GI. The Henry Word List was, on the other hand, to the best of the authors' knowledge, the first list created for financial texts specifically, but due the its limited number of words (e.g. only 85 negative words), its use has been limited (Loughran and McDonald, 2016).

The corpus of literature carries several examples of researchers that, instead of using pre classified word lists in order to measure e.g. sentiment, focus on keywords, terms, or concepts that are discriminating for a specific type of event or state. Researchers either identify these keywords manually or by using machine learning or other advanced approaches (beyond simply selecting words based on expert knowledge). However, as argued by Oliveira et al. (2016), the adoption of a manual approach for producing dictionaries, such as done by Loughran and McDonald (2011), is costly and manual and thus, not always feasible. Examples of constructions of context-specific word lists include Cecchini et al. (2010), Bodnaruk et al. (2015), Brau et al. (2016), Ibriyamova et al. (2017), and Hajek (2017). According to Kearney and Liu (2014), equal weighting of words is the method that most studies apply, but some researchers, e.g. Loughran and McDonald (2011), argue that it is not necessarily the best measure of a word's information content. Other widely used schemes are term weighting schemes, labeled term frequency-inverse document frequency (tf-idf). Using tf-idf takes into account the fact that some words generally appear more frequently, as the frequency of a word in a certain document is offset by the number of times it appears within the corpus of documents. When comparing the traditional equal weighting with tf-idf, Loughran and McDonald (2011) find that using tf-idf results in regressions with better fit than using simple proportion. Hajek (2017), Oliveira et al. (2016), and Cecchini et al. (2010), use tf-idf, whereas e.g. Bodnaruk et al. (2015) merely use a measure defined as the percentage of constraining words. Cecchini et al. (2010) create an automated methodology for defining an ontology[16] of key terms that does not involve human intervention and that can be used to detect bankruptcies and fraudulent companies. Their methodology varies from that of other researchers, as they use WordNet to group words into concepts (so that e.g. car and automobile are gathered as one concept instead of used as two separate words). Ibriyamova et al. (2017) apply a similar methodology, but use semantic fingerprints instead of WordNet. Whereas both Cecchini et al. (2010) and Ibriyamova et al. (2017) focus on fraud prediction (and Cecchini et al. (2010) on

[16] Defined as *"a set of concepts based on a particular area of interest"* (Cecchini et al., 2010, p. 167)

bankruptcy too), Bodnaruk et al. (2015) focus on the prediction of liquidity events such as dividend omissions or underfunded pensions. Brau et al. (2016) vary from the research above by not using machine learning for the construction of their wordlist. The authors apply a survey based approach for creating two strategy related word libraries, arguing that it mitigates researcher bias, is context specific, and contains a continuous measure of a word's tone relative to the dichotomous categories such as negative and positive.

Readability measures are similarly widely used among researchers. Researchers have e.g. found that less readable reports are associated with lower accuracy in analyst earnings forecasts, more analyst following, and greater analyst effort (Lehavy et al., 2011). Furthermore, it is found that more readable reports have more persistent positive earnings and that less readable reports are associated with lower profitability (Li, 2008). Several measures of readability exist, and the corpus of literature shows examples of the use of e.g. the Fog Index, the Bog Index, and measures based on writing attributes. According to Loughran and McDonald (2014), the Fog Index (Gunning, 1969) is one of the most popular measures of readability across many research fields. It consists of two components:

$$Fog\ Index = 0.4(average\ number\ of\ words\ per\ sentence + percentage\ of\ complex\ words)$$

In spite of its many applications, Loughran and McDonald (2014) criticize the Fog Index, stating that the first of its components, average sentence length, is substantially less precise in the context of financial disclosures than in traditional prose. In regards to the second component, percentage of complex words, they argue that it does not either provide any useful insights in the case of financial documents, as multi-syllable words, that are not considered hard for investors to comprehend, are commonly used to describe business operations (Ibid., p. 1645)[17]. However, such words would decrease readability if the Fog Index were used as the measure of readability. In their study, Loughran and McDonald (2014) show that the top quartile of multi-syllable words are likely to be known by a typical investor or analyst, arguing that the use of such words does not necessarily indicate that a document is less readable for its intended readers.

[17] The authors provide examples of commonly used business words: Corporation, company, agreement, management, and operations

While a measure such as the Fog Index is constructed to capture the clarity-component of readability, other measures focus on the writing attributes in order to capture the quantity of disclosure-component (Bonsall et al., 2017). Such a measure is constructed by Loughran and McDonald (2014), who propose a simple proxy for readability defined as the document file size of the 10-K. This measure requires a minimum of work, as no parsing is required, it is correlated with other readability measures, and the authors' results show that it outperforms the Fog Index (Ibid.). Bonsall et al. (2017) emphasize the potential downsides of using quantity-based readability measures, stating that a decrease in readability could possibly be due to the inclusion of other constructs: separate exhibits are included in the filings due to legislative requirements and while adding to the file size, they do not necessarily decrease the readability. Furthermore, different file types (HTML, XML, PDF etc.) could lead to substantial variations in file size (Ibid.).

Bonsall et al. (2017) supply an alternative measure defined as the Bog Index. The authors argue that this measure "*captures the plain English writing attributes recommended by linguistic experts and highlighted in the SEC's Plain English Handbook*" (Ibid., p. 333), and that its word difficulty measure differentiates itself from the Fog Index syllable count by being based on a list of over 200,000 words from which words can be given penalties based on a combination of familiarity and precision. Given the recent publish of the paper introducing this measure, researchers have not yet applied it, nor commented on its advantages and shortcomings.

3 Methodology

This section elaborates on the methodology applied in the study. It is divided into three subsections concerning the data (including a description of the data sample, the data sources used, and how the data was processed prior to modeling), the modeling, and the analysis (including descriptions of model evaluation metrics, the analysis framework, and the methodology applied for robustness tests).

3.1 Data

Besides facilitating understanding and transparency of the data and methodology, the detailed elaboration contained in this section is included in order to ensure replicability of the study. As emphasized by e.g. Loughran and McDonald (2016), researchers must consider the transparency and replicability of their results when conducting textual studies. The method of textual analysis is still considered relatively new (Ibid.), and it is continuously developing. New capabilities constantly evolve due to increases in computational power and an explosion in digitally available textual content. Thus, many methodological choices of previous studies might be altered if the studies were to be conducted today, but if researchers fail to enable replicability and provide transparency, such subsequent research is impossible to conduct. Besides enabling replicability in future research, this study focuses on enabling practical replicability, as its aim is to enhance analysts' EPS forecast accuracy.

3.1.1 Description of Data Sample

The data sample used in this study is obtained by extracting content from the data sources and performing the engineering steps elaborated upon in the following sections. Prior to subsampling based on sectors, it consists of the textual content from 11,788 10-K and 10-Q filings from S&P 500 companies in the time period from 2012-2017.

The number of constituents in the S&P 500 Index can, despite its name, vary from time to time (as of e.g. end-April 2018, the index had 505 constituents) (Standard & Poor's, 2018), and some filings were unobtainable due to reasons that will be addressed further in the following. As earnings surprise data was used to indicate whether the consensus estimate of analysts was above or below actual EPS, filings for which earnings surprise data for the following quarter was unobtainable were left out. The distributions of consensus estimates above, below, or equal to actual EPS, as well as the distribution of filings on quarters can be seen in Figure 3.1.

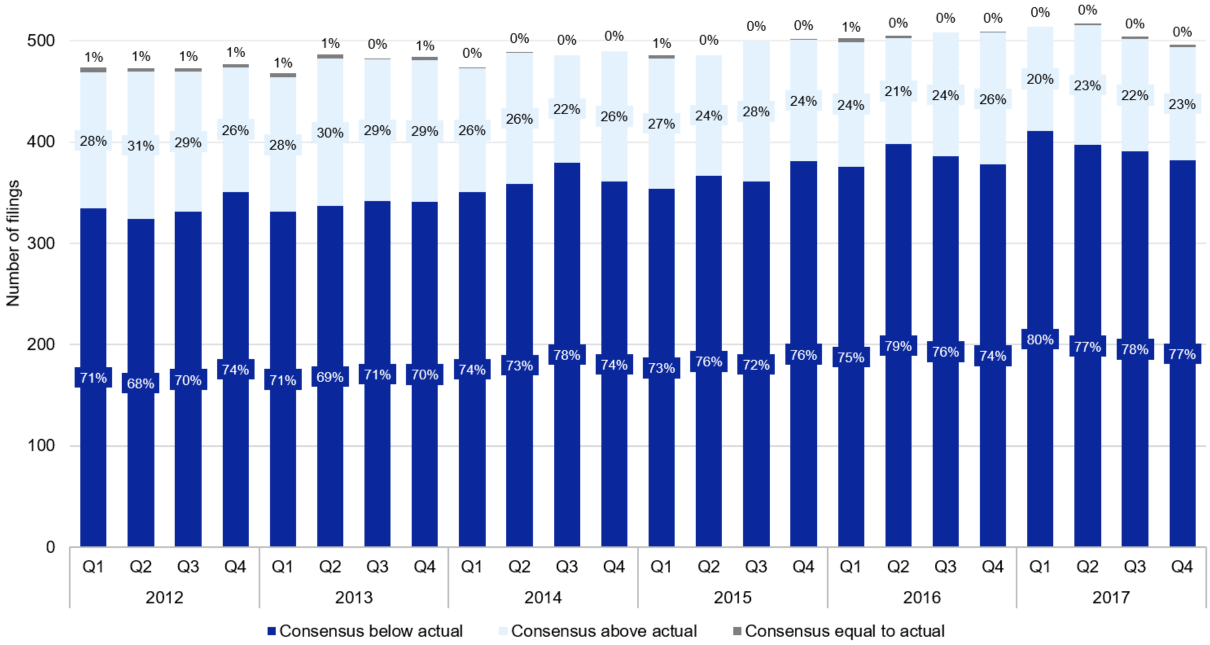


Figure 3.1: Quarterly distribution of observations of estimates above, below, or equal to actual EPS

In order to enable sector-specific modeling, a subsample of data for each sector was created. A description of each sector is included in Appendix III, and the distribution of filings on sectors can be seen in Figure 3.2.

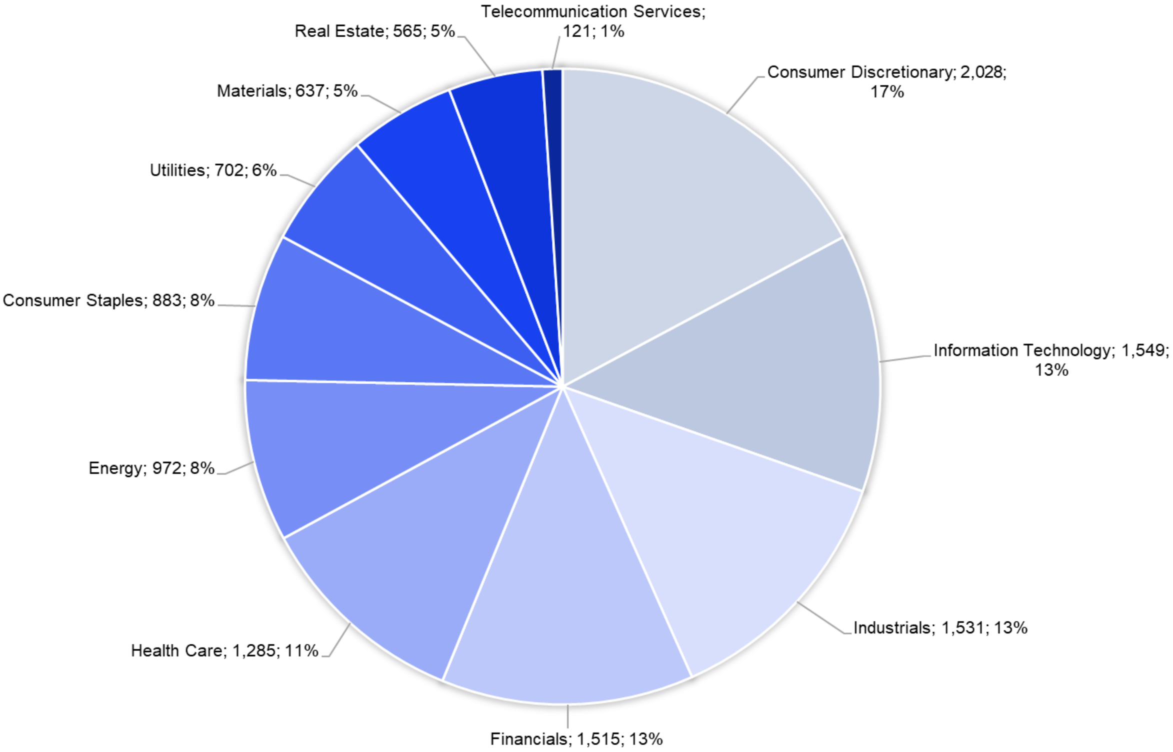


Figure 3.2: Distribution of filings on sectors[18]

Due to the limited number of observations for companies in Energy, Materials, Real Estate, Telecommunication Services, and Utilities (less than 30 observations in one or more quarters), these sector, that together represented approximately 25% of all observations, were excluded from the modeling process. The remaining 8,791 filings were distributed among six sectors: Consumer Discretionary (23.1%), Consumer Staples (10.0%), Financials (17.23%), Health Care (14.6%), Industrials (17.4%), and Information Technology (17.6%). Figure 3.3 shows the number of observations and the distribution of underestimations, overestimations and estimates equal to EPS in each of the included sectors.

[18] Appendix IV includes a table showing the distribution

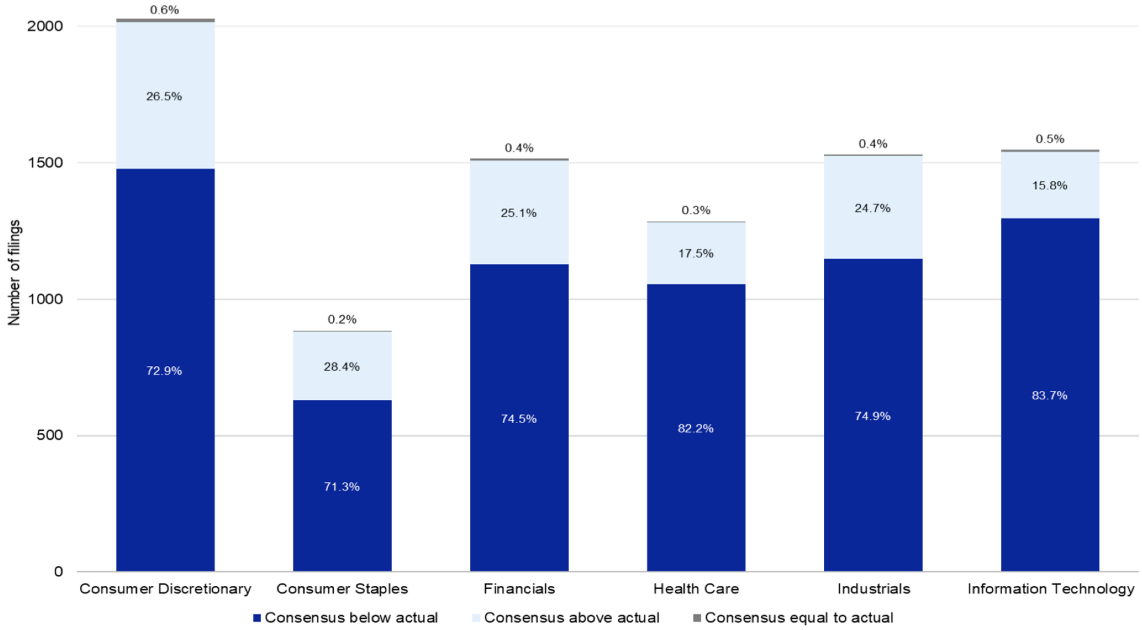


Figure 3.3: Sector distribution of observations of estimates above, below, or equal to actual earnings

In the following sections, the methodology for obtaining, preparing, and joining the data described above is elaborated.

3.1.2 Data Sources

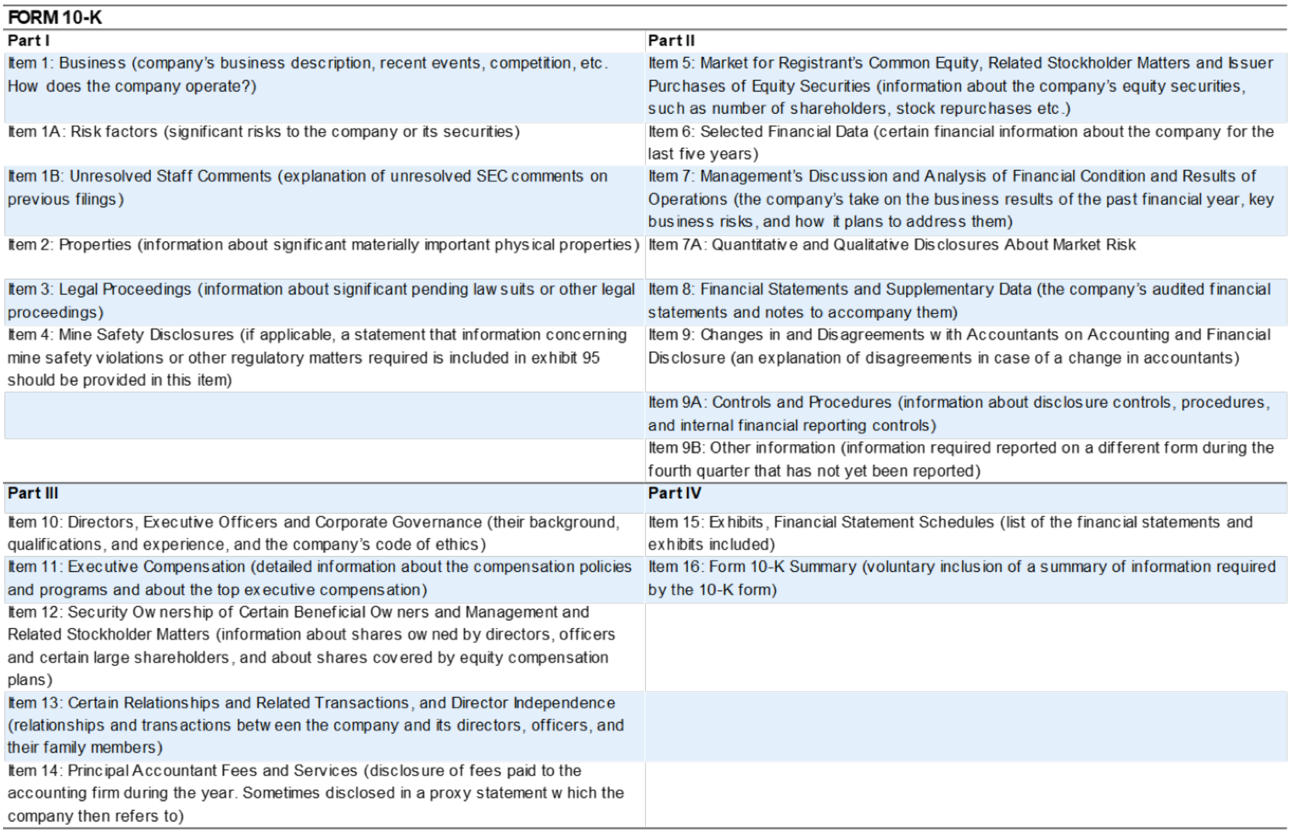
Three sources were used for obtaining the different types of data necessary for the construction of sector-specific subsamples: 1) Bill McDonald's online Google Drive database, Stage One 10-X Parse Data (SRAF, 2018), was used for obtaining the textual content of 10-K and 10-Q filings, 2) Compustat was used for obtaining a list of S&P 500 constituents in each quarter, and 3) Thomson Reuters was used for obtaining earnings surprise data, which was used as proxy for analyst under-/overestimations of EPS, and sector classifications of the companies as well as the company characteristics used in the robustness tests.

3.1.2.1 The Stage One 10-X Parse Data

The data sample consists of the entire textual content of the 10-Ks and 10-Qs filed with the SEC by S&P 500 companies from 2012-2017. Companies that register their securities under the Securities Act must file reports of form 10-K and 10-Q with the U.S. Securities and Exchange

Commission (SEC) unless they fall within one of the following categories: 1) The company has less than 300 shareholders of the class of securities offered, or 2) The company has less than 500 shareholders of the class of securities offered and less than $ 10 million in total assets for each of its last three fiscal years (SEC, 2013). Companies fulfilling the requirements listed above must file three reports of form 10-Q and one report of form 10-K with the SEC annually.

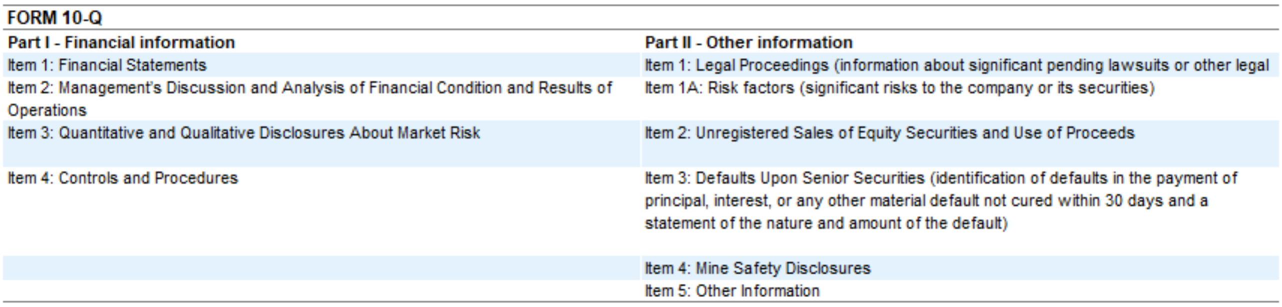
Reports of form 10-K provide readers with a comprehensive overview of the business and financial condition of a company and its audited financial statements (SEC, 2009). The report is divided into four parts and 21 items (SEC, 2018). The content of each of its items is outlined in Table 3.1.



FORM 10-K	
Part I	**Part II**
Item 1: Business (company's business description, recent events, competition, etc. How does the company operate?)	Item 5: Market for Registrant's Common Equity, Related Stockholder Matters and Issuer Purchases of Equity Securities (information about the company's equity securities, such as number of shareholders, stock repurchases etc.)
Item 1A: Risk factors (significant risks to the company or its securities)	Item 6: Selected Financial Data (certain financial information about the company for the last five years)
Item 1B: Unresolved Staff Comments (explanation of unresolved SEC comments on previous filings)	Item 7: Management's Discussion and Analysis of Financial Condition and Results of Operations (the company's take on the business results of the past financial year, key business risks, and how it plans to address them)
Item 2: Properties (information about significant materially important physical properties)	Item 7A: Quantitative and Qualitative Disclosures About Market Risk
Item 3: Legal Proceedings (information about significant pending law suits or other legal proceedings)	Item 8: Financial Statements and Supplementary Data (the company's audited financial statements and notes to accompany them)
Item 4: Mine Safety Disclosures (if applicable, a statement that information concerning mine safety violations or other regulatory matters required is included in exhibit 95 should be provided in this item)	Item 9: Changes in and Disagreements w ith Accountants on Accounting and Financial Disclosure (an explanation of disagreements in case of a change in accountants)
	Item 9A: Controls and Procedures (information about disclosure controls, procedures, and internal financial reporting controls)
	Item 9B: Other information (information required reported on a different form during the fourth quarter that has not yet been reported)
Part III	**Part IV**
Item 10: Directors, Executive Officers and Corporate Governance (their background, qualifications, and experience, and the company's code of ethics)	Item 15: Exhibits, Financial Statement Schedules (list of the financial statements and exhibits included)
Item 11: Executive Compensation (detailed information about the compensation policies and programs and about the top executive compensation)	Item 16: Form 10-K Summary (voluntary inclusion of a summary of information required by the 10-K form)
Item 12: Security Ow nership of Certain Beneficial Ow ners and Management and Related Stockholder Matters (information about shares ow ned by directors, officers and certain large shareholders, and about shares covered by equity compensation plans)	
Item 13: Certain Relationships and Related Transactions, and Director Independence (relationships and transactions betw een the company and its directors, officers, and their family members)	
Item 14: Principal Accountant Fees and Services (disclosure of fees paid to the accounting firm during the year. Sometimes disclosed in a proxy statement w hich the company then refers to)	

Table 3.1: The content of 10-K filings (SEC, 2018)

While the filings of form 10-Q contain less detail than those of form 10-K, they are never the less considered to be comprehensive reports of the performance of companies. The form 10-Q is divided into two parts and ten items, which are outlined in Table 3.2.



FORM 10-Q	
Part I - Financial information	**Part II - Other information**
Item 1: Financial Statements	Item 1: Legal Proceedings (information about significant pending lawsuits or other legal
Item 2: Management's Discussion and Analysis of Financial Condition and Results of Operations	Item 1A: Risk factors (significant risks to the company or its securities)
Item 3: Quantitative and Qualitative Disclosures About Market Risk	Item 2: Unregistered Sales of Equity Securities and Use of Proceeds
Item 4: Controls and Procedures	Item 3: Defaults Upon Senior Securities (identification of defaults in the payment of principal, interest, or any other material default not cured within 30 days and a statement of the nature and amount of the default)
	Item 4: Mine Safety Disclosures
	Item 5: Other Information

Table 3.2: The content of 10-Q filings (SEC, 2018a)

The corporate filings included in the sample were obtained from Bill McDonald's online Google Drive database, Stage One 10-X Parse Data (SRAF, 2018). This database contains zipped versions of 10-X filings from 1994 until 2017 (included). This study includes 10-K and 10-Q forms and limit the time period to 2012-2017. Companies have the possibility to file amended 10-K/As and 10-Q/As in order to provide or correct missing or incorrect information in 10-Q and 10-K filings. In 2017, around 340 10-K/As were filed, and the most common reasons for amendments consisted in subsequent incorporation of the information required in Part III of the filing and in missing signatures and exhibits (Audit Analytics, 2018). As companies are allowed to file the information required in Part III, Items 10-14, at a later point in time, this information will often not be included in the 10-K first submitted. Companies either choose to file a proxy statement or an amended 10-K at a later point in time. As neither of the most common reasons for amending filings is likely to alter the results of this study substantially, and as the number of amended filings is relatively small, amended filings are not included.

The files used were the results of McDonald's "Stage One Parse". This parse consists of cleaning each document of extraneous materials in order to enable researchers to analyze its textual content. McDonald's methodology involves parsing out markup tags, ASCII-encoded graphics, and tables from the text file that embeds the HTML, XBRL, exhibits, and the ASCII-encoded graphics for the given filing[19] (SRAF, 2018). In order to assess whether additional changes were necessary, a sample of 50 random observations was examined. This examination resulted in the following alterations: 1) Page numberings were removed (both standard numbers, numbers surrounded by markers, and roman numbers), 2) All tabulating characters were removed (some remained after the Stage One

[19] According to McDonald, one filing often includes several documents: *"For example, IBM's 10-K filing on 20120228 lists the core 10-K document in HTML format, ten exhibits, four jpg (graphics) files, and six XBRL files"* (SRAF, 2018)

Parsing even though McDonald had removed all of the tables), 3) All appearances of the word "PART" in upper case were removed, 4) All numeric or roman numbers following "PART" were removed, 5) All HTML tagging used by McDonald to indicate the header of a report and any exhibits was removed. The code applied can be seen in Appendix V.

All filings were downloaded in separate text files and given a unique ID as file name. JSON-files containing information on each of the filings, such as submission date and the Central Index Key (CIK) of the company, were created. Furthermore, an index table containing each of the filing IDs and company CIK, submission date, company Committee on Uniform Security Identification Procedures (CUSIP), form (e.g. 10-K), submission year, and submission quarter was created. The entire data sample, including all filings from 1994 to 2017 and of all companies, consisted of 1,029,937 filings.

As it can be derived from the literature review above, researchers consider the MD&A section to be one of the most important sections of 10-K and 10-Q filings. According to a survey conducted by Balakrishnan et al. (2010), the majority of financial analysts indicate that the MD&A section is a very important or extremely important item when they are to evaluate a firm. It contains non-factual content that is perceived with greater interest by users than other parts of the filing (Goel and Uzuner, 2016), and it provides users with superior qualitative information on the performance of a firm and prospects from managers' perspective (Loughran and McDonald, 2011). Thus, one might argue that using MD&As as the textual source for this research would yield better and more accurate results. However, after several unsuccessful attempts to extract the section it was concluded that doing so was not feasible within the scope of this research without compromising the quality of the data. Thus, it was chosen to use the entire textual content of the filings and thereby to avoid the extraction of MD&As. Conferring with leading researchers within the field of textual analysis of financial texts supported such a decision. Bill McDonald, one of the researchers behind the LM word list and the supplier of the online database of corporate filings, stated that the accurate parsing of MD&As is, in his opinion, "*virtually impossible*" (see Appendix VI for the full e-mail). When explaining the obstacles to Petr Hajek, who focuses on MD&As in his paper on mining annual reports for the detection of financial statement fraud, he admitted to having faced the same difficulties. Hajek replied that he had overcome such obstacles by having students manually extract the MD&A sections of the 1,400 reports included in his sample (see Appendix VII for the full e-mail). Glancy and Yadav (2011) similar extract MD&As manually. An elaboration on the extraction approaches attempted and the various challenges resulting from such approaches is available in Appendix VIII.

3.1.2.2 Compustat

The constituents of the S&P 500 index from 2012 to 2017 were determined by using Compustat – Capital IQ. The output included `effective_from_date` and `effective_thru_date`, which made it possible to determine which companies were in the index at each given point in time out of the 657 companies that had been a part of the index within the selected six-year period. Furthermore, the output included a CUSIP codes. These are unique nine character identifiers that are created to be specific to each issuer as well as each issue (CUSIP Global Services, 2018). General Motors e.g. has the same ticker and CIK code now as prior to the delisting during the Global Financial Crisis, but the CUSIP changed when the company was listed in its current form.

3.1.2.3 Thomson Reuters

3.1.2.3.1 Earnings Surprise Data

In line with the approach taken by several researchers focusing on analyst forecast accuracy (e.g. Clement, 1999; Lim, 2001), the earnings surprise data was downloaded from the Institutional Broker Estimate System (I/B/E/S). Besides having a strong record of academic usage, I/B/E/S is integrated in Thomson Reuters and its data is used by more than 70% of the top US and European asset managers (Thomson Reuters, 2018), suggesting that the quality of the data is high.

Historic earnings surprise data was used to construct variables indicating whether EPS had been over- or underestimated, i.e. whether a consensus estimate was above or below actual EPS, respectively. These variables were defined as follows:

$$\text{Cons\_Above\_Act} = \begin{cases} 1 & if\ earnings\ surprise < 0 \\ 0 & if\ earnings\ surprise \geq 0 \end{cases}$$

$$\text{Cons\_Below\_Act} = \begin{cases} 1 & if\ earnings\ surprise > 0 \\ 0 & if\ earnings\ surprise \leq 0 \end{cases}$$

where

$$earnings\ surprise = \frac{actual\ EPS - consensus\ mean\ EPS}{consensus\ mean\ EPS}$$

3.1.2.3.2 Sector Classification

The use of words can be very sector-specific (Feldman et al., 2010). As illustrated in the literature review, the measurement of tone is dependent on e.g. the word lists chosen for classification, and Loughran and McDonald (2011) argue that some words classified as negative on general non-contextual lists (e.g. mine, cancer, or capital) are more likely to identify a specific industry segment than be indicative of a negative financial event. Dyer et al. (2017, p. 239) similarly argue that there are *"substantial differences in disclosure across firms with different underlying industry fundamentals"*. In order to best gain the sector-specific knowledge and exploit the advantages thereof, analysts are typically assigned to specific sectors, which they cover. The aim of this study is to provide analysts with models yielding predictions of the directional shifts that could enhance the accuracy of their EPS estimates. It is argued that the construction of sector-specific models is best suited for fulfilling this objective, since a model covering all sectors might be affected by the differences in use of words across sectors, just as researchers argue that tone measurement is affected by the context of the words. Thus, sector-specific models will be constructed by creating subsamples of data containing observations from each sector. Several researchers construct sector-specific models. The samples of Balakrishnan et al. (2010), Bae (2012), and Wang and Wu (2017) consist of companies in the manufacturing industry only, whereas Cole and Jones (2004) use a sample of companies in the retail industry. Brau et al. (2016) choose to exclude financial companies.

Global Industry Classification Standard (GICS) codes obtainable from Thomson Reuters Eikon were used to assign sectors to the filings. The data contained CUSIP, which enabled a join with the textual content of the filings (see Section 3.1.3.1). MSCI and S&P Global firstly introduced GICS in 1999. It divides companies into 11 sectors, 24 industry groups, 68 industries and 157 sub-industries (MSCI, 2018). In order to retain sample sizes of a certain size, sector classifications are used. GICS is similarly used by e.g. Ibriyamova et al. (2017).

3.1.3 Data Engineering

3.1.3.1 Joining and Preparation

The preparation and joining of the datasets, obtained using the sources above, was performed in Alteryx Designer (Alteryx)[20]. The aim of the data preparation was to create an input file for DataRobot for each sector containing the textual content of the filings, their submission dates, and two variables indicative of whether consensus was above or below the actual EPS. The latter two variables were separately used as target variables in DataRobot, as two directional models for each sector, predicting whether an observation would be above or below actual EPS, respectively, would be constructed. In the following sections, the workflows constructed in Alteryx in each part of the data preparation phase will be included and elaborated.

The first phase was to construct a workflow in Alteryx with the aim of creating an output file with information on each filing (company CIK, company CUSIP, year and quarter of submission, and submission date) and the text of the filing itself. Four inputs were used: the JSON-index file, all files in the folder containing the text files for a given set of IDs, an index file, and an S&P 500 index file. Figure 3.4 shows an example of an entire workflow. Due to the size of the txt-files and the number of filings, it was necessary to the filings into four folders. Workflows similar to the one illustrated in Figure 3.4 were constructed for each of these folders.

[20] Alteryx is an analytics platform that offers a drag and drop workflow environment for data blending and other advanced tools, such as predictive modeling (Alteryx, 2018)

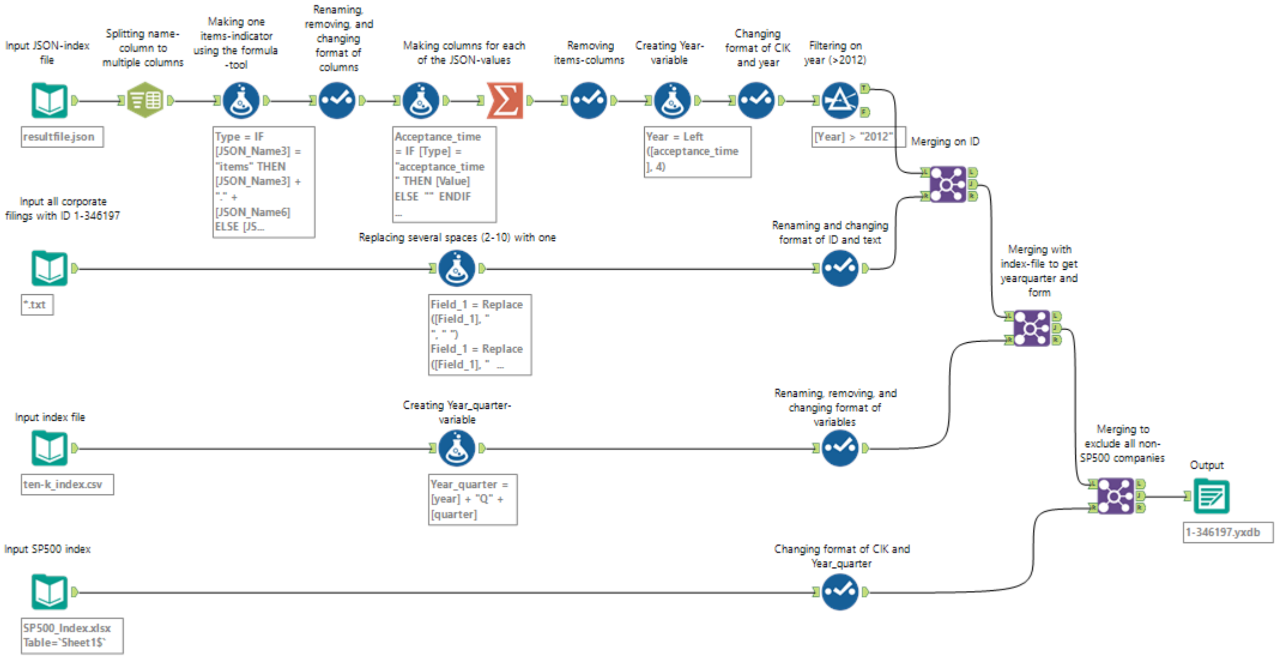

Input JSON-index file
Splitting name-column to multiple columns
Making one items-indicator using the formula -tool
Renaming, removing, and changing format of columns
Making columns for each of the JSON-values
Removing items-columns
Creating Year-variable
Changing format of CIK and year
Filtering on year (>2012)
resultfile.json
Type = IF [JSON_Name3] = "items" THEN [JSON_Name3] + "." + [JSON_Name6] ELSE [JS...
Acceptance_time = IF [Type] = "acceptance_time" THEN [Value] ELSE "" ENDIF ...
Year = Left ([acceptance_time], 4)
[Year] > "2012"
Merging on ID
Input all corporate filings with ID 1-346197
Replacing several spaces (2-10) with one
Renaming and changing format of ID and text
*.txt
Field_1 = Replace ([Field_1], " ", " ") Field_1 = Replace ([Field_1], " ...
Merging with index-file to get yearquarter and form
Input index file
Creating Year_quarter-variable
Renaming, removing, and changing format of variables
ten-k_index.csv
Year_quarter = [year] + "Q" + [quarter]
Merging to exclude all non-SP500 companies
Output
1-346197.yxdb
Input SP500 index
Changing format of CIK and Year_quarter
SP500_Index.xlsx Table=`Sheet1$`

Figure 3.4: Workflow 1 processing text-files with ID 1-346197 (see full size in Appendix X)

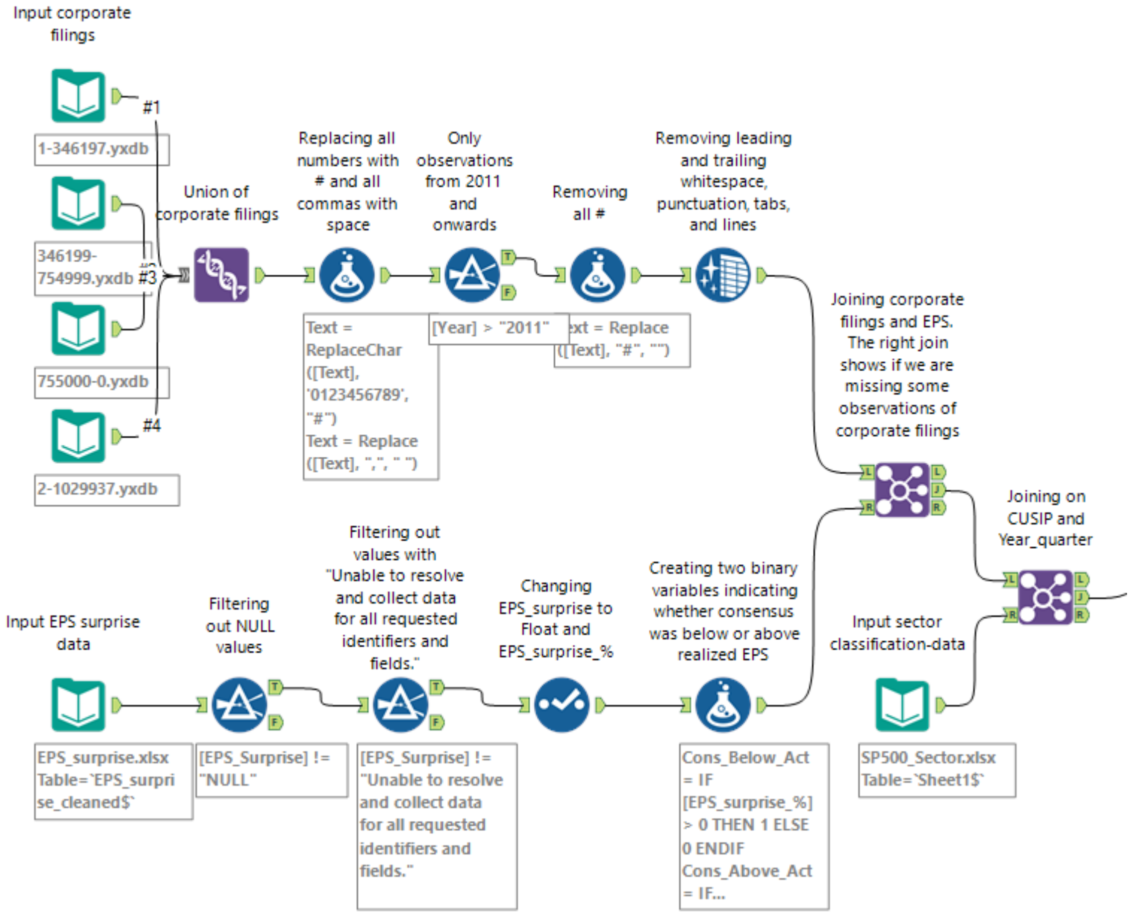
The first row in the workflow inputs the JSON-index file. The file contained two columns: JSON_Name and JSON_ValueString. By applying the tools visualized in Figure 3.4 on the JSON-index file, a list of filings submitted in year 2012 and onwards, containing ID, acceptance_time (equivalent and subsequently renamed to Submission_date), CIK, and Year (year of submission), was obtained.

The second row inputs the folder with txt-files containing the textual content of the filings. Each file in the folder was given the name of its unique ID, and thus, it was possible to input the files in a format containing two columns: one with the name of the file (corresponding to the ID of the relevant filing) and one with all the textual content. Each file (filing) had one row. Examining the filings, it was found that there were many cases of multiple spaces in the texts, and as these do not add value, a formula tool was used to create formulas replacing cases of 2-10 subsequent spaces with one space. After renaming and changing the format of the variables, the input was merged on ID with the JSON-index file inputted in the first row of the workflow.

The third row inputs another index file containing data on the type of submission (10-Q or 10-K) and a variable showing the year and quarter of the filing. Again, every filing had its own unique ID, which was used for joining the inputs.

The fourth row inputs an S&P 500 index. The Compustat data enabled us to create a list stating which companies were in the S&P 500-index in each quarter (and the name, CIK, Ticker, and CUSIP of the companies), and by using `CIK` and `Year_quarter`, it was possible to join this list with the list containing the textual content of the filings. This step significantly reduced the number of observations, as filings that were not from companies in the S&P 500 were excluded. After preparing and joining all of the above inputs, the output tool was used to produce files containing the `text`, `ID`, `CIK`, `Year`, `Submission_date`, `form`, `name`, `ticker`, `CUSIP`, and `Year_quarter` of each of the filings that were submitted by S&P 500 companies between 2012 and 2017.

In order to end up with outputs that could be used as inputs for the modeling in DataRobot, another workflow was constructed. The first part of this workflow is illustrated in Figure 3.5 below.



Input corporate filings
#1
1-346197.yxdb
346199-754999.yxdb #3
755000-0.yxdb
#4
2-1029937.yxdb
Union of corporate filings
Replacing all numbers with # and all commas with space
Text = ReplaceChar ([Text], '0123456789', "#") Text = Replace ([Text], ",", " ")
Only observations from 2011 and onwards
[Year] > "2011"
Removing all #
ext = Replace ([Text], "#", "")
Removing leading and trailing whitespace, punctuation, tabs, and lines
Joining corporate filings and EPS. The right join shows if we are missing some observations of corporate filings
Joining on CUSIP and Year_quarter
Input EPS surprise data
EPS_surprise.xlsx Table=`EPS_surprise_cleaned$`
Filtering out NULL values
[EPS_Surprise] != "NULL"
Filtering out values with "Unable to resolve and collect data for all requested identifiers and fields."
[EPS_Surprise] != "Unable to resolve and collect data for all requested identifiers and fields."
Changing EPS_surprise to Float and EPS_surprise_%
Creating two binary variables indicating whether consensus was below or above realized EPS
Cons_Below_Act = IF [EPS_surprise_%] > 0 THEN 1 ELSE 0 ENDIF Cons_Above_Act = IF...
Input sector classification-data
SP500_Sector.xlsx Table=`Sheet1$`

Figure 3.5: Part one of Workflow 2 (see full size in Appendix XI)

The first step was inputting and unioning (stacking) the four files computed in the workflow described above. The total number of filings was 11,923. Subsequently, all numbers in the text extracted from the corporate filings were replaced with #. Additional steps included removal of all #s, leading and trailing whitespaces, punctuation, tabs, and lines, and replacement of duplicate spaces with single spaces.

Subsequently, the earnings surprise data extracted from Thomson Reuters was inputted. Observations, where no earnings surprise data was available, were removed using the filter tool filtering out NULL-values. This resulted in an exclusion of 165 observations. Furthermore, there were 17 observations that had *"Unable to resolve and collect data for all requested identifiers and fields."* as the value in the earnings surprise field extracted from Thomson Reuters. These observations were similarly removed from the sample. Thus, all 182 observations that, for various reasons (such as lack of analyst coverage), did not have corresponding earnings surprises were excluded.

Based on the percentage earnings surprise variable (`EPS_surprise_%`) extracted from Thomson Reuters, two binary variables, indicating whether consensus underestimated or overestimated EPS (`Cons_Below_Act` and `Cons_Above_Act,` respectively), were constructed. The formulas stated that the variable should be equal to 1 if the value in `EPS_surprise_%` was above zero or below zero, respectively, and 0 otherwise (see Section 3.1.2.3.1 on Earnings Surprise Data for the definition). The aim was to conduct two model constructions for each sector (one indicative of underestimation and one of overestimation), and by creating two such variables, it was possible to construct one data sample suitable for both tasks. When using the data as input in DataRobot, the target variable relevant to the given model was simply selected, while the other was excluded from the feature list.

The two inputs were joined on `CIK` and `Year_quarter`. The year and quarter of the textual content input corresponded to the submission date of the filing. Thus, if a submission date is in 2016 Q2, the filing contains information about Q1. In regards to the earnings surprise-data, the `Year_quarter`-variable corresponded to the period that the surprise concerned. Thus, if the EPS surprise in 2016 Q2 was 10%, it meant that actual EPS in 2016 Q2 exceeded analyst consensus with 10%. Therefore, the `Year_quarter`-variables were matched directly: the aim is to use the textual content regarding Q1 to predict whether there will be a positive or negative earnings surprise in Q2.

The join tool in Alteryx enabled us to identify observations from both inputs that had not been joined with anything and that had therefore been left out. 135 observations were left out from the left input (the input containing the text from the filings). These were the filings that had no corresponding earnings surprise observations. Performing the merge resulted in a right leave out of 295 filings. Further analysis showed that the main reason for this consisted in the companies being European, and thereby not obligated to file with the SEC, or in the companies being subject to M&A activities.

To create sector-specific subsamples, the constructed sector-index was input and the two data sets were joined on `CUSIP` and `Year_quarter`. Performing this join resulted in a data set containing the `text`, `ID`, `CIK`, `Submission_date`, `form`, and `Year_quarter` of 11,788 filings. Furthermore, the output contained variables indicating whether the following quarter resulted in an over- or underestimation of EPS. Filter tools were used to create subsamples for each of the sectors. In order to use the data for predictive modeling in DataRobot, output tools were used to create a CSV-file for each sector containing only the filings' submission dates, textual content, and variables indicating whether the consensus had been above or below actual EPS in the following quarter. This part of the workflow is illustrated in Figure 3.6, and the full workflow is included in Appendix XI.

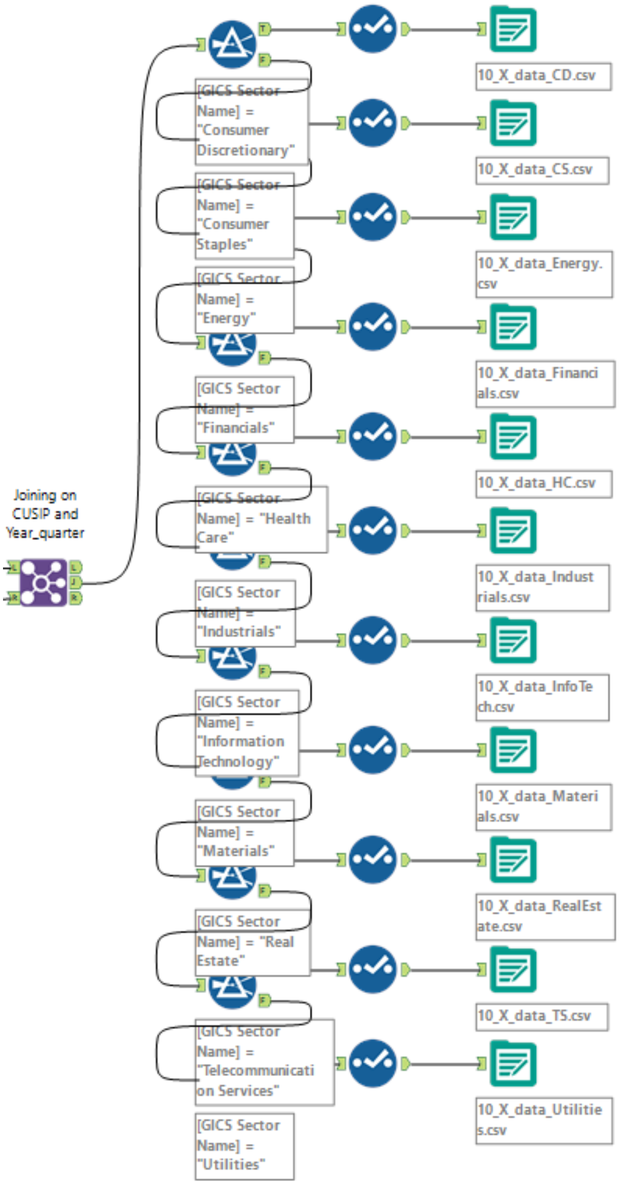

Joining on CUSIP and Year_quarter
[GICS Sector Name] = "Consumer Discretionary"
[GICS Sector Name] = "Consumer Staples"
[GICS Sector Name] = "Energy"
[GICS Sector Name] = "Financials"
[GICS Sector Name] = "Health Care"
[GICS Sector Name] = "Industrials"
[GICS Sector Name] = "Information Technology"
[GICS Sector Name] = "Materials"
[GICS Sector Name] = "Real Estate"
[GICS Sector Name] = "Telecommunication Services"
[GICS Sector Name] = "Utilities"
10_X_data_CD.csv
10_X_data_CS.csv
10_X_data_Energy.csv
10_X_data_Financials.csv
10_X_data_HC.csv
10_X_data_Industrials.csv
10_X_data_InfoTech.csv
10_X_data_Materials.csv
10_X_data_RealEstate.csv
10_X_data_TS.csv
10_X_data_Utilities.csv

Figure 3.6: Part two of Workflow 2 (see full size in Appendix XI)

3.1.3.2 Stop word removal

Additional to the feature engineering described above, stop word removal was performed. Stop words are non-contextual words that are not relevant to the interpretation of a text (Das, 2014). They are furthermore not believed to add meaning to a sentence, as their function is merely grammatical (Li, 2010). Removing stop words is believed to enhance the quality of the analysis of textual content, as noise is reduced (Das, 2014).

The use of stop word lists vary among researchers: Chan and Franklin (2011) e.g. use the Brown List, whereas Li (2010) uses the Lingua::EN::Stopwords list. As no list seems to be considered the standard for removing stop words, this study applies the Generic list provided and recommended by McDonald (SRAF, 2018a). It is argued that a recommendation from McDonald, who is considered to be one of the leading researchers within the field of textual analysis of corporate disclosures, provides certainty that such a list is applicable. The removal of stop words was performed on each of the sector subsamples in R, and the R-script, which also shows the list of stop words, is included in Appendix XII.

3.2 Modeling

The following section includes three subsections: a description of DataRobot, an elaboration of time-aware modeling, and a description of the optimization metric applied.

3.2.1 DataRobot

The construction of models is performed in DataRobot, which is an automated machine learning platform that empowers users of all skill levels to make better predictions faster. Founders and employees of the platform are among the highest ranked data scientists on Kaggle[21], and it incorporates a library of hundreds of the most powerful machine learning algorithms. DataRobot automates, trains, and evaluates predictive models in parallel, making it possible to construct and compare several models in order to choose the one best suited for a given task. According to the DataRobot team, the platform "*provides the fastest path to data science success for organizations of all sizes*" and it enables users to "*build and deploy highly accurate predictive models in a fraction of the time of traditional methods*" (Goh, 2017).

DataRobot is chosen as the modeling tool for several reasons. First, using DataRobot coincides with the goal of this study, as it appears to be an easy to implement machine learning platform that is applicable for analysts who are not data scientists. As stated by the DataRobot team, the platform "*puts the power of machine learning into the hands of any business user*" (Goh, 2017). On Gartner's annual Data & Analytics Summit 2018, automated machine learning was repeatedly positioned as the key technology that would enable "Quantitative Professionals" (i.e. advanced

[21] Kaggle is an online platform with more than 430,000 users hosting worldwide data science competitions. 12 members of the DataRobot team have been ranked in Kaggle's top 100 data scientists, and six are Kaggle Grandmasters (users who have consistently demonstrated outstanding performance in one or more categories of expertise on Kaggle)

Excel users) to become “Citizen Data Scientists” (i.e. business analysts who want to advance their career by incorporating elements of data science, including predictive modeling, into their analyses) (Laurent, 2018). Furthermore, tools for automated machine learning are predicted to be one of the technologies that, according to Forrester Predictions 2017, will enable AI-driven companies to take \$1.2 trillion from competitors by 2020 (Forrester, 2016). As DataRobot was named the pioneer in automated machine learning on Artificial Intelligence 100 list by CB Insights[22] (CB Insights, 2018), this study sets out to investigate whether analysts, by using this platform to increase the accuracy of their earnings estimates, could get a piece of the cake.

From the literature review it follows that no one model is best for a set of problems, and thus, it is argued that choosing the best model in advance is not possible. A neural network, a decision tree, or an ensemble model could have been chosen for predicting whether analysts would over- or underestimate EPS in the following quarter, just as many researchers choose specific models for their research. Parameters could have been optimized until this one selected model was performing as good as possible with the given data. However, spending numerous hours writing advanced codes to construct and optimize this single model, would leave no time for testing other models. Furthermore, it would not be certain that the selected model was in fact the one best suited for the predictive task. DataRobot allows the construction, testing, and optimization of several models in order to identify the one(s) outperforming the others.

Using DataRobot, 12 model constructions[23] will be performed (two directional models for each sector), and furthermore, five permutation tests for each model construction will be made (see Section 3.3.2.1 on Success criteria). The only variables included in the modeling are `text` (the textual content of the filings), `Submission_date`, and `Cons_Above_Act` or `Cons_Below_Act`. The date of submission is included in order to enable time-aware modeling (see Section XX on Time-aware modeling below), either `Cons_Above_Act` or `Cons_Below_Act` is used as target variable, the `text`–variable is used by the models to provide predictions.

[22] CB Insights annually rank the 100 most promising private artificial intelligence companies in the world that will revolutionize the industries from drug discovery and cybersecurity to robotics and legal tech. DataRobot was chosen as one of these top 100 companies among the 2,000+ that CB Insights considered

[23] A model construction is the process of constructing several types of models based on a single data sample uploaded in DataRobot

3.2.2 Time-Aware Modeling

Because the economy evolves over time and financial market conditions change, equity analysts continuously obtain new knowledge. When the content fed to a model contains future reports of a company, there is a risk that value is added to certain characteristics of the text of that company. This can be illustrated using an example: Apple launched the first iPhone in year 2007. Assuming that analysts have been unable to foresee the massive success that the iPhone ended up becoming, and that they have therefore continuously underestimated the EPS of Apple since the launch of the product, using future reports that extensively use the word "iPhone", could possibly enable the model to make predictions that EPS will be underestimated when the word "iPhone" is used in a present report. Reality is, though, that no future reports and observations of analyst accuracy, based on these reports, are available in the present. Thus, applying methods, that do not take time into consideration, entails the risk that findings in terms of model accuracy might be over- or understated. Thus, by incorporating the date of submission of the filings as a time factor in the data sample, the risk of target leakage is reduced, as doing so will enable the software to read the reports in chronological order. The time-aware modeling will ensure that validation and holdout sets consist of future observations that are not used for training the models. This type of validation is oftentimes referred to as out-of-time validation (OTV).

While Kraus and Feuerriegel (2017) similarly use time-aware modeling, it is not common practice in automated textual analysis to take time into consideration. Instead, training and holdout sets are constructed from the entire data sample (typically randomly assigning 70% of the observations to the training set and 30% of them to the holdout set as done by e.g. Delen et al. (2013)), without taking time into consideration. Thereby, two samples are obtained; one sample (training) that can be used for constructing the model and one sample (holdout) that is withheld from the training process. After constructing a model, its performance on the holdout set can be tested to simulate how it would perform on new, unknown data. In order to construct the best possible models, it is furthermore common to construct one or several validation sets from the training set. Doing so enables the user to perform several tests and get an average test error rate before unlocking the hold out set. Thereby, an estimate for the test error rate that will result from testing on the holdout set is obtained, but as opposed to the holdout set, validation sets consist of observations that are also used for training the model at some point. A common approach for cross-validation is leave-one-out cross-validation (used by e.g. Cecchini et al., (2010))[24].

Assigning observations to training and holdout sets independent of any time frames is suitable when constructing models on data that is not time sensitive, but as the intention of this study is to construct models that are able to predict future events based on past data, it is argued that time-aware modeling is necessary.

Time-aware modeling is, in some ways, similar to the common approach described in the above, as the data sample is split into a holdout set and a training set and as several tests are performed on different validation sets (this procedure is known as backtesting in time-aware modeling), but certain characteristics differ: 1) Whereas the holdout set consists of random observations when using the common approach, it only consists of the most recent observations when applying time-aware modeling, 2) Similarly, the training data must consist of all observations up until the holdout data, 3) A rolling time frame is defined, such that validation is performed chronologically (thereby, no observations, occurring subsequent to the prediction, are used when making the prediction).

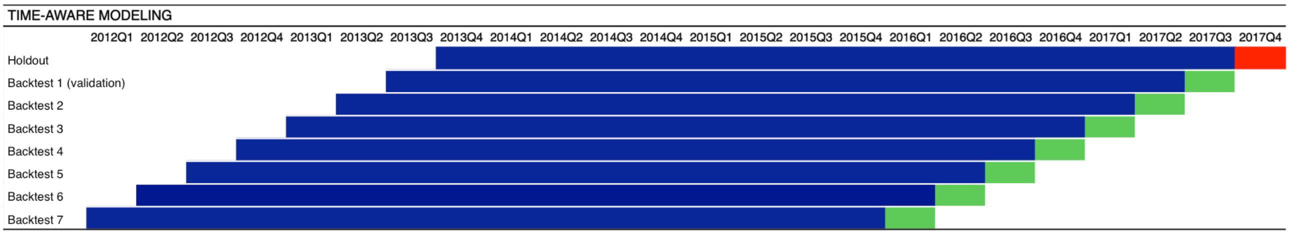


Figure 3.7: Time-aware modeling

The sample in this study consists of six years of data (24 quarters). The observations from the most recent quarter (2017 Q4) are assigned to the hold out set. The rest of the observations are assigned to the training data. In this study, the time-aware modeling is constructed so that the

[24] Leave-one-out cross-validation (LOOCV) is similar to a regular validation set approach, but the approaches differ in the size and number of validation sets. A regular validation set approach splits the training data in two samples of comparable size (a training set and a validation set) and tests the model on the validation set in order to obtain an estimate of the test error rate that will result when applying the model on the holdout data. When applying LOOCV only a single observation is used as the validation set, and the remaining observations are used as training set. This procedure can be repeated n times (when n is the number of observations). The average of all the test error rates constitute the LOOCV estimate for the test error rate (James et al., 2013, pp. 178–179).

model is trained on the observations of the most recent 16 quarters in order to provide predictions for the following quarter. By choosing this split, seven backtests (BTs) can be performed while one quarter (the most recent) is left out as a holdout set for final testing (see illustration in Figure 3.7). Branson et al. (1995) similarly use eight quarters for validation and holdout sets. When choosing the most suitable split, three criteria were considered: 1) The training set needed to consist of a fair amount of quarters to construct well-performing models, 2) On the other hand, in order to avoid noise from outdated observations, the time frame of the training data needed to be limited, and 3) A certain amount of BTs were necessary in order to obtain enough knowledge about the models to draw conclusions in regards to their performance.

When performing time-aware modeling in DataRobot, the platform constructs several types of models (different models with different predetermined feature engineering), which are all trained on the chosen optimization metric. In this study, Logarithmic Loss (Log Loss) is chosen as the optimization metric (see Section 3.2.3 for more on Log Loss). When performing backtesting, a model of each model type is constructed for each of the BTs and the Log Loss for each is computed. An average Log Loss for all of these BTs can then be obtained. The models will be ranked and selected based on their average performance in BTs, and after deciding which models are the best, the holdout sample will be unlocked. The holdout should only be regarded as a test of how the model would work one period ahead, and it should never lead to any changes in regards to model selection.

3.2.3 Optimization Metric: Logarithmic Loss (Log Loss)

Log Loss is a commonly used optimization metric and performance measure for evaluating predictive classification models (used by e.g. Cesa-Bianchi et al. (1997)). It provides a measure of the difference between the predicted probability of a specific event assigned by the model and the actual outcome. Mathematically, the Log Loss for a binary classifier is given by

$$Log\ Loss = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log p_i + (1-y_i)\log(1-p_i)\ ]$$

Where N is the number of observations, y_i is a binary variable, indicating whether the classification of observation i is correct, and p_i is the classification of observation i given by the model.

As opposed to an accuracy measure (see Section 3.3.1.1 on Confusion Matrix for more on Accuracy), Log Loss penalizes classification errors to a higher degree when a model's level of confidence in a wrong prediction is high (Cesa-Bianchi et al., 1997). Log Loss is perceived to be superior as a model evaluation measure and as an optimization metric. Based on the structure of the data and the binary target variables, it is the metric recommended for the models by DataRobot (DataRobot, 2018a). As Log Loss is a soft measure, there is no general rule distinguishing a good score from a bad score, but since it is a loss measure, a perfect classifier would have a Log Loss of 0 (Cesa-Bianchi et al., 1997), and the lower a Log Loss is, the better the model is. However, the possible difficulty in interpretation that follows from being such a soft measure is considered to be the major disadvantage of using Log Loss as an evaluation measure. Whether a score is good, simply depends on the complexity of the problem.

In this study, Log Loss is applied in two ways: 1) As the optimization metric (thus, models are trained to minimize Log Loss) and 2) As a relative evaluation metric (see Section 3.3.2.1 on Success Criteria).

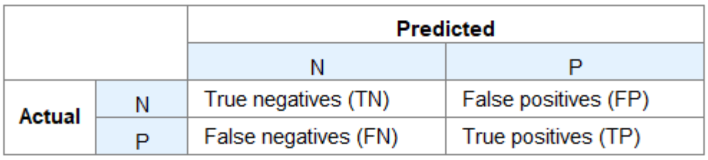
3.3 Analysis

The following sections will elaborate on the different measures used for evaluating the constructed models, and they will be followed by sections describing the framework of the conducted analysis and the methodology applied for testing the robustness of the results.

3.3.1 Model Evaluation

3.3.1.1 Confusion Matrix

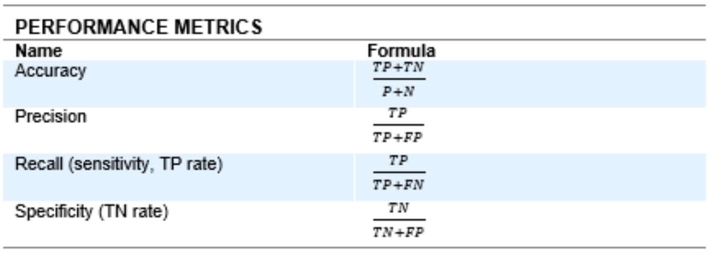
A confusion matrix operates on two axes; 1) Predictions and 2) Actual outcomes. The matrix thereby illustrates the accuracy of the model predictions and provides the number of two types of errors: False positives (also known as type I errors) and false negatives (also known as type II errors). While a false positive error refers to the case in which an observation is predicted to be positive but the actual outcome is in fact negative, a false negative error refer to the case in which an observation is predicted to be negative but the actual outcome is positive.



		Predicted	
		N	P
Actual	N	True negatives (TN)	False positives (FP)
	P	False negatives (FN)	True positives (TP)

Figure 3.8: Confusion matrix

A set of performance metrics based on the confusion matrix is commonly calculated when evaluating a model (James et al., 2013, p. 145). These measures consist of accuracy, precision, recall, and specificity, and as they will similarly implicitly be used for model evaluation in this study, they will be described in the following. Each metric and the formula it is given by can be seen in Table 3.3 below.



PERFORMANCE METRICS

Name	Formula
Accuracy	$\frac{TP+TN}{P+N}$
Precision	$\frac{TP}{TP+FP}$
Recall (sensitivity, TP rate)	$\frac{TP}{TP+FN}$
Specificity (TN rate)	$\frac{TN}{TN+FP}$

Table 3.3: Performance metrics

Accuracy is the percentage of correctly classified outcomes out of all predictions. Whether an accuracy score is good or bad, depends on the probabilities of the specific outcomes. An accuracy score that exceeds 50% is good when predicting the outcome of a coin flip, but if e.g. 70% of the observations in a sector subsample are cases of underestimations of EPS and a constructed model is only able to predict the correct outcome in 70% of all cases (or less), equally good (or better) accuracy could have been obtained by simply assigning all observations to the majority class (also referred to as a majority classifier). If, on the other hand, 75% of the observations are predicted correctly, the model is able to correctly predict some cases of overestimated or perfect forecasts too, which means that the model performs better than both random and a majority classifier.

Thus, there is no general threshold distinguishing good accuracy scores from bad ones across all predictive tasks (Das, 2014).

The drawback of placing too much emphasis on the accuracy score is the risk of ignoring the different costs of false predictions (i.e. false negatives and false positives). In many cases, such costs are asymmetric, meaning that the losses resulting from false negative and false positive predictions differ. Thus, it might be preferable to reduce the number of the more costly kind of false predictions at the expense of some model accuracy. This can be illustrated by observing costs in bankruptcy prediction: the costs of not foreseeing a bankruptcy (a false negative) are larger than the returns lost from not investing in a company that was predicted to bankrupt, but ended up not bankrupting (a false positive). Thus, while accuracy can be an important evaluation metric that should be taken into account when evaluating predictive models, it should not be the only measure that is examined, and therefore, the following measures should be taken into account too. Precision is the percentage of correctly classified positives out all predicted positives (James et al., 2013, p. 149). Recall (sometimes referred to as sensitivity or TP rate) is a measure of the percentage of correctly classified positives of all actual positives (Ibid.). Specificity (sometimes referred to as TN rate) is the percentage of correctly classified negatives of all actual negatives.

Due to an easily interpretable output, confusion matrices are considered helpful tools when optimizing predictive models to fit specific requirements. When classifying observations, a model effectively assigns probabilities (from 0-1) that the event assigned as class 1 will occur. Thus, a threshold for the minimum probability that must be assigned to an observation for it to be classified as positive (1) must be determined. If a model should only assign positive predictions in cases where the probability of the occurrence of (in this case) an underestimation is above 90%, the threshold must be set at 0.9. While setting a threshold manually is possible (James et al., 2013, p. 147), a neutral approach is to set the threshold so that the F_1-score (sometimes referred to the as F-score or F-measure) is maximized, as this score takes into account both the precision rate and the recall rate (Hajek and Henriques, 2017). The F_1-score is given by

$$F_1 - score = 2 * \frac{precision * recall}{precision + recall}$$

When a model has been constructed, neither its accuracy nor any of the measures above are locked, as all of these measures depend on the threshold that has been chosen. When choosing the optimal threshold, one should consider the precision-recall tradeoff, which is illustrated in the expression for F_1-score. When optimizing a model, one can choose to emphasize either precision or recall, depending on the aim of the model. Increasing the threshold, in order to only classify observations as positive when the model is really confident that they are in fact positive, is likely to increase the precision rate. On the other hand, the recall rate is likely to decrease, as some observations that were in fact positive, will no longer be classified as positive, since their predicted probability did not exceed the threshold. Consider again the bankruptcy example: if the costs of investing in a company that ends up going bankrupt are larger than the returns lost from not investing in a company that was predicted to go bankrupt, but that did not go bankrupt, one might want to minimize the recall rate and avoid false negative predictions since these are more costly.

However, the asymmetry in the costs of the two types of false predictions is not always clear. When predicting whether a consensus estimate will result in an underestimation (overestimation) of EPS, it is argued that the costs to an analyst differ: a false negative occurs when it is predicted that consensus will not underestimate (overestimate) EPS, and the analyst therefore maintains his estimate, but consensus turns out to be underestimating (overestimating) EPS. In such a case, the analyst will not have decreased his accuracy based – it will be the same, as if had the model not been constructed. If, on the other hand, an analyst shifts his forecast upwards (downwards) based on a prediction stating that consensus will underestimate (overestimate) EPS, and the prediction turns out to be wrong, the analyst will have provided a more inaccurate estimate than he would have, had he not acted according to the predictions of the model. Thus, false positives are more costly to analysts than false negatives.

3.3.1.2 Receiver Operating Characteristics (ROC) Curves and Area Under the Curve (AUC)

The name of the "receiver operating characteristics" (ROC) curve stems from the original purpose of the methodology, when developed under World War II, which was to evaluate the performance of radars by measuring whether the classification of a blip on the screen (as either an enemy or as noise) was correct (Fan et al., 2006). A ROC curve depicts the trade-off between the recall rate $\left(\frac{TP}{TP+FN}\right)$ and 1-specificity $\left(1-\frac{TN}{TN+FP}\right)$.

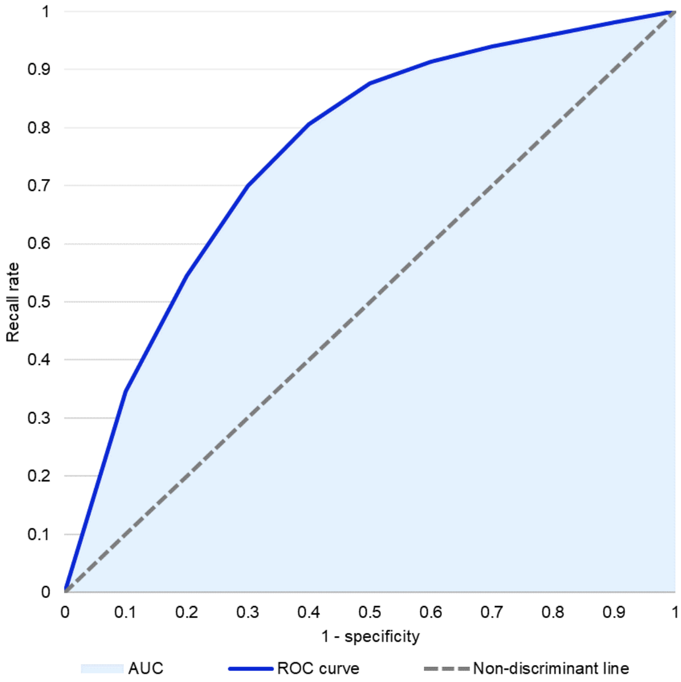


Figure 3.9: Illustration of ROC curve, AUC, and a non-discriminant line

The non-discriminant line represents a line of random guesses (Jones, 2017). As a ROC curve adjusts for an unequal number of positives and negatives, it is considered more appropriate to use the area under the curve (AUC) rather than accuracy to evaluate a model (Hajek, 2017; James et al., 2013, p. 147). The non-discriminant line has an AUC of 0.5, and a model having the same AUC value does no better than random guessing. Thus, 0.5 is the least desired AUC score for a model. A model with an AUC score of 1 is considered to have a perfect AUC score. While a model with an AUC score of 0 is completely incorrect and all of its predictions are completely wrong, the exact opposite of the given predictions must be right, and thus, the model is just as useful as a model with an AUC of 1 (James et al., 2013, p. 147). In general, there is no threshold for when an AUC score is considered to be good, as it depends on the difficulty of the predictions. Whiting et al. (2012, p. 511) are among the few, who use the absolute level of the AUC, and they refer to a general rule of thumb when examining AUC scores: "*As a rough rule of thumb, $AUC > 0.9$ indicates excellent test accuracy, while $0.8 < AUC < 0.9$ indicates good test accuracy; AUC values* *in the 0.7 range are generally considered fair*". Oftentimes, researchers use AUC as a comparative measure, trying to maximize the AUC scores of their models (e.g. Cecchini et al., 2010; du Jardin, 2017; Hajek, 2017; Weng et al., 2017). AUC is used by researchers mainly due to its strengths when working with imbalanced datasets with uneven sizes of groups as well as due to the possibilities to estimate pure model performance independent of the asymmetric costs of misclassifications (du Jardin, 2017). However, the latter can also constitute a model drawback: the ROC curves of different models might cross each other. This scenario could lead to misleading conclusions, as AUC scores do not take such asymmetric misclassification costs scores into consideration (Ibid.). Thus, solely relying on absolute AUC scores to choose among models should be avoided.

3.3.1.3 Lift Charts

In lift charts, predictions are sorted by their assigned probabilities and assigned to a specified number of bins. Each bin contains observations that are assigned based on the predicted probability of an event. Thus, if ten is the specified number of bins, the first bin on the left side of the curve (bin 1) will illustrate the average of the probabilities assigned to the 10% of the observations with the lowest assigned probabilities. A lift chart depicts two lines: one line with the average predicted probabilities for the specified number of bins, and another line with the actual average outcome for each bin. The distance between the lines is the inaccuracy (DataRobot, 2018).

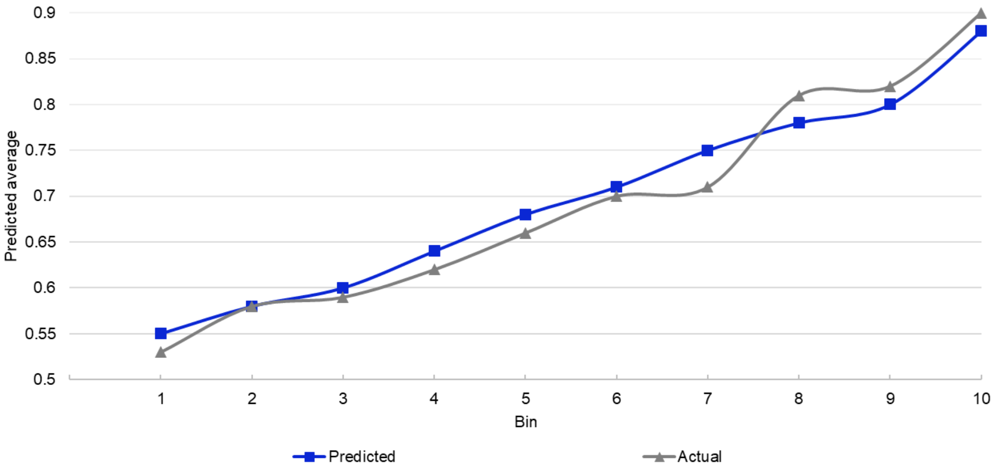


Figure 3.10: Lift chart

As both the actual average outcome and the average predicted probabilities is illustrated, lift charts can be used to evaluate different predictive models based on the objective constructing them. In some cases it might be that a model that performs well in predicting the tails correctly is more suitable for a given task (e.g. in stock pricing where the investable amount is a constraint or in loan default prediction where losses might be very costly). For other objectives, e.g. pricing decisions, it might be more important to be correct for predictions around the middle of the range. Thus, examining lift charts to evaluate models can result in the choice of a predictive model that does not have the lowest Log Loss or accuracy, but that simply fits the goal of the task better at the critical point.

3.3.2 Analysis Framework

The analysis is performed in conjunction with the aim of this study: to examine whether analysts, who are not data scientists, are able to enhance their EPS forecast accuracy by implementing automated textual analysis. For the sake of this analysis, it is assumed that an analyst, due to constraints in regards to both time and data science knowledge, will choose the model concluded to be best in terms of Log Loss (relative to the other models in the model construction). To support the aim of the study these models are examined based on a set of success criteria set out in Section 3.3.2.1 below, and the information obtained is used to determine whether an implementation of the models would lead to enhancements in accuracy.

Furthermore, each model construction section will include an examination of the word cloud output only available for the *Auto-Tuned Word N-Gram Text Modeler* when it is concluded that this model fulfills the criteria, C1 and C2 (see Section 3.3.2.1 below). Word clouds measure two dimensions of the use of words: 1) The color of a word in a word cloud determines its coefficient[25], and 2) The size of a word represents the frequency with which it appears (DataRobot, 2018b). As such, word clouds are graphics of the most relevant words, and they can thereby be used to identify and interpret the words that are deemed as indicative of the specific outcomes. Examinations of word clouds are made as it is argued that analysts might be able to benefit from

[25] The coefficient is a value between -1 and 1 indicating the strength of the negative/positive association that the word has with the targets variable, where -1 is a very negative association and 1 is a very positive association

their highly interpretable informational content even though the models yielding them are not necessarily the best performing models.

3.3.2.1 Success Criteria

In order to evaluate the performance of the highest ranked models, it is examined whether and to what extent they meet the following criteria:

The significance test

C1: The average AUC of all backtests does not fall within the AUC range for random tests (permutation test)

Mason and Graham (2002) apply a permutation test to determine the significance of AUC scores. This study uses a similar method to test whether a given model finds a stronger pattern than what it could have found on datasets with random combinations of the feature and target values. Five datasets with randomly shuffled target values are constructed for each sector subsample, and model constructions are performed for each of them. For each specific model type, the average AUC score for all BTs is compared to the range of the average AUC scores for the five datasets with randomly shuffled target variables (defined as the random range). If none of the models of this specific type, constructed on randomly shuffled data, yield better average BT AUC scores than the model in question, it is argued that the model has been able to identify a significant pattern.

The applicability test

C2: A maximum of one backtest with AUC $<$ 0.5

If more than one out of the seven BTs yield an AUC score below 0.5, it is argued that the individual BT AUC scores of the model are too volatile for the model to be implemented in practice. An average fulfilling C1 could be carried by superior performance in certain quarters and poor performance in others, and such model characteristics are not suitable for practical applications, as analysts are required to perform consistently well.

C3: Subjective evaluation

Based on the measures elaborated in Section 3.3.1 on Model Evaluation, additional evaluation of the performance of the model is conducted. Such an evaluation could include an assessment of the level of the average BT AUC score, of the relative performance of the model (both across and in individual backtests) to other models, and of the performance of the model when predicting the outcomes of holdout and validation observations (measured by accuracy, precision, recall, specificity, and visualized in a lift chart). However, the content of subjective evaluations will vary, and only the measures best supporting the analysis will be included.

3.3.3 Robustness

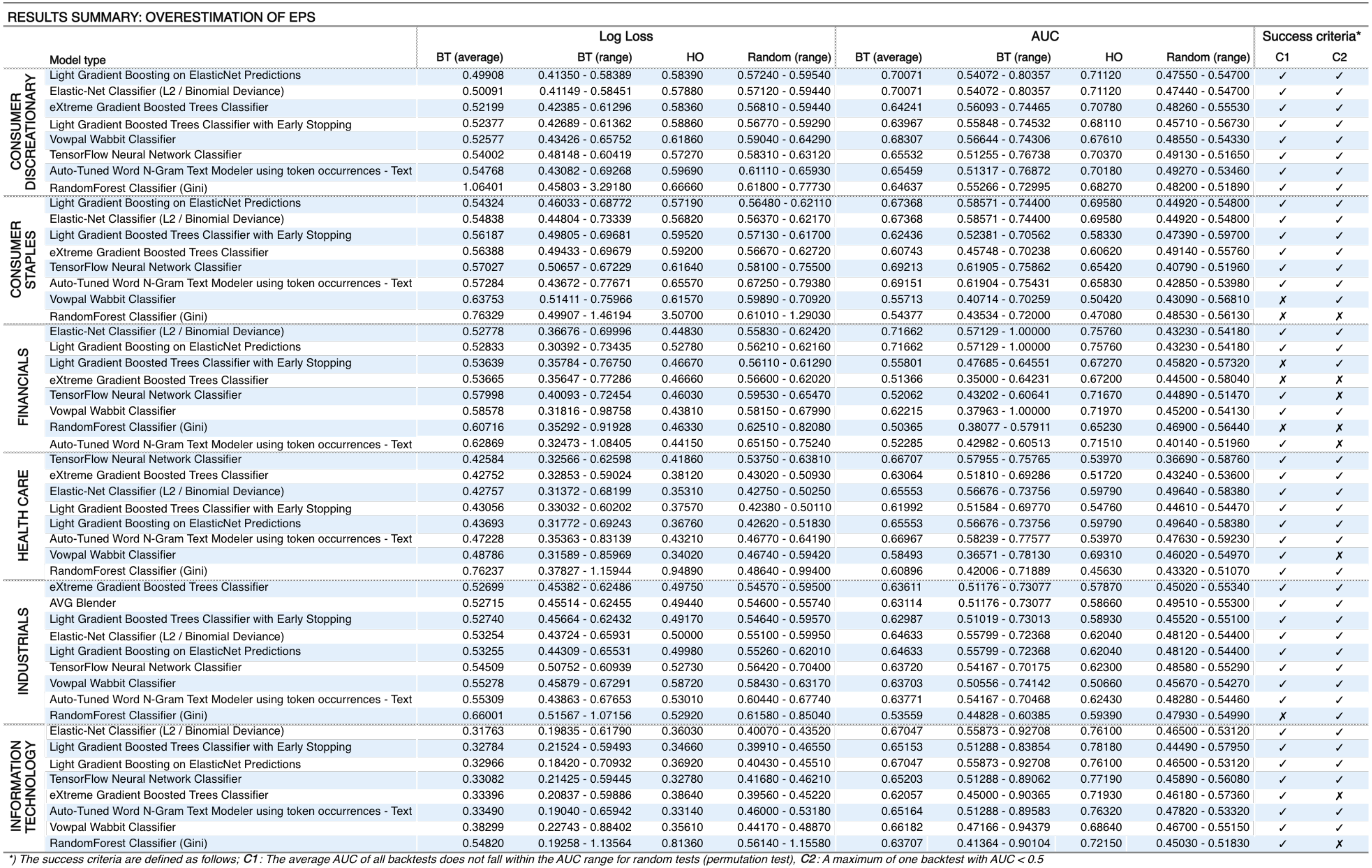
The robustness of the results obtained by applying the methodology described in the previous sections is tested across various measures. In order to examine whether the findings are carried by better predictions for companies that analysts grant less attention, and that they are thereby more likely to assign inaccurate forecasts, robustness in regards to both market capitalization and analysts coverage of the companies is tested (both used as proxies for company attention). To examine whether model predictions are carried by the amount of information available to the models, similar tests, in which the length of the reports is used as a proxy for the information available, are conducted. Lastly, the choices made in regards to the time aware modeling are tested by conducting model constructions where the training data is reduced from 16 quarters to 12 and eight quarters, respectively.

4 Results and Analysis

The Results and Analysis section commences with a summary of the overall results of the model constructions. This is followed by sector specific sections in which the highest ranked model in each model construction is assessed based on the criteria set out above. Furthermore, the word clouds of the constructed *Auto-Tuned Word N-Gram Text Modelers*, will be examined. The sector specific analyses are followed by an overall analysis in which key findings, in regards to whether the implementation of these models could enhance analyst forecast accuracy, will be presented. Findings indicating whether specific types of models outperform others and whether the predictive models perform better in specific scenarios or sectors, will similarly be presented.

4.1 Results Summary

Overall results for the model constructions are presented in Table 4.1 and Table 4.2 below. The tables provide the backtesting average, backtesting range, holdout score, and the range of average scores for random samples for Log Loss and AUC, respectively, for all models in each model construction. The two columns on the right indicate whether the models fulfill criteria C1 and C2. Table 4.3 concerns only the models ranked best in terms of average Log Loss within each model construction. It depicts their ability to meet all three criteria (C1, C2, and C3, which are described in Section 3.3.2.1). The additional evaluation measures, based on which it is decided whether they meet C3, will be presented in the sector specific sections when adding value to the given analysis. All results can be seen in Appendix XIII-Appendix XXIV. These appendices include Log Loss and AUC values and rankings, random ranges, holdout performance, confusion matrices for holdout and validation predictions, and lift charts.



RESULTS SUMMARY: OVERESTIMATION OF EPS

		Log Loss				AUC				Success criteria*	
	Model type	BT (average)	BT (range)	HO	Random (range)	BT (average)	BT (range)	HO	Random (range)	C1	C2
CONSUMER DISCREATIONARY	Light Gradient Boosting on ElasticNet Predictions	0.49908	0.41350 - 0.58389	0.58390	0.57240 - 0.59540	0.70071	0.54072 - 0.80357	0.71120	0.47550 - 0.54700	✓	✓
	Elastic-Net Classifier (L2 / Binomial Deviance)	0.50091	0.41149 - 0.58451	0.57880	0.57120 - 0.59440	0.70071	0.54072 - 0.80357	0.71120	0.47440 - 0.54700	✓	✓
	eXtreme Gradient Boosted Trees Classifier	0.52199	0.42385 - 0.61296	0.58360	0.56810 - 0.59440	0.64241	0.56093 - 0.74465	0.70780	0.48260 - 0.55530	✓	✓
	Light Gradient Boosted Trees Classifier with Early Stopping	0.52377	0.42689 - 0.61362	0.58860	0.56770 - 0.59290	0.63967	0.55848 - 0.74532	0.68110	0.45710 - 0.56730	✓	✓
	Vowpal Wabbit Classifier	0.52577	0.43426 - 0.65752	0.61860	0.59040 - 0.64290	0.68307	0.56644 - 0.74306	0.67610	0.48550 - 0.54330	✓	✓
	TensorFlow Neural Network Classifier	0.54002	0.48148 - 0.60419	0.57270	0.58310 - 0.63120	0.65532	0.51255 - 0.76738	0.70370	0.49130 - 0.51650	✓	✓
	Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.54768	0.43082 - 0.69268	0.59690	0.61110 - 0.65930	0.65459	0.51317 - 0.76872	0.70180	0.49270 - 0.53460	✓	✓
	RandomForest Classifier (Gini)	1.06401	0.45803 - 3.29180	0.66660	0.61800 - 0.77730	0.64637	0.55266 - 0.72995	0.68270	0.48200 - 0.51890	✓	✓
CONSUMER STAPLES	Light Gradient Boosting on ElasticNet Predictions	0.54324	0.46033 - 0.68772	0.57190	0.56480 - 0.62110	0.67368	0.58571 - 0.74400	0.69580	0.44920 - 0.54800	✓	✓
	Elastic-Net Classifier (L2 / Binomial Deviance)	0.54838	0.44804 - 0.73339	0.56820	0.56370 - 0.62170	0.67368	0.58571 - 0.74400	0.69580	0.44920 - 0.54800	✓	✓
	Light Gradient Boosted Trees Classifier with Early Stopping	0.56187	0.49805 - 0.69681	0.59520	0.57130 - 0.61700	0.62436	0.52381 - 0.70562	0.58330	0.47390 - 0.59700	✓	✓
	eXtreme Gradient Boosted Trees Classifier	0.56388	0.49433 - 0.69679	0.59200	0.56670 - 0.62720	0.60743	0.45748 - 0.70238	0.60620	0.49140 - 0.55760	✓	✓
	TensorFlow Neural Network Classifier	0.57027	0.50657 - 0.67229	0.61640	0.58100 - 0.75500	0.69213	0.61905 - 0.75862	0.65420	0.40790 - 0.51960	✓	✓
	Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.57284	0.43672 - 0.77671	0.65570	0.67250 - 0.79380	0.69151	0.61904 - 0.75431	0.65830	0.42850 - 0.53980	✓	✓
	Vowpal Wabbit Classifier	0.63753	0.51411 - 0.75966	0.61570	0.59890 - 0.70920	0.55713	0.40714 - 0.70259	0.50420	0.43090 - 0.56810	✗	✓
	RandomForest Classifier (Gini)	0.76329	0.49907 - 1.46194	3.50700	0.61010 - 1.29030	0.54377	0.43534 - 0.72000	0.47080	0.48530 - 0.56130	✗	✗
FINANCIALS	Elastic-Net Classifier (L2 / Binomial Deviance)	0.52778	0.36676 - 0.69996	0.44830	0.55830 - 0.62420	0.71662	0.57129 - 1.00000	0.75760	0.43230 - 0.54180	✓	✓
	Light Gradient Boosting on ElasticNet Predictions	0.52833	0.30392 - 0.73435	0.52780	0.56210 - 0.62160	0.71662	0.57129 - 1.00000	0.75760	0.43230 - 0.54180	✓	✓
	Light Gradient Boosted Trees Classifier with Early Stopping	0.53639	0.35784 - 0.76750	0.46670	0.56110 - 0.61290	0.55801	0.47685 - 0.64551	0.67270	0.45820 - 0.57320	✗	✓
	eXtreme Gradient Boosted Trees Classifier	0.53665	0.35647 - 0.77286	0.46660	0.56600 - 0.62020	0.51366	0.35000 - 0.64231	0.67200	0.44500 - 0.58040	✗	✗
	TensorFlow Neural Network Classifier	0.57998	0.40093 - 0.72454	0.46030	0.59530 - 0.65470	0.52062	0.43202 - 0.60641	0.71670	0.44890 - 0.51470	✓	✗
	Vowpal Wabbit Classifier	0.58578	0.31816 - 0.98758	0.43810	0.58150 - 0.67990	0.62215	0.37963 - 1.00000	0.71970	0.45200 - 0.54130	✓	✓
	RandomForest Classifier (Gini)	0.60716	0.35292 - 0.91928	0.46330	0.62510 - 0.82080	0.50365	0.38077 - 0.57911	0.65230	0.46900 - 0.56440	✗	✗
	Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.62869	0.32473 - 1.08405	0.44150	0.65150 - 0.75240	0.52285	0.42982 - 0.60513	0.71510	0.40140 - 0.51960	✓	✗
HEALTH CARE	TensorFlow Neural Network Classifier	0.42584	0.32566 - 0.62598	0.41860	0.53750 - 0.63810	0.66707	0.57955 - 0.75765	0.53970	0.36690 - 0.58760	✓	✓
	eXtreme Gradient Boosted Trees Classifier	0.42752	0.32853 - 0.59024	0.38120	0.43020 - 0.50930	0.63064	0.51810 - 0.69286	0.51720	0.43240 - 0.53600	✓	✓
	Elastic-Net Classifier (L2 / Binomial Deviance)	0.42757	0.31372 - 0.68199	0.35310	0.42750 - 0.50250	0.65553	0.56676 - 0.73756	0.59790	0.49640 - 0.58380	✓	✓
	Light Gradient Boosted Trees Classifier with Early Stopping	0.43056	0.33032 - 0.60202	0.37570	0.42380 - 0.50110	0.61992	0.51584 - 0.69770	0.54760	0.44610 - 0.54470	✓	✓
	Light Gradient Boosting on ElasticNet Predictions	0.43693	0.31772 - 0.69243	0.36760	0.42620 - 0.51830	0.65553	0.56676 - 0.73756	0.59790	0.49640 - 0.58380	✓	✓
	Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.47228	0.35363 - 0.83139	0.43210	0.46770 - 0.64190	0.66967	0.58239 - 0.77577	0.53970	0.47630 - 0.59230	✓	✓
	Vowpal Wabbit Classifier	0.48786	0.31589 - 0.85969	0.34020	0.46740 - 0.59420	0.58493	0.36571 - 0.78130	0.69310	0.46020 - 0.54970	✓	✗
	RandomForest Classifier (Gini)	0.76237	0.37827 - 1.15944	0.94890	0.48640 - 0.99400	0.60896	0.42006 - 0.71889	0.45630	0.43320 - 0.51070	✓	✓
INDUSTRIALS	eXtreme Gradient Boosted Trees Classifier	0.52699	0.45382 - 0.62486	0.49750	0.54570 - 0.59500	0.63611	0.51176 - 0.73077	0.57870	0.45020 - 0.55340	✓	✓
	AVG Blender	0.52715	0.45514 - 0.62455	0.49440	0.54600 - 0.55740	0.63114	0.51176 - 0.73077	0.58660	0.49510 - 0.55300	✓	✓
	Light Gradient Boosted Trees Classifier with Early Stopping	0.52740	0.45664 - 0.62432	0.49170	0.54640 - 0.59570	0.62987	0.51019 - 0.73013	0.58930	0.45520 - 0.55100	✓	✓
	Elastic-Net Classifier (L2 / Binomial Deviance)	0.53254	0.43724 - 0.65931	0.50000	0.55100 - 0.59950	0.64633	0.55799 - 0.72368	0.62040	0.48120 - 0.54400	✓	✓
	Light Gradient Boosting on ElasticNet Predictions	0.53255	0.44309 - 0.65531	0.49980	0.55260 - 0.62010	0.64633	0.55799 - 0.72368	0.62040	0.48120 - 0.54400	✓	✓
	TensorFlow Neural Network Classifier	0.54509	0.50752 - 0.60939	0.52730	0.56420 - 0.70400	0.63720	0.54167 - 0.70175	0.62300	0.48580 - 0.55290	✓	✓
	Vowpal Wabbit Classifier	0.55278	0.45879 - 0.67291	0.58720	0.58430 - 0.63170	0.63703	0.50556 - 0.74142	0.50660	0.45670 - 0.54270	✓	✓
	Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.55309	0.43863 - 0.67653	0.53010	0.60440 - 0.67740	0.63771	0.54167 - 0.70468	0.62430	0.48280 - 0.54460	✓	✓
	RandomForest Classifier (Gini)	0.66001	0.51567 - 1.07156	0.52920	0.61580 - 0.85040	0.53559	0.44828 - 0.60385	0.59390	0.47930 - 0.54990	✗	✓
INFORMATION TECHNOLOGY	Elastic-Net Classifier (L2 / Binomial Deviance)	0.31763	0.19835 - 0.61790	0.36030	0.40070 - 0.43520	0.67047	0.55873 - 0.92708	0.76100	0.46500 - 0.53120	✓	✓
	Light Gradient Boosted Trees Classifier with Early Stopping	0.32784	0.21524 - 0.59493	0.34660	0.39910 - 0.46550	0.65153	0.51288 - 0.83854	0.78180	0.44490 - 0.57950	✓	✓
	Light Gradient Boosting on ElasticNet Predictions	0.32966	0.18420 - 0.70932	0.36920	0.40430 - 0.45510	0.67047	0.55873 - 0.92708	0.76100	0.46500 - 0.53120	✓	✓
	TensorFlow Neural Network Classifier	0.33082	0.21425 - 0.59445	0.32780	0.41680 - 0.46210	0.65203	0.51288 - 0.89062	0.77190	0.45890 - 0.56080	✓	✓
	eXtreme Gradient Boosted Trees Classifier	0.33396	0.20837 - 0.59886	0.38640	0.39560 - 0.45220	0.62057	0.45000 - 0.90365	0.71930	0.46180 - 0.57360	✓	✗
	Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.33490	0.19040 - 0.65942	0.33140	0.46000 - 0.53180	0.65164	0.51288 - 0.89583	0.76320	0.47820 - 0.53320	✓	✓
	Vowpal Wabbit Classifier	0.38299	0.22743 - 0.88402	0.35610	0.44170 - 0.48870	0.66182	0.47166 - 0.94379	0.68640	0.46700 - 0.55150	✓	✓
	RandomForest Classifier (Gini)	0.54820	0.19258 - 1.13564	0.81360	0.56140 - 1.15580	0.63707	0.41364 - 0.90104	0.72150	0.45030 - 0.51830	✓	✗

) The success criteria are defined as follows;* ***C1**: The average AUC of all backtests does not fall within the AUC range for random tests (permutation test),* ***C2****: A maximum of one backtest with AUC < 0.5*

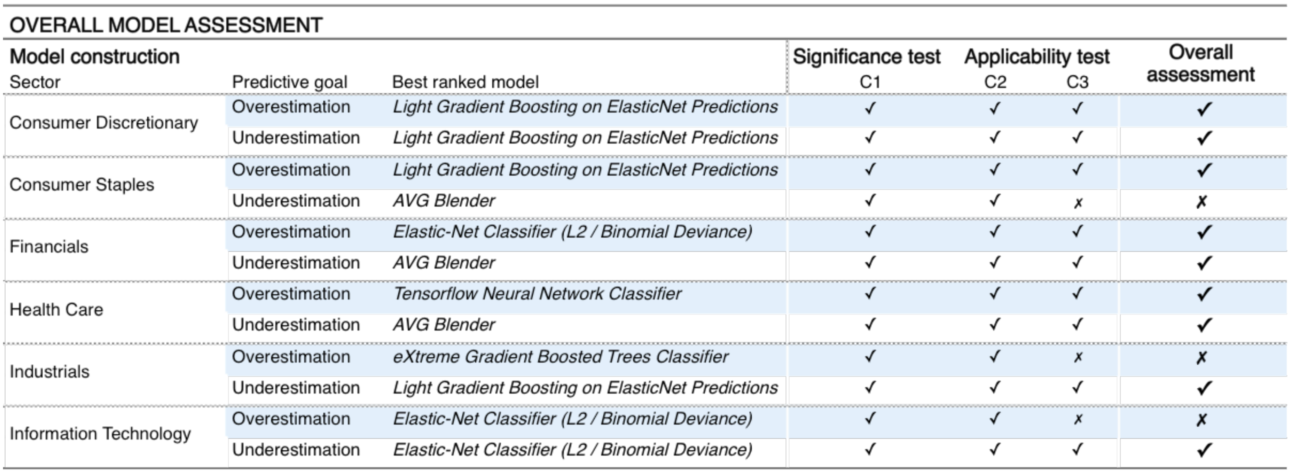
Table 4.1: Results summary - Overestimation of EPS

RESULTS SUMMARY: UNDERESTIMATION OF EPS

		Log Loss				AUC				Success criteria*	
	Model type	BT (average)	BT (range)	HO	Random (range)	BT (average)	BT (range)	HO	Random (range)	C2	C3
CONSUMER DISCREATIONARY	Light Gradient Boosting on ElasticNet Predictions	0.50981	0.41748 - 0.60499	0.58550	0.57680 - 0.59880	0.68530	0.51250 - 0.79241	0.70940	0.47340 - 0.54590	✓	✓
	Elastic-Net Classifier (L2 / Binomial Deviance)	0.51184	0.41488 - 0.60544	0.57500	0.57600 - 0.60000	0.68530	0.51250 - 0.79241	0.70940	0.47340 - 0.54590	✓	✓
	Light Gradient Boosted Trees Classifier with Early Stopping	0.52891	0.46643 - 0.59974	0.59030	0.58100 - 0.60030	0.63856	0.51131 - 0.77515	0.69620	0.47420 - 0.52820	✓	✓
	eXtreme Gradient Boosted Trees Classifier	0.53012	0.46610 - 0.60871	0.59080	0.58850 - 0.59920	0.63888	0.50625 - 0.78066	0.69620	0.48020 - 0.51050	✓	✓
	Vowpal Wabbit Classifier	0.54347	0.43517 - 0.71761	0.61620	0.59500 - 0.64800	0.68132	0.56131 - 0.74940	0.67360	0.49530 - 0.54030	✓	✓
	Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.55942	0.45240 - 0.71424	0.59270	0.61430 - 0.67180	0.63968	0.49464 - 0.77936	0.71250	0.47820 - 0.54250	✓	✓
	TensorFlow Neural Network Classifier	0.59486	0.41682 - 0.88201	0.69370	0.57700 - 0.62150	0.64042	0.49286 - 0.78585	0.71000	0.45840 - 0.53080	✓	✓
	RandomForest Classifier (Gini)	1.00221	0.46253 - 2.33231	0.94320	0.66180 - 0.69480	0.61188	0.50216 - 0.69565	0.70900	0.49300 - 0.54570	✓	✓
CONSUMER STAPLES	AVG Blender	0.53835	0.44741 - 0.68351	0.60280	0.59710 - 0.62170	0.67383	0.57857 - 0.73200	0.61250	0.49370 - 0.51410	✓	✓
	Light Gradient Boosting on ElasticNet Predictions	0.54162	0.45189 - 0.69149	0.57580	0.56480 - 0.62290	0.68226	0.57857 - 0.74400	0.68750	0.44570 - 0.54580	✓	✓
	eXtreme Gradient Boosted Trees Classifier	0.54558	0.45130 - 0.68464	0.64240	0.56670 - 0.62550	0.61469	0.52857 - 0.69800	0.47500	0.49140 - 0.54050	✓	✓
	Elastic-Net Classifier (L2 / Binomial Deviance)	0.54673	0.45025 - 0.72473	0.56890	0.56370 - 0.62100	0.68226	0.57857 - 0.74400	0.68750	0.44570 - 0.54580	✓	✓
	Light Gradient Boosted Trees Classifier with Early Stopping	0.56141	0.47745 - 0.69279	0.64770	0.57130 - 0.61960	0.56698	0.40357 - 0.68800	0.46670	0.45880 - 0.53550	✓	✗
	Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.57826	0.45473 - 0.77959	0.65150	0.67250 - 0.79150	0.69040	0.61224 - 0.76293	0.65830	0.43320 - 0.54210	✓	✓
	RandomForest Classifier (Gini)	0.58883	0.45203 - 0.75772	0.71830	0.61010 - 0.80050	0.55780	0.28698 - 0.74200	0.47080	0.48730 - 0.53750	✓	✗
	TensorFlow Neural Network Classifier	0.59372	0.43040 - 0.82550	0.59000	0.58100 - 0.65630	0.68377	0.60204 - 0.76786	0.62500	0.45800 - 0.57030	✓	✓
	Vowpal Wabbit Classifier	0.63775	0.51190 - 0.75797	0.61640	0.59890 - 0.70890	0.55681	0.39286 - 0.71121	0.50420	0.43040 - 0.58100	✗	✓
FINANCIALS	AVG Blender	0.51839	0.33057 - 0.70697	0.45950	0.56900 - 0.61000	0.71613	0.56941 - 1.00000	0.75910	0.44950 - 0.54150	✓	✓
	Light Gradient Boosting on ElasticNet Predictions	0.52435	0.29936 - 0.71344	0.50951	0.57660 - 0.63070	0.71613	0.56941 - 1.00000	0.75910	0.44810 - 0.56060	✓	✓
	Elastic-Net Classifier (L2/Binomial Deviance)	0.52909	0.36692 - 0.70093	0.45300	0.56920 - 0.63250	0.71613	0.56941 - 1.00000	0.75910	0.44940 - 0.56060	✓	✓
	Light Gradient Boosted Trees Classifier with Early Stopping	0.53223	0.35522 - 0.74128	0.46450	0.57070 - 0.62080	0.56862	0.49671 - 0.80000	0.69320	0.44660 - 0.51590	✓	✓
	eXtreme Gradient Boosted Trees Classifier	0.53251	0.35702 - 0.74073	0.46570	0.57450 - 0.62130	0.56487	0.47368 - 0.80000	0.67650	0.47260 - 0.50720	✓	✓
	TensorFlow Neural Network Classifier	0.57575	0.46951 - 0.74096	0.53240	0.57720 - 0.63170	0.53644	0.38462 - 0.80000	0.64850	0.44420 - 0.57890	✗	✗
	Vowpal Wabbit Classifier	0.58672	0.33120 - 0.96721	0.44020	0.59200 - 0.68570	0.62377	0.37963 - 1.00000	0.70300	0.43680 - 0.54150	✓	✓
	Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.61082	0.28463 - 1.02130	0.44400	0.65340 - 0.76100	0.57743	0.47588 - 0.80000	0.70910	0.41760 - 0.51720	✓	✓
	RandomForest Classifier (Gini)	0.63529	0.36116 - 1.15531	0.53820	0.60240 - 0.89510	0.53012	0.27500 - 0.60111	0.52420	0.42370 - 0.55810	✗	✓
HEALTH CARE	AVG Blender	0.42620	0.31610 - 0.63627	0.36170	0.42980 - 0.50230	0.66191	0.58807 - 0.71342	0.57940	0.45390 - 0.59310	✓	✓
	Light Gradient Boosted Trees Classifier with Early Stopping	0.43005	0.32691 - 0.60852	0.37300	0.42970 - 0.52900	0.64542	0.58446 - 0.69111	0.54230	0.44960 - 0.57720	✓	✓
	eXtreme Gradient Boosted Trees Classifier	0.43053	0.32862 - 0.61095	0.37280	0.43060 - 0.50420	0.63595	0.58239 - 0.69351	0.53440	0.45230 - 0.55860	✓	✓
	Elastic-Net Classifier (L2 / Binomial Deviance)	0.43198	0.31517 - 0.67406	0.35360	0.43620 - 0.50280	0.64953	0.56735 - 0.73152	0.60050	0.48890 - 0.58850	✓	✓
	Light Gradient Boosting on ElasticNet Predictions	0.44140	0.32289 - 0.68862	0.36770	0.43350 - 0.52450	0.64953	0.56735 - 0.73152	0.60050	0.48890 - 0.58850	✓	✓
	TensorFlow Neural Network Classifier	0.46726	0.32246 - 0.73208	0.36100	0.46630 - 0.50480	0.60838	0.35143 - 0.74661	0.55290	0.46140 - 0.59910	✓	✓
	Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.47681	0.35739 - 0.79216	0.43710	0.48300 - 0.65270	0.65943	0.60408 - 0.75264	0.52650	0.47040 - 0.58960	✓	✓
	Vowpal Wabbit Classifier	0.51014	0.31471 - 0.97076	0.34100	0.46810 - 0.60720	0.57378	0.42571 - 0.67572	0.69310	0.46340 - 0.55210	✓	✓
	RandomForest Classifier (Gini)	0.62556	0.37468 - 0.99693	0.89470	0.71110 - 2.57890	0.58675	0.43608 - 0.70553	0.54500	0.47810 - 0.56090	✓	✗
INDUSTRIALS	Light Gradient Boosting on ElasticNet Predictions	0.53052	0.43696 - 0.64611	0.50000	0.55710 - 0.62140	0.65305	0.55799 - 0.72807	0.61770	0.46830 - 0.54090	✓	✓
	AVG Blender	0.53083	0.43521 - 0.64925	0.50010	0.55610 - 0.58450	0.65305	0.55799 - 0.72807	0.61770	0.44950 - 0.53240	✓	✓
	Elastic-Net Classifier (L2 / Binomial Deviance)	0.53128	0.43352 - 0.65261	0.50020	0.55580 - 0.60840	0.65305	0.55799 - 0.72807	0.61770	0.46830 - 0.54090	✓	✓
	Light Gradient Boosted Trees Classifier with Early Stopping	0.53166	0.44763 - 0.63186	0.48920	0.55760 - 0.60000	0.65172	0.55347 - 0.70175	0.63160	0.48410 - 0.55140	✓	✓
	eXtreme Gradient Boosted Trees Classifier	0.53288	0.44708 - 0.62816	0.49320	0.56080 - 0.59830	0.64010	0.55278 - 0.70249	0.62100	0.48410 - 0.55280	✓	✓
	Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.54903	0.43889 - 0.66572	0.52880	0.61010 - 0.67550	0.64681	0.54444 - 0.70614	0.62570	0.47740 - 0.54270	✓	✓
	Vowpal Wabbit Classifier	0.56695	0.50128 - 0.67051	0.59150	0.58780 - 0.62670	0.64457	0.51667 - 0.77176	0.52120	0.45620 - 0.56020	✓	✓
	TensorFlow Neural Network Classifier	0.56945	0.42534 - 0.76992	0.53670	0.55950 - 0.61430	0.64442	0.53611 - 0.70204	0.62700	0.51180 - 0.55100	✓	✓
	RandomForest Classifier (Gini)	0.63041	0.47632 - 1.09074	0.53200	0.64710 - 0.87970	0.57804	0.49514 - 0.69038	0.61240	0.46270 - 0.51300	✓	✗
INFORMATION TECHNOLOGY	Elastic-Net Classifier (L2 / Binomial Deviance)	0.32841	0.23604 - 0.64385	0.40110	0.42410 - 0.49230	0.66134	0.55556 - 0.86979	0.72420	0.46630 - 0.56670	✓	✓
	Light Gradient Boosted Trees Classifier with Early Stopping	0.34191	0.25375 - 0.61550	0.38140	0.42250 - 0.46040	0.60161	0.37002 - 0.86979	0.70140	0.47090 - 0.51710	✓	✓
	eXtreme Gradient Boosted Trees Classifier	0.34205	0.25199 - 0.62068	0.38280	0.42280 - 0.46210	0.61228	0.46956 - 0.86979	0.70640	0.44590 - 0.56460	✓	✓
	Light Gradient Boosting on ElasticNet Predictions	0.34546	0.22478 - 0.75167	0.41290	0.42790 - 0.49280	0.66134	0.55556 - 0.86979	0.72420	0.46630 - 0.53150	✓	✓
	TensorFlow Neural Network Classifier	0.34915	0.16785 - 0.77951	0.47060	0.42480 - 0.48530	0.64275	0.50351 - 0.89063	0.75600	0.42590 - 0.51790	✓	✓
	Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.34951	0.19792 - 0.70381	0.38780	0.47110 - 0.63500	0.64453	0.50351 - 0.88021	0.70830	0.47270 - 0.53090	✓	✓
	Vowpal Wabbit Classifier	0.40539	0.22784 - 0.96678	0.38520	0.48970 - 0.52160	0.65520	0.44189 - 0.94145	0.66860	0.47220 - 0.54310	✓	✓
	RandomForest Classifier (Gini)	1.31224	0.25855 - 3.11317	0.83600	0.62650 - 0.99080	0.55028	0.35363 - 0.76563	0.73110	0.43110 - 0.57150	✗	✗

*) The success criteria are defined as follows; ***C1***: The average AUC of all backtests does not fall within the AUC range for random tests (permutation test), ***C2***: A maximum of one backtest with AUC < 0.5

Table 4.2: Results summary - Underestimation of EPS



OVERALL MODEL ASSESSMENT

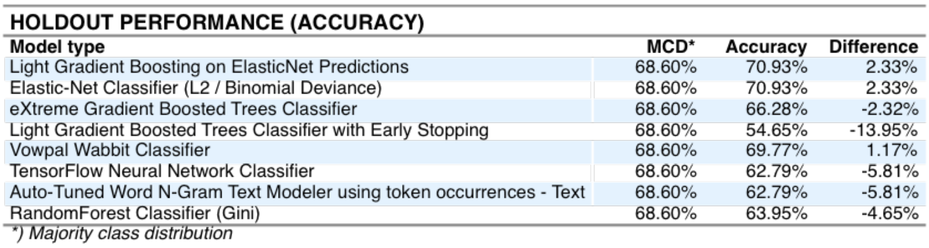
Model construction			Significance test	Applicability test		Overall assessment
Sector	Predictive goal	Best ranked model	C1	C2	C3	
Consumer Discretionary	Overestimation	*Light Gradient Boosting on ElasticNet Predictions*	✓	✓	✓	✓
	Underestimation	*Light Gradient Boosting on ElasticNet Predictions*	✓	✓	✓	✓
Consumer Staples	Overestimation	*Light Gradient Boosting on ElasticNet Predictions*	✓	✓	✓	✓
	Underestimation	*AVG Blender*	✓	✓	✗	✗
Financials	Overestimation	*Elastic-Net Classifier (L2 / Binomial Deviance)*	✓	✓	✓	✓
	Underestimation	*AVG Blender*	✓	✓	✓	✓
Health Care	Overestimation	*Tensorflow Neural Network Classifier*	✓	✓	✓	✓
	Underestimation	*AVG Blender*	✓	✓	✓	✓
Industrials	Overestimation	*eXtreme Gradient Boosted Trees Classifier*	✓	✓	✗	✗
	Underestimation	*Light Gradient Boosting on ElasticNet Predictions*	✓	✓	✓	✓
Information Technology	Overestimation	*Elastic-Net Classifier (L2 / Binomial Deviance)*	✓	✓	✗	✗
	Underestimation	*Elastic-Net Classifier (L2 / Binomial Deviance)*	✓	✓	✓	✓

Table 4.3: Overall assessment of all models ranked as number one on cumulative average BT Log Loss

4.2 Consumer Discretionary

4.2.1 Overestimation of EPS

All eight models, built with the aim of predicting cases in which consensus overestimates EPS in the following quarter, fulfill criteria C1 and C2 (see Table 4.1). However, when observing the accuracy scores obtained by the models when predicting the outcomes of the holdout observations (see Table 4.4), it appears that only three models obtain accuracy scores that exceed the score that could be obtained by simply assigning all predictions to the majority class (a model doing so is also referred to as a majority classifier).



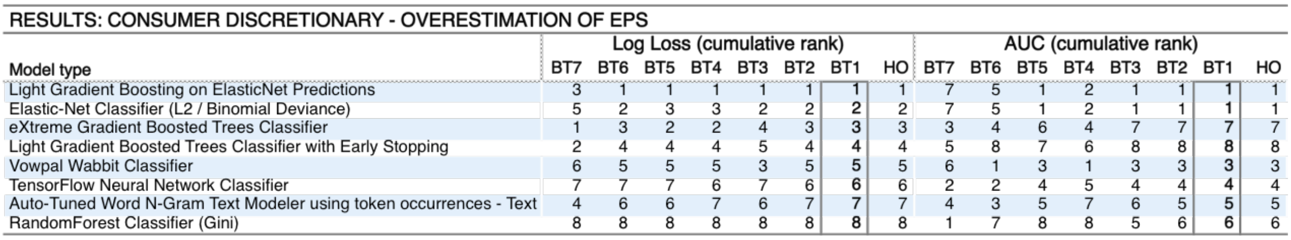
HOLDOUT PERFORMANCE (ACCURACY)

Model type	MCD*	Accuracy	Difference
Light Gradient Boosting on ElasticNet Predictions	68.60%	70.93%	2.33%
Elastic-Net Classifier (L2 / Binomial Deviance)	68.60%	70.93%	2.33%
eXtreme Gradient Boosted Trees Classifier	68.60%	66.28%	-2.32%
Light Gradient Boosted Trees Classifier with Early Stopping	68.60%	54.65%	-13.95%
Vowpal Wabbit Classifier	68.60%	69.77%	1.17%
TensorFlow Neural Network Classifier	68.60%	62.79%	-5.81%
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	68.60%	62.79%	-5.81%
RandomForest Classifier (Gini)	68.60%	63.95%	-4.65%

**) Majority class distribution*

Table 4.4: Holdout performance (Consumer Discretionary - Overestimation of EPS)

The *Light Gradient Boosting on ElasticNet Predictions* model demonstrates performance superior to the performance of the rest of the models in terms of both average BT Log Loss (0.49908) and average BT AUC score (0.70071), which causes it to rank as number one in the cumulative ranking on both measures (see Table 4.5).



RESULTS: CONSUMER DISCRETIONARY - OVERESTIMATION OF EPS

	Log Loss (cumulative rank)								AUC (cumulative rank)							
Model type	BT7	BT6	BT5	BT4	BT3	BT2	BT1	HO	BT7	BT6	BT5	BT4	BT3	BT2	BT1	HO
Light Gradient Boosting on ElasticNet Predictions	3	1	1	1	1	1	**1**	1	7	5	1	2	1	1	**1**	1
Elastic-Net Classifier (L2 / Binomial Deviance)	5	2	3	3	2	2	**2**	2	7	5	1	2	1	1	**1**	1
eXtreme Gradient Boosted Trees Classifier	1	3	2	2	4	3	**3**	3	3	4	6	4	7	7	**7**	7
Light Gradient Boosted Trees Classifier with Early Stopping	2	4	4	4	5	4	**4**	4	5	8	7	6	8	8	**8**	8
Vowpal Wabbit Classifier	6	5	5	5	3	5	**5**	5	6	1	3	1	3	3	**3**	3
TensorFlow Neural Network Classifier	7	7	7	6	7	6	**6**	6	2	2	4	5	4	4	**4**	4
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	4	6	6	7	6	7	**7**	7	4	3	5	7	6	5	**5**	5
RandomForest Classifier (Gini)	8	8	8	8	8	8	**8**	8	1	7	8	8	5	6	**6**	6

Table 4.5: Cumulative rank of models (Consumer Discretionary – Overestimation of EPS)

The model obtains an average BT AUC score above 0.7 (0.70071), which, according to Whiting et al. (2012), is generally considered a fair score for a model[26] (see Table 4.1). Furthermore, with its accuracy score of 70.93% on the holdout, which is 2.33 percentage points (pp) better than a majority classifier, the model is among the three models yielding better predictions than such a classifier.

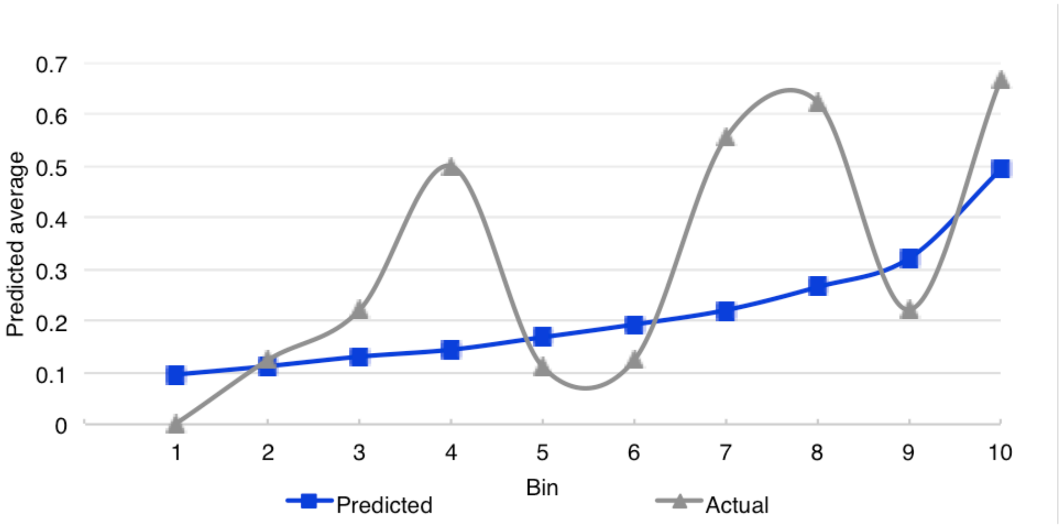


Figure 4.1: Lift chart of *Light Gradient Boosting on ElasticNet Predictions* (Consumer Discretionary - Overestimation of EPS)

An analysis of the lift chart depicted in Figure 4.1 reveals that the model performs well when predicting the 10% of the observations that were assigned the highest probabilities, as the average actual outcome of these observations is the highest among all bins. On average across all sectors, the sample used in this study indicates that consensus underestimates EPS in approximately 75% of all cases. Thus, lowering an estimate would be a rather bold move

[26] Whiting et al. (2012, p. 511)state that, *"As a rough rule of thumb, AUC > 0.9 indicates excellent test accuracy, while 0.8 < AUC < 0.9 indicates good test accuracy; AUC values in the 0.7 range are generally considered fair"*

for an analyst. It is likely that the analyst would only do so based on a model prediction, if the given model was very confident in such a prediction. It is argued that a model that is right when it assigns high probabilities to observations and on the other hand assigns wrong predictions in cases in which it is less certain, is preferable to a model that possesses the opposite characteristics.

Even though the *Light Gradient Boosting on ElasticNet Predictions* model struggles to assign the correct probabilities in the bins between the tails, it seems fairly robust: the model performs consistently well in backtests, it has an accuracy score exceeding the majority class distribution, and its lift chart shows good results for the specific task. Thus, based on the analysis, it is concluded that the model fulfills C3 as well.

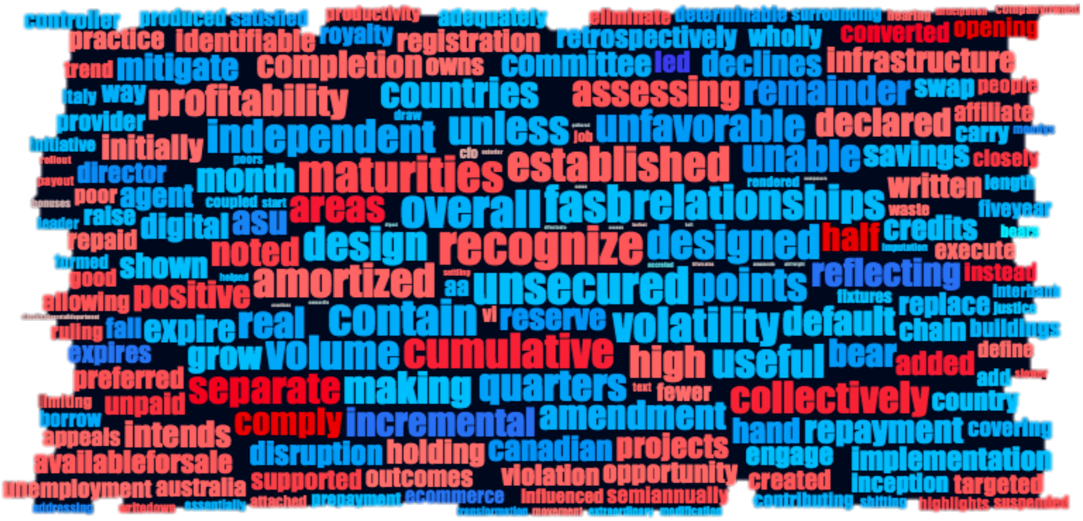
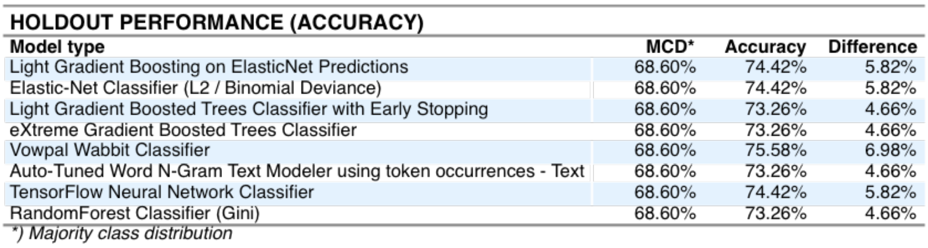


Figure 4.2: Word cloud (Consumer Discretionary - Overestimation of EPS)

While the *Auto-Tuned Word N-Gram Text Modeler* has the weakest results in terms of average BT Log Loss, its word cloud is still considered, as the model fulfills C1 and C2. The word “half” is considered as having a highly discriminant value when predicting overestimations of EPS by consensus, as it has the highest coefficient (0.9875), and as it appears in 790 of the filings. However, an examination of the use of “half” offers no intuitive explanation as to why the word shows such a discriminative value. In regards to the word “opening”, which has the second highest coefficient (0.9795), it is easier to provide possible explanations for its discriminant value. An examination of sentences that include “opening” reveals that it in many cases is used in relation to store openings. Thus, the results might indicate that analysts, on average, overestimate the positive impact, store openings have on EPS. Another explanation might be that companies with physical stores use “opening” more often, and that the EPS of such companies have been below consensus expectations due to increased competition from online sales.

4.2.2 Underestimation of EPS

All models in this model construction fulfill both C2 and C3. Furthermore, when assigning predictions to the holdout observations, all models obtain an accuracy score that is 5-7 pp above the accuracy that could have been obtained by simply assigning all predictions to the majority class (see Table 4.6).



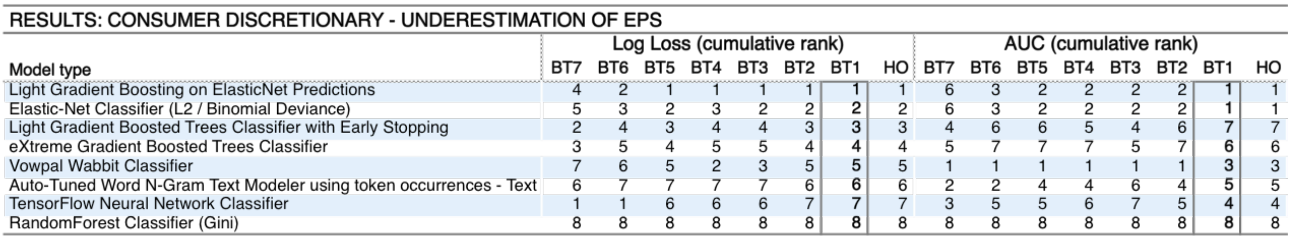
HOLDOUT PERFORMANCE (ACCURACY)

Model type	MCD*	Accuracy	Difference
Light Gradient Boosting on ElasticNet Predictions	68.60%	74.42%	5.82%
Elastic-Net Classifier (L2 / Binomial Deviance)	68.60%	74.42%	5.82%
Light Gradient Boosted Trees Classifier with Early Stopping	68.60%	73.26%	4.66%
eXtreme Gradient Boosted Trees Classifier	68.60%	73.26%	4.66%
Vowpal Wabbit Classifier	68.60%	75.58%	6.98%
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	68.60%	73.26%	4.66%
TensorFlow Neural Network Classifier	68.60%	74.42%	5.82%
RandomForest Classifier (Gini)	68.60%	73.26%	4.66%

**) Majority class distribution*

Table 4.6: Holdout performance (Consumer Discretionary – Underestimation of EPS)

The *Light Gradient Boosting on ElasticNet Predictions* model ranks as number one in terms of both average BT Log Loss and average BT AUC score (see Table 4.7). An examination of the underlying ranks in each backtest (see Table 4.8) reveals volatility in the relative performance of the model, but as it has consistently been ranked in the upper half, such volatility is deemed to be acceptable.



RESULTS: CONSUMER DISCRETIONARY - UNDERESTIMATION OF EPS

	Log Loss (cumulative rank)								AUC (cumulative rank)							
Model type	BT7	BT6	BT5	BT4	BT3	BT2	BT1	HO	BT7	BT6	BT5	BT4	BT3	BT2	BT1	HO
Light Gradient Boosting on ElasticNet Predictions	4	2	1	1	1	1	**1**	1	6	3	2	2	2	2	**1**	1
Elastic-Net Classifier (L2 / Binomial Deviance)	5	3	2	3	2	2	**2**	2	6	3	2	2	2	2	**1**	1
Light Gradient Boosted Trees Classifier with Early Stopping	2	4	3	4	4	3	**3**	3	4	6	6	5	4	6	**7**	7
eXtreme Gradient Boosted Trees Classifier	3	5	4	5	5	4	**4**	4	5	7	7	7	5	7	**6**	6
Vowpal Wabbit Classifier	7	6	5	2	3	5	**5**	5	1	1	1	1	1	1	**3**	3
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	6	7	7	7	7	6	**6**	6	2	2	4	4	6	4	**5**	5
TensorFlow Neural Network Classifier	1	1	6	6	6	7	**7**	7	3	5	5	6	7	5	**4**	4
RandomForest Classifier (Gini)	8	8	8	8	8	8	**8**	8	8	8	8	8	8	8	**8**	8

Table 4.7: Cumulative rank of models (Consumer Discretionary - Underestimation of EPS)

RESULTS: CONSUMER DISCRETIONARY - UNDERESTIMATION OF EPS

	Log Loss (rank)								AUC (rank)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
Light Gradient Boosting on ElasticNet Predictions	4	3	2	3	2	1	**2**	2	6	2	1	3	1	1	**1**	3
Elastic-Net Classifier (L2 / Binomial Deviance)	5	2	3	4	1	2	**1**	1	6	2	1	3	1	1	**1**	3
Light Gradient Boosted Trees Classifier with Early Stopping	2	5	5	2	8	3	**5**	3	4	5	5	5	6	8	**6**	6
eXtreme Gradient Boosted Trees Classifier	3	4	4	5	7	4	**3**	4	5	6	6	6	5	7	**4**	6
Vowpal Wabbit Classifier	7	1	1	1	3	7	**4**	6	1	1	7	1	3	3	**7**	8
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	6	7	6	7	4	6	**6**	5	2	4	4	7	7	6	**5**	1
TensorFlow Neural Network Classifier	1	6	7	6	6	8	**7**	7	3	7	3	8	8	5	**3**	2
RandomForest Classifier (Gini)	8	8	8	8	5	5	**8**	8	8	8	8	2	4	4	**8**	5

Table 4.8: Rank of models (Consumer Discretionary - Underestimation of EPS)

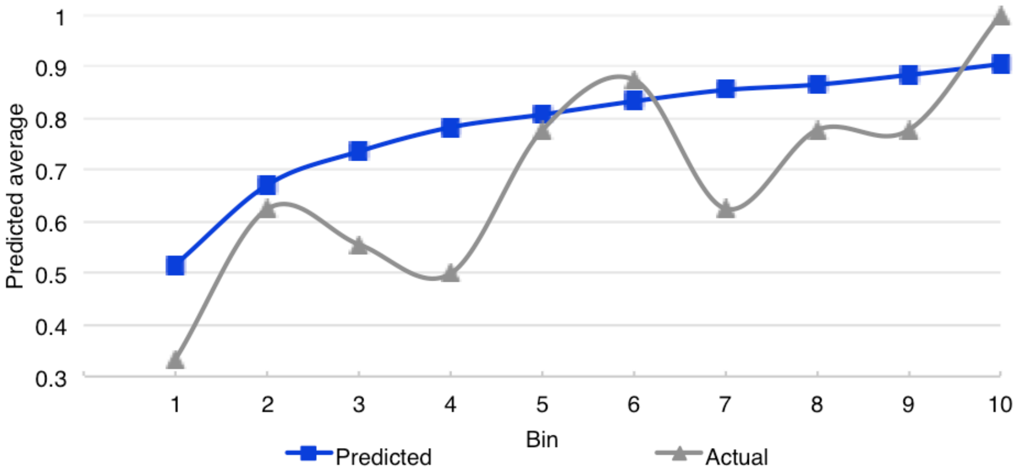


Figure 4.3: Lift chart of *Light Gradient Boosting on ElasticNet Predictions* (Consumer Discretionary - Underestimation of EPS)

The lift chart of the model shows steepness of the left side of the curve depicting the average of assigned probabilities. This indicates that the model performs well when predicting the outcome of observations where estimates should not be lifted (negative predictions), and illustrates that the model is in fact able to outperform a majority classifier as it is able to detect observations that belong to the minority class. The observations in the bin with the 10% of the observations that have been assigned the lowest probabilities have the lowest average actual outcome of all bins. As the average of actual outcomes is below 0.5, less than 50% of the observations in the bin have an actual outcome of an underestimation. Thus, analysts would be able to enhance their accuracy more than by using a majority classifier by lifting their estimates for all filings but the ones in this particular bin.

Based on the analysis outlined in the above, it is argued that the performance of the *Light Gradient Boosting on ElasticNet Predictions* model is sufficient to fulfill the subjective criterion, C3.

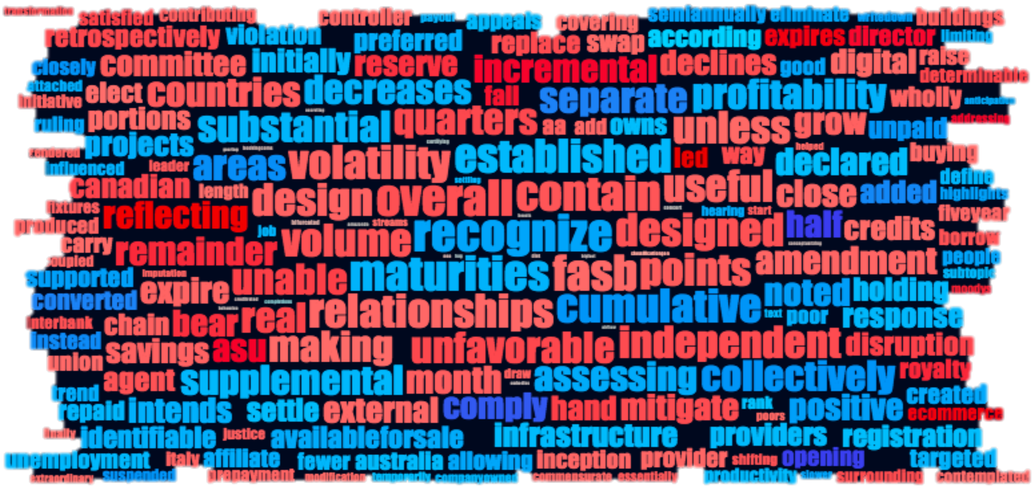

satisfied contributing controller appeals covering semiannually eliminate buildings
retrospectively violation preferred replace swap according expires director limiting
closely committee initially reserve incremental declines good digital raise
attached determinable
initiative elect countries decreases fall separate profitability wholly
ruling portions substantial quarters aa add owns unless grow unpaid addressing
projects established led way declared buying
influenced leader areas volatility
canadian length design overall contain useful close added define highlights
fixtures
produced reflecting half credits fiveyear borrow
carry remainder volume recognize designed
coupled unable maturities fasb points amendment people subtopic
supported expire
converted real relationships cumulative noted holding
interbank chain bear poor response
instead savings asu making unfavorable independent disruption
union agent supplemental month draw assessing collectively royalty
trend intends settle external comply hand mitigate rank positive created
repaid ecommerce
identifiable justice availableforsale infrastructure providers registration
unemployment italy affiliate fewer australia allowing inception provider shifting opening targeted
suspended prepayment productivity surrounding contemplated

Figure 4.4: Word cloud (Consumer Discretionary - Underestimation of EPS)

An analysis of the word cloud from the *Auto-Tuned Word N-Gram Text Modeler* (which fulfills C1 and C2) reveals that “led” and “reflecting” have the highest coefficients of 1.00 and 0.8363, respectively. Further examination of the contexts of the words reveals that they are commonly used by companies when providing insights in regards to deviations from normal conditions or previous quarters[27]. Thus, an explanation for the high coefficients of the words might be that analysts underestimate the positive impact on EPS it could entail when conditions start normalize subsequent to such deviations.

“ecommerce” has the third highest coefficient (0.7890). This could potentially be explained by the fact that worldwide retail ecommerce has grown by approximately 72% from 2014-2017 (Statista, 2018), and that this growth has most likely been difficult for analysts to foresee ex ante. However, it could be argued that analysts should now be aware of the importance of the ecommerce market, and that they are therefore likely to pay more attention to the field in the future. Thus, the word “ecommerce” might not have the same indicative value going forward.

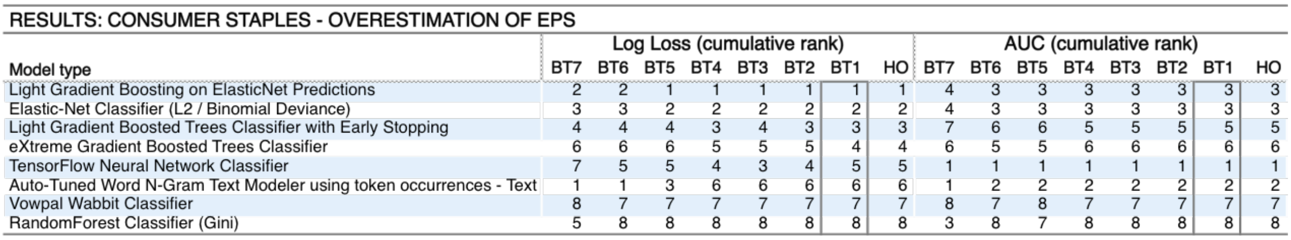
[27] E.g. *”lower revenues from australian newspapers reflecting impact from foreign currency fluctuations”*

4.3 Consumer Staples

4.3.1 Overestimation of EPS

Two of the eight models built for predicting analyst overestimations of EPS have an average BT AUC score that falls within the random range. Moreover, one of these models has more than one BT AUC score with a value below 0.5. Thus, neither of these models fulfills both C1 and C2. Six models do, on the other hand, fulfill the criteria (see Table 4.1 above).

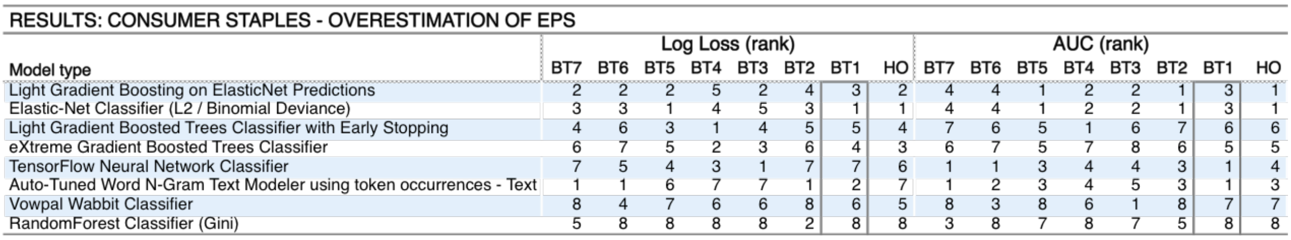
Based on the cumulative ranking listed in Table 4.9, the *Light Gradient Boosting on ElasticNet Predictions* model is the best model. However, an assessment of the relative performance of the model in each backtest reveals that it has not ranked as number one in any individual backtest (see Table 4.10). Thus, the model ends as number one on the cumulative ranking due to the consistency characterizing it (and not its peers).



RESULTS: CONSUMER STAPLES - OVERESTIMATION OF EPS

	Log Loss (cumulative rank)								AUC (cumulative rank)							
Model type	BT7	BT6	BT5	BT4	BT3	BT2	BT1	HO	BT7	BT6	BT5	BT4	BT3	BT2	BT1	HO
Light Gradient Boosting on ElasticNet Predictions	2	2	1	1	1	1	1	1	4	3	3	3	3	3	3	3
Elastic-Net Classifier (L2 / Binomial Deviance)	3	3	2	2	2	2	2	2	4	3	3	3	3	3	3	3
Light Gradient Boosted Trees Classifier with Early Stopping	4	4	4	3	4	3	3	3	7	6	6	5	5	5	5	5
eXtreme Gradient Boosted Trees Classifier	6	6	6	5	5	5	4	4	6	5	5	6	6	6	6	6
TensorFlow Neural Network Classifier	7	5	5	4	3	4	5	5	1	1	1	1	1	1	1	1
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	1	1	3	6	6	6	6	6	1	2	2	2	2	2	2	2
Vowpal Wabbit Classifier	8	7	7	7	7	7	7	7	8	7	8	7	7	7	7	7
RandomForest Classifier (Gini)	5	8	8	8	8	8	8	8	3	8	7	8	8	8	8	8

Table 4.9: Cumulative rank of models (Consumer Staples - Overestimation of EPS)

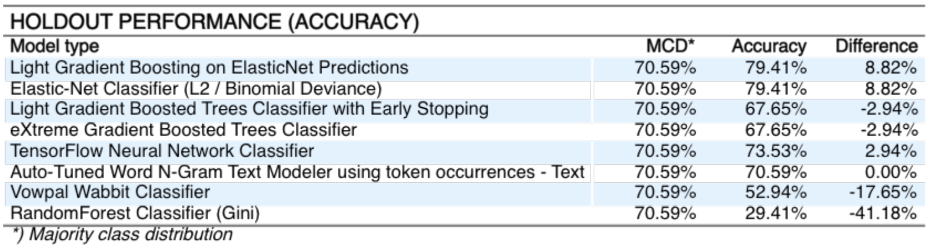


RESULTS: CONSUMER STAPLES - OVERESTIMATION OF EPS

	Log Loss (rank)								AUC (rank)							
Model type	BT7	BT6	BT5	BT4	BT3	BT2	BT1	HO	BT7	BT6	BT5	BT4	BT3	BT2	BT1	HO
Light Gradient Boosting on ElasticNet Predictions	2	2	2	5	2	4	3	2	4	4	1	2	2	1	3	1
Elastic-Net Classifier (L2 / Binomial Deviance)	3	3	1	4	5	3	1	1	4	4	1	2	2	1	3	1
Light Gradient Boosted Trees Classifier with Early Stopping	4	6	3	1	4	5	5	4	7	6	5	1	6	7	6	6
eXtreme Gradient Boosted Trees Classifier	6	7	5	2	3	6	4	3	6	7	5	7	8	6	5	5
TensorFlow Neural Network Classifier	7	5	4	3	1	7	7	6	1	1	3	4	4	3	1	4
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	1	1	6	7	7	1	2	7	1	2	3	4	5	3	1	3
Vowpal Wabbit Classifier	8	4	7	6	6	8	6	5	8	3	8	6	1	8	7	7
RandomForest Classifier (Gini)	5	8	8	8	8	2	8	8	3	8	7	8	7	5	8	8

Table 4.10: Rank of models (Consumer Staples - Overestimation of EPS)

The model performs well when providing predictions for the holdout sample. The accuracy score of 79.41 obtained by the model is approximately 9 pp better than the score that could have been obtained if it had simply assigned all observations to the majority class (see Table 4.11).

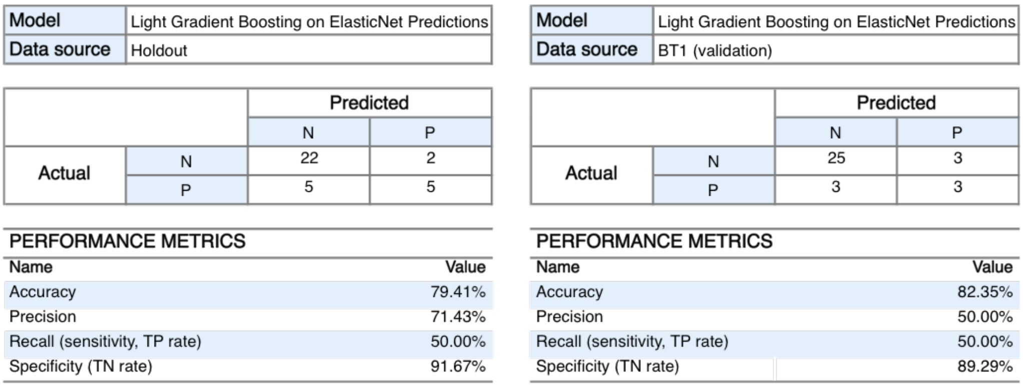


HOLDOUT PERFORMANCE (ACCURACY)			
Model type	MCD*	Accuracy	Difference
Light Gradient Boosting on ElasticNet Predictions	70.59%	79.41%	8.82%
Elastic-Net Classifier (L2 / Binomial Deviance)	70.59%	79.41%	8.82%
Light Gradient Boosted Trees Classifier with Early Stopping	70.59%	67.65%	-2.94%
eXtreme Gradient Boosted Trees Classifier	70.59%	67.65%	-2.94%
TensorFlow Neural Network Classifier	70.59%	73.53%	2.94%
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	70.59%	70.59%	0.00%
Vowpal Wabbit Classifier	70.59%	52.94%	-17.65%
RandomForest Classifier (Gini)	70.59%	29.41%	-41.18%

*) Majority class distribution

Table 4.11: Holdout performance (Consumer Staples - Overestimation of EPS)

As displayed in the confusion matrices of the *Light Gradient Boosting on ElasticNet Predictions* model depicted in Figure 4.5 below, the accuracy score obtained by the model is even higher when providing predictions for the observations in BT1 (82.35%). Such an accuracy score is, however, equal to what could have been obtained by assigning all observations to the majority class.



Model	Light Gradient Boosting on ElasticNet Predictions
Data source	Holdout

		Predicted	
		N	P
Actual	N	22	2
	P	5	5

PERFORMANCE METRICS	
Name	Value
Accuracy	79.41%
Precision	71.43%
Recall (sensitivity, TP rate)	50.00%
Specificity (TN rate)	91.67%

Model	Light Gradient Boosting on ElasticNet Predictions
Data source	BT1 (validation)

		Predicted	
		N	P
Actual	N	25	3
	P	3	3

PERFORMANCE METRICS	
Name	Value
Accuracy	82.35%
Precision	50.00%
Recall (sensitivity, TP rate)	50.00%
Specificity (TN rate)	89.29%

Figure 4.5: Confusion matrices of *Light Gradient Boosting on ElasticNet Predictions* - holdout and BT1 (Consumer Staples - Overestimation of EPS)

It is emphasized that obtaining the same accuracy score as a majority classifier does not necessarily equal poor performance. A model cannot be expected to outperform a majority classifier in all cases (and even less so in cases where the majority class holds so many of the observations). Furthermore, from the confusion matrices it follows that the model is in fact able to correctly identify some of the minority class observations. The distribution of true and false positive predictions shows that an analyst would obtain an overall enhancement of his accuracy by adjusting his estimates according to the model predictions, as he would correctly shift his estimates downwards in 5 and 3 cases, respectively, versus incorrectly shift them downwards in 2 and 3 cases, respectively.

It is emphasized that models performing consistently well are preferable to volatile models that either perform very well or very poor. *The Light Gradient Boosting on ElasticNet Predictions* model has consistent performance across all backtests in a model construction where other models display a high degree of volatility. Furthermore, the model obtains an accuracy score equal to or better than a majority classifier would have in the validation and the HO set, and it fulfills both C1 and C2. Thus, based on the analysis, it is argued that the model similarly passes the subjective evaluation and thereby fulfills C3.

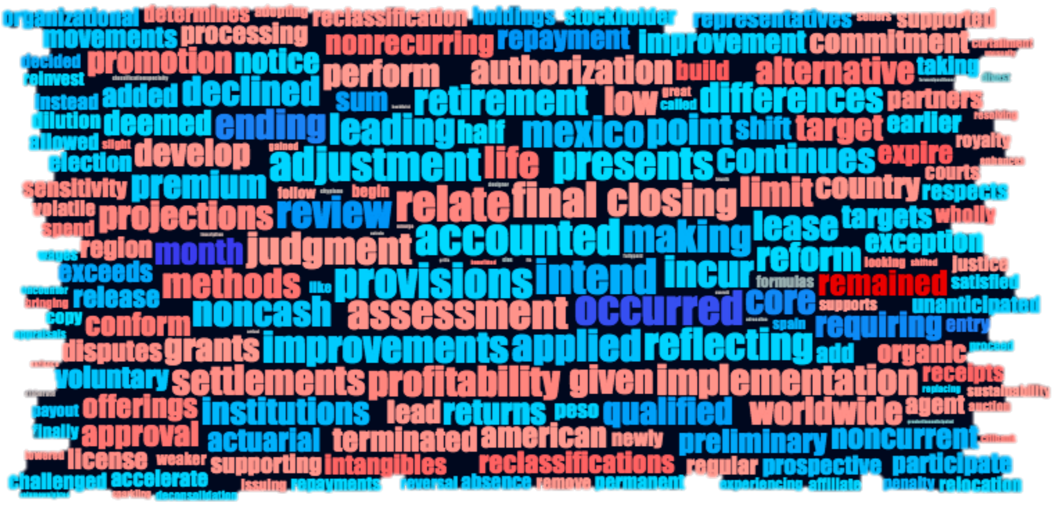
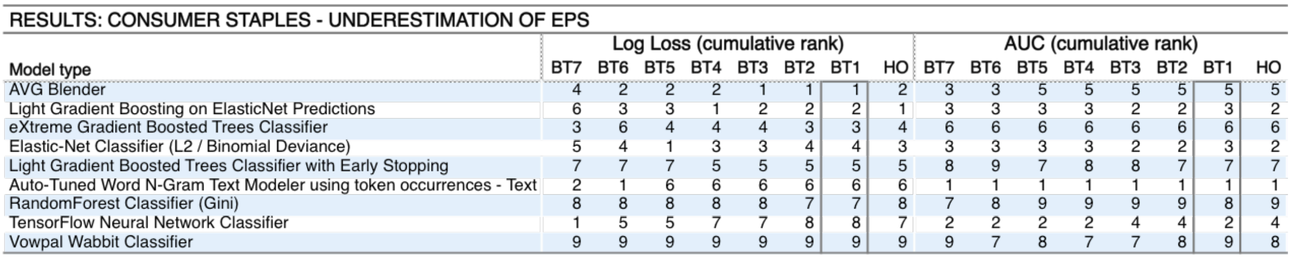


Figure 4.6: Word cloud (Consumer Staples - Overestimation of EPS)

The word cloud of the *Auto-Tuned Word N-Gram Text Modeler*, which fulfills C1 and C2, is examined. From such examination it follows that the word "remained" has the highest coefficient of all words (1.00). Further investigation of the filings reveals that the word is used in several different contexts, which makes it difficult to provide an intuitive explanation as to what could cause its strong discriminant value. However, a possible explanation might be that companies use "remained" when referring to both recurring positive news and recurring negative news, but that analysts tend to extrapolate recurring positive news while expecting negative news not to persist. Such behavior would cause "remained" to be indicative of overestimations.

4.3.2 Underestimation of EPS

As can be seen in Table 4.2, six of the nine models, built for predicting underestimation of EPS, in the Consumer Staples sector fulfill C1 and C2. Among the constructed models, the best performing one, measured on average BT Log Loss, is an *AVG Blender* (see Table 4.12). The *AVG Blender* is an ensemble model consisting of a *Light Gradient Boosting on ElasticNet Predictions* model and an *eXtreme Gradient Boosted Trees Classifier*. The predicted probability for a given observation is the average of the probabilities assigned by the two underlying models. However, in terms of average BT AUC score the *AVG Blender* ranks as the fifth best model.

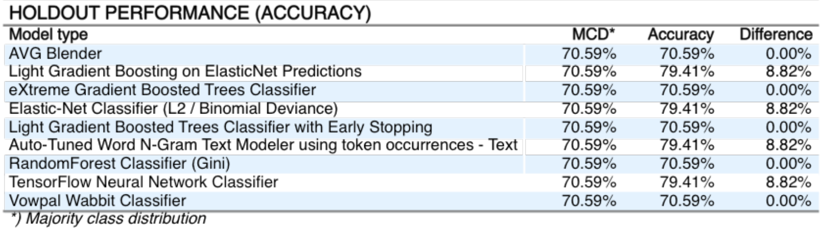


RESULTS: CONSUMER STAPLES - UNDERESTIMATION OF EPS

	Log Loss (cumulative rank)								AUC (cumulative rank)							
Model type	BT7	BT6	BT5	BT4	BT3	BT2	BT1	HO	BT7	BT6	BT5	BT4	BT3	BT2	BT1	HO
AVG Blender	4	2	2	2	1	1	1	2	3	3	5	5	5	5	5	5
Light Gradient Boosting on ElasticNet Predictions	6	3	3	1	2	2	2	1	3	3	3	3	2	2	3	2
eXtreme Gradient Boosted Trees Classifier	3	6	4	4	4	3	3	4	6	6	6	6	6	6	6	6
Elastic-Net Classifier (L2 / Binomial Deviance)	5	4	1	3	3	4	4	3	3	3	3	3	2	2	3	2
Light Gradient Boosted Trees Classifier with Early Stopping	7	7	7	5	5	5	5	5	8	9	7	8	8	7	7	7
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	2	1	6	6	6	6	6	6	1	1	1	1	1	1	1	1
RandomForest Classifier (Gini)	8	8	8	8	8	7	7	8	7	8	9	9	9	9	8	9
TensorFlow Neural Network Classifier	1	5	5	7	7	8	8	7	2	2	2	2	4	4	2	4
Vowpal Wabbit Classifier	9	9	9	9	9	9	9	9	9	7	8	7	7	8	9	8

Table 4.12: Cumulative rank of models (Consumer Staples - Underestimation of EPS)

From the accuracy obtained in the holdout sample (see Table 4.13), it can be inferred that the model performs no better than a majority classifier when providing these predictions. Furthermore, the confusion matrices depicted in Figure 4.7 reveal that the model does not outperform a majority classifier in BT1 either, as it simply assigns positive predictions to all observations (predictions of underestimations). Thus, the performance of the model in HO and BT1 indicates that it has difficulties spotting observations belonging to the minority class.

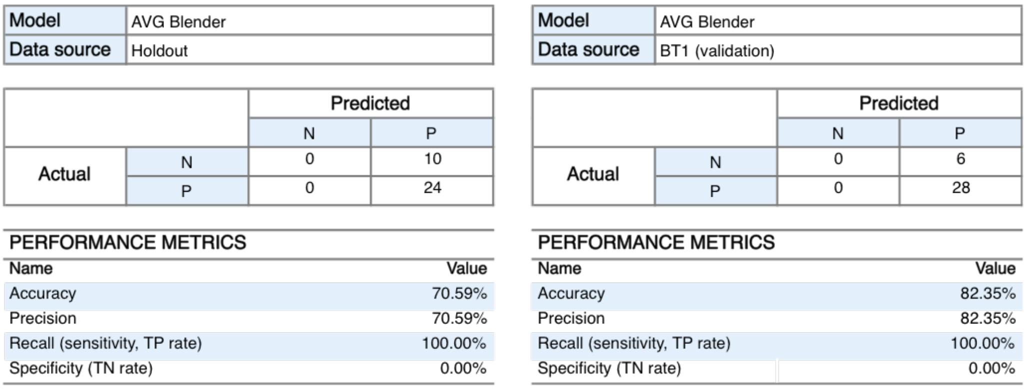


HOLDOUT PERFORMANCE (ACCURACY)

Model type	MCD*	Accuracy	Difference
AVG Blender	70.59%	70.59%	0.00%
Light Gradient Boosting on ElasticNet Predictions	70.59%	79.41%	8.82%
eXtreme Gradient Boosted Trees Classifier	70.59%	70.59%	0.00%
Elastic-Net Classifier (L2 / Binomial Deviance)	70.59%	79.41%	8.82%
Light Gradient Boosted Trees Classifier with Early Stopping	70.59%	70.59%	0.00%
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	70.59%	79.41%	8.82%
RandomForest Classifier (Gini)	70.59%	70.59%	0.00%
TensorFlow Neural Network Classifier	70.59%	79.41%	8.82%
Vowpal Wabbit Classifier	70.59%	70.59%	0.00%

*) Majority class distribution

Table 4.13: Holdout performance (Consumer Staples - Underestimation of EPS)



Model	AVG Blender
Data source	Holdout

		Predicted	
		N	P
Actual	N	0	10
	P	0	24

PERFORMANCE METRICS	
Name	Value
Accuracy	70.59%
Precision	70.59%
Recall (sensitivity, TP rate)	100.00%
Specificity (TN rate)	0.00%

Model	AVG Blender
Data source	BT1 (validation)

		Predicted	
		N	P
Actual	N	0	6
	P	0	28

PERFORMANCE METRICS	
Name	Value
Accuracy	82.35%
Precision	82.35%
Recall (sensitivity, TP rate)	100.00%
Specificity (TN rate)	0.00%

Figure 4.7: Confusion matrices of *AVG Blender* - holdout and BT1 (Consumer Staples - Underestimation of EPS)

Even though the model ranks as number one in terms of Log Loss, it is not able to identify a pattern enabling it to spot minority class observations in neither the HO or in BT1. This results in accuracy scores that could similarly have been obtained by a majority classifier. Furthermore, its ranking in terms of average BT AUC is not impressive. Based on these findings, it is argued that the model does not fulfill C3.

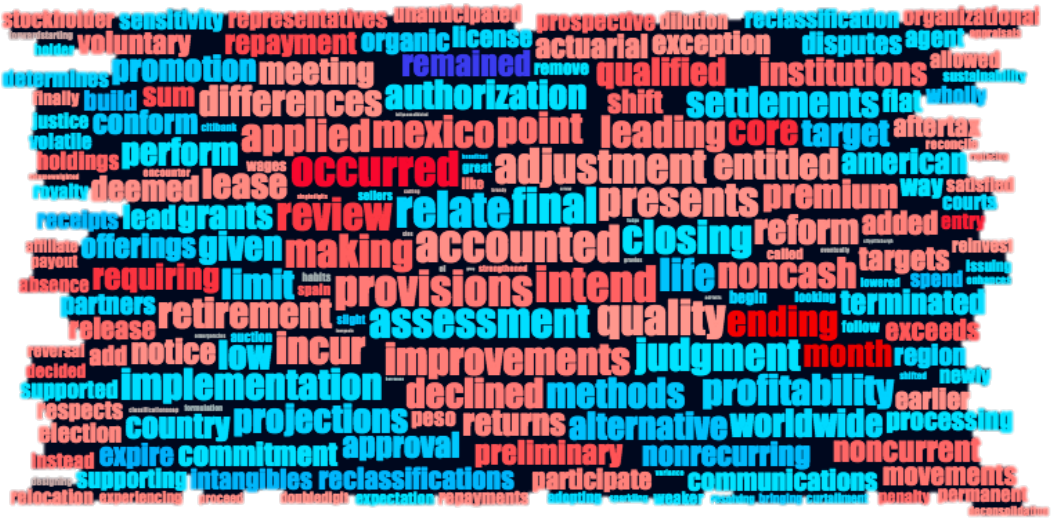


Figure 4.8: Word cloud (Consumer Staples - Underestimation of EPS)

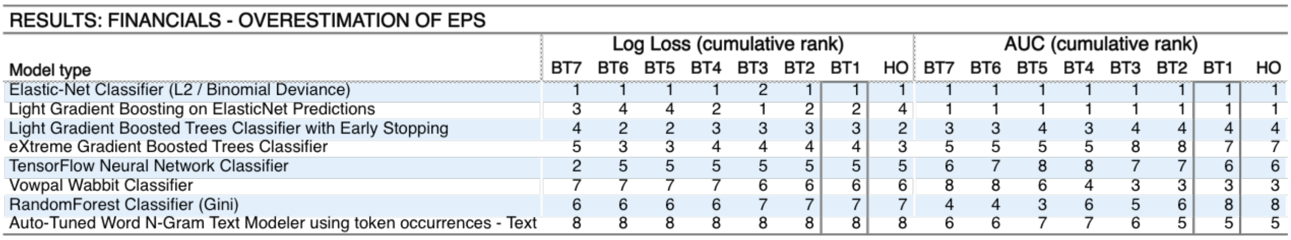
The word cloud from the *Auto-Tuned Word N-Gram Text Modeler* reveals that the word "month" has the highest coefficient (0.6728). While an explanation for such a coefficient might not seem apparent, it is found from further analysis that the word is oftentimes used in conjunction with referrals to hedging activities, FX movements, and changes in short term interest rates (especially the three month LIBOR rate). All of these add complexity to the forecasting task, an analyst faces, and furthermore, they most likely impact many of the companies in the sector, which e.g. include large global ones such as P&G, Coca Cola, and Philip Morris. In regards to the use of the word “month” in relation to LIBOR, a possible explanation for its coefficient might be that analysts are likely to overestimate the negative impact of the increase in the interest rate caused by Brexit[28] - either because they fail to consider the fact that the depreciation in the British Pound also resulting from Brexit offsets some of this negative impact, or because they fail to take into consideration that the hedging activities of companies could similarly offset some of the negative impact.

4.4 Financials

4.4.1 Overestimation of EPS

Only three of the eight models constructed to predict overestimations in the Financials sector fulfill both C1 and C2. While one models fail to pass only the significance test (stating that the average BT AUC should be outside the random range), two models fail to pass the applicability test (stating that no more than one BT AUC score should be below 0.5), and two fail to pass both. However, it is emphasized that out of the three models fulfilling the criteria, two have an average BT AUC score above 0.7 (see Table 4.1). As stated above, models with such score levels are considered fair models (Whiting et al., 2012).

Based on the cumulative average BT Log Loss, the *Elastic-Net Classifier (L2 / Binomial Deviance)* displays the best performance of the models fulfilling the criteria. This holds for the average BT AUC score as well (see Table 4.14).



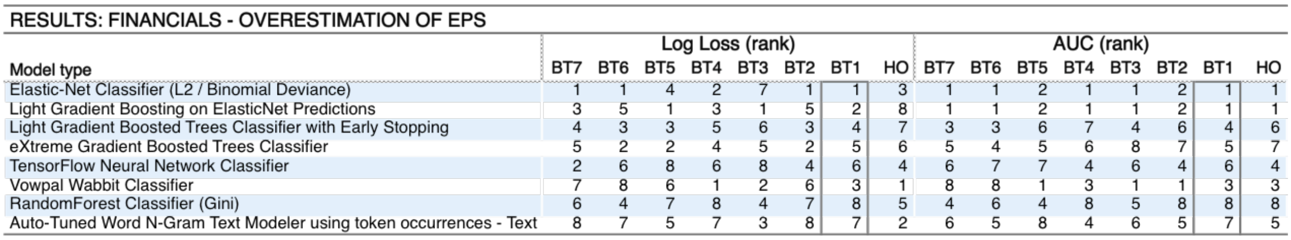
RESULTS: FINANCIALS - OVERESTIMATION OF EPS

	Log Loss (cumulative rank)								AUC (cumulative rank)							
Model type	BT7	BT6	BT5	BT4	BT3	BT2	BT1	HO	BT7	BT6	BT5	BT4	BT3	BT2	BT1	HO
Elastic-Net Classifier (L2 / Binomial Deviance)	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1
Light Gradient Boosting on ElasticNet Predictions	3	4	4	2	1	2	2	4	1	1	1	1	1	1	1	1
Light Gradient Boosted Trees Classifier with Early Stopping	4	2	2	3	3	3	3	2	3	3	4	3	4	4	4	4
eXtreme Gradient Boosted Trees Classifier	5	3	3	4	4	4	4	3	5	5	5	5	8	8	7	7
TensorFlow Neural Network Classifier	2	5	5	5	5	5	5	5	6	7	8	8	7	7	6	6
Vowpal Wabbit Classifier	7	7	7	7	6	6	6	6	8	8	6	4	3	3	3	3
RandomForest Classifier (Gini)	6	6	6	6	7	7	7	7	4	4	3	6	5	6	8	8
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	8	8	8	8	8	8	8	8	6	6	7	7	6	5	5	5

Table 4.14: Cumulative rank of models (Financials - Overestimation of EPS)

[28] Brexit refers to the withdrawal of the United Kingdom from the European Union. In June 2016 a majority of the UK electorate voted to leave the EU, and due the following invokement of Article 50 of the Lisbon Treaty, the UK is to leave the EU in March 2019

The ranking of models on individual BTs reveals that the model was the best scoring model in four out of seven BTs in terms of Log Loss and five in terms of AUC score. Thus, its relatively good performance seems to be fairly consistent (see Table 4.15).



RESULTS: FINANCIALS - OVERESTIMATION OF EPS	Log Loss (rank)								AUC (rank)							
Model type	BT7	BT6	BT5	BT4	BT3	BT2	BT1	HO	BT7	BT6	BT5	BT4	BT3	BT2	BT1	HO
Elastic-Net Classifier (L2 / Binomial Deviance)	1	1	4	2	7	1	1	3	1	1	2	1	1	2	1	1
Light Gradient Boosting on ElasticNet Predictions	3	5	1	3	1	5	2	8	1	1	2	1	1	2	1	1
Light Gradient Boosted Trees Classifier with Early Stopping	4	3	3	5	6	3	4	7	3	3	6	7	4	6	4	6
eXtreme Gradient Boosted Trees Classifier	5	2	2	4	5	2	5	6	5	4	5	6	8	7	5	7
TensorFlow Neural Network Classifier	2	6	8	6	8	4	6	4	6	7	7	4	6	4	6	4
Vowpal Wabbit Classifier	7	8	6	1	2	6	3	1	8	8	1	3	1	1	3	3
RandomForest Classifier (Gini)	6	4	7	8	4	7	8	5	4	6	4	8	5	8	8	8
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	8	7	5	7	3	8	7	2	6	5	8	4	6	5	7	5

Table 4.15: Rank of models (Financials - Overestimation of EPS)

The average BT AUC score of the model (0.71662) is emphasized, as this score makes the model one of the two models in this model construction that are considered fair, according to the rule of thumb described by Whiting et al. (2012).

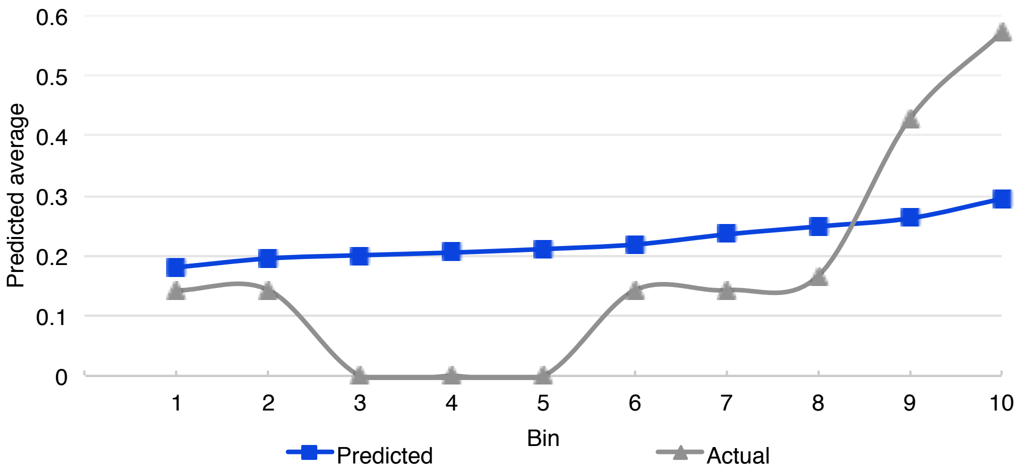
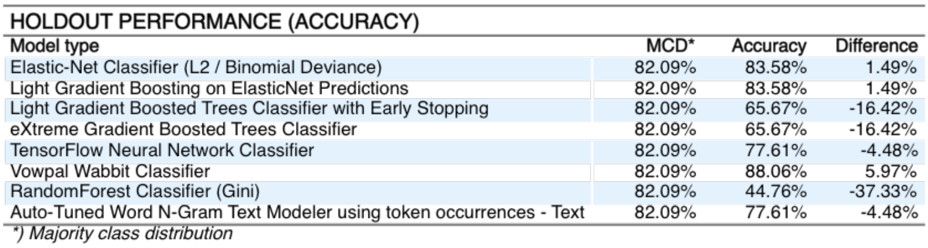


Figure 4.9: Lift chart of *Elastic-Net Classifier (L2 / Binomial Deviance)* (Financials - Overestimation of EPS)

The lift chart of the model further underlines its good performance: the bin with the 10% of the observations with the highest assigned probabilities is the bin with the 10% highest average actual outcome. As consensus overestimates EPS in more than 50% of the observations in this bin, analysts would improve their accuracy if they simply chose to lower their estimates for these particular observations. The model in fact assigns positive predictions (i.e. predictions stating that EPS should be lowered) to all observations above the threshold of 0.2571, corresponding to the 19.4% of all observations with highest assigned probabilities. This leads to seven correct downwards shifts of EPS estimates, while six estimates are incorrectly shifted downwards.

When providing predictions for the HO, the accuracy score obtained by the model is 1.49 pp above the one that could have been obtained if all predictions had been assigned to the majority class (see Table 4.16). From the table it appears that several models struggle to reach similar performance, indicating that it is not an easy predictive task. Thus, it can be inferred that the model performs well when providing predictions for the HO observations.



HOLDOUT PERFORMANCE (ACCURACY)

Model type	MCD*	Accuracy	Difference
Elastic-Net Classifier (L2 / Binomial Deviance)	82.09%	83.58%	1.49%
Light Gradient Boosting on ElasticNet Predictions	82.09%	83.58%	1.49%
Light Gradient Boosted Trees Classifier with Early Stopping	82.09%	65.67%	-16.42%
eXtreme Gradient Boosted Trees Classifier	82.09%	65.67%	-16.42%
TensorFlow Neural Network Classifier	82.09%	77.61%	-4.48%
Vowpal Wabbit Classifier	82.09%	88.06%	5.97%
RandomForest Classifier (Gini)	82.09%	44.76%	-37.33%
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	82.09%	77.61%	-4.48%

*) Majority class distribution

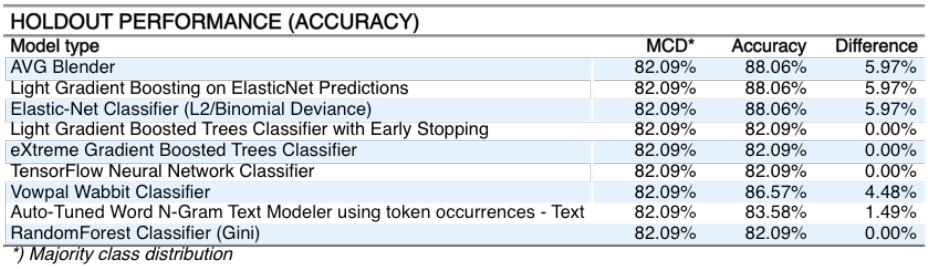
Table 4.16: Holdout performance (Financials - Overestimation of EPS)

As the model is able to correctly identify observations belonging to the minority class (overestimation-cases), and as it displays consistently good performance in terms of both Log Loss and AUC score, it is argued that it fulfills C3.

As the *Auto-Tuned Word N-Gram Text Modeler* does not fulfill C2, its word cloud will not be examined.

4.4.2 Underestimation of EPS

Seven models fulfill both criteria and the remaining two models fail to fulfill either one or both (see Table 4.2). Three of the models fulfilling both criteria obtain average BT AUC scores above 0.7. Furthermore, these three models obtain accuracy scores of 88.06% when providing predictions for the holdout sample, which is approximately 6 pp better than what could have been obtained by using a majority classifier (see Table 4.17).



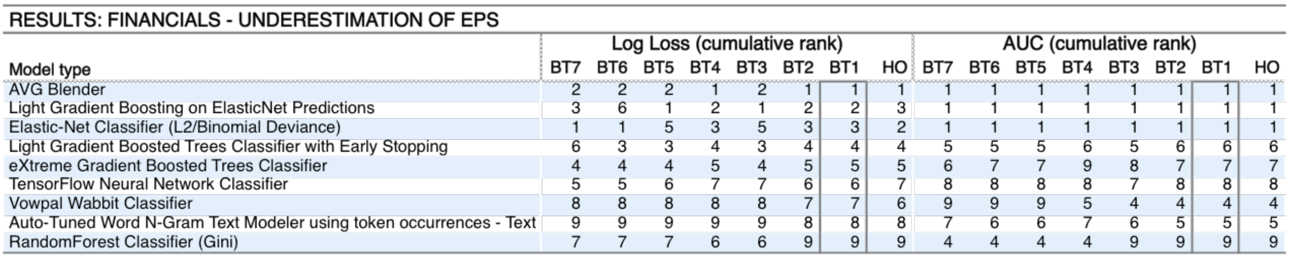
HOLDOUT PERFORMANCE (ACCURACY)

Model type	MCD*	Accuracy	Difference
AVG Blender	82.09%	88.06%	5.97%
Light Gradient Boosting on ElasticNet Predictions	82.09%	88.06%	5.97%
Elastic-Net Classifier (L2/Binomial Deviance)	82.09%	88.06%	5.97%
Light Gradient Boosted Trees Classifier with Early Stopping	82.09%	82.09%	0.00%
eXtreme Gradient Boosted Trees Classifier	82.09%	82.09%	0.00%
TensorFlow Neural Network Classifier	82.09%	82.09%	0.00%
Vowpal Wabbit Classifier	82.09%	86.57%	4.48%
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	82.09%	83.58%	1.49%
RandomForest Classifier (Gini)	82.09%	82.09%	0.00%

**) Majority class distribution*

Table 4.17: Holdout performance (Financials - Underestimation of EPS)

Measured on both average BT Log Loss and average BT AUC score, the best performing model throughout the backtests is the *AVG Blender* built on the *Light Gradient Boosting on ElasticNet Predictions* model and the *Elastic-Net Classifier (L2 / Binomial Deviance)* (see Table 4.18 below). This model is among the three models in this model construction obtaining an average BT AUC score above 0.7.



RESULTS: FINANCIALS - UNDERESTIMATION OF EPS

	Log Loss (cumulative rank)								AUC (cumulative rank)							
Model type	BT7	BT6	BT5	BT4	BT3	BT2	BT1	HO	BT7	BT6	BT5	BT4	BT3	BT2	BT1	HO
AVG Blender	2	2	2	1	2	1	1	1	1	1	1	1	1	1	1	1
Light Gradient Boosting on ElasticNet Predictions	3	6	1	2	1	2	2	3	1	1	1	1	1	1	1	1
Elastic-Net Classifier (L2/Binomial Deviance)	1	1	5	3	5	3	3	2	1	1	1	1	1	1	1	1
Light Gradient Boosted Trees Classifier with Early Stopping	6	3	3	4	3	4	4	4	5	5	5	6	5	6	6	6
eXtreme Gradient Boosted Trees Classifier	4	4	4	5	4	5	5	5	6	7	7	9	8	7	7	7
TensorFlow Neural Network Classifier	5	5	6	7	7	6	6	7	8	8	8	8	7	8	8	8
Vowpal Wabbit Classifier	8	8	8	8	8	7	7	6	9	9	9	5	4	4	4	4
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	9	9	9	9	9	8	8	8	7	6	6	7	6	5	5	5
RandomForest Classifier (Gini)	7	7	7	6	6	9	9	9	4	4	4	4	9	9	9	9

Table 4.18: Cumulative rank of models (Financials - Underestimation of EPS)

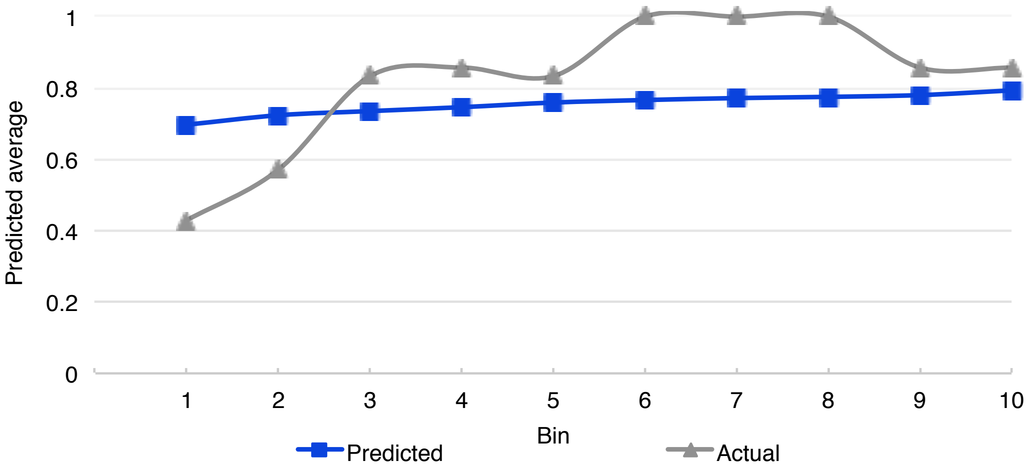


Figure 4.10: Lift chart of *AVG Blender* (Financials - Underestimation of EPS)

An analysis of the lift chart adds additional confidence in the performance of the *AVG Blender*, as it appears that the bin for which the model assigns the 10% lowest probabilities is the bin with the 10% lowest average actual outcomes (see Figure 4.10). The second bin (with the 10-20% lowest average assigned probabilities) is similarly the bin with the second lowest average actual outcomes. Based on the lift chart, it is argued that the model seems to be able to identify an underlying pattern in the observations, which it successfully applies when assigning predictions. This ability to identify minority class observations enables it to outperform a majority classifier by approximately 6 pp when providing predictions for the HO (see Table 4.17 above).

Based on the analysis of the performance of the *AVG Blender*, it is concluded to be a good model for predicting underestimations of EPS by consensus in the Financials sector. Thus, it fulfills C3.

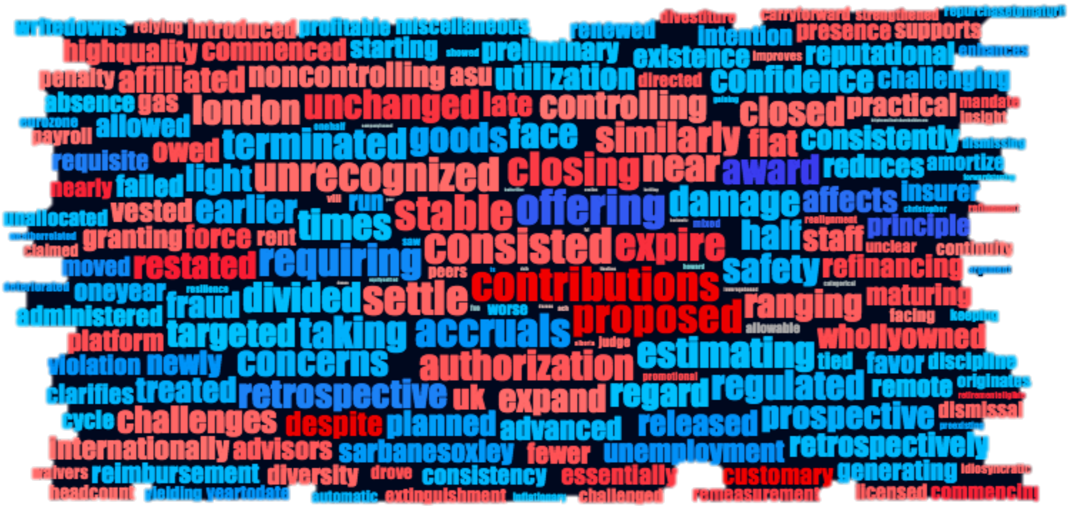


Figure 4.11: Word Cloud (Financials - Underestimation of EPS)

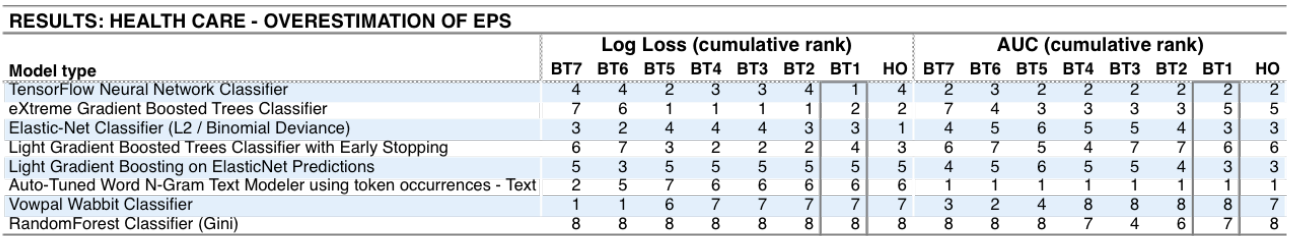
An examination of word cloud from the *Auto-Tuned Word N-Gram Text Modeler* reveals that the words “despite” and “proposed” have the highest coefficients (1.00 and 0.9710, respectively). The use of “despite” appears to be frequently used when explaining the lack of positive impact on EPS of an underlying positive trend or event. Thus, one could hypothesize that 1) companies use “despite” more often when positive trends and events have not (yet) translated into positive earnings, as they want to stress this fact to readers of the report, and 2) analysts do not buy into such explanations from management, which results in an underestimation of EPS in the following quarter when the positive trends do in fact commence to impact EPS positively.

An examination of the use of “proposed” reveals that it is often used in conjunction with proposals of new regulation. Analysts might become either uncertain and stick to conservative estimates or they might choose to include any negative impact that the new regulation could possibly eventually entail before its actual impact. Both types of behavior could lead to underestimations of EPS, and they could therefore be explanatory of the high coefficient of “proposed”.

4.5 Health Care

4.5.1 Overestimation of EPS

All models with the aim of predicting cases of overestimation for companies within the Health Care sector, except one, fulfill both C1 and C2 (see Section Table 4.1). Based on the cumulative ranking of the models shown in Table 4.19, the *Tensorflow Neural Network Classifier* outperforms the other models, ranking as number one in terms of average BT Log Loss and as number two in terms of average BT AUC.



RESULTS: HEALTH CARE - OVERESTIMATION OF EPS

	Log Loss (cumulative rank)								AUC (cumulative rank)							
Model type	BT7	BT6	BT5	BT4	BT3	BT2	BT1	HO	BT7	BT6	BT5	BT4	BT3	BT2	BT1	HO
TensorFlow Neural Network Classifier	4	4	2	3	3	4	1	4	2	3	2	2	2	2	2	2
eXtreme Gradient Boosted Trees Classifier	7	6	1	1	1	1	2	2	7	4	3	3	3	3	5	5
Elastic-Net Classifier (L2 / Binomial Deviance)	3	2	4	4	4	3	3	1	4	5	6	5	5	4	3	3
Light Gradient Boosted Trees Classifier with Early Stopping	6	7	3	2	2	2	4	3	6	7	5	4	7	7	6	6
Light Gradient Boosting on ElasticNet Predictions	5	3	5	5	5	5	5	5	4	5	6	5	5	4	3	3
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	2	5	7	6	6	6	6	6	1	1	1	1	1	1	1	1
Vowpal Wabbit Classifier	1	1	6	7	7	7	7	7	3	2	4	8	8	8	8	7
RandomForest Classifier (Gini)	8	8	8	8	8	8	8	8	8	8	8	7	4	6	7	8

Table 4.19: Cumulative rank of models (Health Care - Overestimation of EPS)

However, when observing only the relative performance of the model in the holdout, it appears that it ranks as number six on Log Loss and as number five on AUC (see Table 4.20). Due to this performance, the model is not able to uphold its cumulative Log Loss rank (including the HO, the model ranks as number 4). Furthermore, as can be seen in Table 4.20, the model is not able to obtain a rank as the best model in any single BT. In most cases, it ranks as the fourth best model.

RESULTS: HEALTH CARE - OVERESTIMATION OF EPS

	Log Loss (rank)								AUC (rank)							
Model type	BT7	BT6	BT5	BT4	BT3	BT2	BT1	HO	BT7	BT6	BT5	BT4	BT3	BT2	BT1	HO
TensorFlow Neural Network Classifier	4	4	3	4	5	6	2	6	2	4	5	2	4	5	3	5
eXtreme Gradient Boosted Trees Classifier	7	5	1	3	3	4	6	5	7	3	1	1	6	4	6	7
Elastic-Net Classifier (L2 / Binomial Deviance)	3	1	4	1	2	1	5	2	4	6	6	3	2	1	4	2
Light Gradient Boosted Trees Classifier with Early Stopping	6	6	2	2	4	3	7	4	6	8	2	6	7	3	7	4
Light Gradient Boosting on ElasticNet Predictions	5	3	5	7	1	2	4	3	4	6	6	3	2	1	4	2
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	2	7	6	6	6	7	3	7	1	4	4	3	5	5	2	5
Vowpal Wabbit Classifier	1	2	7	8	7	8	1	1	3	2	8	8	8	8	1	1
RandomForest Classifier (Gini)	8	8	8	5	8	5	8	8	8	1	3	7	1	7	8	8

Table 4.20: Rank of models (Health Care - Overestimation of EPS)

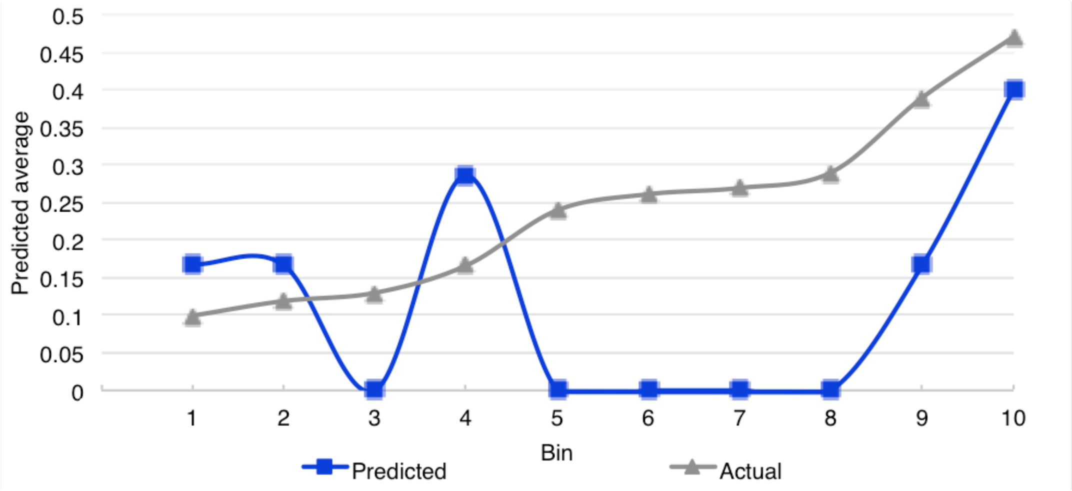
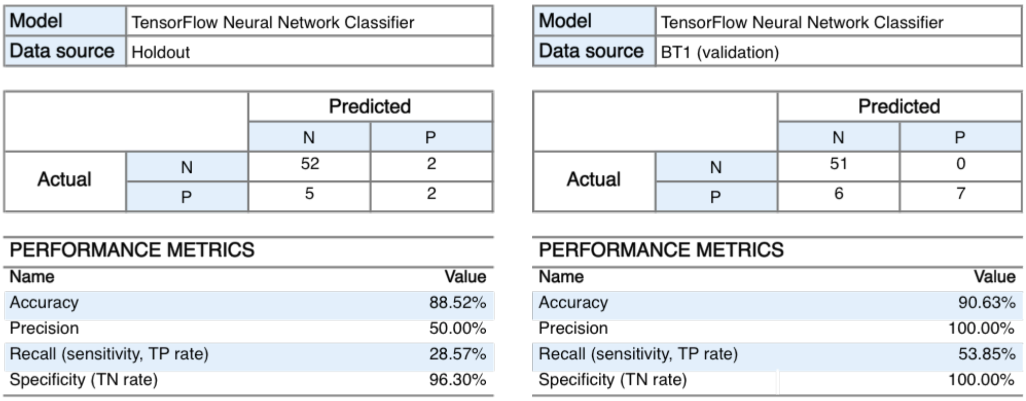


Figure 4.12: Lift chart of *TensorFlow Neural Network Classifier* (Health Care - Overestimation of EPS)

The lift chart of the model, depicted Figure 4.12, shows that the 10% of observations assigned the highest average probabilities also have the highest average actual outcomes. However, less than 50% of these observations are actual cases of overestimations, and thus, lowering estimates for all of these particular observations, would result in a decrease in accuracy. From an examination of threshold that maximizes the F_1-score, it follows that, if an analyst were to follow the recommendations of the model, he would only shift 6.5% of his estimates downward. These would be the 4 observations that had been assigned the highest probabilities. Doing so would lead to an accuracy score of 88.52% - the same accuracy that could have been obtained by a majority classifier. However, when assigning predictions to the observations in BT1 (validation), the model is able to obtain an accuracy score that is approximately 11 pp above what a majority classifier would obtain (see Figure 4.13).



Model	TensorFlow Neural Network Classifier
Data source	Holdout

		Predicted	
		N	P
Actual	N	52	2
	P	5	2

PERFORMANCE METRICS	
Name	Value
Accuracy	88.52%
Precision	50.00%
Recall (sensitivity, TP rate)	28.57%
Specificity (TN rate)	96.30%

Model	TensorFlow Neural Network Classifier
Data source	BT1 (validation)

		Predicted	
		N	P
Actual	N	51	0
	P	6	7

PERFORMANCE METRICS	
Name	Value
Accuracy	90.63%
Precision	100.00%
Recall (sensitivity, TP rate)	53.85%
Specificity (TN rate)	100.00%

Figure 4.13: Confusion matrices of *TensorFlow Neural Network Classifier* - holdout and BT1 (Health Care - Overestimation of EPS)

As the performance of the model when providing predictions for HO and BT1 seems to be good, it is argued that the model passes the subjective evaluation and thus fulfills C3. However, it is emphasized that one should remain cautious when applying it, as its BT rankings illustrate its volatile performance,

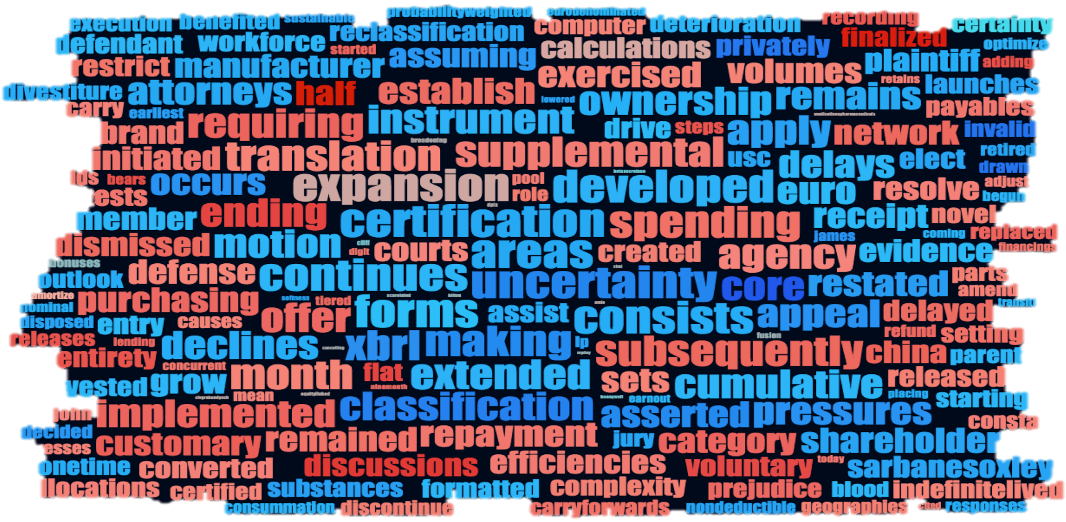
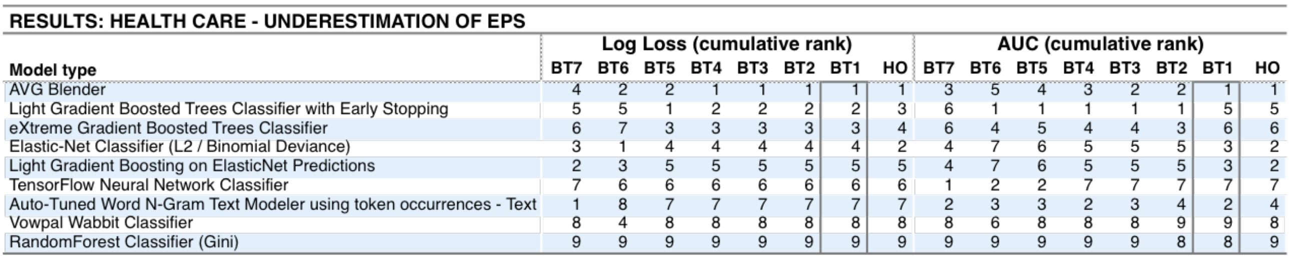


Figure 4.14: Word cloud (Health Care - Overestimation of EPS)

An examination of the word cloud of the *Auto-Tuned Word N-Gram Text Modeler* reveals that “half” and “finalized” have the highest coefficients (1.00 and 0.9497, respectively). Even though “half” has the highest coefficient, the conclusion in regards to its use is the same as the one drawn in Section 4.2 on the Consumer Discretionary sector: there is no intuitive explanation as to why this word shows such a discriminative value. An analysis of contexts in which the word “finalized” is used reveals that the word is mostly used in conjunction with allocation of fair value of acquisitions[29]. Thus, a hypothesis could be that the upcoming fair value adjustments of acquisitions hold a high level of complexity and limited visibility in regards to timing, which could cause analysts to underestimate the negative impact on EPS that such adjustments could entail when eventually appearing. Thereby analysts would overestimate EPS.

4.5.2 Underestimation of EPS

Out of the nine constructed models, eight fulfill C1 and C2 (see Table 4.2). Overall results indicate that the best performing model is the *AVG Blender*, which is an ensemble model of the *Light Gradient Boosted Trees Classifier with Early Stopping* and an *Elastic-Net Classifier (L2 / Binomial Deviance)*. As it appears in the cumulative ranking of the models, depicted in Table 4.21, this model is superior in terms of both average BT Log Loss and average BT AUC score (the model ends up ranking as number one on both metrics). Furthermore, the model upholds this rank when model performance on predicting the HO observations is included.



RESULTS: HEALTH CARE - UNDERESTIMATION OF EPS

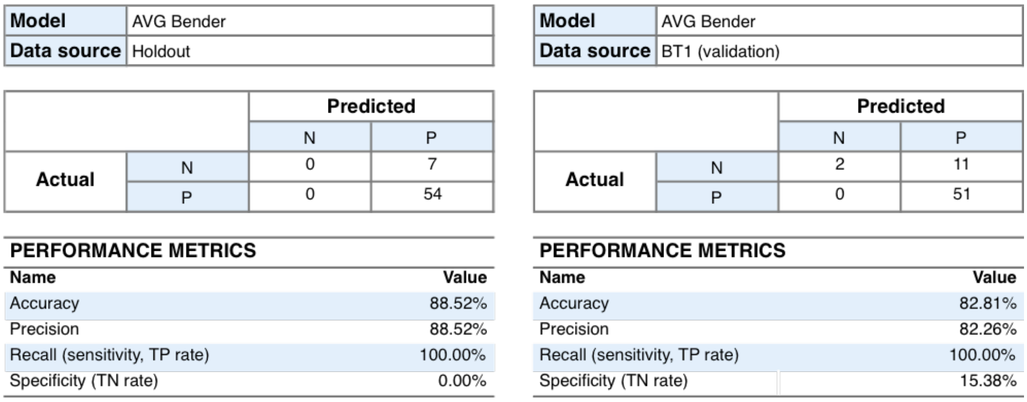
	Log Loss (cumulative rank)								AUC (cumulative rank)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
AVG Blender	4	2	2	1	1	1	1	1	3	5	4	3	2	2	1	1
Light Gradient Boosted Trees Classifier with Early Stopping	5	5	1	2	2	2	2	3	6	1	1	1	1	1	5	5
eXtreme Gradient Boosted Trees Classifier	6	7	3	3	3	3	3	4	6	4	5	4	4	3	6	6
Elastic-Net Classifier (L2 / Binomial Deviance)	3	1	4	4	4	4	4	2	4	7	6	5	5	5	3	2
Light Gradient Boosting on ElasticNet Predictions	2	3	5	5	5	5	5	5	4	7	6	5	5	5	3	2
TensorFlow Neural Network Classifier	7	6	6	6	6	6	6	6	1	2	2	7	7	7	7	7
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	1	8	7	7	7	7	7	7	2	3	3	2	3	4	2	4
Vowpal Wabbit Classifier	8	4	8	8	8	8	8	8	8	6	8	8	8	9	9	8
RandomForest Classifier (Gini)	9	9	9	9	9	9	9	9	9	9	9	9	9	8	8	9

Table 4.21: Cumulate rank of models (Health Care - Underestimation of EPS)

When testing the *AVG Blender* on the holdout data, it performs no better than a majority classifier would: both have an accuracy score of 88.52% (see Figure 4.15 below). However, when examining the AUC scores obtained on the holdout sets by all models, it appears that all, but one, have scores below their average BT AUC scores when providing predictions on the holdout observations (see Table 4.2 above), and that four models even have AUC scores

[29] An example of its use include e.g. *“the allocation of the fair value of the acquisition will be finalized when the valuation is completed”*

that are below their entire BT range (including the *AVG Blender*). Thus, it might be that the outcomes of this quarter were particularly difficult to predict.



Model	AVG Bender
Data source	Holdout

		Predicted	
		N	P
Actual	N	0	7
	P	0	54

PERFORMANCE METRICS

Name	Value
Accuracy	88.52%
Precision	88.52%
Recall (sensitivity, TP rate)	100.00%
Specificity (TN rate)	0.00%

Model	AVG Bender
Data source	BT1 (validation)

		Predicted	
		N	P
Actual	N	2	11
	P	0	51

PERFORMANCE METRICS

Name	Value
Accuracy	82.81%
Precision	82.26%
Recall (sensitivity, TP rate)	100.00%
Specificity (TN rate)	15.38%

Figure 4.15: Confusion matrices of *AVG Blender* - holdout and BT1 (Health Care - Underestimation of EPS)

As can be derived from Figure 4.15 above, the *AVG Blender* is capable of assigning negative predictions in BT1 even though the majority class still holds 79.69% of all observations. Thereby the model outperforms a majority classifier by approximately 3 pp in terms of accuracy in this BT. It is therefore emphasized that even when predicting a difficult quarter (the holdout), the *AVG Blender* performs as good as a majority classifier in terms of accuracy. Thus, even when the model performs its worst in terms of AUC score, its performance is moderate.

Based on the analysis provided above, it is concluded that the *AVG Blender*, besides fulfilling C1 and C2, fulfills the subjective C3.

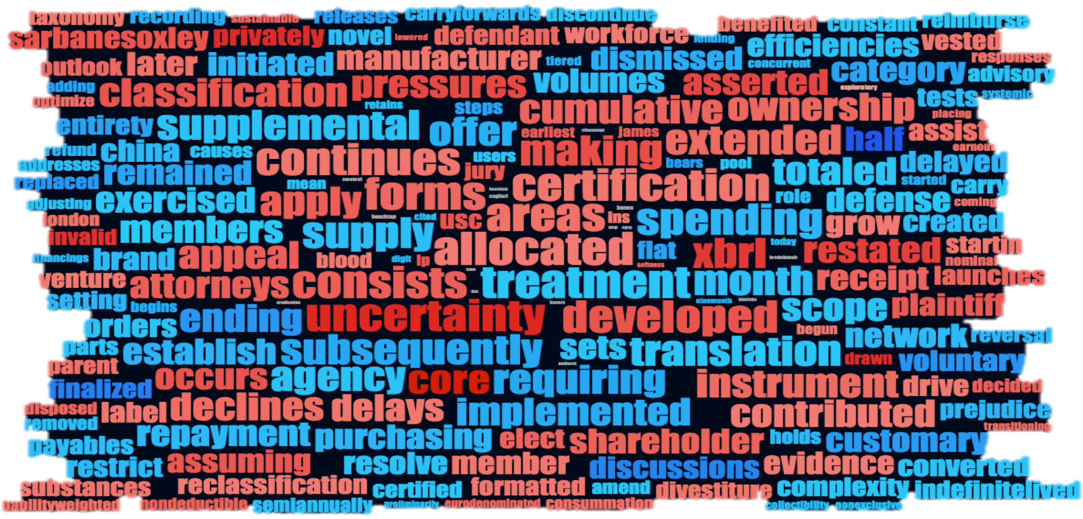


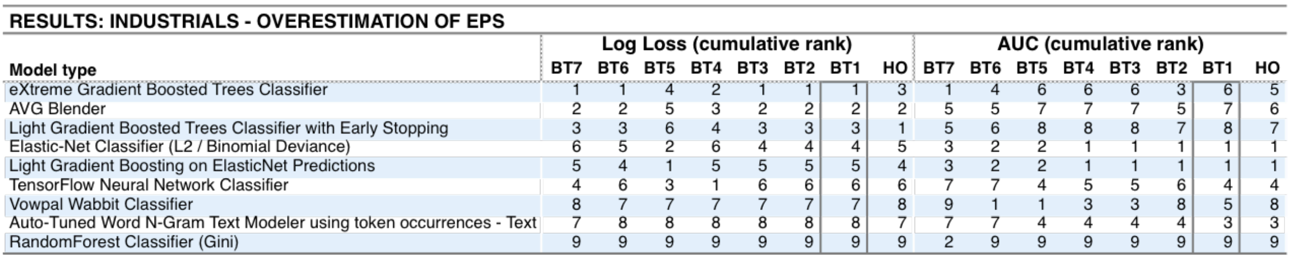
Figure 4.16: Word Cloud (Health Care - Underestimation of EPS)

While the *Auto-Tuned Word N-Gram Text Modeler* does not outperform the other models in terms of Log Loss and AUC score, it still fulfills both C2 and C3, and thus, further analysis of its outcome will be included. From the word cloud, depicted in Figure 4.16, it follows that the words, “core” and “uncertainty” have the highest coefficients (0.7708 and 0.6625, respectively). Further examination reveals that “core” is used in a variety of conflicting contexts, and thus, no proper explanation for its highly indicative coefficient value can be provided. From an examination of the filings it appears that examples of the use of “uncertainty” include e.g. *“the effect of the continuing worldwide macroeconomic uncertainty on our business and results of operations”* and *“in addition brexit could lead to legal uncertainty”*. Thus, it could be argued that the word is used in the kind of statements that might cause analysts to be cautious about future estimates, which could results in too conservative estimates and thus, underestimations of EPS.

4.6 Industrials

4.6.1 Overestimation of EPS

All models except one fulfill C1 and C2 (see Table 4.1). In terms of average BT Log Loss, the *eXtreme Gradient Boosted Trees Classifier* outperforms the other models (see Table 4.22 below). However, when observing the cumulative ranking based on average BT AUC score, the performance of the model appears to be poorer than the performance of its peers (ranking as number 6). Further examination reveals that none of the constructed models seem to have consistently good performance across Log Loss and AUC score, as models ranking as number one, two, and three in terms of average BT Log Loss, rank as number six, seven, and eight, respectively, in terms of average BT AUC score.

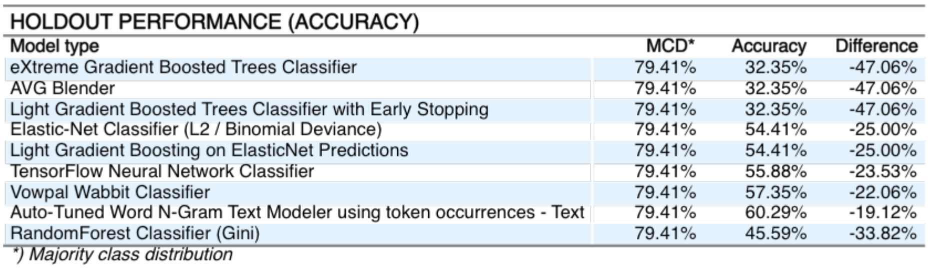


RESULTS: INDUSTRIALS - OVERESTIMATION OF EPS

	Log Loss (cumulative rank)								AUC (cumulative rank)							
Model type	BT7	BT6	BT5	BT4	BT3	BT2	BT1	HO	BT7	BT6	BT5	BT4	BT3	BT2	BT1	HO
eXtreme Gradient Boosted Trees Classifier	1	1	4	2	1	1	1	3	1	4	6	6	6	3	6	5
AVG Blender	2	2	5	3	2	2	2	2	5	5	7	7	7	5	7	6
Light Gradient Boosted Trees Classifier with Early Stopping	3	3	6	4	3	3	3	1	5	6	8	8	8	7	8	7
Elastic-Net Classifier (L2 / Binomial Deviance)	6	5	2	6	4	4	4	5	3	2	2	1	1	1	1	1
Light Gradient Boosting on ElasticNet Predictions	5	4	1	5	5	5	5	4	3	2	2	1	1	1	1	1
TensorFlow Neural Network Classifier	4	6	3	1	6	6	6	6	7	7	4	5	5	6	4	4
Vowpal Wabbit Classifier	8	7	7	7	7	7	7	8	9	1	1	3	3	8	5	8
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	7	8	8	8	8	8	8	7	7	7	4	4	4	4	3	3
RandomForest Classifier (Gini)	9	9	9	9	9	9	9	9	2	9	9	9	9	9	9	9

Table 4.22: Cumulative rank of models (Industrials - Overestimation of EPS)

Model performance on the holdout set is poor for all models. As displayed in Table 4.23 below, none of the models are able to obtain an accuracy score above the one that would have been obtained by simply assigning all observations to the majority class. The *eXtreme Gradient Boosted Trees Classifier* even obtains a score that is 47.06 pp below such a majority classifier score.



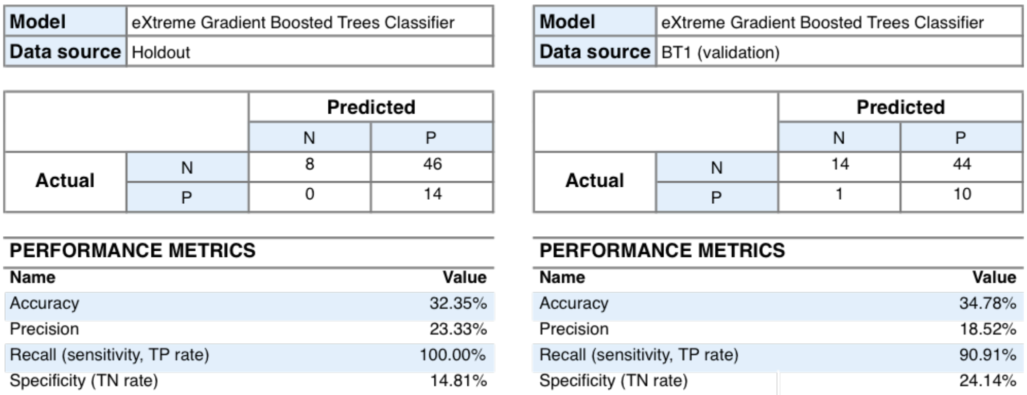
HOLDOUT PERFORMANCE (ACCURACY)

Model type	MCD*	Accuracy	Difference
eXtreme Gradient Boosted Trees Classifier	79.41%	32.35%	-47.06%
AVG Blender	79.41%	32.35%	-47.06%
Light Gradient Boosted Trees Classifier with Early Stopping	79.41%	32.35%	-47.06%
Elastic-Net Classifier (L2 / Binomial Deviance)	79.41%	54.41%	-25.00%
Light Gradient Boosting on ElasticNet Predictions	79.41%	54.41%	-25.00%
TensorFlow Neural Network Classifier	79.41%	55.88%	-23.53%
Vowpal Wabbit Classifier	79.41%	57.35%	-22.06%
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	79.41%	60.29%	-19.12%
RandomForest Classifier (Gini)	79.41%	45.59%	-33.82%

**) Majority class distribution*

Table 4.23: Holdout performance (Industrials - Overestimation of EPS)

Such performance might be a result indicating that it was particularly difficult to provide accurate predictions for the HO. However, as can be seen in Figure 4.17, the performance of the *eXtreme Gradient Boosted Trees Classifier* was equally bad when predicting the outcome of observations in BT1 (49.28 pp below the accuracy that could have been obtained by a majority classifier).



Model	eXtreme Gradient Boosted Trees Classifier
Data source	Holdout

		Predicted	
		N	P
Actual	N	8	46
	P	0	14

PERFORMANCE METRICS	
Name	**Value**
Accuracy	32.35%
Precision	23.33%
Recall (sensitivity, TP rate)	100.00%
Specificity (TN rate)	14.81%

Model	eXtreme Gradient Boosted Trees Classifier
Data source	BT1 (validation)

		Predicted	
		N	P
Actual	N	14	44
	P	1	10

PERFORMANCE METRICS	
Name	**Value**
Accuracy	34.78%
Precision	18.52%
Recall (sensitivity, TP rate)	90.91%
Specificity (TN rate)	24.14%

Figure 4.17: Confusion matrices of *eXtreme Gradient Boosted Trees Classifier* - holdout and BT1 (Industrials - Overestimation of EPS)

Based on the conducted analysis, it is argued that the performance of the *eXtreme Gradient Boosted Trees Classifier* does not display performance sufficient enough to justify its implementation. Thus, it does not fulfill C3.

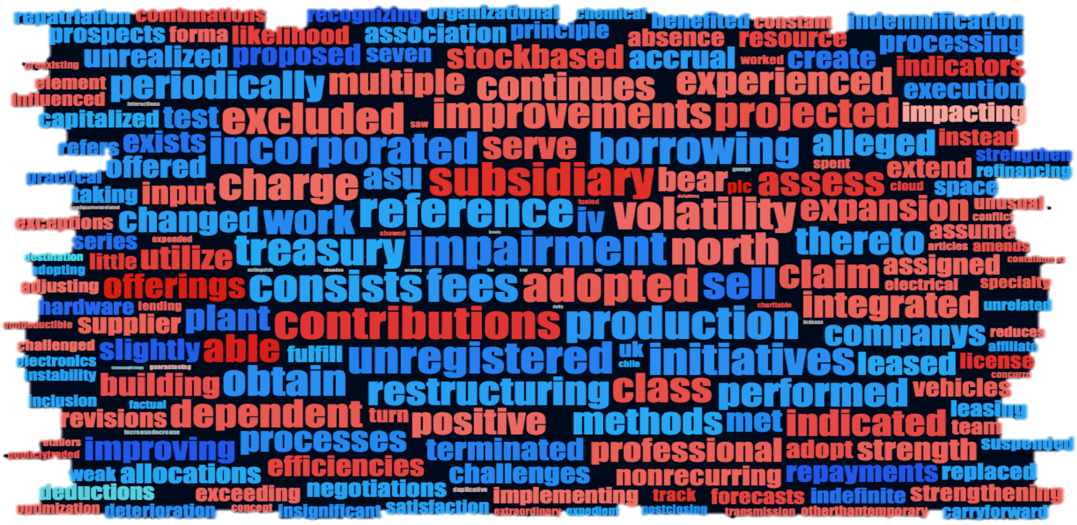


Figure 4.18: Word cloud (Industrials - Overestimation of EPS)

The word cloud available from the output of the *Auto-Tuned Word N-Gram Text Modeler* is examined as the model fulfills both C1 and C2. While the frequency of “plc” is relatively low (the word appears in 246 filings) when compared to other high-coefficient words such as “offerings” or “subsidiary” (they appear in 800 and 568 rows, respectively), it is the word with the highest coefficient (1.00). Its occurrences are therefore examined. “plc” is the legal abbreviation of public limited company[30], and it is generally used in UK company names the same way that "Ltd." and "Inc." are used in US company names. The coefficient of the word could be a result of Brexit: US companies consolidate the earnings of their UK subsidiaries, and the coefficient of "plc" could indicate that analysts underestimate the effect that the depreciation of the British Pound, resulting from Brexit and the election of President Trump, has had on the EPS of the US parent company. However, in regards to these findings it must be emphasized that, even though the effect lags over time due to hedging activities, Brexit is a non recurring event. Thus, one should thus be cautious in relying on the coefficient value of "plc" in future predictions.

"offerings" has a coefficient of 0.9451. In some cases, it is used in conjunction with the elaboration of the effect of product offerings[31], and in some cases it is merely used when describing how a specific product category is performing[32]. As the use of the word varies such, no intuitive reason as to why it seems to be indicative of overestimations of EPS can be provided.

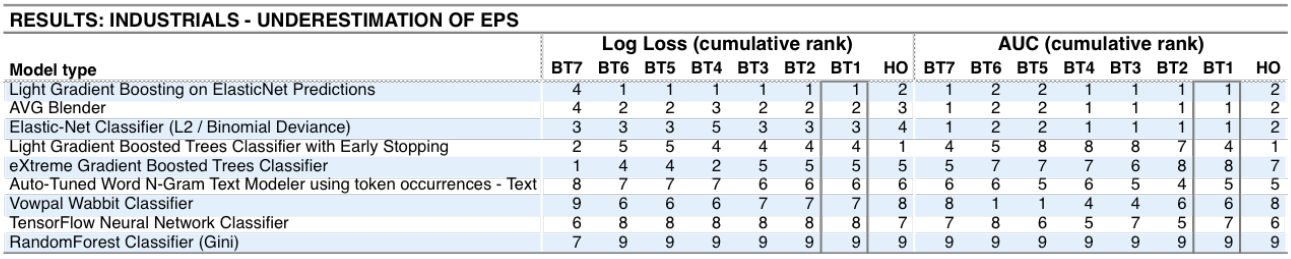
4.6.2 Underestimation of EPS

All models except one fulfill both the significance test (C1) and the first criterion in the applicability test (C2) (see Table 4.2). Based on the average BT Log Loss of the models, the *Light Gradient Boosting on ElasticNet Predictions* model performs best with a cumulative average of 0.53052 (see Table 4.24). Likewise, the model has the highest average BT AUC score.

[30] A public limited company has shares that are publicly available and the company has allotted share capital with a nominal value of at least £50,000 (Thomson Reuters Practical Law, 2018)

[31] Examples include e.g. *"revenue ... continued to experience solid organic recurring and nonrecurring growth led by our automotive product offerings"* and *"we believe that our comprehensive global coverage and product and service offerings provide a competitive advantage"*

[32] Examples include e.g. *"our chemicals and opis product offerings continue to perform well in the fourth quarter"*



RESULTS: INDUSTRIALS - UNDERESTIMATION OF EPS

	Log Loss (cumulative rank)								AUC (cumulative rank)							
Model type	BT7	BT6	BT5	BT4	BT3	BT2	BT1	HO	BT7	BT6	BT5	BT4	BT3	BT2	BT1	HO
Light Gradient Boosting on ElasticNet Predictions	4	1	1	1	1	1	1	2	1	2	2	1	1	1	1	2
AVG Blender	4	2	2	3	2	2	2	3	1	2	2	1	1	1	1	2
Elastic-Net Classifier (L2 / Binomial Deviance)	3	3	3	5	3	3	3	4	1	2	2	1	1	1	1	2
Light Gradient Boosted Trees Classifier with Early Stopping	2	5	5	4	4	4	4	1	4	5	8	8	8	7	4	1
eXtreme Gradient Boosted Trees Classifier	1	4	4	2	5	5	5	5	5	7	7	7	6	8	8	7
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	8	7	7	7	6	6	6	6	6	6	5	6	5	4	5	5
Vowpal Wabbit Classifier	9	6	6	6	7	7	7	8	8	1	1	4	4	6	6	8
TensorFlow Neural Network Classifier	6	8	8	8	8	8	8	7	7	8	6	5	7	5	7	6
RandomForest Classifier (Gini)	7	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9

Table 4.24: *Cumulative rank of models (Industrials - Underestimation of EPS)*

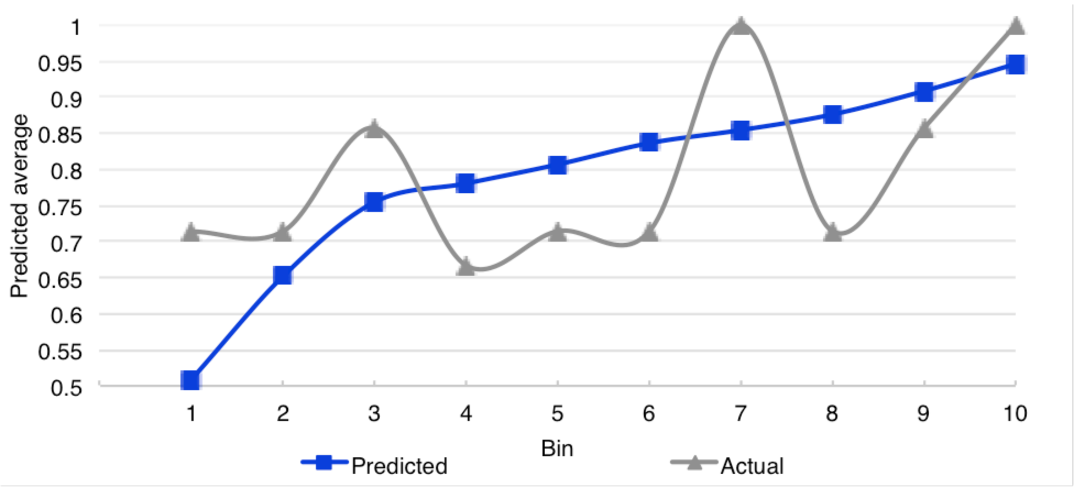


Figure 4.19: Lift chart of *Light Gradient Boosting on ElasticNet Predicions* (Industrials - Underestimation of EPS)

The lift chart of the model reveals that it performs well when assigning its highest probabilities. This is confirmed by its accuracy score on the HO, which is slightly (1.47 pp) better than the accuracy that could be obtained by a majority classifier, and it indicates that the model is able to correctly identify observations belonging to the minority class.

Generally, the average AUC scores of the models in this construction are in the low end (when compared to the AUC scores of models in other model constructions), and the Log Losses are mediocre. Hence, it could be argued that the models face a difficult predictive task, but as the analysis above shows, the *Light Gradient Boosting on ElasticNet Predictions* model does well when performing it. Thus, the model is concluded to fulfill C3.

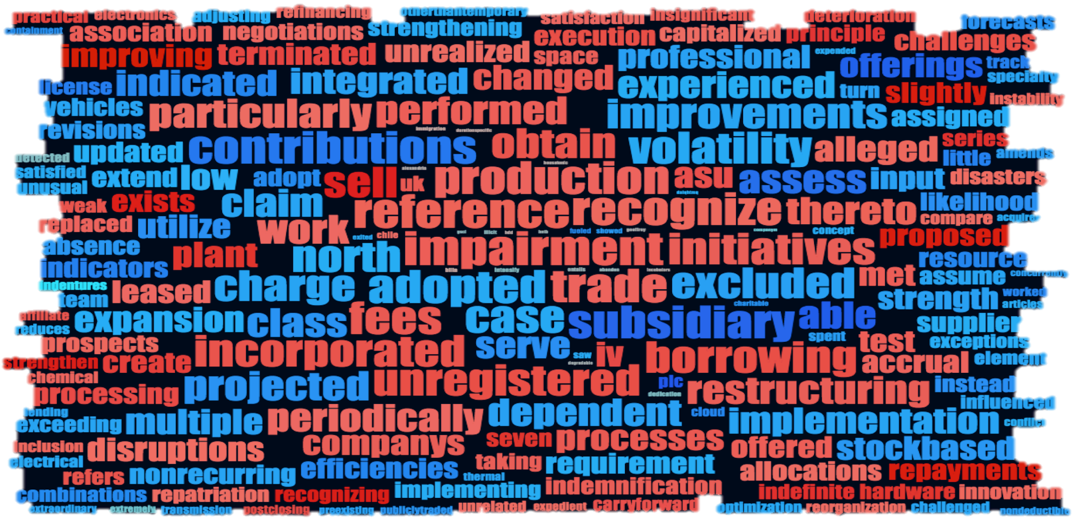


Figure 4.20: Word cloud (Industrials - Underestimation of EPS)

The word cloud of the *Auto-Tuned Word N-Gram Text Modeler* shows that the words with the highest coefficients are “improving” (1.00) and “slightly” (0.9545). “improving” appears in 538 filings, and it is generally used to (as the meaning of the word indicates) describe the intention to do or to make something better (examples include product quality, manufacturing efficiency, performance-to-cost ratio, operating results, the handling of customers etc.). Thus, possible interpretations might be that analysts either underestimate the ability of companies to execute intended improvements, or that they underestimate the positive effect of such improvements on EPS. “slightly” is used in relation to both increases and decreases in e.g. costs, profit, sales, and revenue[33]. As the use of the word varies such, no intuitive reason as to why “slightly” seems to be indicative of underestimations of EPS can be provided. “repayments” has the third highest coefficient (0.9312), indicating that it is similarly highly discriminative of underestimations. The word is oftentimes used to explain repayments of debt[34]. Thus, the fact that analysts tend to underestimate EPS, when

[33] Examples include e.g. *“payroll costs declined slightly primarily due to a lower management incentive award”*, *“air product volume decreased slightly after five previous quarters of double digit growth”*, *“gross margin declined slightly to and for the three and nine months ended september respectively”*, and *“we expect flat to slightly negative revenue growth from residential building applications over the balance of the year”*

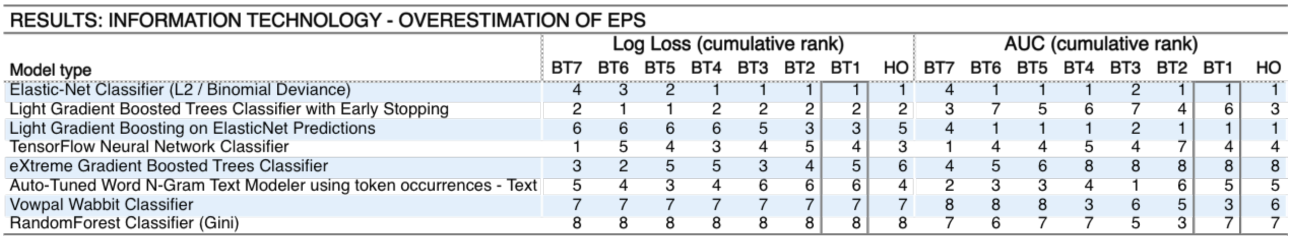
[34] Examples include e.g. *“the change in net cash used in financing activities is primarily due to ... XX million of dividend payments made to ordinary shareholders during the three months ended march and XX million of repayments of our longterm debt”* and *“in the current year we made XX million net repayments on the senior secured credit facilities and XX million of repayments of other borrowings”*

"repayments" is used in a report, could indicate that analysts do not grasp the magnitude of the effect that lowering debt, and thereby financing costs, will have on EPS.

4.7 Information Technology

4.7.1 Overestimation of EPS

All models but two fulfill criteria C1 and C2. The *Elastic-Net Classifier (L2 / Binomial Deviance)* displays performance superior to that of the additional models, ranking as number one in the cumulative ranking in terms of average Log Loss. Likewise, the model ranks as number one in terms of average BT AUC score (see Table 4.25).



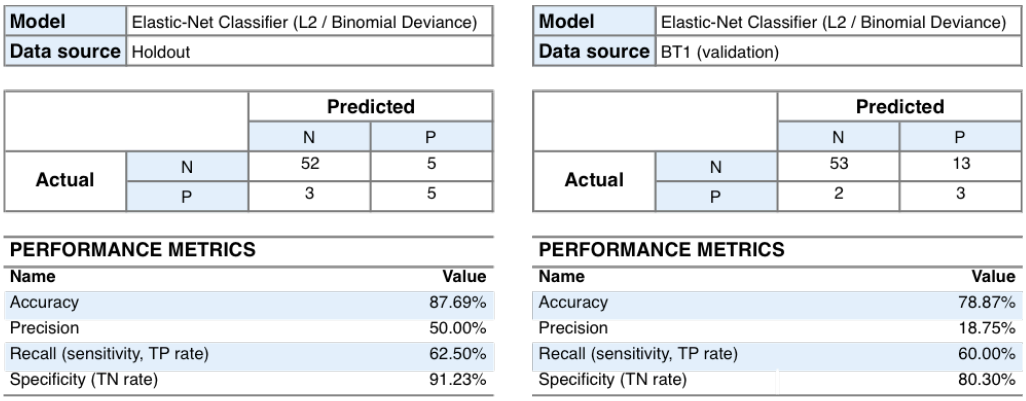
RESULTS: INFORMATION TECHNOLOGY - OVERESTIMATION OF EPS

	Log Loss (cumulative rank)								AUC (cumulative rank)							
Model type	BT7	BT6	BT5	BT4	BT3	BT2	BT1	HO	BT7	BT6	BT5	BT4	BT3	BT2	BT1	HO
Elastic-Net Classifier (L2 / Binomial Deviance)	4	3	2	1	1	1	1	1	4	1	1	1	2	1	1	1
Light Gradient Boosted Trees Classifier with Early Stopping	2	1	1	2	2	2	2	2	3	7	5	6	7	4	6	3
Light Gradient Boosting on ElasticNet Predictions	6	6	6	6	5	3	3	5	4	1	1	1	2	1	1	1
TensorFlow Neural Network Classifier	1	5	4	3	4	5	4	3	1	4	4	5	4	7	4	4
eXtreme Gradient Boosted Trees Classifier	3	2	5	5	3	4	5	6	4	5	6	8	8	8	8	8
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	5	4	3	4	6	6	6	4	2	3	3	4	1	6	5	5
Vowpal Wabbit Classifier	7	7	7	7	7	7	7	7	8	8	8	3	6	5	3	6
RandomForest Classifier (Gini)	8	8	8	8	8	8	8	8	7	6	7	7	5	3	7	7

Table 4.25: Cumulative rank of models (Information Technology - Overestimation of EPS)

In spite of the relatively high AUC score of 0.7610 obtained by the model when predicting the HO, the corresponding accuracy score of 87.69% is no better than the score that could have been obtained by a majority classifier. Further analysis reveals that the five observations assigned the highest probabilities by the model have actual negative outcomes (they are predicted to be cases of overestimations but turn out not to be). Thus, trusting the model predictions in relation to these five predictions for the holdout observations would result in wrong downward shifts of estimates, decreasing the accuracy of the analysts. However, the threshold maximizing the F_1-score causes the model to assign positive predictions to the following observations as well (the observations with the sixth to tenth highest assigned probabilities), and thereby the model accuracy ends up being equal to what a majority classifier could have obtained.

The confusion matrix depicting the distribution of the HO predictions in Figure 4.21 shows that, in total, five estimates would be wrongfully shifted downwards while five would be correctly shifted downwards, and an analyst acting according to the model predictions would neither increase not increase his accuracy by doing so. However, the confusion matrix depicting the model's BT1 predictions shows that following these predictions would result in 13 wrongful upwards shifts of estimates and only three correct ones. Thus, as the number of false positive predictions exceeds the number of true positive predictions, the accuracy of an analyst acting according to the model predictions would be lower than if he had not acted accordingly.



Model	Elastic-Net Classifier (L2 / Binomial Deviance)
Data source	Holdout

		Predicted	
		N	P
Actual	N	52	5
	P	3	5

PERFORMANCE METRICS

Name	**Value**
Accuracy	87.69%
Precision	50.00%
Recall (sensitivity, TP rate)	62.50%
Specificity (TN rate)	91.23%

Model	Elastic-Net Classifier (L2 / Binomial Deviance)
Data source	BT1 (validation)

		Predicted	
		N	P
Actual	N	53	13
	P	2	3

PERFORMANCE METRICS

Name	**Value**
Accuracy	78.87%
Precision	18.75%
Recall (sensitivity, TP rate)	60.00%
Specificity (TN rate)	80.30%

Figure 4.21: Confusion matrices of *Elastic-Net Classifier (L2 / Binomial Deviance)* - holdout and BT1 (Information Technology - Overestimation of EPS)

Overall, the performance of the *Elastic-Net Classifier (L2 / Binomial Deviance)* seems to be quite volatile (BT Log Loss and AUC ranges of 0.19835-0.61790 and 0.55873-0.92708, respectively), and its predictive performance is not impressive in neither BT1 nor HO. Thus, it is argued that the model is not suited for implementation in practice, and that it does not fulfill C3.

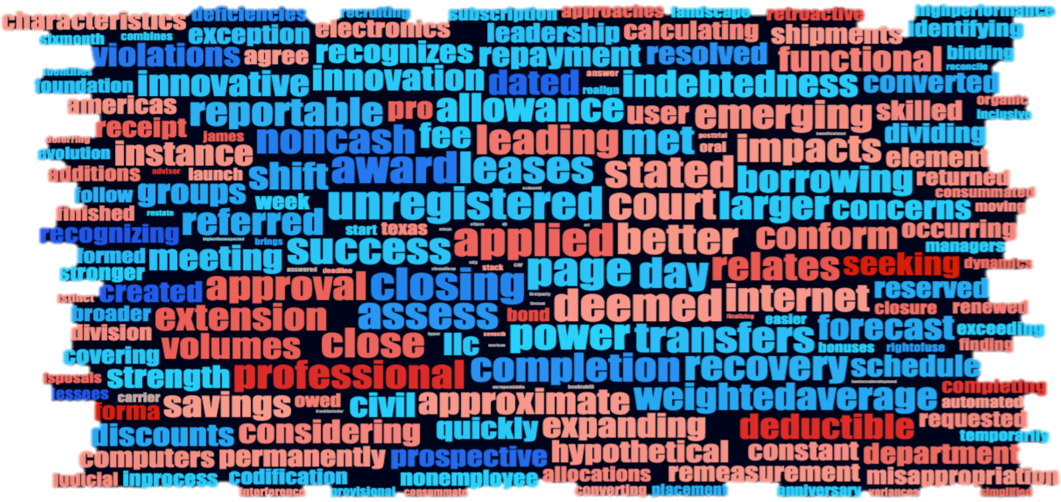


Figure 4.22: Word cloud (Information Technology - Overestimation of EPS)

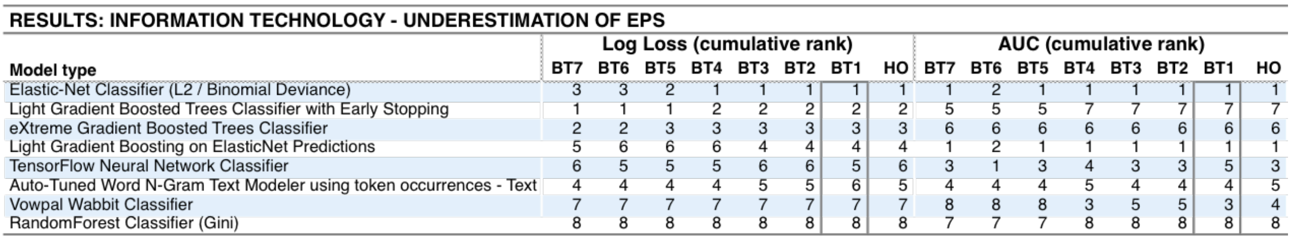
By examining the word cloud of the *Auto-Tuned Word N-Gram Text Modeler*, which fulfills C1 and C2, it is found that the word “seeking” has the highest coefficient (0.5493) closely followed by “forma”, which is has a coefficient of 0.5328. However, the level of the coefficients must be emphasized, as neither is high when compared to previously described words.

Further analysis reveals that “seeking” is generally used when addressing legal issues[35]. Thus, the coefficient of the word could indicate that analysts tend to underestimate the negative effect on EPS of legal proceedings. “forma” appears as part of the expression “pro forma”, which is used to refer to e.g. *“pro forma disclosures”*, *“pro forma results of operations for certain acquisitions”*, *“pro forma deferred tax assets”*, and *“pro forma adjustments of net tax”*. An explanation for the high coefficient of the word might be that analysts do not grasp that companies are positively biased when they use “forma” (“pro forma”) to depict how they perceive underlying trends, which causes analysts to end up overestimating EPS.

[35] Examples include e.g. *“seeking damages”*, *“seeking a declaration that it acted lawfully”*, *“a claim seeking to enforce promises that oracle relied upon in providing services”*, *“new subpoenas from ofac seeking additional information about certain of these transactions”*, *“the doj is also seeking information regarding the company s global fcpa compliance program”*, and *“we have received subpoenas from the us department of justice doj seeking the production of certain information related to our historical antimoney laundering program”*

4.7.2 Underestimation of EPS

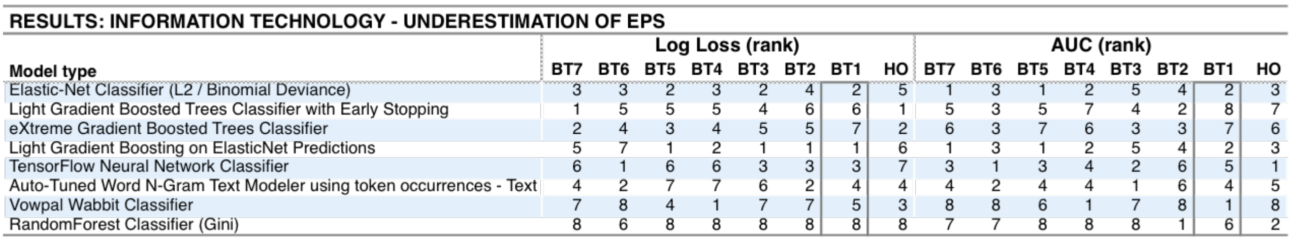
In this model construction, all but one model fulfill C1 and C2. When comparing the models, the *Elastic-Net Classifier (L2 / Binomial Deviance)* is found to consistently outperform its peers. As displayed in the cumulative ranking in Table 4.27, the model ranks number one in terms of both average BT Log Loss and average BT AUC score. The *Elastic-Net Classifier (L2 / Binomial Deviance)* still upholds its cumulative ranking as number one in terms of both Log Loss and AUC when providing predictions for the holdout, even though its actual rankings for this particular prediction set are five (Log Loss) and three (AUC score) (see Table 4.27).



RESULTS: INFORMATION TECHNOLOGY - UNDERESTIMATION OF EPS

	Log Loss (cumulative rank)								AUC (cumulative rank)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
Elastic-Net Classifier (L2 / Binomial Deviance)	3	3	2	1	1	1	1	1	1	2	1	1	1	1	1	1
Light Gradient Boosted Trees Classifier with Early Stopping	1	1	1	2	2	2	2	2	5	5	5	7	7	7	7	7
eXtreme Gradient Boosted Trees Classifier	2	2	3	3	3	3	3	3	6	6	6	6	6	6	6	6
Light Gradient Boosting on ElasticNet Predictions	5	6	6	6	4	4	4	4	1	2	1	1	1	1	1	1
TensorFlow Neural Network Classifier	6	5	5	5	6	6	5	6	3	1	3	4	3	3	5	3
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	4	4	4	4	5	5	6	5	4	4	4	5	4	4	4	5
Vowpal Wabbit Classifier	7	7	7	7	7	7	7	7	8	8	8	3	5	5	3	4
RandomForest Classifier (Gini)	8	8	8	8	8	8	8	8	7	7	7	8	8	8	8	8

Table 4.26: Cumulative rank of models (Information Technology - Underestimation of EPS)



RESULTS: INFORMATION TECHNOLOGY - UNDERESTIMATION OF EPS

	Log Loss (rank)								AUC (rank)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
Elastic-Net Classifier (L2 / Binomial Deviance)	3	3	2	3	2	4	2	5	1	3	1	2	5	4	2	3
Light Gradient Boosted Trees Classifier with Early Stopping	1	5	5	5	4	6	6	1	5	3	5	7	4	2	8	7
eXtreme Gradient Boosted Trees Classifier	2	4	3	4	5	5	7	2	6	3	7	6	3	3	7	6
Light Gradient Boosting on ElasticNet Predictions	5	7	1	2	1	1	1	6	1	3	1	2	5	4	2	3
TensorFlow Neural Network Classifier	6	1	6	6	3	3	3	7	3	1	3	4	2	6	5	1
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	4	2	7	7	6	2	4	4	4	2	4	4	1	6	4	5
Vowpal Wabbit Classifier	7	8	4	1	7	7	5	3	8	8	6	1	7	8	1	8
RandomForest Classifier (Gini)	8	6	8	8	8	8	8	8	7	7	8	8	8	1	6	2

Table 4.27: Rank of models (Information Technology - Underestimation of EPS)

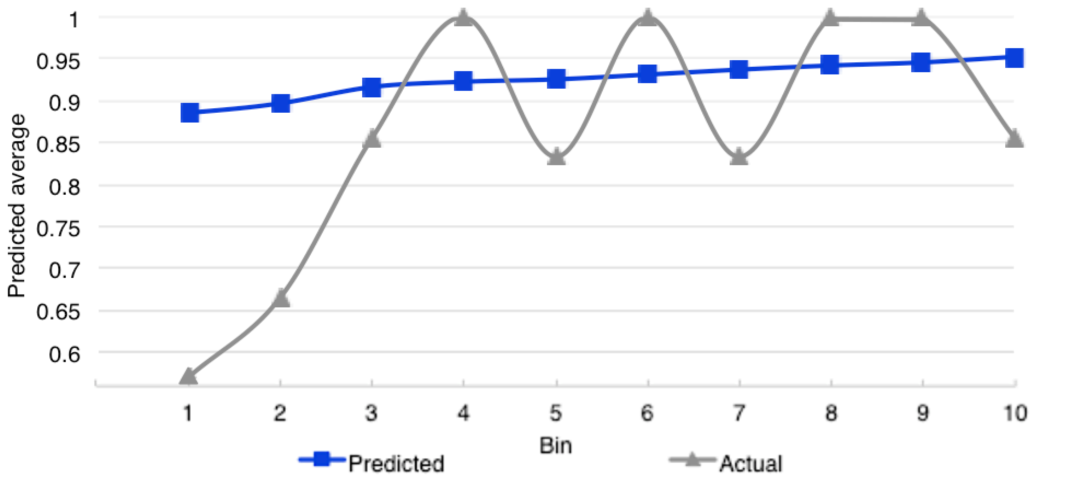


Figure 4.23: Lift chart *of Elastic-Net Classifier (L2 / Binomial Deviance)* (Information Technology - Underestimation of EPS)

The lift chart of the model illustrates that the 10% of the observations assigned with the lowest probabilities are the observations with the lowest average actual outcome. While this seems positive, it is emphasized that the average actual outcome of these observations is still above 0.5, meaning that more than half of the observations have positive actual outcomes. Assigning negative predictions to all of these observations, and thereby arguing to leave the analyst estimates unchanged, will thus result in more false than true predictions, and an accuracy score lower than could have been achieved by a majority classifier stating that all estimates should be shifted upwards. However, further investigation reveals that out of the eight observations that are assigned the lowest probabilities (predicted not to be cases of underestimations), five turn out to be true. Thus, it seems like the model is able to identify observations belonging to the minority class. The threshold that optimizes the F_1-score for the *Elastic-Net Classifier (L2 / Binomial Deviance)* is 0.8901, which means that the probability assigned by the model must be above 89.01% for an observation to be classified as an underestimation. The holdout accuracy of 87.69% (1.54 pp above the majority class distribution) achieved with this threshold illustrates that the model has the ability to capture true negatives and thus, it outperforms a majority classifier.

Due to the consistency of the rankings of the model and its demonstrated ability to capture minority class observations, it is argued that it fulfills C3 as well.

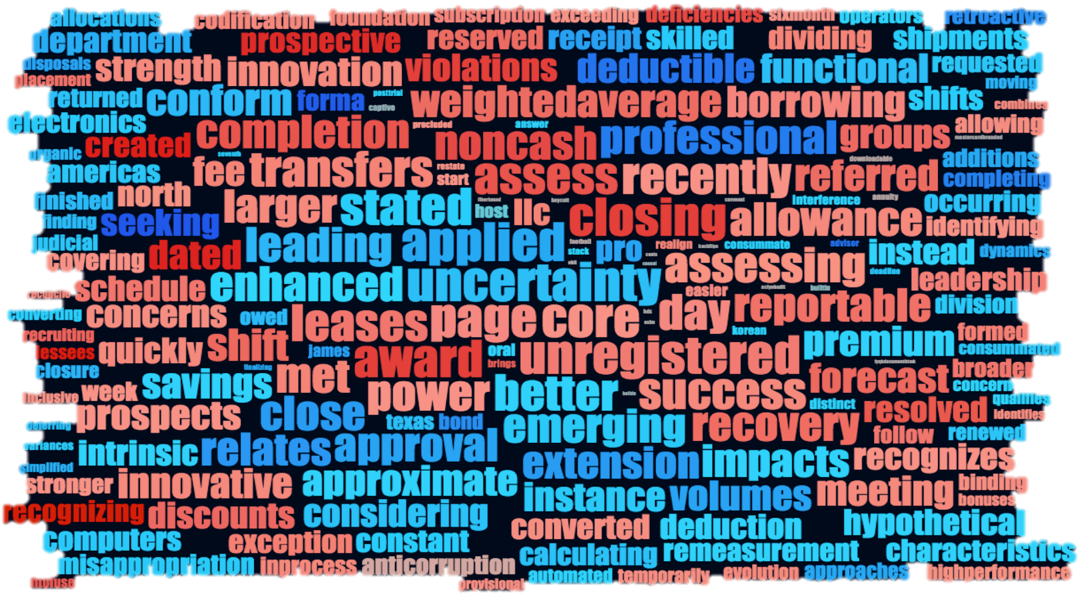


Figure 4.24: Word cloud (Information technology - Underestimation of EPS)

The word cloud of the *Auto-Tuned Word N-Gram Text Modeler* shows that the words with the highest coefficients are "recognizing" (0.5414) and "lessees" (0.5122). As in the overestimation-word cloud, the coefficients of these words are relatively low when compared to previously examined words, but as they are the words with the highest coefficients, they are examined.

An examination of the filings reveals that "recognizing" often refers to the recognition of e.g. revenue, lease assets, impairments, or tax consequences. The use of the word "lessees" coincides with that of "recognizing": further investigation reveals that it is generally used in 2016-reports when referring to IFRS 16[36]. This standard, issued in January 2016, requires lessees to recognize assets and liabilities on their balance sheets for most leases from January 2019 (EY, 2016). The coefficient of the word could possibly result from two things: either, analysts struggle when estimating the companies that have lease as a part of their operations, or they, in cases where the IFRS 16 has already been implemented under early

[36] The legislation was issued by the International Accounting Standards Board (IASB) and the Financial Accounting Standards Board (FASB). It e.g. requires a lessee to classify a lease as either a finance or operating lease in which lessees will need to recognize a right-of-use asset and a lease liability for their leases. While adoption is not required for annual periods beginning before 1 January 2019, early implementation is permitted

adoption, struggle to estimate its effect on EPS, as costs are moved from operating costs to depreciation and finance costs in the P&L statement.

4.8 Findings

The analyses of the results of the twelve model constructions lead to several overall conclusions. The main goal of this study is to examine whether using automated textual analysis of the content of corporate filings can enable analysts to enhance their forecast accuracy. While enhancing accuracy is a term that could be interpreted in multiple ways, the simplest form of the term must be that any enhancement, even the slightest, is an enhancement. Thus, an enhancement would occur if an analyst could identify and apply a predictive model that yields just one more true than false positive prediction, causing him to shift just one more forecast in the right direction than in the wrong direction. Based on the results presented above, the answer to such a question is affirmative: automated textual analysis can be used to enhance the accuracy of analysts' EPS forecasts. As the analyses reveal that several models displayed superior performance and that several were able to outperform majority classifiers, greater enhancements in accuracy are even made possible in some cases. That said, the magnitude of such improvements of accuracy vary across sectors, the type of event one tries to predict, and the models applied.

In terms of model performance within the specific sectors, the analyses reveal that at least one model within each sector complies with all three established criteria. While it seems that fewer high performance models are constructed within some sectors than others, the framework is not concluded to be inapplicable in any of the sectors. However, more caution is recommended in some than in others: overall model assessment (see Table 4.3 above) reveals that the performance of the best ranked models in the model constructions for underestimation for Consumer Staples, overestimation for Industrials, and overestimation for Information Technology is not deemed sufficient for implementation in practice. From further examination in regards to the possible causes for such failures to fulfill the criteria, it appears that, within each of these sectors, the word clouds revealed that event-specific words carried highly indicative coefficients[37]. Thus, it could be argued that, in cases where sectors have been significantly affected by specific events, models might struggle to identify other general patterns. On the other hand, some word clouds revealed words with high coefficients for which explanations based on more general analyst tendencies could be provided. This is the case for e.g. the Health Care sector, the Consumer Discretionary sector, and the Financials sector[38]. The latter sector is the one in which the best AUC scores are obtained (the constructions yield five models with an average BT AUC score above 0.7). However, it is moreover the sector with fewest models fulfilling the criteria (six out of seventeen models are rejected based on either C1 or C2), indicating that the construction of well performing models is not just a result of an easy predictive task. Based on the good results of these particular model constructions, it could be hypothesized that analysts struggle incorporating some of the content of the filings in their forecasts due to e.g. the complexity of the companies in the sector, or the magnitude of the level of information contained in their reports (which could partially be a result of legislative requirements). If the software is able to identify patterns in the content, which analysts fail to (correctly) incorporate, it is more likely to provide good models. Thus, the complexity of the companies and their filings might be the reason that the models within this particular sector perform well.

Predicting outcomes for EPS estimates of companies in the Industrials sector seems, on the other hand, to be a difficult task. The best ranking model of the models constructed for predicting cases of overestimation does not display sufficient performance, and while the *Light Gradient Boosting on ElasticNet Predictions* model is concluded to be applicable for predictions of underestimations, its performance is not as convincing as the performance of

[37] The Consumer Staples underestimation word cloud revealed that "month" could be indicative of an underestimation as analysts to fail to adjust for the fact that the negative impact of e.g. an increase in the interest rate could be offset by the depreciation in the Pound resulting from Brexit. The Industrials overestimation word cloud revealed another possible Brexit-effect ("plc" could be indicative of an overestimation due to the fact that analysts fail to incorporate the negative effect of the depreciation in the Pound when consolidating UK daughter company EPS in US parent company EPS). In regards to Information Technology, the underestimation word cloud showed that "leessees" had a high coefficient indicative of an underestimation, and it was hypothesized that this could be the case because of analysts struggle to estimate the effects of IFRS 16 on EPS

[38] The words "proposed" and "invalid" are often used in legal contexts, and it is hypothesized that analysts make conservative forecasts in case of legal uncertainty. "opening" similarly has a high coefficient, and a possible explanation might be that analysts overestimate the positive effect on EPS of store openings

some of the models constructed on other sector subsamples. Thus, analysts should be cautious when attempting to apply textual analysis for enhancing accuracy within this sector.

Based on the above, it is contended that the complexity of companies and their reports could play a key role in determining when automated textual analysis is most beneficial for analysts: the more complex a forecasting task, the more mistakes are likely to be made, and the more the software is able to assist in improving accuracy by identifying patterns in the information that analysts fail to (correctly) interpret and incorporate. Thus, models might perform better in complex or highly regulated sectors. Furthermore, the models in sectors with companies that are more alike seem to perform better. It is argued that the variation in the content of company filings is greater in a sector such as Industrials than in e.g. Financials. This may partly be because the companies are more different in terms of the services they provide[39], and partly because of the regulatory requirements that financial companies are subject to when compiling a report. Thus, the software might be more capable of identifying patterns if there is less variation in the content of the reports across companies within a sector.

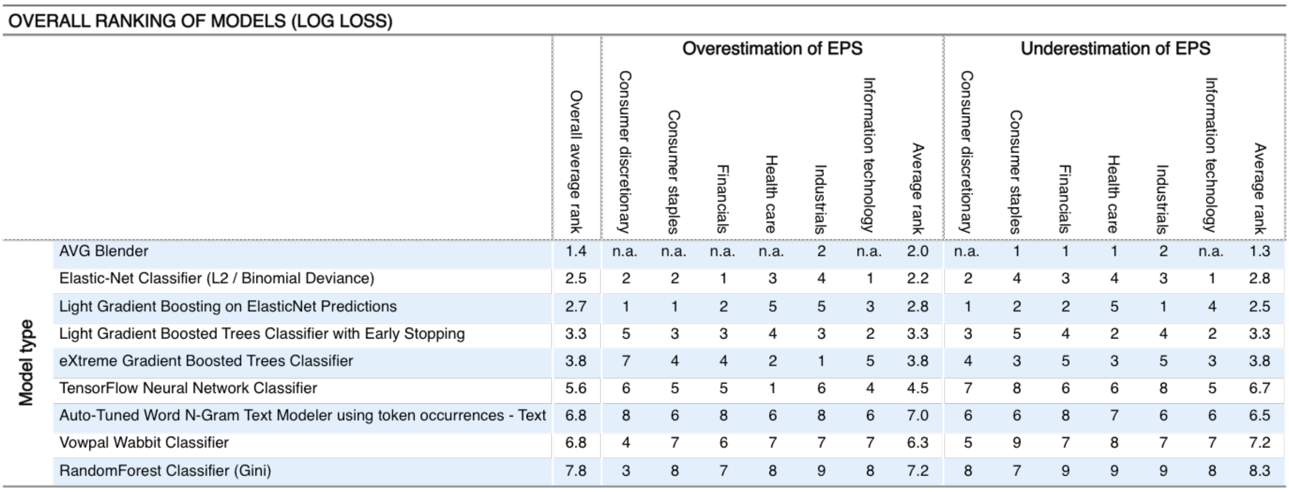
In terms of differences in general model performance of overestimation and underestimation models, respectively, neither is concluded to outperform the other. However, it is emphasized that there are less overestimation observations, and thus, that the chances of finding patterns in these observations might be lower than the chances of finding patterns in underestimation observations.

Furthermore, one could argue that analysts would be able to enhance their accuracy significantly more by choosing to implement underestimation models if they were to choose between the two types. The models are constructed to predict binary outcomes: in regards to underestimation models, the models are to predict whether analysts would enhance the

[39] As goes for Financials companies, one bank is very similar to another bank and an insurance company is similarly to another insurance company. The Industrials sector, on the other hand, includes numerous different types of companies, which are argued to be very different (such as electrical equipment manufacturers and human resource service providers)

accuracy of their estimates by shifting their forecasts upwards, or whether they should avoid changing their forecasts (because consensus overestimates or is equal to the actual EPS). The opposite goes for overestimation models. When relying on such models, an enhancement in accuracy only occurs if the model suggests a downward shift in the EPS forecast. As stated above, analysts have underestimated EPS of companies in approximately 75% of the observations included in the sample, and several models are able to identify these cases (some are even able to identify the cases in which the forecast should not be changed). Thus, implementing an underestimation model (and following its recommendations) would result in more forecast shifts than implementing an overestimation model (even though the accuracy scores of these models might be the same).

While no one type of model is consistently superior across all model constructions, certain types are distinctly better than others - and some worse. From the ranking of the models depicted in Table 4.28, it follows that, in cases where the *AVG Blender* was constructed, it consistently ranked first or second. However, it is emphasized that the *AVG Blender* is only constructed when DataRobot deems that there is a possibility that the model might outperform the models already constructed, and thus, one could argue that such rankings are biased. The fact that the ensemble models do not always rank as number one is consistent with the findings in the literature review in Section 2.2: not all researchers conclude ensemble models to be superior (examples include Tsai and Wu (2008), Geng et al. (2015), and Kirkos et al. (2007)), but in many cases they are (Patel et al., 2015; Whiting et al., 2012).



OVERALL RANKING OF MODELS (LOG LOSS)

			Overestimation of EPS							Underestimation of EPS						
		Overall average rank	Consumer discretionary	Consumer staples	Financials	Health care	Industrials	Information techonology	Average rank	Consumer discretionary	Consumer staples	Financials	Health care	Industrials	Information techonology	Average rank
Model type	AVG Blender	1.4	n.a.	n.a.	n.a.	n.a.	2	n.a.	2.0	n.a.	1	1	1	2	n.a.	1.3
	Elastic-Net Classifier (L2 / Binomial Deviance)	2.5	2	2	1	3	4	1	2.2	2	4	3	4	3	1	2.8
	Light Gradient Boosting on ElasticNet Predictions	2.7	1	1	2	5	5	3	2.8	1	2	2	5	1	4	2.5
	Light Gradient Boosted Trees Classifier with Early Stopping	3.3	5	3	3	4	3	2	3.3	3	5	4	2	4	2	3.3
	eXtreme Gradient Boosted Trees Classifier	3.8	7	4	4	2	1	5	3.8	4	3	5	3	5	3	3.8
	TensorFlow Neural Network Classifier	5.6	6	5	5	1	6	4	4.5	7	8	6	6	8	5	6.7
	Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	6.8	8	6	8	6	8	6	7.0	6	6	8	7	6	6	6.5
	Vowpal Wabbit Classifier	6.8	4	7	6	7	7	7	6.3	5	9	7	8	7	7	7.2
	RandomForest Classifier (Gini)	7.8	3	8	7	8	9	8	7.2	8	7	9	9	9	8	8.3

Table 4.28: Overall ranking of models based on cumulative Log Loss

The average rankings within overestimation and underestimation models are in line with the overall average ranking, besides the fact that the *Light Gradient Boosting on ElasticNet Predictions* model ranks three in overestimation models while the *Elastic-Net Classifier (L2 / Binomial Deviance)* ranks two, and vice versa for underestimation models. Moreover, it follows from Table 4.28 that the model type ranking first in most model constructions is the *Light Gradient Boosting on ElasticNet Predictions*. While it has an overall average rank below that of the *Elastic-Net Classifier (L2 / Binomial Deviance)*, it is the model that (according to the assumptions made on analyst behavior in this study) would be considered for implementation in most cases. However, the volatility in its ranking should be emphasized. As the model is not consistent in its performance across all model constructions, no general recommendation to apply it can be made.

Three model types consistently have lower end rankings across all model constructions: the *RandomForest Classifier (Gini)*, the *Auto-Tuned Word N-Gram Modeler*, and the *Wowpal Wabbit Classifier*. They all rank 6-8 in all model constructions but three (in which they rank three, four, and five). Thus, neither type of model would be implemented once.

As previously emphasized in Section 3.2.1 on DataRobot, this study postulates that no one model type is best for a set of problems, and thus, that choosing the best model for a given problem in advance is not possible. The findings of this analysis support this: in sum, the analyses of the individual model constructions reveal that, while some types of models can generally be rejected based on consistent weak performance, it is not possible to solely recommend one type. Similar findings were made in the literature review in Section 2.2.2, as it appeared that researchers vary in their model choice and model recommendations within and across different predictive tasks.

4.9 Robustness of results

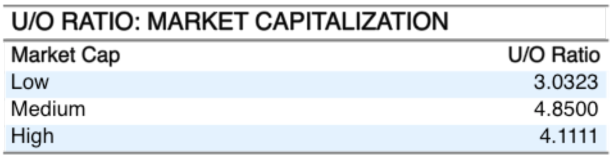
In order to test whether the predictions provided by the models are highly affected by underlying characteristics of the data that have not been included as variables in the modeling, a robustness check is performed.

4.9.1 Firm Size

Analysts are constrained (Clement, 1999), and thus, it is argued that not all companies receive the same level of attention. Investors, who seek to mirror the market, or who are tightly benchmarked to the market, should hold larger investments in companies with larger market capitalization. As such investors are therefore more likely to prefer that analysts focus on large companies, one could argue that analyst attention varies across companies depending on their size. Thus, when constrained analysts choose which companies to cover and how much effort to put into such coverage, it is argued that relatively large companies are likely to receive more analyst attention than relatively small companies. As market capitalization is a widely used measure of company size, it is applied as a proxy for analyst coverage.

Brown et al. (1987) find that there is a positive relationship between the size of a company and analyst forecast accuracy. Hence, one could argue that, if analysts are more accurate when predicting the EPS of larger companies, due to the higher degree of attention allocated to such coverage, the ratio of the number of underestimations to the number of overestimations (U/O ratio) should be close to one for these companies and very different from one for companies of smaller size. The argumentation for such a statement goes as follows: the closer an estimate is to the actual EPS, the higher the probability of slightly overestimating in one quarter and slightly underestimating in another quarter. Thus, it follows that the number of overestimations and the number of underestimations are expected to be roughly equal if analysts are very accurate, which would yield an U/O ratio of approximately one. Given that this holds, automated textual analysis models are expected to have the most confident predictions for observations belonging to the class in which analyst estimates are the least accurate (as measured by the U/O ratio: the group with the highest U/O ratio). In order to test whether this is the case, the HO predictions are labeled as either Low, Medium, or High, depending on whether the market capitalization of the company issuing the filing in question is in the upper, middle, or lower third of the sample.

As depicted in Table 4.29 below, companies belonging to the low market capitalization class have the U/O ratio closest to one. This finding thus contradicts the stated hypothesis of high market capitalization companies having the ratio closest to one.

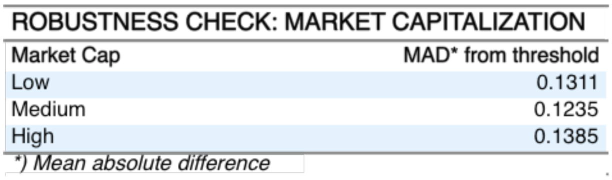


U/O RATIO: MARKET CAPITALIZATION

Market Cap	U/O Ratio
Low	3.0323
Medium	4.8500
High	4.1111

Table 4.29: Ratio of underestimations to overestimations - Market capitalization

To test whether the models are more confident when providing predictions for the class with the highest U/O ratio, a confidence measure, defined as the mean of all absolute differences between assigned probabilities and the thresholds set in the specific models[40], is constructed. From Table 4.30 it follows that, on average, the highest ranked models from each of the model constructions provide the most confident predictions for companies with high market capitalization (meaning that the models assign probabilities furthest from the threshold to high market capitalization companies). The least confident predictions are assigned to the companies with in the low market capitalization class.



ROBUSTNESS CHECK: MARKET CAPITALIZATION

Market Cap	MAD* from threshold
Low	0.1311
Medium	0.1235
High	0.1385

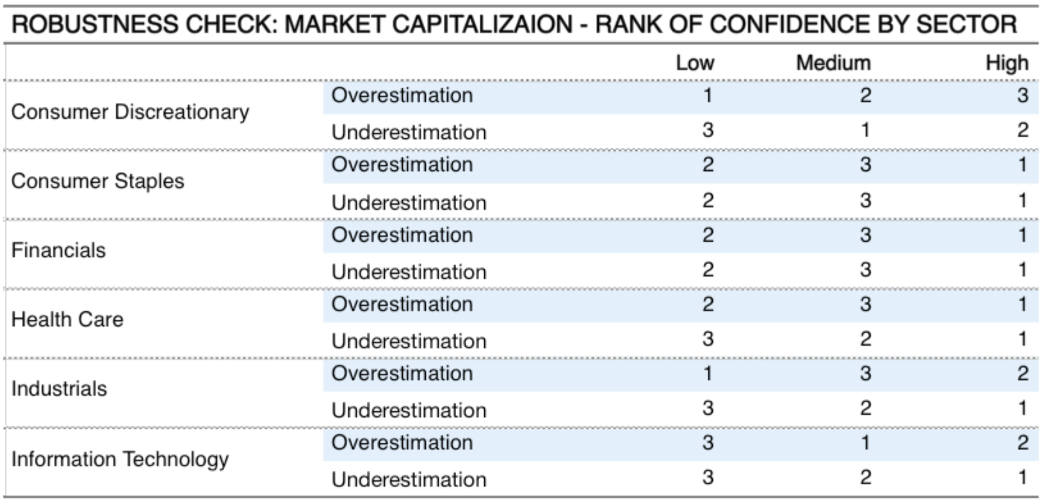
*) Mean absolute difference

Table 4.30: Robustness check - Market capitalization

[40] The model thresholds are chosen as these are the probability levels at which the models are the least confident in their predictions: a small alteration in the probability could result in the model assigning a different prediction

Thus, the expectation that the models would assign more confident predictions (making applications of the models more beneficial), the higher the U/O ratio, does not seem to be fulfilled. These results indicate that one would not benefit more from applying the models when the U/O ratio is high (when analysts are less accurate). However, it should be emphasized that the deviations in the mean absolute difference from the threshold between the market capitalization classes are fairly small, and that the value of using automated textual analysis models does not differ significantly across different market capitalizations classes.

Further analysis of the highest ranking models in each sector subsamples confirms that, in most cases, the most confident predictions are assigned to observations belonging to the high market capitalization class (see Table 4.31). The least confident predictions are, in most cases, assigned to observations belonging to the medium market capitalization class. Despite the tendency to assign slightly more confident predictions to filings of companies with high market capitalization, it must again be emphasized that the differences are minor.



ROBUSTNESS CHECK: MARKET CAPITALIZAION - RANK OF CONFIDENCE BY SECTOR

		Low	Medium	High
Consumer Discreationary	Overestimation	1	2	3
	Underestimation	3	1	2
Consumer Staples	Overestimation	2	3	1
	Underestimation	2	3	1
Financials	Overestimation	2	3	1
	Underestimation	2	3	1
Health Care	Overestimation	2	3	1
	Underestimation	3	2	1
Industrials	Overestimation	1	3	2
	Underestimation	3	2	1
Information Technology	Overestimation	3	1	2
	Underestimation	3	2	1

Table 4.31: Robustness check - Market capitalization - Rank of confidence by sector

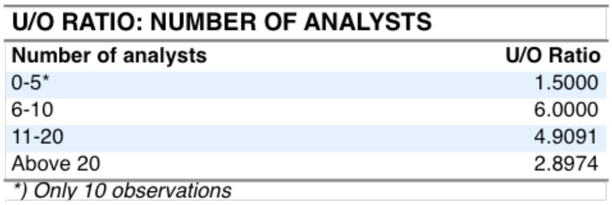
Based on the results of the tests that have been presented in the above, the findings of this study seem to be robust to differences in market capitalization.

4.9.2 Analyst coverage

In the tests performed above, market capitalization is used as a proxy for the amount of attention companies receive. However, one could argue that various reasons could make smaller companies more interesting and appealing to investors (and thereby analysts) than larger companies. If this were the case, market capitalization would not be applicable as a proxy. To overcome such an issue, and to test whether the findings in this study are robust to differences in analyst coverage, a similar robustness test, in which the number of analysts covering a company is used as proxy for company attention, is performed. Lang and Lundholm (1996) find that a positive relationship between the number of analysts covering a firm and analyst forecast accuracy exists. Thus, the U/O ratio should be closer to one for companies with high coverage than for companies with less coverage[41]. In order to perform the robustness test, HO predictions are divided into four classes characterized by the number of analysts covering the company issuing the filing in question: 0-5, 6-10, 11-20 and above 20.

The results of the test are presented in Table 4.32. Companies covered by 0-5 analysts obtain the U/O ratio closest to one. However, it is emphasized that the number of observations in this particular group is low, and that one must not draw any conclusions from such a finding. When disregarding the 0-5 group, further examination of the results reveals that companies covered by more than 20 analysts obtain the U/O ratio closest to one (2.8974), closely followed by the group of companies covered by 11-20 analysts. These findings are, as opposed the findings in the previously conducted test, in line with the expectations: the more attention a company gets (given by analyst coverage), the more accurate analysts are, i.e. the less the number of under- and overestimations differs (the closer to one the U/O ratio is).

[41] The argumentation for such a postulate is the same as for the previously performed test with market capitalization used as proxy

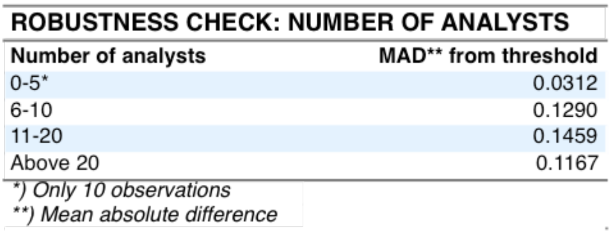


U/O RATIO: NUMBER OF ANALYSTS	
Number of analysts	**U/O Ratio**
0-5*	1.5000
6-10	6.0000
11-20	4.9091
Above 20	2.8974

*) Only 10 observations

Table 4.32: Ratio of underestimations to overestimations - Number of Analysts

Table 4.33 reveals that the models are the least confident when providing predictions based on the filings submitted by companies covered by more than 20 analysts (when disregarding the 0-5 class). This is similarly in line with expectations. However, it is once again emphasized that the deviations between classes are small, indicating the results of this study are fairly robust to the differences in analyst coverage.

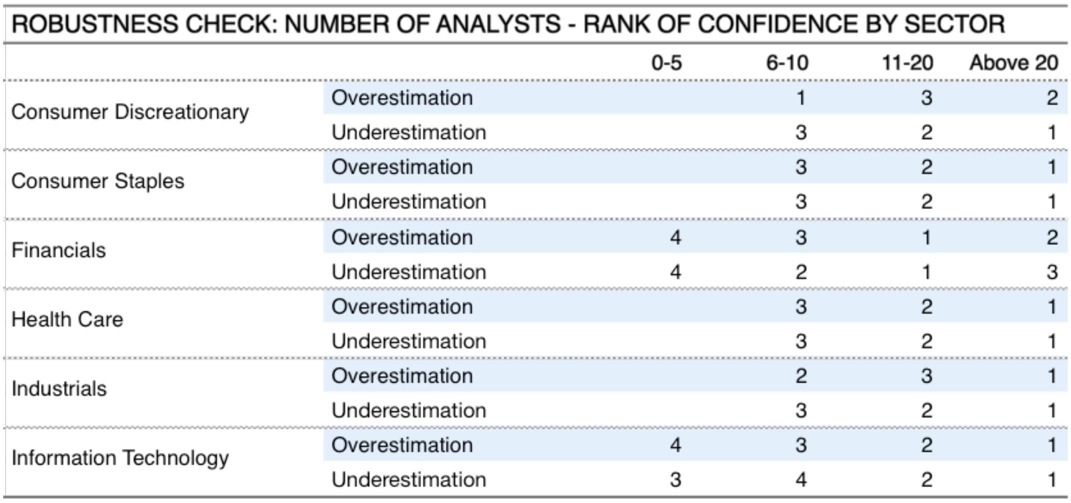


ROBUSTNESS CHECK: NUMBER OF ANALYSTS	
Number of analysts	**MAD** from threshold**
0-5*	0.0312
6-10	0.1290
11-20	0.1459
Above 20	0.1167

*) Only 10 observations
**) Mean absolute difference

Table 4.33: Robustness check - Number of Analysts

The fact that the best performing model in each model construction most often assigns its most confident prediction to the filings of companies covered by more than 20 analysts supports the robustness of the findings. These results, depicted in Table 4.34, contradict the findings above indicating that the models are more confident when providing predictions for companies covered by 11-20 and 6-10 analysts than by more than 20 analysts.



ROBUSTNESS CHECK: NUMBER OF ANALYSTS - RANK OF CONFIDENCE BY SECTOR

		0-5	6-10	11-20	Above 20
Consumer Discreationary	Overestimation		1	3	2
	Underestimation		3	2	1
Consumer Staples	Overestimation		3	2	1
	Underestimation		3	2	1
Financials	Overestimation	4	3	1	2
	Underestimation	4	2	1	3
Health Care	Overestimation		3	2	1
	Underestimation		3	2	1
Industrials	Overestimation		2	3	1
	Underestimation		3	2	1
Information Technology	Overestimation	4	3	2	1
	Underestimation	3	4	2	1

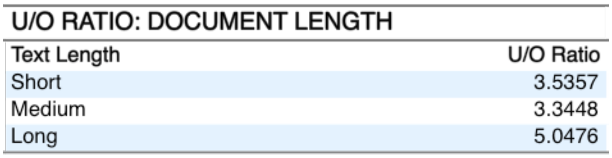
Table 4.34: Robustness check - Number of Analysts - Rank of confidence by sector

As no significant differences in the performance across classes are found, the results in regards to the applicability of automated textual analysis models are considered to be robust to differences in market capitalization and analyst coverage, both used as proxies for company attention. However, the rankings of confidence levels grouped on sectors (Table 4.31 and Table 4.34 above) indicate that models provide relatively more confident predictions for high market capitalization companies and companies covered by more than 20 analysts. Thus, any third factor positively correlated with both characteristics could be driving prediction confidence.

4.9.3 The Amount of Information Available

Hassell et al. (1988) and Lang and Lundholm (1996) find that both the size of a company and the number of analysts covering it are positively correlated with the amount of disclosure available. Thus, as this might be the common factor called for, it is tested whether it has significant impact on the results obtained in this study. To perform such a test, the number of characters in each filing is used as a measure of the amount of disclosure available, and filings are assigned to one of three classes (Short, Medium, or Long), depending on whether the length of the filing in question is in the upper, middle, or lower third of the sample. It has been argued in previous research that analysts incorporate the information available to them (see e.g. Hassell et al. (1988)). Thus, as a higher amount of available disclosure must yield a higher amount of available information, it is hypothesized that the forecasts of analysts are better for companies with high amounts of disclosure available (long reports) than for companies with low amounts of disclosure available. The expectation is similar to the ones in previous tests: the more accurate, analysts are, the closer to one the U/O ratio is.

However, as can be seen in Table 4.35, results are not in line with such an expectation, as the U/O ratio of filings classified as long is the furthest from one, while the medium length filings have the ratio closest to one.

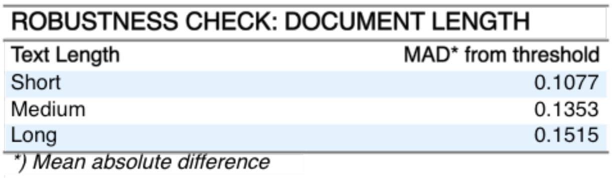


U/O RATIO: DOCUMENT LENGTH

Text Length	U/O Ratio
Short	3.5357
Medium	3.3448
Long	5.0476

Table 4.35: Ratio of underestimations to overestimations - Document length

From Table 4.36 it follows that the models, in line with expectations, provide the most confident predictions for the class with the highest U/O ratio (“Long”). However, the least confident predictions are provided for “Short” filings, while it was expected that this would be the case for the “Medium” filings as analysts, according to the findings above, are more accurate when providing these estimates. The findings indicate that the models perform slightly better, the longer the reports are.



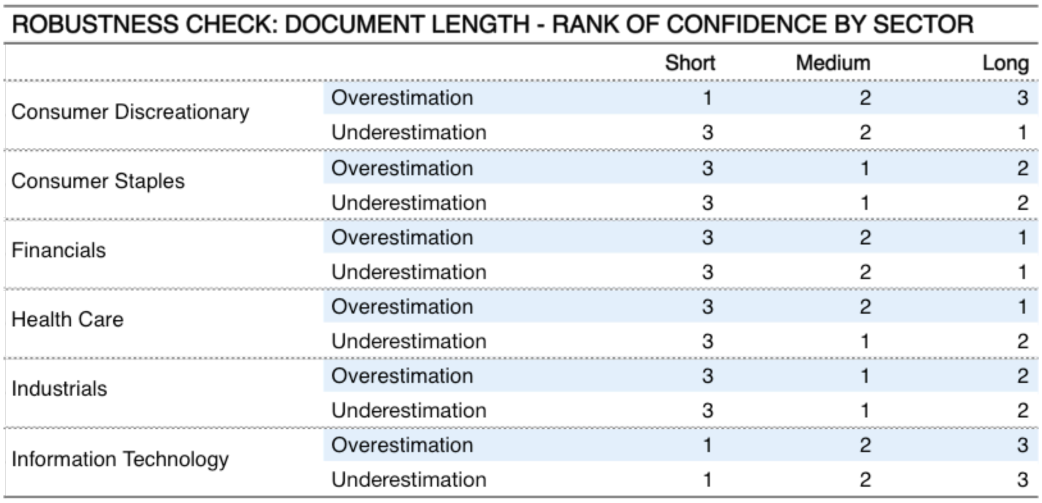
ROBUSTNESS CHECK: DOCUMENT LENGTH

Text Length	MAD* from threshold
Short	0.1077
Medium	0.1353
Long	0.1515

*) Mean absolute difference

Table 4.36: Robustness check - Document length

In most cases (five out of twelve), the highest ranked models in each model construction provide the most confident predictions when the filings are of “Medium” length (see Table 4.37).



ROBUSTNESS CHECK: DOCUMENT LENGTH - RANK OF CONFIDENCE BY SECTOR

		Short	Medium	Long
Consumer Discreationary	Overestimation	1	2	3
	Underestimation	3	2	1
Consumer Staples	Overestimation	3	1	2
	Underestimation	3	1	2
Financials	Overestimation	3	2	1
	Underestimation	3	2	1
Health Care	Overestimation	3	2	1
	Underestimation	3	1	2
Industrials	Overestimation	3	1	2
	Underestimation	3	1	2
Information Technology	Overestimation	1	2	3
	Underestimation	1	2	3

Table 4.37: Robustness check - Document length - Rank of confidence by sector

As predictions in one category do not significantly outperform those in another category, the results obtained in this study are concluded to be robust to differences in document length.

Based on the results obtained in the conducted robustness tests, indicating that none of the generated subcategories carry significantly more confident predictions, it is argued that the results obtained in this study are robust to variations in both company attention and document length.

4.9.4 Length of Training Data

Lastly, it is tested whether the results are robust to variations in the number of quarters included in the training data. The goal is to examine whether changing the number of quarters included in the training data (which, as described in Section 3.2.2 on Time-Aware Modeling, is 16) would result in changes in model performance. A decrease in model performance could indicate that 16 is the appropriate number of quarters to include for the given task, but it is emphasized that equal performance across variations could indicate that the models simply adjust according to the data that is provided to them.

Tests are performed conducting 12 new model constructions consisting of both under- and overestimation models for three sector subsamples (Consumer Discretionary, Financials and Industrials) and either 12 quarters or eight used for training data. To enable a comparison, the tests are constructed so that the same seven quarters are used for backtesting. The results (included in Appendix XXV) show that a change from 16 to 12 quarters would decrease the average BT Log Loss by 0.0037 and increase the average BT AUC score by 0.0083 (across all sectors and including both overestimation and underestimation models). Thus, the results indicate that model performance would increase slightly by such an alteration. However, it is emphasized that the magnitude of the improvement is minor. Furthermore, an examination of the results reveals that all highest ranked models would suffer a slight decrease in AUC score, and as these are the main focus of this study, the results obtained would be slightly weakened.

The impact of using eight quarters instead of 16 quarters has similarly been assessed, and the tests show that it doing so would have resulted in an average BT Log Loss 0.0032 above the one obtained in this study, and an average BT AUC score 0.0008 below the one obtained in this study.

As the results of the tests indicate that the differences in model performance are minor when altering the number of quarters included in the training data, it is argued that the results obtained in this study are robust to changes that could be made in this regard. Major increases in performance would have caused concern, but as the decreases are merely minor, it is argued that it seems as if the models are able to adjust to the data they are given.

Additional work will be required to test the robustness of the findings in this study across other characteristics. Such characteristics could include e.g. different time periods, different data sources (e.g. using company collected consensus estimates), and the readability of filings (using e.g. the measures described in Section 2.2.3.2 on Linguistic Measures: the Fog Index, filing document size, or the Bog Index).

5 Discussion

5.1 Relating Findings to Previous Studies

The results of this study reveal that analysts are in fact able to enhance their forecast accuracy by applying automated textual analysis. Hence, the findings indicate that 10-Q and 10-K filings contain information that analysts fail to fully incorporate in their EPS estimates. A reason for such lack of information incorporation could be the constraints, analysts are found to be subject to (see e.g. Clement (1999)). However, it is argued that analysts could alleviate the impact of such constraints and increase their forecasting accuracy by applying automated textual analysis as part of their forecasting task. This is in line with Lobo and Nair (1990), who find that combinations of statistical and judgmental forecasts result in higher forecasting accuracy than any of the two would yield individually.

In regards to the constraints mentioned in the above, Clement (1999) and Amir and Sougiannis (1999) find that they consist of factors such as limited resources and portfolio complexity. Furthermore, the authors argue that such constraints entail that analysts struggle to fully grasp the information available to them, if their coverage portfolio is complex or if the information is complex by nature. Their findings might, to a certain degree, explain some of the results obtained in this study. It is found that the models constructed on the Financials subsamples and on the Health Care subsamples perform well. As both of these sectors are heavily regulated, and as good analyst coverage of them requires extensive sector-specific knowledge, such model performance could be driven by the complexity characterizing the firms, the sectors include (i.e. the models are able to identify patterns that analysts, due to complexity, fail to fully grasp).

Moreover, several researchers have found that specific complex items are difficult for analysts to fully comprehend: Amir and Sougiannis (1999), Plumlee (2003), and Picconi (2006) find that items such as tax carry forwards, tax law changes, and pension information, respectively, are correlated with lower analyst forecast accuracy. The word clouds analyzed in Section 4 similarly indicate that this could be the case, as some words used in conjunction with complex items (such as the words e.g. “lessees” and “finalized”) were seen to carry highly indicative values. Hence, it is argued that, besides providing analysts with recommendations in regards to directional shifts in forecasts, the output of the models can be beneficial to analysts by identifying such complex items and enabling analysts to grasp their impact on EPS.

Lim (2001) finds that analysts tend to be positively biased when forecasting earnings in order to gain access to management. Such findings are not in line with the characteristics of the data used in this study, as the majority of observations are cases of underestimations. However, Lim (2001) states that the positive bias diminishes, the closer companies get to the filing date, and that analysts tend to overestimate e.g. one year ahead-earnings, while this is not necessarily the case when forecasting more short term earnings. Thus, one could argue that the tendency of being positively biased that Lim (2001) finds to characterize analysts does not necessarily hold when examining a horizon of one quarter. Moreover, companies do not necessarily prefer high estimates to estimates that are reachable: Bartov et al. (2002) e.g. find that companies that meet or beat expectations are able to enjoy greater stock returns after controlling for the underlying performance than companies that do not reach expectations.

As can be seen in the literature review, researchers apply several types of models for predictions of specific events, and even when their predictive tasks are alike, they seem to disagree on the type of model most suitable. Within e.g. fraud prediction, Cecchini et al. (2010) apply a support vector machine, whereas Whiting et al. (2012) argue that ensemble models are best suited for the task. The review furthermore reveals that the comparisons of different types of models seldomly result in similar recommendations: within bankruptcy prediction, Geng et al. (2015) and Jo et al. (1997) both find neural networks to be superior, whereas du Jardin (2016) argues that ensemble techniques such as blending or boosting are more accurate. Such disagreements are, to some degree, the motivation of this study, as they illustrate that not even specialized researchers are able to, with certainty, prove the superior performance of one specific type of model for a given task. Even less so for several different tasks. Thus, one cannot expect practitioners, such as constrained analysts, to be able to do so. The results of this study show that the choice of model should depend on the given task and data, as it is illustrated that no single model is superior in all of the model constructions. Thus, the advantages of simultaneously constructing several models in order to enable comparisons of their performance are stressed.

5.2 Limitations and Suggestions for Future Research

This section concerns the limitations characterizing the study and addresses how such limitations could be alleviated in future research. Firstly, due to the sample size cap of 500 MB for academic licenses in DataRobot, the data sample of this study only consists of observations from 24 quarters in the years 2012-2017. It is generally recommended to use data that covers an entire business cycle, as upturns as well as downturns are thereby covered (Petersen and Plenborg, 2012, p. 66). Thus, the time frame used in this study might be considered a limitation, as it does not contain enough years to capture such an entire business cycle. One could argue that this affects the results obtained. As stated in the analysis, approximately 75% of the observations in the data sample are cases in which analysts underestimate consensus. This might have been different if the sample had included years prior and subsequent to 2008, as it is likely that analysts were more prone to overestimate EPS during the Global Financial Crisis. Given that this holds, future research could test whether the practical applicability would still hold for a time period prior to the one chosen in this study. Further research could moreover perform similar tests on a data sample including both time periods (thus include an entire business cycle). Such research would test whether models are able to adjust to changes in the underlying economy over time in the observations included. If the results of such a test were affirmative, one could argue that the models constructed in this study are (even more) applicable.

Secondly, a limitation might exist in the fact that the findings of this study are based on a data sample that includes only companies in the S&P 500 index. While it is emphasized that including only S&P 500 companies seems to be common practice among researchers[42], the

[42] See e.g. Tetlock et al. (2008, p. 1441), who justify their choice by stating that they choose the S&P 500 constituents "*for reasons of importance and tractability*". Other examples include e.g. Huang et al. (2014) and Chen and Vincent (2016)

limitations that might result from excluding other (smaller) companies are acknowledged. Moreover, using only S&P 500 companies resulted in the exclusion of five sector subsamples due to an insufficient number of observations in each quarter. However, it is argued that, given the constraints, this study is subject to, the models are still constructed on the basis of a sample that would benefit the majority of analysts and other stakeholders, as the constituents of S&P 500 represent approximately 80% of available market capitalization in the US (Standard & Poor's, 2018). That said, future research could potentially mitigate the possible limitations resulting from such an exclusion by conducting similar tests on a sample consisting of e.g. S&P 1500 companies. Doing so would most likely enable similar model constructions on the (in this study) excluded sectors. Furthermore, it would increase the amount of training data, which could potentially increase model performance. However, such a sample would most likely entail a substantial increase in the proportion of low market capitalization companies that are covered by few analysts. Thus, the importance of performing robustness tests for such research in regards to analyst coverage (and company size) is stressed.

Thirdly, the fact that the constructed models are directional, binary classification models, entails that no findings in regards to the magnitude of the under- and overestimations are provided. However, such information would enable identification of the cases in which the models would be of most benefit to analysts: the ones where their estimates are most likely to be furthest from the actual outcome. Moreover, the models would enable greater increases in accuracy by indicating what the magnitude of the shifts in forecasts should be, instead of merely indicating the directions. Thus, as the results obtained in this study indicate that an accuracy enhancement is in fact possible, it is suggested that future research attempts to construct models that are able to provide such additional information.

Lastly, suggestions for future research include an examination of whether model performance could be further enhanced if additional features, such as financial variables (e.g. EPS, change in EPS, or P/E ratio) or the textual content from other sources (e.g. microblogs, news, or earnings conference calls), were added. In regards to the inclusion of financial variables, researchers have found in previous studies that they are able to increase their model accuracy when combining linguistic and financial variables (e.g. Cecchini et al., 2010; Hajek, 2017; Mayew et al., 2015). In regards to the inclusion of other textual sources, Kearney and Liu (2014), in their assessment of the advantages and disadvantages of different textual sources, advise that researchers employ as many information sources as possible. Thus, researchers could attempt to add the textual content of other sources or to construct models based on a different source solely, in order to examine whether the use of other sources would yield better results.

5.3 Implications for Practitioners and Suggestions for Future Research

This section concerns the implications that the findings of this study might entail for practitioners and addresses the future research, such implications could motivate.

Some of the main findings include that analysts are able to enhance their forecast accuracy by applying automated textual analysis as illustrated in this study, and that they do not necessarily need advanced data science skills to benefit from such applications. In practice, a consensus estimate is the average of the individual estimates provided by analysts. The models included in this study are constructed to predict cases of under- and overestimations, and these variables are based on whether EPS turns out to be above or below consensus at the time of the announcement of earnings. As analysts do not know the exact level of the final consensus estimate, on which the model predictions are based, determining whether their own estimate should be shifted might be difficult. However, it is argued that the models still add significant value by acting as indicators or warnings: if a model predicts that there is a 90% probability that consensus will turn out to be an underestimation of EPS, an analyst, whose estimate is below the current consensus, might infer that his estimate is likely to be too low as well. Shifting the estimate upwards would thus, most likely, result in an increase in accuracy. However, one should keep in mind that changes in the underlying, individual estimates lead to changes in consensus. Thus, one might argue that a first mover advantage (or a last mover disadvantage) exists: if several analysts have shifted their estimates upwards based on predictions provided by the model, consensus will have increased as well, and an analyst, whose estimate was above the original consensus but who, subsequent to the consensus shift, has an estimate below consensus, could decide to adjust his estimate upwards - this could potentially decrease his accuracy. Moreover, it is emphasized that if all analysts were to implement automated textual analysis, overall accuracy would be expected to increase over time, which would most likely cause the predictive power of the models to diminish. It is suggested that future research examines if similar models are applicable for predictions of whether the estimates of individual analysts will be under- or overestimations of EPS. Such an application would most likely be of more interest to analysts and mitigate some of the implications in regards to practical applicability that might follow from the model being based on consensus.

In regards to analysts, one could speculate whether enhancements of accuracy are always in their interest. According to Bartov et al. (2002), companies meeting or beating expectations enjoy excess stock returns. In order to support their long-term positively biased estimates and thereby positive stock recommendations, analysts could choose to lowball quarterly estimates to trigger such excess returns. If analysts game estimates in practice, it must follow that they consider accuracy to be of less importance, and their implementation of models as the ones constructed in this study might therefore seem unlikely. On the other hand, one could argue that if just one analyst were to use automated textual analysis in the manner, introduced in this study, such an analyst would be able to achieve a significant increase in accuracy relative to his peers. According to Mikhail et al. (1999), this would be of particular interest to analysts, as relative performance is negatively correlated with the risk of being fired.

Due to such lowballing behavior, or to other reasons not addressed in this study, analysts might fail to exploit the potential benefits that the implementation of automated textual analysis would entail. However, it is emphasized that the findings in this study could be of value to other stakeholders, such as investors. Future research could e.g. examine whether exploiting the tools provided in this study to predict cases of under- and overestimations of EPS and creating a trading strategy based on such predictions could lead to excess returns. In doing so, the findings of Bartov et al. (2002) could be applied by creating a strategy that takes into account that companies meeting or beating expectations enjoy return in excess of the underlying beat in the following quarter.

Bannister and Newmann (1996) find that the accruals of companies, whose earnings before accruals do not meet expectations, are larger than the accruals of companies, whose earnings before accruals already exceeds expectations. Such findings indicate that, besides causing analysts to alter their behavior, the excess returns resulting from meeting or beating expectations could additionally intrigue companies to use accruals to reach expectations. If companies exercise such behavior, one could argue that earnings are determined subsequent to consensus estimates, and that implementation of the models suggested in this study, would therefore not result in significant increases in analyst forecast accuracy.

6 Conclusion

This study examines to what extent models, based on automated textual analysis of the content of 10-K and 10-Q filings, can be used to enhance the accuracy of analysts' earnings per share forecasts. Based on the textual content of 10-K and 10-Q fillings of S&P 500 companies from 2012-2017, 12 sector-specific model constructions are created and subsequently tested by using DataRobot, a platform developed by leading data scientists for practical implementations. Such model constructions yield directional models that provide predictions indicating whether analyst consensus is likely to over- or underestimate EPS in the following quarter. In order to assess the practical applicability and the possibility of accuracy enhancements, the 12 best performing models in terms of average Log Loss in backtests are identified (one model from each model construction). These models are subsequently assessed based on a set of defined criteria, and it is examined whether analysts, who act according to the provided predictions, are likely to enhance their EPS forecast accuracy. Moreover, the word clouds generated as output in each of the model constructions are examined in order to assess whether analysts are able benefit from their highly interpretable content.

It is found that nine out of the 12 models fulfill the defined criteria, indicating that analysts, by implementing automated textual analysis as suggested in this study, would be able to significantly enhance the accuracy of their EPS forecasts in each of these cases. These findings are concluded to be robust to differences in market capitalization, analyst coverage, document length and the number of quarters used for training data in the modeling. However, it is found that the magnitude of such possible enhancements varies across sectors and predictive targets. While enhancements are concluded to be possible within all sectors, and particularly good models are constructed for e.g. the Financials sector, the models seem to struggle more when identifying significant patterns in three specific sectors: Consumer Staples, Industrials, and Information Technology. It is concluded that word clouds are exploitable to analysts as well, and by examining their output, analysts are able to gain valuable insights in regards to the general trends and nonrecurring, specific events that cause them to over- or underestimate EPS. Hence, it is argued that, besides providing analysts with recommendations in regards to directional shifts in forecasts, the output of the models can be beneficial to analysts by possibly identifying information that they often fail to fully comprehend.

It is emphasized that this study builds on the notion that no one type of model is best for a set of problems, and that choosing a model in advance when facing a predictive task is virtually impossible. Two findings seem to support this notion: 1) An extensive review of the literature indicates that researchers apply various models for similar predictive tasks, and that their findings in regards to superiority of models differ, 2) No model is consistently superior in any of the model constructions performed. This study illustrates that practitioners can avoid making such a choice among model types and still build strong predictive models and gain valuable insights, by applying software that enables simultaneous model constructions.

The study contributes to the existing literature by finding that the accuracy of analysts' EPS forecasts can be enhanced by the implementation of automated textual analysis, and by finding that analysts are not required to possess advanced data science skills to obtain the potential benefits. To the best of the authors' knowledge, similar research has not been conducted previously. Furthermore, the study contributes to the literature concerning analyst forecasts by finding that 10-K and 10-Q filings contain information that users of the filings do not fully comprehend, while automated textual analysis software seems to be able to identify specific patterns that can improve forecast accuracy. Lastly, it contributes to general research within the field of machine learning and automated textual analysis by underlining the importance of testing different types of models when attempting to predict specific outcomes, as both the literature review and the results indicate that models cannot be chosen in advance when attempting to solve a predictive task.

7 References

AghaeiRad, A., Chen, N., Ribeiro, B., 2017. Improve credit scoring using transfer of learned knowledge from self-organizing map. Neural Comput. Appl. 28, 1329–1342. Available at: https://doi.org/10.1007/s00521-016-2567-2

Alteryx, 2018. Alteryx Designer [ONLINE]. Available at: https://www.alteryx.com/products/alteryx-designer (accessed 5.11.18).

Amani, F.A., Fadlalla, A.M., 2017. Data mining applications in accounting: A review of the literature and organizing framework. Int. J. Account. Inf. Syst. 24, 32–58. Available at: https://doi.org/10.1016/j.accinf.2016.12.004

Amir, E., Sougiannis, T., 1999. Analysts' Interpretation and Investors' Valuation of Tax Carryforwards. Contemp. Account. Res. 16, 1–33. Available at: https://doi.org/10.1111/j.1911-3846.1999.tb00572.x

Antweiler, W., Frank, M.Z., 2004. Is All That Talk Just Noise? The Information Content of Internet Stock Message Boards. J. Finance 59, 1259–1294.

Araújo, R. de A., 2011. Translation Invariant Morphological Time-lag Added Evolutionary Forecasting method for stock market prediction. Expert Syst. Appl. 38, 2835–2848. Available at: https://doi.org/10.1016/j.eswa.2010.08.076

Audit Analytics, 2018. Reasons for Amended 10-K: 2017 [ONLINE]. Available at: https://www.auditanalytics.com/blog/reasons-for-an-amended-10-k-2017/ (accessed 5.11.18).

Ayres, D., Huang, X. (Sharon), Myring, M., 2017. Fair value accounting and analyst forecast accuracy. Adv. Account. 37, 58–70. Available at: https://doi.org/10.1016/j.adiac.2016.12.004

Bae, J.K., 2012. Predicting financial distress of the South Korean manufacturing industries. Expert Syst. Appl. 39, 9159–9165. Available at: https://doi.org/10.1016/j.eswa.2012.02.058

Balakrishnan, R., Qiu, X.Y., Srinivasan, P., 2010. On the predictive ability of narrative disclosures in annual reports. Eur. J. Oper. Res. 202, 789–801. Available at: https://doi.org/10.1016/j.ejor.2009.06.023

Bannister, J.W., Newman, H.A., 1996. Accrual usage to manage earnings toward financial analysts' forecasts. Rev. Quant. Finance Account. 7, 259–278. Available at: https://doi.org/10.1007/BF00245253

Bartov, E., Givoly, D., Hayn, C., 2002. The rewards to meeting or beating earnings expectations. J. Account. Econ. 32.

Behn, B.K., Choi, J.-H., Rang, T., 2008. Audit Quality and Properties of Analyst Earnings Forecasts. Account. Rev. 24.

Bhandari, A., Mammadov, B., Thevenot, M., 2017. The impact of executive inside debt on sell-side financial analyst forecast characteristics. Rev. Quant. Finance Account. Available at: https://doi.org/10.1007/s11156-017-0671-8

Bodnaruk, A., Loughran, T., McDonald, B., 2015. Using 10-K Text to Gauge Financial Constraints. J. Financ. Quant. Anal. 50, 623–646. Available at: https://doi.org/10.1017/S0022109015000411

Bollen, J., Mao, H., Zeng, X., 2011. Twitter mood predicts the stock market. J. Comput. Sci. 2, 1–8. Available at: https://doi.org/10.1016/j.jocs.2010.12.007

Bonsall, S.B., Leone, A.J., Miller, B.P., Rennekamp, K., 2017. A plain English measure of financial reporting readability. J. Account. Econ. 63, 329–357. Available at: https://doi.org/10.1016/j.jacceco.2017.03.002

Boyacioglu, M.A., Kara, Y., Baykan, Ö.K., 2009. Predicting bank financial failures using neural networks, support vector machines and multivariate statistical methods: A comparative analysis in the sample of savings deposit insurance fund (SDIF) transferred banks in Turkey. Expert Syst. Appl. 36, 3355–3366. Available at: https://doi.org/10.1016/j.eswa.2008.01.003

Branson, B.C., Lorek, K.S., Pagach, D.P., 1995. Evidence on the Superiority of Analysts Quarterly Earnings Forecasts for Small Capitalization Firms. Decis. Sci. 26, 243–263. Available at: https://doi.org/10.1111/j.1540-5915.1995.tb01428.x

Brau, J.C., Cicon, J., McQueen, G., 2016. Soft Strategic Information and IPO Underpricing. J. Behav. Finance 17, 1–17. Available at: https://doi.org/10.1080/15427560.2016.1133619

Brown, L.D., Hagerman, R.L., Griffin, P.A., Zmijewski, M.E., 1987. Security analyst superiority relative to univariate time-series models in forecasting quarterly earnings. J. Account. Econ. 9, 61–87. Available at: https://doi.org/10.1016/0165-4101(87)90017-6

CB Insights, 2018. The AI 100 [ONLINE]. Available at: https://www.cbinsights.com/research-ai-100 (accessed 5.11.18).

Cecchini, M., Aytug, H., Koehler, G.J., Pathak, P., 2010. Making words work: Using financial text as a predictor of financial events. Decis. Support Syst. 50, 164–175. Available at: https://doi.org/10.1016/j.dss.2010.07.012

Cecchini, M., Aytug, H., Koehler, G.J., Pathak, P., 2010a. Detecting Management Fraud in Public Companies. Manag. Sci. 56, 1146–1160. Available at: https://doi.org/10.1287/mnsc.1100.1174

Cesa-Bianchi, N., Freund, Y., Haussler, D., Helmbold, D.P., Schapire, R.E., Warmuth, M.K., 1997. How to Use Expert Advice 59.

Chan, S.W.K., Franklin, J., 2011. A text-based decision support system for financial sequence prediction. Decis. Support Syst. 52, 189–198. Available at: https://doi.org/10.1016/j.dss.2011.07.003

Checkley, M.S., Higón, D.A., Alles, H., 2017. The hasty wisdom of the mob: How market sentiment predicts stock market behavior. Expert Syst. Appl. 77, 256–263. Available at: https://doi.org/10.1016/j.eswa.2017.01.029

Chen, C.-L., Liu, C.-L., Chang, Y.-C., Tsai, H.-P., 2013. Opinion Mining for Relating Subjective

Expressions and Annual Earnings in US Financial Statements 22.

Chen, M.-Y., 2014. Using a hybrid evolution approach to forecast financial failures for Taiwan-listed companies. Quant. Finance 14, 1047–1058. Available at: https://doi.org/10.1080/14697688.2011.618458

Chen, W.-S., Du, Y.-K., 2009. Using neural networks and data mining techniques for the financial distress prediction model. Expert Syst. Appl. 36, 4075–4086. Available at: https://doi.org/10.1016/j.eswa.2008.03.020

Chen, Y., Vincent, K., 2016. The Role of Momentum, Sentiment, and Economic Fundamentals in Forecasting Bear Stock Market: Bear Stock Market Forecasting. J. Forecast. 35, 504–527. Available at: https://doi.org/10.1002/for.2392

Clement, M.B., 1999. Analyst forecast accuracy: Do ability, resources, and portfolio complexity matter? J. Account. Econ. 19.

Clement, M.B., Tse, S.Y., 2005. Financial Analyst Characteristics and Herding Behavior in Forecasting. J. Finance 60, 307–341.

Cole, C.J., Jones, C.L., 2004. The Usefulness of MD&A Disclosures in the Retail Industry. J. Account. Audit. Finance 19, 361–388. Available at: https://doi.org/10.1177/0148558X0401900401

CUSIP Global Services, 2018. About CGS Identifiers [ONLINE]. Available at: https://www.cusip.com/cusip/about-cgs-identifiers.htm (accessed 5.11.18).

Das, S.R., 2014. Text and context: language analytics in finance, Foundations and trends in finance. Now Publ, Boston, Mass.

Das, S.R., Chen, M.Y., 2007. Yahoo! for Amazon: Sentiment Extraction from Small Talk on the Web. Manag. Sci. 53, 1375–1388. Available at: https://doi.org/10.1287/mnsc.1070.0704

DataRobot, 2018. Leaderboard Tab reference: Lift Chart tab.

DataRobot, 2018a. Using the Show Advanced Options link: Changing the optimization metric.

DataRobot, 2018b. Using DataRobot's Insights link: Using Word Cloud insights.

Davis, A.K., Tama-Sweet, I., 2012. Managers' Use of Language Across Alternative Disclosure Outlets: Earnings Press Releases versus MD&A*: Language in Earnings Press Releases vs. MD&A. Contemp. Account. Res. 29, 804–837. Available at: https://doi.org/10.1111/j.1911-3846.2011.01125.x

De Andrés, J., Lorca, P., de Cos Juez, F.J., Sánchez-Lasheras, F., 2011. Bankruptcy forecasting: A hybrid approach using Fuzzy c-means clustering and Multivariate Adaptive Regression Splines (MARS). Expert Syst. Appl. 38, 1866–1875. Available at: https://doi.org/10.1016/j.eswa.2010.07.117

Delen, D., Kuzey, C., Uyar, A., 2013. Measuring firm performance using financial ratios: A decision tree approach. Expert Syst. Appl. 40, 3970–3983. Available at:

https://doi.org/10.1016/j.eswa.2013.01.012

Deutsche Bank, 2018. How automatic text analysis and artificial intelligence lead to new investment products [ONLINE]. Available at: https://www.db.com/newsroom_news/2018/how-automatic-text-analysis-and-artificial-intelligence-lead-to-new-investment-products-en-11449.htm (accessed 5.11.18).

du Jardin, P., 2017. Dynamics of firm financial evolution and bankruptcy prediction. Expert Syst. Appl. 75, 25–43. Available at: https://doi.org/10.1016/j.eswa.2017.01.016

du Jardin, P., 2016. A two-stage classification technique for bankruptcy prediction. Eur. J. Oper. Res. 254, 236–252. Available at: https://doi.org/10.1016/j.ejor.2016.03.008

du Jardin, P., 2015. Bankruptcy prediction using terminal failure processes. Eur. J. Oper. Res. 242, 286–303. Available at: https://doi.org/10.1016/j.ejor.2014.09.059

Dyer, T., Lang, M., Stice-Lawrence, L., 2017. The evolution of 10-K textual disclosure: Evidence from Latent Dirichlet Allocation. J. Account. Econ. 64, 221–245. Available at: https://doi.org/10.1016/j.jacceco.2017.07.002

Eliacik, A.B., Erdogan, N., 2018. Influential user weighted sentiment analysis on topic based microblogging community. Expert Syst. Appl. 92, 403–418. Available at: https://doi.org/10.1016/j.eswa.2017.10.006

Elsevier, 2017. Scopus Content Coverage Guide [ONLINE]. Available at: https://www.elsevier.com/__data/assets/pdf_file/0007/69451/0597-Scopus-Content-Coverage-Guide-US-LETTER-v4-HI-singles-no-ticks.pdf (accessed 5.11.18).

EY, 2016. A summary of IFRS 16 and its effects [ONLINE]. Available at: http://www.ey.com/Publication/vwLUAssets/ey-leases-a-summary-of-ifrs-16-and-its-effects-may-2016/$FILE/ey-leases-a-summary-of-ifrs-16-and-its-effects-may-2016.pdf (accessed 5.11.18).

Fan, J., Upadhye, S., Worster, A., 2006. Understanding receiver operating characteristic (ROC) curves. CJEM 8, 19–20. Available at: https://doi.org/10.1017/S1481803500013336

Feldman, R., Govindaraj, S., Livnat, J., Segal, B., 2010. Management's tone change, post earnings announcement drift and accruals. Rev. Account. Stud. 15, 915–953. Available at: https://doi.org/10.1007/s11142-009-9111-x

Florez-Lopez, R., Ramon-Jeronimo, J.M., 2015. Enhancing accuracy and interpretability of ensemble strategies in credit risk assessment. A correlated-adjusted decision forest proposal. Expert Syst. Appl. 42, 5737–5753. Available at: https://doi.org/10.1016/j.eswa.2015.02.042

Forrester, 2016. Predictions 2017: Artificial Intelligence Will Drive The Insights Revolution Advanced - Insights Will Spark Digital Transformation In The Year Ahead [ONLINE]. Available at: https://go.forrester.com/wp-content/uploads/Forrester_Predictions_2017_-Artificial_Intelligence_Will_Drive_The_Insights_Revolution.pdf (accessed 5.11.18).

Geng, R., Bose, I., Chen, X., 2015. Prediction of financial distress: An empirical study of listed Chinese companies using data mining. Eur. J. Oper. Res. 241, 236–247. Available at:

https://doi.org/10.1016/j.ejor.2014.08.016

Gepp, A., Kumar, K., Bhattacharya, S., 2010. Business failure prediction using decision trees. J. Forecast. 29, 536–555. Available at: https://doi.org/10.1002/for.1153

Glancy, F.H., Yadav, S.B., 2011. A computational model for financial reporting fraud detection. Decis. Support Syst. 50, 595–601. Available at: https://doi.org/10.1016/j.dss.2010.08.010

Goel, S., Uzuner, O., 2016. Do Sentiments Matter in Fraud Detection? Estimating Semantic Orientation of Annual Reports: Do Sentiments Matter in Fraud Detection: Estimating Semantic Orientation of Annual Reports. Intell. Syst. Account. Finance Manag. 23, 215–239. Available at: https://doi.org/10.1002/isaf.1392

Goh, G., 2017. DataRobot Named to 2018 ‘AI 100’ by CB Insights [ONLINE]. Available at: https://www.datarobot.com/news/datarobot-named-2018-ai-100-cb-insights/ (accessed 5.11.18).

Goldman Sachs Asset Management, 2016. Text, Tone and Topic - Exploring Potential Benefits of Natural Language Processing [ONLINE]. Available at: https://www.gsam.com/content/dam/gsam/pdfs/common/en/public/articles/perspectives/2016/big-data/GSAMPerspectives_TextToneTopic.pdf?sa=n&rd=n (accessed 5.11.18).

Gunning, R., 1969. The Fog Index After Twenty Years. J. Bus. Commun. 6, 3–13. Available at: https://doi.org/10.1177/002194366900600202

Hajek, P., 2017. Combining bag-of-words and sentiment features of annual reports to predict abnormal stock returns. Neural Comput. Appl. 29, 343–358. Available at: https://doi.org/10.1007/s00521-017-3194-2

Hajek, P., Henriques, R., 2017. Mining corporate annual reports for intelligent detection of financial statement fraud – A comparative study of machine learning methods. Knowl.-Based Syst. 128, 139–152. Available at: https://doi.org/10.1016/j.knosys.2017.05.001

Hajizadeh, E., Seifi, A., Fazel Zarandi, M.H., Turksen, I.B., 2012. A hybrid modeling approach for forecasting the volatility of S&P 500 index return. Expert Syst. Appl. 39, 431–436. Available at: https://doi.org/10.1016/j.eswa.2011.07.033

Hassell, J.M., Jennings, R.H., Lasser, D.J., 1988. Management earnings forecasts:Their usefullness as a source of firm-specific information to security analysts. J. Financ. Res. 4, 303–319.

Houlihan, P., Creamer, G.G., 2017. Can Sentiment Analysis and Options Volume Anticipate Future Returns? Comput. Econ. 50, 669–685. Available at: https://doi.org/10.1007/s10614-017-9694-4

Hribar, P., McInnis, J., 2012. Investor Sentiment and Analysts’ Earnings Forecast Errors. Manag. Sci. 58, 293–307. Available at: https://doi.org/10.1287/mnsc.1110.1356

Huang, A.H., Zang, A.Y., Zheng, R., 2014. Evidence on the Information Content of Text in Analyst Reports. Account. Rev. 89, 2151–2180. Available at: https://doi.org/10.2308/accr-50833

Huang, C.-J., Liao, J.-J., Yang, D.-X., Chang, T.-Y., Luo, Y.-C., 2010. Realization of a news dissemination agent based on weighted association rules and text mining techniques. Expert Syst. Appl. 37, 6409–6413. Available at: https://doi.org/10.1016/j.eswa.2010.02.078

Humpherys, S.L., Moffitt, K.C., Burns, M.B., Burgoon, J.K., Felix, W.F., 2011. Identification of fraudulent financial statements using linguistic credibility analysis. Decis. Support Syst. 50, 585–594. Available at: https://doi.org/10.1016/j.dss.2010.08.009

Ibriyamova, F., Kogan, S., Salganik-Shoshan, G., Stolin, D., 2017. Using semantic fingerprinting in finance. Appl. Econ. 49, 2719–2735. Available at: https://doi.org/10.1080/00036846.2016.1245844

Ittoo, A., Nguyen, L.M., van den Bosch, A., 2015. Text analytics in industry: Challenges, desiderata and trends. Comput. Ind. 78, 96–107. Available at: https://doi.org/10.1016/j.compind.2015.12.001

James, G., Witten, D., Hastie, T., Tibshirani, R. (Eds.), 2013. An introduction to statistical learning: with applications in R, Springer texts in statistics. Springer, New York.

Jo, H., Han, I., Lee, H., 1997. Bankruptcy prediction using case-based reasoning, neural networks, and discriminant analysis. Expert Syst. Appl. 13, 97–108. Available at: https://doi.org/10.1016/S0957-4174(97)00011-0

Jones, S., 2017. Corporate bankruptcy prediction: a high dimensional analysis. Rev. Account. Stud. 22, 1366–1422. Available at: https://doi.org/10.1007/s11142-017-9407-1

Kaastra, I., Boyd, M., 1996. Designing a neural network for forecasting financial and economic time series. Neurocomputing 10, 215–236. Available at: https://doi.org/10.1016/0925-2312(95)00039-9

Kearney, C., Liu, S., 2014. Textual sentiment in finance: A survey of methods and models. Int. Rev. Financ. Anal. 33, 171–185. Available at: https://doi.org/10.1016/j.irfa.2014.02.006

Kerl, A., Ohlert, M., 2015. Star-Analysts' Forecast Accuracy and the Role of Corporate Governance 29.

Kim, K., 2003. Financial time series forecasting using support vector machines. Neurocomputing 55, 307–319. Available at: https://doi.org/10.1016/S0925-2312(03)00372-2

Kim, S.H., Noh, H.J., 1997. Predictability of Interest Rates Using Data Mining Tools: A Comparative Analysis of Korea and the US 11.

Kim, Y., Song, M., 2015. Management Earnings Forecasts and Value of Analyst Forecast Revisions. Manag. Sci. 61, 1663–1683. Available at: https://doi.org/10.1287/mnsc.2014.1920

Kirkos, E., Spathis, C., Manolopoulos, Y., 2007. Data Mining techniques for the detection of fraudulent financial statements. Expert Syst. Appl. 32, 995–1003. Available at: https://doi.org/10.1016/j.eswa.2006.02.016

Kodogiannis, V., Lolis, A., 2002. Forecasting Financial Time Series using Neural Network and Fuzzy System-based Techniques. Neural Comput. Appl. 11, 90–102. Available at: https://doi.org/10.1007/s005210200021

Kraus, M., Feuerriegel, S., 2017. Decision support from financial disclosures with deep neural networks and transfer learning. Decis. Support Syst. 104, 38–48. Available at:

https://doi.org/10.1016/j.dss.2017.10.001

Kräussl, R., Mirgorodskaya, E., 2017. Media, sentiment and market performance in the long run. Eur. J. Finance 23, 1059–1082. Available at: https://doi.org/10.1080/1351847X.2016.1226188

Lang, M.H., Lundholm, R.J., 1996. Corporate Disclosure Policy and Analyst Behavior. Account. Rev. 71, 467–492.

Laurent, B., 2018. Gartner Data & Analytics Summit attendees rate "Augmented Data Science" as transformational [ONLINE]. Available at: https://blog.datarobot.com/gartner-data-analytics-summit-attendees-rate-augmented-data-science-as-transformational?utm_medium=referral&utm_source=datarobot.com&utm_campaign=(referral)&utm_content=/newsroom/ (accessed 5.11.18).

Lehavy, R., Li, F., Merkley, K., 2011. The Effect of Annual Report Readability on Analyst Following and the Properties of Their Earnings Forecasts. Account. Rev. 86, 1087–1115. Available at: https://doi.org/10.2308/accr.00000043

Li, F., 2010. The Information Content of Forward-Looking Statements in Corporate Filings—A Naïve Bayesian Machine Learning Approach. J. Account. Res. 48, 1049–1102.

Li, F., 2008. Annual report readability, current earnings, and earnings persistence. J. Account. Econ. 45, 221–247. Available at: https://doi.org/10.1016/j.jacceco.2008.02.003

Li, F., 2010a. Textual Analysis of Corporate Disclosures: A survey of the literature. J. Account. Lit. 29, 143–165.

Li, Q., Wang, T., Gong, Q., Chen, Y., Lin, Z., Song, S., 2014. Media-aware quantitative trading based on public Web information. Decis. Support Syst. 61, 93–105. Available at: https://doi.org/10.1016/j.dss.2014.01.013

Lim, T., 2001. Rationality and Analysts' Forecast Bias. J. Finance 56, 369–385. Available at: https://doi.org/10.1111/0022-1082.00329

Liu, C., Maheu, J.M., 2009. Forecasting Realized Volatility: A Bayesian Model-Averaging Approach. J. Appl. Econom. 24, 709–733.

Lobo, G.J., Kwon, S.S., Ndubizu, G.A., 1998. The Impact of SFAS No. 14 Segment Information on Price Variability and Earnings Forecast Accuracy. J. Bus. Finance Htmlent Glyphamp Asciiamp Account. 25, 969–985. Available at: https://doi.org/10.1111/1468-5957.00221

Lobo, G.J., Nair, R.D., 1990. Combining Judgmental and Statistical Forecasts: An Application to Earnings Forecasts. Decis. Sci. 21, 446–460. Available at: https://doi.org/10.1111/j.1540-5915.1990.tb01696.x

Loughran, T., McDonald, B., 2016. Textual Analysis in Accounting and Finance: A Survey: Textual Analysis in Accounting and Finance. J. Account. Res. 54, 1187–1230. Available at: https://doi.org/10.1111/1475-679X.12123

Loughran, T., McDonald, B., 2014. Measuring Readability in Financial Disclosures: Measuring Readability in Financial Disclosures. J. Finance 69, 1643–1671. Available at:

https://doi.org/10.1111/jofi.12162

Loughran, T., McDonald, B., 2011. When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks. J. Finance 66, 35–65.

Loughran, T., McDonald, B., 2011a. Barron's Red Flags: Do They Actually Work? J. Behav. Finance 12, 90–97. Available at: https://doi.org/10.1080/15427560.2011.575971

Lu, C.-J., 2013. Hybridizing nonlinear independent component analysis and support vector regression with particle swarm optimization for stock index forecasting. Neural Comput. Appl. 23, 2417–2427. Available at: https://doi.org/10.1007/s00521-012-1198-5

Lu, C.-J., Lee, T.-S., Chiu, C.-C., 2009. Financial time series forecasting using independent component analysis and support vector regression. Decis. Support Syst. 47, 115–125. Available at: https://doi.org/10.1016/j.dss.2009.02.001

Mangee, N., 2018. Stock Returns and the Tone of Marketplace Information: Does Context Matter? J. Behav. Finance 1–11. Available at: https://doi.org/10.1080/15427560.2018.1405268

Mason, S.J., Graham, N.E., 2002. Areas beneath the relative operating characteristics (ROC) and relative operating levels (ROL) curves: Statistical significance and interpretation 22.

Mayew, W.J., Sethuraman, M., Venkatachalam, M., 2015. MD&A Disclosure and the Firm's Ability to Continue as a Going Concern. Account. Rev. 90, 1621–1651. Available at: https://doi.org/10.2308/accr-50983

Mayew, W.J., Venkatachalam, M., 2012. The Power of Voice: Managerial Affective States and Future Firm Performance. J. Finance 67, 1–43.

Michel, J.-S., 2017. Investor Overreaction to Analyst Reference Points. J. Behav. Finance 18, 329–343. Available at: https://doi.org/10.1080/15427560.2017.1342646

Mikhail, M.B., Walther, B.R., Willis, R.H., 1999. Does Forecast Accuracy Matter to Security Analysts? Account. Rev. 74, 185–200.

Milian, J.A., Smith, A.L., 2017. An Investigation of Analysts' Praise of Management During Earnings Conference Calls. J. Behav. Finance 18, 65–77. Available at: https://doi.org/10.1080/15427560.2017.1276068

Moshiri, S., Cameron, N.E., Scuse, D., 1999. Static, Dynamic, and Hybrid Neural Networks in Forecasting Inflation 17.

MSCI, 2018. GICS [ONLINE]. Available at: https://www.msci.com/gics (accessed 5.11.18).

MSCI, 2016. Global Industry Classification Standard [ONLINE]. Available at: https://www.msci.com/documents/10199/4547797/GICS+Sector+definitions-Sep+2016.pdf/7e5236a8-2ddd-4e29-a8bf-18f394c7f0fb (accessed 5.11.18).

Murphy, P.R., Purda, L., Skillicorn, D., 2018. Can Fraudulent Cues Be Transmitted by Innocent Participants? J. Behav. Finance 19, 1–15. Available at: https://doi.org/10.1080/15427560.2017.1365367

Nag, A.K., Mitra, A., 2002. Forecasting daily foreign exchange rates using genetically optimized neural networks. J. Forecast. 21, 501–511. Available at: https://doi.org/10.1002/for.838

Nekrasov, A., Ogneva, M., 2011. Using earnings forecasts to simultaneously estimate firm-specific cost of equity and long-term growth. Rev. Account. Stud. 16, 414–457. Available at: https://doi.org/10.1007/s11142-011-9159-2

Ni, H., Yin, H., 2009. Exchange rate prediction using hybrid neural networks and trading indicators. Neurocomputing 72, 2815–2823. Available at: https://doi.org/10.1016/j.neucom.2008.09.023

O'Connor, N., Madden, M.G., 2006. A neural network approach to predicting stock exchange movements using external factors. Knowl.-Based Syst. 19, 371–378. Available at: https://doi.org/10.1016/j.knosys.2005.11.015

Öğüt, H., Aktaş, R., Alp, A., Doğanay, M.M., 2009. Prediction of financial information manipulation by using support vector machine and probabilistic neural network. Expert Syst. Appl. 36, 5419–5423. Available at: https://doi.org/10.1016/j.eswa.2008.06.055

Oliveira, N., Cortez, P., Areal, N., 2017. The impact of microblogging data for stock market prediction: Using Twitter to predict returns, volatility, trading volume and survey sentiment indices. Expert Syst. Appl. 73, 125–144. Available at: https://doi.org/10.1016/j.eswa.2016.12.036

Oliveira, N., Cortez, P., Areal, N., 2016. Stock market sentiment lexicon acquisition using microblogging data and statistical measures. Decis. Support Syst. 85, 62–73. Available at: https://doi.org/10.1016/j.dss.2016.02.013

Patel, J., Shah, S., Thakkar, P., Kotecha, K., 2015. Predicting stock market index using fusion of machine learning techniques. Expert Syst. Appl. 42, 2162–2172. Available at: https://doi.org/10.1016/j.eswa.2014.10.031

Petersen, C.V., Plenborg, T., 2012. Financial statement analysis: valuation, credit analysis, executive compensation. Financial Times/Prentice Hall, Harlow, England ; New York.

Picconi, M., 2006. The Perils of Pensions: Does Pension Accounting Lead Investors and Analysts Astray? Account. Rev. 32.

Plumlee, M.A., 2003. The Effect of Information Complexity on Analysts' Use of That Information. Account. Rev. 78, 275–296. Available at: https://doi.org/10.2308/accr.2003.78.1.275

Price, S.M., Doran, J.S., Peterson, D.R., Bliss, B.A., 2012. Earnings conference calls and stock returns: The incremental informativeness of textual tone. J. Bank. Finance 36, 992–1011. Available at: https://doi.org/10.1016/j.jbankfin.2011.10.013

Ravisankar, P., Ravi, V., Raghava Rao, G., Bose, I., 2011. Detection of financial statement fraud and feature selection using data mining techniques. Decis. Support Syst. 50, 491–500. Available at: https://doi.org/10.1016/j.dss.2010.11.006

Rönnqvist, S., Sarlin, P., 2017. Bank distress in the news: Describing events through deep learning. Neurocomputing 264, 57–70. Available at: https://doi.org/10.1016/j.neucom.2016.12.110

Schebesch, K.B., Stecking, R., 2005. Support Vector Machines for Classifying and Describing Credit Applicants: Detecting Typical and Critical Regions. J. Oper. Res. Soc. 56.

SEC, 2018. Form 10-K [ONLINE]. Available at: https://www.sec.gov/files/form10-k.pdf (accessed 5.11.18).

SEC, 2013. Information About Some Companies Not Available From the SEC [ONLINE]. Available at: https://www.sec.gov/answers/noinfo.htm (accessed 5.11.18).

SEC, 2009. Form 10-K [ONLINE]. Available at: https://www.sec.gov/fast-answers/answers-form10khtm.html (accessed 5.11.18).

SEC, 2018a. Form 10-Q [ONLINE]. Available at: https://www.sec.gov/files/form10-q.pdf (accessed 5.11.18).

Shen, F., Chao, J., Zhao, J., 2015. Forecasting exchange rate using deep belief networks and conjugate gradient method. Neurocomputing 167, 243–253. Available at: https://doi.org/10.1016/j.neucom.2015.04.071

Shie, F.S., Chen, M.-Y., Liu, Y.-S., 2012. Prediction of corporate financial distress: an application of the America banking industry. Neural Comput. Appl. 21, 1687–1696. Available at: https://doi.org/10.1007/s00521-011-0765-5

SRAF, 2018. Stage One 10-X Parse Data [ONLINE]. Notre Dame Softw. Repos. Account. Finance. Available at: http://sraf.nd.edu/data/stage-one-10-x-parse-data/ (accessed 5.11.18).

SRAF, 2018a. Resources [ONLINE]. Notre Dame Softw. Repos. Account. Finance. Available at: https://sraf.nd.edu/textual-analysis/resources/ (accessed 5.11.18).

Standard & Poor's, 2018. S&P 500 [ONLINE]. Available at: https://us.spindices.com/indices/equity/sp-500 (accessed 5.11.18).

Statista, 2018. Retail e-commerce sales worldwide from 2014 to 2021 (in billion U.S. dollars) [ONLINE]. Available at: https://www.statista.com/statistics/379046/worldwide-retail-e-commerce-sales/ (accessed 5.11.18).

Taylor, J.W., 2005. Generating Volatility Forecasts from Value at Risk Estimates. Manag. Sci. 51, 712–725. Available at: https://doi.org/10.1287/mnsc.1040.0355

Tetlock, P.C., 2007. Giving Content to Investor Sentiment: The Role of Media in the Stock Market. J. Finance 62, 1139–1168.

Tetlock, P.C., Saar-Tsechansky, M., Macskassy, S., 2008. More than Words: Quantifying Language to Measure Firms' Fundamentals. J. Finance 63, 1437–1467.

The Big 4 Accounting Firms, 2018. Who Are The Big 5 Accounting Firms [ONLINE]. Available at: http://big4accountingfirms.org/big-5-accounting-firms/ (accessed 5.11.18).

Thomson Reuters, 2018. I/B/E/S Estimates [ONLINE]. Available at: https://financial.thomsonreuters.com/en/products/data-analytics/company-data/ibes-estimates.html (accessed 5.11.18).

Thomson Reuters Practical Law, 2018. Public limited company (PLC) [ONLINE]. Available at: https://uk.practicallaw.thomsonreuters.com/4-107-7082?transitionType=Default&contextData=(sc.Default)&firstPage=true&bhcp=1 (accessed 5.11.18).

Tobback, E., Bellotti, T., Moeyersoms, J., Stankova, M., Martens, D., 2017. Bankruptcy prediction for SMEs using relational data. Decis. Support Syst. 102, 69–81. Available at: https://doi.org/10.1016/j.dss.2017.07.004

Tsai, C., Wu, J., 2008. Using neural network ensembles for bankruptcy prediction and credit scoring. Expert Syst. Appl. 34, 2639–2649. Available at: https://doi.org/10.1016/j.eswa.2007.05.019

Tsai, M.-F., Wang, C.-J., 2017. On the risk prediction and analysis of soft information in finance reports. Eur. J. Oper. Res. 257, 243–250. Available at: https://doi.org/10.1016/j.ejor.2016.06.069

Tsukioka, Y., Yanagi, J., Takada, T., 2017. Investor sentiment extracted from internet stock message boards and IPO puzzles. Int. Rev. Econ. Finance. Available at: https://doi.org/10.1016/j.iref.2017.10.025

Tung, W.L., Quek, C., Cheng, P., 2004. GenSo-EWS: a novel neural-fuzzy based early warning system for predicting bank failures. Neural Netw. 17, 567–587. Available at: https://doi.org/10.1016/j.neunet.2003.11.006

Uhl, M.W., 2014. Reuters Sentiment and Stock Returns. J. Behav. Finance 15, 287–298. Available at: https://doi.org/10.1080/15427560.2014.967852

Wang, G., Hao, J., Ma, J., Jiang, H., 2011. A comparative assessment of ensemble learning for credit scoring. Expert Syst. Appl. 38, 223–230. Available at: https://doi.org/10.1016/j.eswa.2010.06.048

Wang, Jie, Wang, Jun, 2015. Forecasting stock market indexes using principle component analysis and stochastic time effective neural networks. Neurocomputing 156, 68–78. Available at: https://doi.org/10.1016/j.neucom.2014.12.084

Wang, L., Wu, C., 2017. A Combination of Models for Financial Crisis Prediction: Integrating Probabilistic Neural Network with Back-Propagation based on Adaptive Boosting. Int. J. Comput. Intell. Syst. 10, 507. Available at: https://doi.org/10.2991/ijcis.2017.10.1.35

Weng, B., Ahmed, M.A., Megahed, F.M., 2017. Stock market one-day ahead movement prediction using disparate data sources. Expert Syst. Appl. 79, 153–163. Available at: https://doi.org/10.1016/j.eswa.2017.02.041

Whiting, D.G., Hansen, J.V., McDonald, J.B., Albrecht, C., Albrecht, W.S., 2012. Manchine Learning Methods for Detecting Patterns of Management Fraud. Comput. Intell. 28, 505–527. Available at: https://doi.org/10.1111/j.1467-8640.2012.00425.x

Xu, X., Wang, Y., 2009. Financial failure prediction using efficiency as a predictor. Expert Syst. Appl. 36, 366–373. Available at: https://doi.org/10.1016/j.eswa.2007.09.040

Yang, S.Y., Mo, S.Y.K., Liu, A., Kirilenko, A.A., 2017. Genetic programming optimization for a sentiment feedback strength based trading strategy. Neurocomputing 264, 29–41. Available at: https://doi.org/10.1016/j.neucom.2016.10.103

Yao, J., Tan, C.L., 2000. A case study on using neural networks to perform technical forecasting of forex. Neurocomputing 34, 79–98. Available at: https://doi.org/10.1016/S0925-2312(00)00300-3

Yeh, C.-Y., Huang, C.-W., Lee, S.-J., 2011. A multiple-kernel support vector regression approach for stock market price forecasting. Expert Syst. Appl. 38, 2177–2186. Available at: https://doi.org/10.1016/j.eswa.2010.08.004

Yu, Q., Miche, Y., Séverin, E., Lendasse, A., 2014. Bankruptcy prediction using Extreme Learning Machine and financial expertise. Neurocomputing 128, 296–302. Available at: https://doi.org/10.1016/j.neucom.2013.01.063

Zhang, G., Hu, M.Y., Patuwo, B.E., Indro, D.C., 1999. Artificial neural networks in bankruptcy prediction: General framework and cross-validation analysis. Eur. J. Oper. Res. 17.

Zhang, W., Cao, Q., Schniederjans, M.J., 2004. Neural Network Earnings per Share Forecasting Models: A Comparative Analysis of Alternative Methods. Decis. Sci. 35, 205–237. Available at: https://doi.org/10.1111/j.00117315.2004.02674.x

Zhang, Y., Swanson, P.E., Prombutr, W., 2012. Measuring Effects on Stock Returns of Sentiment Indexes Created From Stock Message Boards. J. Financ. Res. 35, 79–114. Available at: https://doi.org/10.1111/j.1475-6803.2011.01310.x

Zhong, X., Enke, D., 2017. Forecasting daily stock market return using dimensionality reduction. Expert Syst. Appl. 67, 126–139. Available at: https://doi.org/10.1016/j.eswa.2016.09.027

Zhou, L., Burgoon, J.K., Twitchell, D.P., Qin, T., Nunamaker Jr., J.F., 2004. A Comparison of Classification Methods for Predicting Deception in Computer-Mediated Communication. J. Manag. Inf. Syst. 20, 139–166. Available at: https://doi.org/10.1080/07421222.2004.11045779

Zhou, W., Kapoor, G., 2011. Detecting evolutionary financial statement fraud. Decis. Support Syst. 50, 570–575. Available at: https://doi.org/10.1016/j.dss.2010.08.007

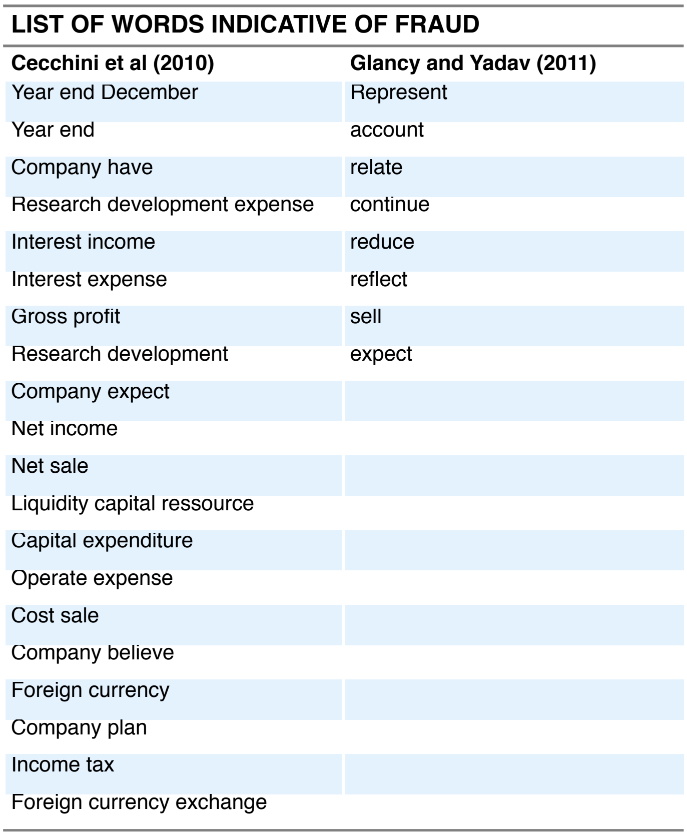
8 Appendix

Appendix I. Final Scopus-search String

TITLE-ABS-KEY ("analyst forecast* accuracy*" OR "analyst accuracy*" OR "stock return*" OR "financial accounting" OR "finance" OR "financial reporting" OR "financial report*" OR "earning*" OR "earning* surprise*" OR "10-K" OR "10-Q" OR "annual report*" OR "management discussion and analysis" OR "MD&A" OR "corporate disclosure*" AND "prediction*" OR "predictive" OR "machine learning" OR "data mining" OR "neural network*" OR "textual analysis" OR "forecasting" OR "artificial intelligence" OR "artificial neural network" OR "text mining" OR "word list*") AND (LIMIT-TO (SRCTYPE , "j ")) AND (LIMIT-TO (SUBJAREA , "COMP ") OR LIMIT-TO (SUBJAREA , " MATH ") OR LIMIT-TO (SUBJAREA , " BUSI ") OR LIMIT-TO (SUBJAREA , " ECON ") OR LIMIT-TO (SUBJAREA , " DECI ")) AND (LIMIT-TO (EXACTSRCTITLE , "Expert Systems With Applications ") OR LIMIT-TO (EXACTSRCTITLE , " European Journal Of Operational Research ") OR LIMIT-TO (EXACTSRCTITLE , " Journal Of Forecasting ") OR LIMIT-TO (EXACTSRCTITLE , " Neurocomputing ") OR LIMIT-TO (EXACTSRCTITLE , " Journal Of Finance ") OR LIMIT-TO (EXACTSRCTITLE , " Decision Support Systems ") OR LIMIT-TO (EXACTSRCTITLE , " Knowledge Based Systems ") OR LIMIT-TO (EXACTSRCTITLE , " Journal Of Financial Research ") OR LIMIT-TO (EXACTSRCTITLE , " Journal Of The Operational Research Society ") OR LIMIT-TO (EXACTSRCTITLE , " Neural Computing And Applications ") OR LIMIT-TO (EXACTSRCTITLE , " International Journal Of Forecasting ") OR LIMIT-TO (EXACTSRCTITLE , " Management Science ") OR LIMIT-TO (EXACTSRCTITLE , " Neural Network World ") OR LIMIT-TO (EXACTSRCTITLE , " Journal Of Behavioral Finance ") OR LIMIT-TO (EXACTSRCTITLE , " European Journal Of Finance ") OR LIMIT-TO (EXACTSRCTITLE , " Quantitative Finance ") OR LIMIT-TO (EXACTSRCTITLE , " Expert Systems ") OR LIMIT-TO (EXACTSRCTITLE , " Review Of Quantitative Finance And Accounting ") OR LIMIT-TO (EXACTSRCTITLE , " Review Of Accounting Studies ") OR LIMIT-TO (EXACTSRCTITLE , " Applied Economics ") OR LIMIT-TO (EXACTSRCTITLE , " Computational Economics ") OR LIMIT-TO (EXACTSRCTITLE , " Decision Sciences ") OR LIMIT-TO (EXACTSRCTITLE , " Financial Review ") OR LIMIT-TO (EXACTSRCTITLE , " Intelligent Data Analysis ") OR LIMIT-TO (EXACTSRCTITLE , " Journal Of Information And Computational Science ") OR LIMIT-TO (EXACTSRCTITLE , " Managerial Finance ") OR LIMIT-TO (EXACTSRCTITLE , " International Journal Of Computational Intelligence Systems ") OR LIMIT-TO (EXACTSRCTITLE , " International Journal Of Technology Management ") OR LIMIT-TO (EXACTSRCTITLE , " Journal Of Applied Business Research ") OR LIMIT-TO (EXACTSRCTITLE , " Journal Of Banking And Finance ") OR LIMIT-TO (EXACTSRCTITLE , " Neural Networks ") OR LIMIT-TO (EXACTSRCTITLE , " Applied Artificial Intelligence ") OR LIMIT-TO (EXACTSRCTITLE , " Applied Economics Letters ") OR LIMIT-TO (EXACTSRCTITLE , " Applied Financial Economics ") OR LIMIT-TO (EXACTSRCTITLE , " Applied Mathematics And Computation ") OR LIMIT-TO (EXACTSRCTITLE , " Journal Of Accounting And Economics ") OR LIMIT-TO (EXACTSRCTITLE , " Journal Of Applied Econometrics ") OR LIMIT-TO (EXACTSRCTITLE , " Journal Of Computational Information Systems ") OR LIMIT-TO (EXACTSRCTITLE , " Mathematical Problems In Engineering ") OR LIMIT-TO (EXACTSRCTITLE , " Accounting Finance ") OR LIMIT-TO (EXACTSRCTITLE , " Accounting Review ") OR LIMIT-TO (EXACTSRCTITLE , " Economic Modelling ") OR LIMIT-TO (EXACTSRCTITLE , " International Review Of Finance ") OR LIMIT-TO (EXACTSRCTITLE , " Journal Of Business And Economic Statistics ") OR LIMIT-TO (EXACTSRCTITLE , " Journal Of Machine Learning Research ") OR LIMIT-TO (EXACTSRCTITLE , " Accounting Research Journal ") OR LIMIT-TO (EXACTSRCTITLE , " Artificial Intelligence Review ") OR LIMIT-TO (EXACTSRCTITLE , " Computational Statistics And Data Analysis ") OR LIMIT-TO (EXACTSRCTITLE , " Contemporary Accounting Research ") OR LIMIT-TO (EXACTSRCTITLE , " Economic Computation And Economic Cybernetics Studies And Research ") OR LIMIT-TO (EXACTSRCTITLE , " International Journal Of Computational Intelligence And Applications ")

OR LIMIT-TO (EXACTSRCTITLE , " International Journal Of Pattern Recognition And Artificial Intelligence ") OR LIMIT-TO (EXACTSRCTITLE , " International Review Of Economics And Finance ") OR LIMIT-TO (EXACTSRCTITLE , " Journal Of Business Finance And Accounting ") OR LIMIT-TO (EXACTSRCTITLE , " Journal Of Financial Economics ") OR LIMIT-TO (EXACTSRCTITLE , " Review Of Accounting And Finance ") OR LIMIT-TO (EXACTSRCTITLE , " Review Of Finance ") OR LIMIT-TO (EXACTSRCTITLE , " Academy Of Accounting And Financial Studies Journal ") OR LIMIT-TO (EXACTSRCTITLE , " Banking And Finance Review ") OR LIMIT-TO (EXACTSRCTITLE , " Econometric Reviews ") OR LIMIT-TO (EXACTSRCTITLE , " IBM Journal Of Research And Development ") OR LIMIT-TO (EXACTSRCTITLE , " Intelligent Decision Technologies ") OR LIMIT-TO (EXACTSRCTITLE , " Intelligent Systems In Accounting Finance And Management ") OR LIMIT-TO (EXACTSRCTITLE , " International Journal Of Accounting ") OR LIMIT-TO (EXACTSRCTITLE , " International Research Journal Of Finance And Economics ") OR LIMIT-TO (EXACTSRCTITLE , " Journal Of Business Ethics ") OR LIMIT-TO (EXACTSRCTITLE , " Journal Of Computational Science ") OR LIMIT-TO (EXACTSRCTITLE , " Journal Of Construction Engineering And Management ") OR LIMIT-TO (EXACTSRCTITLE , " Journal Of Information Science And Engineering ") OR LIMIT-TO (EXACTSRCTITLE , " North American Journal Of Economics And Finance ") OR LIMIT-TO (EXACTSRCTITLE , " AI Magazine ") OR LIMIT-TO (EXACTSRCTITLE , " Accounting And Business Research ") OR LIMIT-TO (EXACTSRCTITLE , " Accounting And Finance ") OR LIMIT-TO (EXACTSRCTITLE , " Accounting Horizons ") OR LIMIT-TO (EXACTSRCTITLE , " Advances In Accounting ") OR LIMIT-TO (EXACTSRCTITLE , " Computational Intelligence ") OR LIMIT-TO (EXACTSRCTITLE , " Computational Intelligence And Neuroscience ") OR LIMIT-TO (EXACTSRCTITLE , " Computers And Mathematics With Applications ") OR LIMIT-TO (EXACTSRCTITLE , " International Journal Of Computer Applications In Technology ") OR LIMIT-TO (EXACTSRCTITLE , " International Journal Of Managerial Finance ") OR LIMIT-TO (EXACTSRCTITLE , " International Journal Of Theoretical And Applied Finance ") OR LIMIT-TO (EXACTSRCTITLE , " International Review Of Financial Analysis ") OR LIMIT-TO (EXACTSRCTITLE , " Investment Management And Financial Innovations ") OR LIMIT-TO (EXACTSRCTITLE , " Journal Of Accounting Auditing Finance ") OR LIMIT-TO (EXACTSRCTITLE , " Journal Of Accounting Research ") OR LIMIT-TO (EXACTSRCTITLE , " Journal Of Business Research ") OR LIMIT-TO (EXACTSRCTITLE , " Journal Of Computational And Theoretical Nanoscience ") OR LIMIT-TO (EXACTSRCTITLE , " Journal Of Economic Methodology ") OR LIMIT-TO (EXACTSRCTITLE , " Journal Of Financial And Quantitative Analysis "))

Appendix II. List of Words Indicative of Fraud



LIST OF WORDS INDICATIVE OF FRAUD

Cecchini et al (2010)	Glancy and Yadav (2011)
Year end December	Represent
Year end	account
Company have	relate
Research development expense	continue
Interest income	reduce
Interest expense	reflect
Gross profit	sell
Research development	expect
Company expect	
Net income	
Net sale	
Liquidity capital ressource	
Capital expenditure	
Operate expense	
Cost sale	
Company believe	
Foreign currency	
Company plan	
Income tax	
Foreign currency exchange	

Appendix Table I: List of words indicative of fraud

Appendix III. MSCI Global Industry Classification Standard (GICS)

Definitions of Sectors (MSCI, 2016):

*"**Energy Sector:** The Energy Sector comprises companies engaged in exploration & production, refining & marketing, and storage & transportation of oil & gas and coal & consumable fuels. It also includes companies that offer oil & gas equipment and services.*

***Materials Sector:** The Materials Sector includes companies that manufacture chemicals, construction materials, glass, paper, forest products and related packaging products, and metals, minerals and mining companies, including producers of steel.*

***Industrials Sector:** The Industrials Sector includes manufacturers and distributors of capital goods such as aerospace & defense, building products, electrical equipment and machinery and companies that offer construction & engineering services. It also includes providers of commercial & professional services including printing, environmental and facilities services, office services & supplies, security & alarm services, human resource & employment services, research & consulting services. It also includes companies that provide transportation services.*

***Consumer Discretionary Sector:** The Consumer Discretionary Sector encompasses those businesses that tend to be the most sensitive to economic cycles. Its manufacturing segment includes automotive, household durable goods, leisure equipment and textiles & apparel. The services segment includes hotels, restaurants and other leisure facilities, media production and services, and consumer retailing and services.*

***Consumer Staples Sector:** The Consumer Staples Sector comprises companies whose businesses are less sensitive to economic cycles. It includes manufacturers and distributors of food, beverages and tobacco and producers of non-durable household goods and personal products. It also includes food & drug retailing companies as well as hypermarkets and consumer super centers.*

***Health Care Sector:** The Health Care Sector includes health care providers & services, companies that manufacture and distribute health care equipment & supplies, and health care technology companies.*

It also includes companies involved in the research, development, production and marketing of pharmaceuticals and biotechnology products.

Financials Sector: *The Financials Sector contains companies involved in banking, thrifts & mortgage finance, specialized finance, consumer finance, asset management and custody banks, investment banking and brokerage and insurance. It also includes Financial Exchanges & Data and Mortgage REITs.*

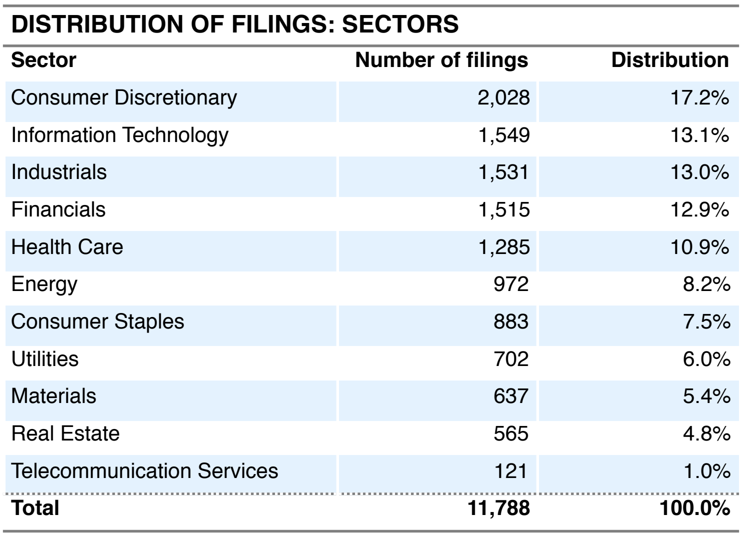
Information Technology Sector: *The Information Technology Sector comprises companies that offer software and information technology services, manufacturers and distributors of technology hardware & equipment such as communications equipment, cellular phones, computers & peripherals, electronic equipment and related instruments, and semiconductors.*

Telecommunication Services Sector: *The Telecommunication Services Sector contains companies that provide communications services primarily through a fixed-line, cellular or wireless, high bandwidth and/or fiber optic cable network.*

Utilities Sector: *The Utilities Sector comprises utility companies such as electric, gas and water utilities. It also includes independent power producers & energy traders and companies that engage in generation and distribution of electricity using renewable sources.*

Real Estate Sector: *The Real Estate Sector contains companies engaged in real estate development and operation. It also includes companies offering real estate related services and Equity Real Estate Investment Trusts (REITs)."*

Appendix IV. Table Showing Distribution of Filings on Sectors



DISTRIBUTION OF FILINGS: SECTORS

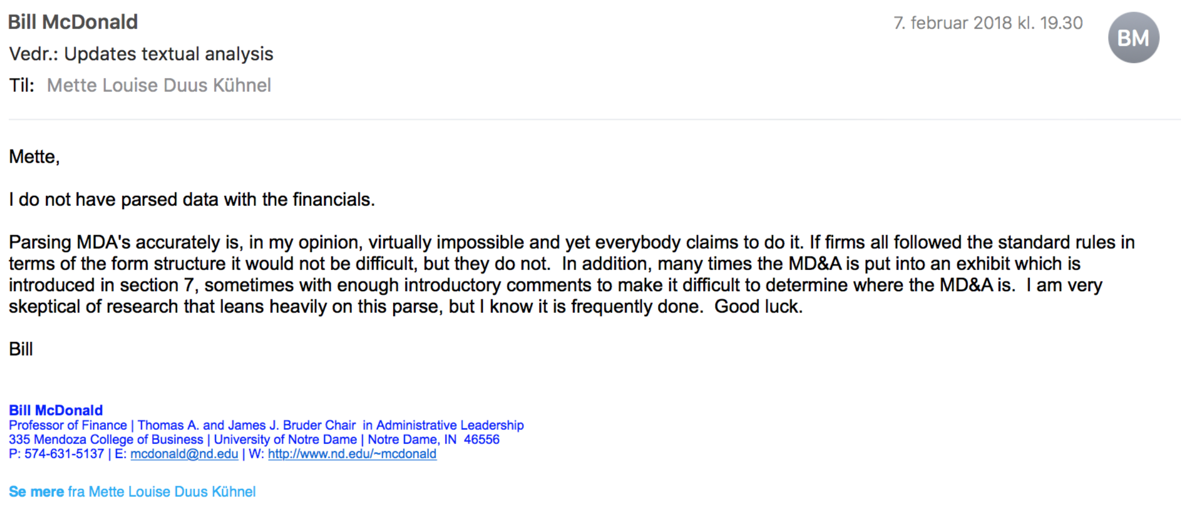
Sector	Number of filings	Distribution
Consumer Discretionary	2,028	17.2%
Information Technology	1,549	13.1%
Industrials	1,531	13.0%
Financials	1,515	12.9%
Health Care	1,285	10.9%
Energy	972	8.2%
Consumer Staples	883	7.5%
Utilities	702	6.0%
Materials	637	5.4%
Real Estate	565	4.8%
Telecommunication Services	121	1.0%
Total	**11,788**	**100.0%**

Appendix Table II: Distribution of filings on all sectors

Appendix V. 10-X Parse String

```
def pre_parse_mcdonald(txt, module_logger):
    # Parses the whole mcdonald report parsing out other noisy and non-important parts
    try:
        module_logger.debug('Running pre parsing of a McDonald report')
        regex_remove_numeric_on_newline = re.compile(r'^[0-9]*$')
        result_string = regex_remove_numeric_on_newline.sub('', txt)
        regex_remove_rome_numbers = re.compile(r'^[IVX\.]*$')
        result_string = regex_remove_rome_numbers.sub('', result_string,)
        regex_remove_numeric_on_newline_plusbind = re.compile(r'\-\d+\-')
        result_string = regex_remove_numeric_on_newline_plusbind.sub('', result_string)
        regex_remove_numeric_on_newline_plus2bind = re.compile(r'\-\s\d{1,3}\s\-')
        result_string = regex_remove_numeric_on_newline_plus2bind.sub('', result_string)
        regex_remove_alone_num = re.compile(r'\s\s+\d')
        result_string = regex_remove_alone_num.sub('', result_string)
        regex_remove_tabs = re.compile(r'\t')
        result_string = regex_remove_tabs.sub('', result_string)
        regex_parts = re.compile(r'PART\s\d{1}')
        result_string = regex_parts.sub('', result_string)
        regex_parts_rome =
re.compile(r'PART\s(M{0,4}(CM|CD|D?C{0,3})(XC|XL|L?X{0,3})(IX|IV|V?I{0,3}))')
        result_string = regex_parts_rome.sub('', result_string)
        regex_remove_exhibits = re.compile(r'(<EX\-(\d{1,2}|\d{1,2}\.\d{1,2}))((.|[\n\r])*)')
        result_string = regex_remove_exhibits.sub('', result_string)
        regex_remove_backslash = re.compile(r'\\\\')
        result_string = regex_remove_backslash.sub('', result_string)
        regex_remove_newline_between_item_and_num = re.compile(r'(Item|item|ITEM)(\n|\r)')
        result_string = regex_remove_newline_between_item_and_num.sub('Item ', result_string)

    except Exception as e:
        module_logger.error('Error precleaning the report', e)
        result_string = ''

    return result_string
def extract_all_items(txt, module_logger):


    txt = txt.lower()

    remove_tags = re.compile(r'<[^>]+>')

    txt = remove_tags.sub('', txt)
    regex_remove_newline = re.compile(r'(\n|\r)')
    result_string = regex_remove_newline.sub(' ', txt)
    result_length = 0
    appear = 'item 7' in txt
    result_dict = {}

    result_dict['all'] = {'appearance': appear, 'number_of_items': result_length}

    return result_dict, result_string
```

Appendix VI. E-mail from Bill McDonald



Bill McDonald

7. februar 2018 kl. 19.30

Vedr.: Updates textual analysis

Til: Mette Louise Duus Kühnel

Mette,

I do not have parsed data with the financials.

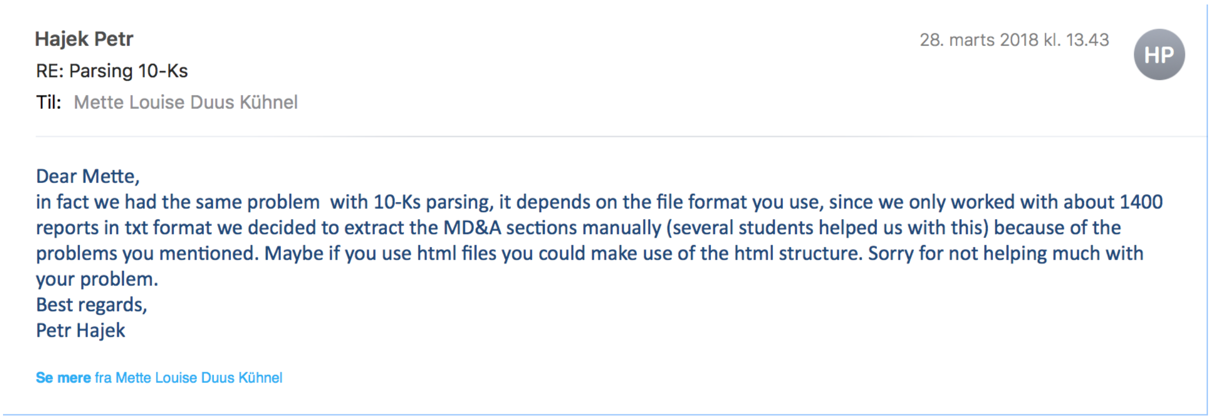
Parsing MDA's accurately is, in my opinion, virtually impossible and yet everybody claims to do it. If firms all followed the standard rules in terms of the form structure it would not be difficult, but they do not. In addition, many times the MD&A is put into an exhibit which is introduced in section 7, sometimes with enough introductory comments to make it difficult to determine where the MD&A is. I am very skeptical of research that leans heavily on this parse, but I know it is frequently done. Good luck.

Bill

Bill McDonald
Professor of Finance | Thomas A. and James J. Bruder Chair in Administrative Leadership
335 Mendoza College of Business | University of Notre Dame | Notre Dame, IN 46556
P: 574-631-5137 | E: mcdonald@nd.edu | W: http://www.nd.edu/~mcdonald

Se mere fra Mette Louise Duus Kühnel

Appendix VII. E-mail from Petr Hajek



Hajek Petr

28. marts 2018 kl. 13.43

RE: Parsing 10-Ks

Til: Mette Louise Duus Kühnel

Dear Mette,
in fact we had the same problem with 10-Ks parsing, it depends on the file format you use, since we only worked with about 1400 reports in txt format we decided to extract the MD&A sections manually (several students helped us with this) because of the problems you mentioned. Maybe if you use html files you could make use of the html structure. Sorry for not helping much with your problem.
Best regards,
Petr Hajek

Se mere fra Mette Louise Duus Kühnel

Appendix VIII. Attempts Made to Extract MD&As

Several extraction approaches were attempted. The first was to define a logic that stated that the section to be extracted had "Item 7" as a starting point and a later item (such as "Item 7A" or "Item 8" - these items were defined in a list used in the code) as an ending point. Such a logic had several shortfalls. In a majority of the cases, an outcome of more than one result per filing would appear. Oftentimes a piece of text from the table of content would show up as an outcome. This could be handled by setting up a criteria demanding that the string had a minimum length. The more significant problem consisted in the fact that such a logic did not take into consideration the cases in which Item 7 was referred to in other parts of the filing[43]. Such a referral would cause the constructed logic to believe that item 7 had begun, but unless a later item was similarly referred to (before the actual Item 7), it would not stop the inclusion until a later item actually occurred or was referred to - thus, after the actual appearance of item 7. Even if a later item was referred to earlier, which was observed in some cases, it was difficult to define a logic that would be able to choose between the outcomes, as length would no longer be a suitable determinant. An attempt to add "Item 7" as an ending term to avoid outcomes that e.g. started with a referral in Item 2 and continued until after Item 7 (thus resulting in more outputs) was subsequently made, as such a logic would cause the inclusion to stop when the actual Item 7 occurred. Similarly, it was not possible to construct a logic suitable to identify the correct MD&A outcome. Lastly, an attempt, in which the methodology used in previous research focusing on MD&As was relied on, was made. Li (2008) describes the extraction method applied in his study in detail (see Appendix IX). When assessing a sample of the outcome resulting from using the Li's method, it was found that its quality seemed to be good. However, it was only possible to extract MD&As from about 25% of the filings. As the modeling required continuous data for the companies included, using Li's method did not result in a sample suitable for the purpose of this study.

Davis and Tama-Sweet (2012) elaborate on their struggle to extract MD&As. Examining a random sample of 150 reports, they discover that in about 10% of them, their string has been unable to exactly identify and extract the right section. They state that in most of these cases, no text has been extracted at all. The authors furthermore state they choose to leave out reports with incorrect extraction. They do not elaborate on cases with other wrongful extractions than missing texts or on how they are able to identify these cases when leaving out reports with incorrect extractions.

[43] PharMerica e.g. write in Item 1 and 6, respectively, of their 10-K report from 2010-Q1: "*For information about the corporation's practices relating to working capital items, see Item 7, Management's Discussion and Analysis of Financial Condition and Results of Operations*" and "*The following table presents our selected historical consolidated financial and operating data. The selected historical financial and operating data should be read in conjunction with, and is qualified in its entirety by reference to, Item 7, Management's Discussion and Analysis of Financial Condition and Results of Operations...*"

Appendix IX.“Steps to extract MD&A and Notes to the financial statements” (Li, 2008, Appendix B)

An elaboration of the methodology applied by Li (2008) when parsing out MD&As from 10-K reports.

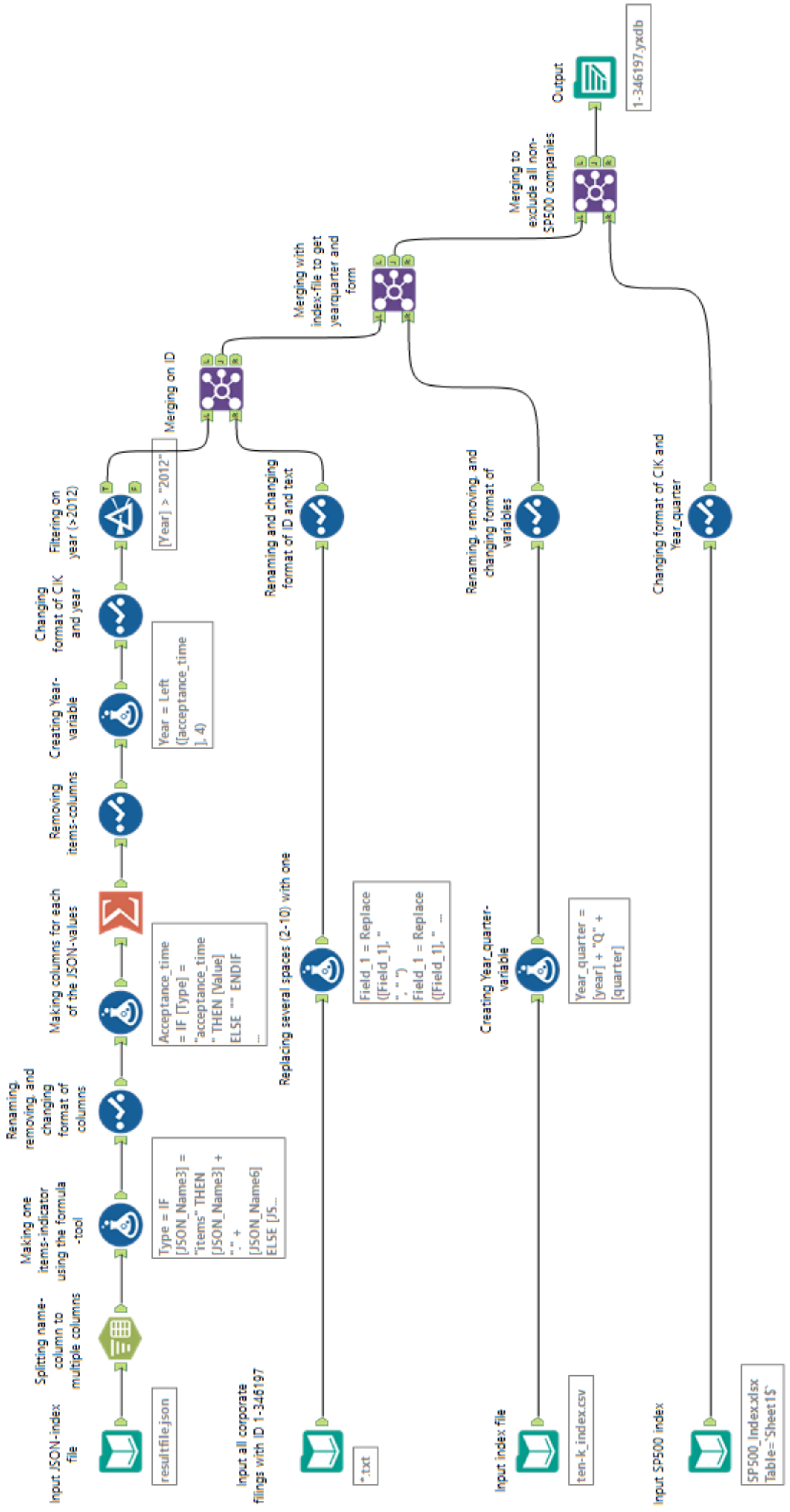
“This appendix explains the details of extracting the MD&A section and Notes from 10-K filings. Starting with the raw 10-K file, I first delete the SEC-header information, all the contents between <TABLE> and </TABLE> text, the paragraphs that contain <S> or <C>, all the tags in the format of <...> and <&...> are removed using the same process described in Appendix A.

Within the remaining text, the program identifies a line that satisfies one of the following criteria as the beginning of the MD&A section: (1) the line starts with ‘‘management’s discussion’’ or ‘‘management’s discussion’’ following some white spaces; (2) the line contains ‘‘management’s discussion’’ and (‘‘item’’+one or more white space+‘‘7’’) and does not contain the word ‘‘see’’; (3) the line starts with some white spaces followed by ‘‘managements discussion’’ or ‘‘managements discussion’’; or (4) the line contains ‘‘managements discussion’’ and (‘‘item’’+one or more white space+‘‘7’’) and does not contain the word ‘‘see.’’ Since many firms refer to the MD&A section in the front-matter of the annual reports, the word ‘‘see’’ serves to identify all such situations. The program identifies a line that satisfies one of the following criteria as the ending of the MD&A section: (1) the line begins with some white spaces followed by ‘‘Financial Statements’’ or ‘‘Financial Statements’’; (2) the line contains ‘‘item’’ followed by one or more white spaces and the number ‘‘8’’; (3) the line contains ‘‘Supplementary Data’’; or (4) the line begins with some white spaces followed by ‘‘SUMMARY OF SELECTED FINANCIAL DATA’’ or ‘‘SUMMARY OF SELECTED FINANCIAL DATA.’’ Most firms have a table of contents listing the main sections of the 10-K filing. In some instances, this table of contents is not embedded between <TABLE> and </TABLE> and therefore is not cleaned in the previous steps. As a result, the line in the table of contents about MD&A will also be picked up by the program as part of the MD&A.

Similarly, the program identifies a line as the beginning of the Notes, if: (1) the line starts with ‘‘NOTES TO’’ or some white spaces followed by ‘‘NOTES TO’’; and (2) the line does not contain any number except when it follows ‘‘for the years ended.’’ The program identifies a line that satisfies one of the following criteria as the ending of the Notes: (1) the line contains ‘‘Changes in and Disagreements with Accountants’’ or ‘‘DISAGREEMENTS ON ACCOUNTING’’; (2) the line contains ‘‘DIRECTORS AND EXECUTIVE OFFICERS’’; or (3) the line contains ‘‘exhibit index.’’

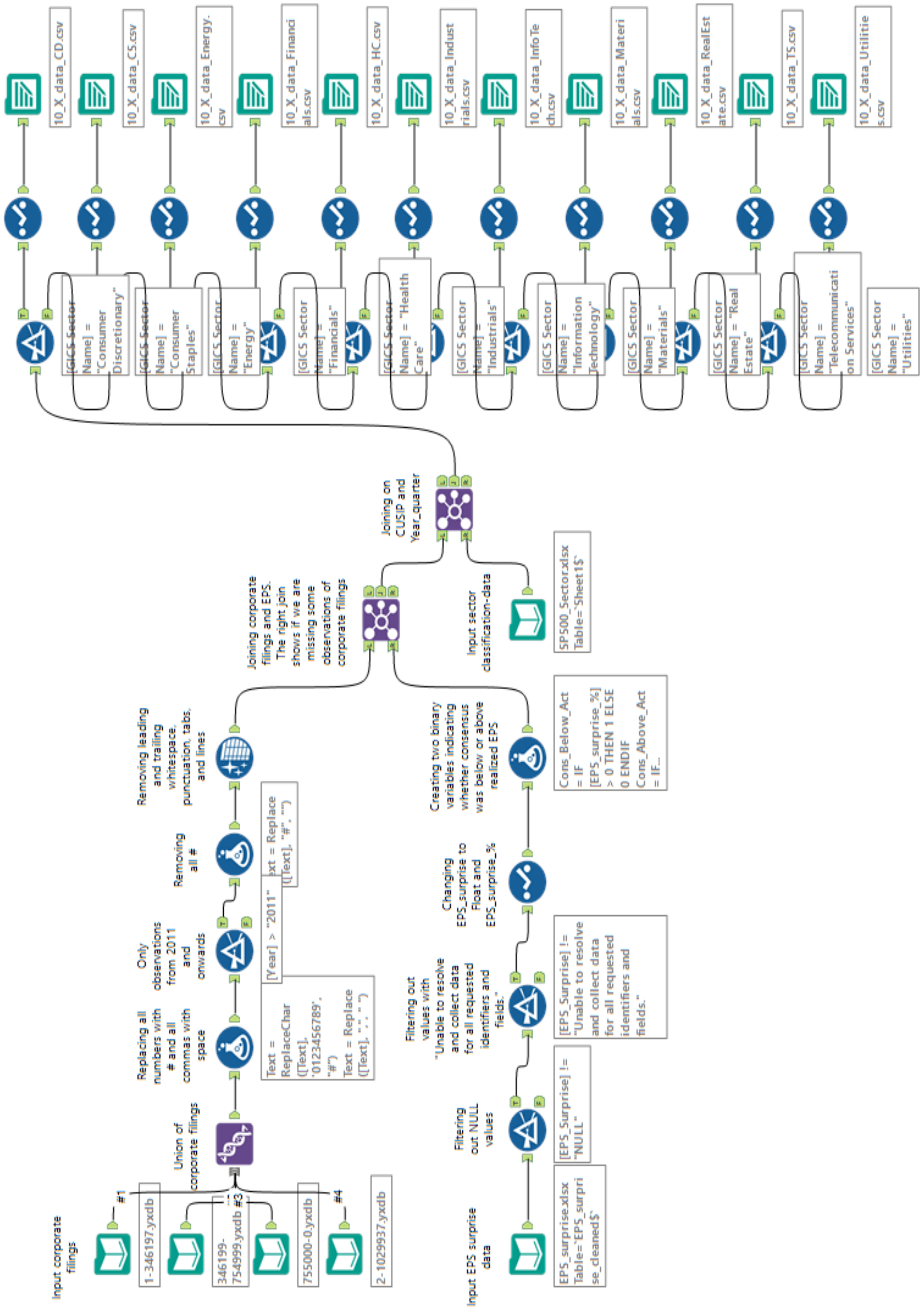
After the MD&A and the Notes are identified, all the paragraphs with more than 50% of non-alphabetic characters (e.g., white spaces or numbers) are deleted. Finally, the Fathom package is used to calculate the readability measures.”

Appendix X. Workflow 1 Processing txt-files With ID 1-34619 (Full Size)


Input JSON-index file
resultfile.json
Splitting name-column to multiple columns
Making one items-indicator using the formula -tool
Type = IF [JSON_Name3] = "items" THEN [JSON_Name3] + "." + [JSON_Name6] ELSE [JS...
Renaming, removing, and changing format of columns
Making columns for each of the JSON-values
Acceptance_time = IF [Type] = "acceptance_time" THEN [Value] ELSE "" ENDIF ...
Removing items-columns
Creating Year-variable
Year = Left ([acceptance_time], 4)
Changing format of CIK and year
Filtering on year (>2012)
[Year] > "2012"
T
F
Merging on ID
L
J
R
Input all corporate filings with ID 1-346197
*.txt
Replacing several spaces (2-10) with one
Field_1 = Replace ([Field_1], " ", " ") Field_1 = Replace ([Field_1], " ...
Renaming and changing format of ID and text
Merging with index-file to get yearquarter and form
Input index file
ten-k_index.csv
Creating Year_quarter-variable
Year_quarter = [year] + "Q" + [quarter]
Renaming, removing, and changing format of variables
Merging to exclude all non-SP500 companies
Output
1-346197.yxdb
Input SP500 index
SP500_Index.xlsx Table=`Sheet1$`
Changing format of CIK and Year_quarter

Appendix Figure I: Workflow 1 Processing txt-files With ID 1-34619 (Full Size)

Appendix XI. Workflow 2 (Full Size)



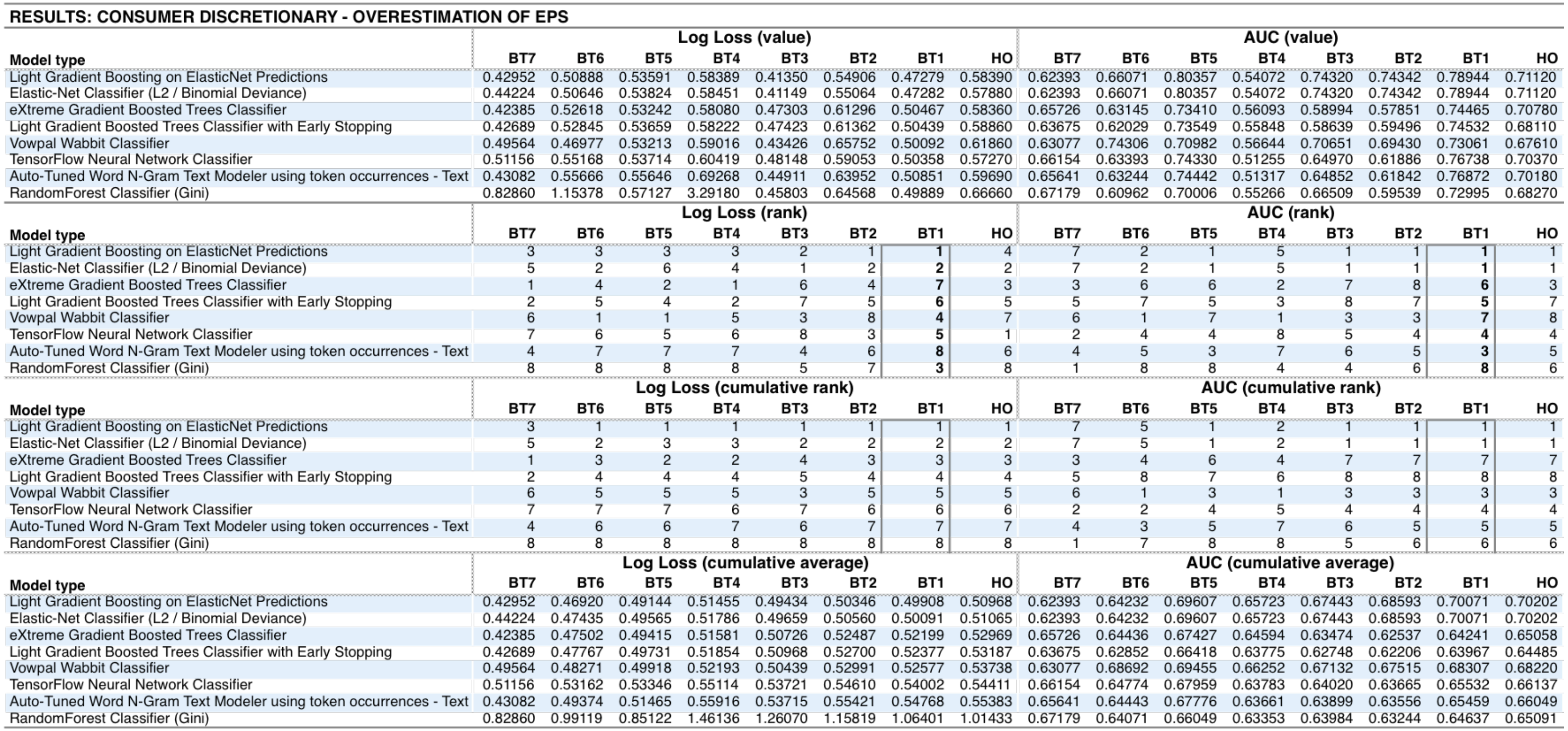
Input corporate filings
1-346197.yxdb
346199-754999.yxdb #3
755000-0.yxdb
2-1029937.yxdb
#1
#4
Union of corporate filings
Replacing all numbers with # and all commas with space
Text = ReplaceChar ([Text], '0123456789', "#") Text = Replace ([Text], ",", " ")
Only observations from 2011 and onwards
[Year] > "2011"
Removing all #
Text = Replace ([Text], "#", "")
Removing leading and trailing whitespace, punctuation, tabs, and lines
Joining corporate filings and EPS. The right join shows if we are missing some observations of corporate filings
Input EPS surprise data
EPS_surprise.xlsx Table='EPS_surprise_cleaned$'
Filtering out NULL values
[EPS_Surprise] != "NULL"
Filtering out values with "Unable to resolve and collect data for all requested identifiers and fields."
[EPS_Surprise] != "Unable to resolve and collect data for all requested identifiers and fields."
Changing EPS_surprise to Float and EPS_surprise_%
Creating two binary variables indicating whether consensus was below or above realized EPS
Cons_Below_Act = IF [EPS_surprise_%] > 0 THEN 1 ELSE 0 ENDIF Cons_Above_Act = IF...
Input sector classification-data
SP500_Sector.xlsx Table='Sheet1$'
Joining on CUSIP and Year_quarter
[GICS Sector Name] = "Consumer Discretionary"
[GICS Sector Name] = "Consumer Staples"
[GICS Sector Name] = "Energy"
[GICS Sector Name] = "Financials"
[GICS Sector Name] = "Health Care"
[GICS Sector Name] = "Industrials"
[GICS Sector Name] = "Information Technology"
[GICS Sector Name] = "Materials"
[GICS Sector Name] = "Real Estate"
[GICS Sector Name] = "Telecommunication Services"
[GICS Sector Name] = "Utilities"
10_X_data_CD.csv
10_X_data_CS.csv
10_X_data_Energy.csv
10_X_data_Financials.csv
10_X_data_HC.csv
10_X_data_Industrials.csv
10_X_data_InfoTech.csv
10_X_data_Materials.csv
10_X_data_RealEstate.csv
10_X_data_TS.csv
10_X_data_Utilities.csv
T
F
L
J
R

Appendix Figure II: Workflow 2 (Full Size)

Appendix XII. R-script for Stop Word Removal

```
library(data.table)
library(tidytext)
library(rio)

dat <- fread("[FILEPATH]")
newdat <- dat

removewords <- c(" about", " above", " after", " again", " all", " am", " among", " an",
" and", " any", " are", " as", " at", " be", " because", " been", " before", " being", "
below", " between", " both", " but", " by", " can", " did", " do", " does", " doing", " down",
" during", " each", " few", " for", " from", " further", " had", " has", " have", " having", "
he", " her", " here", " hers", " herself", " him", " himself", " his", " how", " if", " in", "
into", " is", " it", " its", " itself", " just", " me", " more", " most", " my", " myself", "
no", " nor", " not", " now", " of",
" off", " on", " once", " only", " or", " other", " our", " ours", " ourselves", " out", "
over", " own", " same", " she", " should", " so", " some", " such", " than", " that", " the",
" their", " theirs", " them", " themselves", " then", " there", " these", " they", " this", "
those", " through", " to", " too", " under", " until", " up", " very", " was", " we", " were",
" what", " when", " where", " which", " while", " who", " whom", " why", " with", " you", "
your", " yours", " yourself", " yourselves")

newdat$Text <- gsub(x = newdat$Text, pattern = paste0("\\b(",paste(removewords,
collapse="|"),")\\b"),
                 replacement = "")

sum(nchar(dat$Text))
sum(nchar(newdat$Text))

export(newdat, "[FILEPATH]")
```

Appendix XIII. Consumer Discretionary (Overestimation of EPS)



RESULTS: CONSUMER DISCRETIONARY - OVERESTIMATION OF EPS

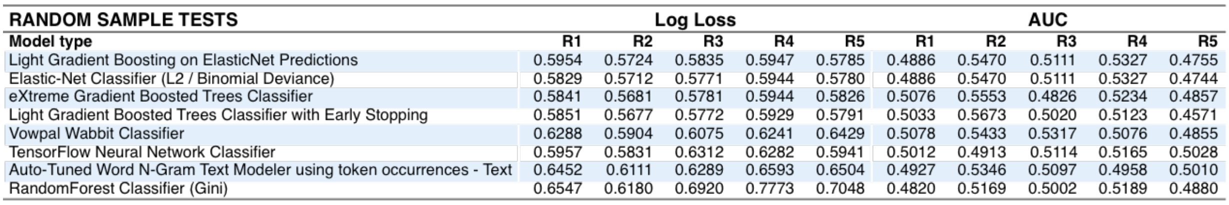
	Log Loss (value)								AUC (value)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
Light Gradient Boosting on ElasticNet Predictions	0.42952	0.50888	0.53591	0.58389	0.41350	0.54906	0.47279	0.58390	0.62393	0.66071	0.80357	0.54072	0.74320	0.74342	0.78944	0.71120
Elastic-Net Classifier (L2 / Binomial Deviance)	0.44224	0.50646	0.53824	0.58451	0.41149	0.55064	0.47282	0.57880	0.62393	0.66071	0.80357	0.54072	0.74320	0.74342	0.78944	0.71120
eXtreme Gradient Boosted Trees Classifier	0.42385	0.52618	0.53242	0.58080	0.47303	0.61296	0.50467	0.58360	0.65726	0.63145	0.73410	0.56093	0.58994	0.57851	0.74465	0.70780
Light Gradient Boosted Trees Classifier with Early Stopping	0.42689	0.52845	0.53659	0.58222	0.47423	0.61362	0.50439	0.58860	0.63675	0.62029	0.73549	0.55848	0.58639	0.59496	0.74532	0.68110
Vowpal Wabbit Classifier	0.49564	0.46977	0.53213	0.59016	0.43426	0.65752	0.50092	0.61860	0.63077	0.74306	0.70982	0.56644	0.70651	0.69430	0.73061	0.67610
TensorFlow Neural Network Classifier	0.51156	0.55168	0.53714	0.60419	0.48148	0.59053	0.50358	0.57270	0.66154	0.63393	0.74330	0.51255	0.64970	0.61886	0.76738	0.70370
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.43082	0.55666	0.55646	0.69268	0.44911	0.63952	0.50851	0.59690	0.65641	0.63244	0.74442	0.51317	0.64852	0.61842	0.76872	0.70180
RandomForest Classifier (Gini)	0.82860	1.15378	0.57127	3.29180	0.45803	0.64568	0.49889	0.66660	0.67179	0.60962	0.70006	0.55266	0.66509	0.59539	0.72995	0.68270

	Log Loss (rank)								AUC (rank)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
Light Gradient Boosting on ElasticNet Predictions	3	3	3	3	2	1	**1**	4	7	2	1	5	1	1	**1**	1
Elastic-Net Classifier (L2 / Binomial Deviance)	5	2	6	4	1	2	**2**	2	7	2	1	5	1	1	**1**	1
eXtreme Gradient Boosted Trees Classifier	1	4	2	1	6	4	**7**	3	3	6	6	2	7	8	**6**	3
Light Gradient Boosted Trees Classifier with Early Stopping	2	5	4	2	7	5	**6**	5	5	7	5	3	8	7	**5**	7
Vowpal Wabbit Classifier	6	1	1	5	3	8	**4**	7	6	1	7	1	3	3	**7**	8
TensorFlow Neural Network Classifier	7	6	5	6	8	3	**5**	1	2	4	4	8	5	4	**4**	4
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	4	7	7	7	4	6	**8**	6	4	5	3	7	6	5	**3**	5
RandomForest Classifier (Gini)	8	8	8	8	5	7	**3**	8	1	8	8	4	4	6	**8**	6

	Log Loss (cumulative rank)								AUC (cumulative rank)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
Light Gradient Boosting on ElasticNet Predictions	3	1	1	1	1	1	1	1	7	5	1	2	1	1	1	1
Elastic-Net Classifier (L2 / Binomial Deviance)	5	2	3	3	2	2	2	2	7	5	1	2	1	1	1	1
eXtreme Gradient Boosted Trees Classifier	1	3	2	2	4	3	3	3	3	4	6	4	7	7	7	7
Light Gradient Boosted Trees Classifier with Early Stopping	2	4	4	4	5	4	4	4	5	8	7	6	8	8	8	8
Vowpal Wabbit Classifier	6	5	5	5	3	5	5	5	6	1	3	1	3	3	3	3
TensorFlow Neural Network Classifier	7	7	7	6	7	6	6	6	2	2	4	5	4	4	4	4
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	4	6	6	7	6	7	7	7	4	3	5	7	6	5	5	5
RandomForest Classifier (Gini)	8	8	8	8	8	8	8	8	1	7	8	8	5	6	6	6

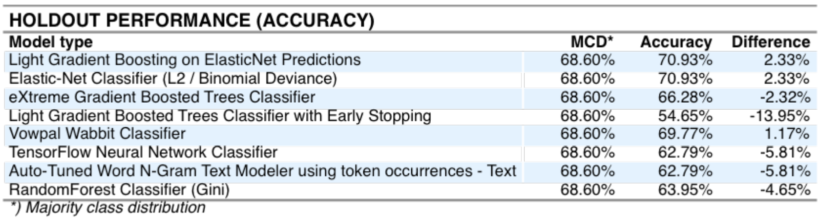
	Log Loss (cumulative average)								AUC (cumulative average)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
Light Gradient Boosting on ElasticNet Predictions	0.42952	0.46920	0.49144	0.51455	0.49434	0.50346	0.49908	0.50968	0.62393	0.64232	0.69607	0.65723	0.67443	0.68593	0.70071	0.70202
Elastic-Net Classifier (L2 / Binomial Deviance)	0.44224	0.47435	0.49565	0.51786	0.49659	0.50560	0.50091	0.51065	0.62393	0.64232	0.69607	0.65723	0.67443	0.68593	0.70071	0.70202
eXtreme Gradient Boosted Trees Classifier	0.42385	0.47502	0.49415	0.51581	0.50726	0.52487	0.52199	0.52969	0.65726	0.64436	0.67427	0.64594	0.63474	0.62537	0.64241	0.65058
Light Gradient Boosted Trees Classifier with Early Stopping	0.42689	0.47767	0.49731	0.51854	0.50968	0.52700	0.52377	0.53187	0.63675	0.62852	0.66418	0.63775	0.62748	0.62206	0.63967	0.64485
Vowpal Wabbit Classifier	0.49564	0.48271	0.49918	0.52193	0.50439	0.52991	0.52577	0.53738	0.63077	0.68692	0.69455	0.66252	0.67132	0.67515	0.68307	0.68220
TensorFlow Neural Network Classifier	0.51156	0.53162	0.53346	0.55114	0.53721	0.54610	0.54002	0.54411	0.66154	0.64774	0.67959	0.63783	0.64020	0.63665	0.65532	0.66137
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.43082	0.49374	0.51465	0.55916	0.53715	0.55421	0.54768	0.55383	0.65641	0.64443	0.67776	0.63661	0.63899	0.63556	0.65459	0.66049
RandomForest Classifier (Gini)	0.82860	0.99119	0.85122	1.46136	1.26070	1.15819	1.06401	1.01433	0.67179	0.64071	0.66049	0.63353	0.63984	0.63244	0.64637	0.65091

Appendix Table III: Log Loss and AUC (Values and ranks)



RANDOM SAMPLE TESTS	Log Loss					AUC				
Model type	R1	R2	R3	R4	R5	R1	R2	R3	R4	R5
Light Gradient Boosting on ElasticNet Predictions	0.5954	0.5724	0.5835	0.5947	0.5785	0.4886	0.5470	0.5111	0.5327	0.4755
Elastic-Net Classifier (L2 / Binomial Deviance)	0.5829	0.5712	0.5771	0.5944	0.5780	0.4886	0.5470	0.5111	0.5327	0.4744
eXtreme Gradient Boosted Trees Classifier	0.5841	0.5681	0.5781	0.5944	0.5826	0.5076	0.5553	0.4826	0.5234	0.4857
Light Gradient Boosted Trees Classifier with Early Stopping	0.5851	0.5677	0.5772	0.5929	0.5791	0.5033	0.5673	0.5020	0.5123	0.4571
Vowpal Wabbit Classifier	0.6288	0.5904	0.6075	0.6241	0.6429	0.5078	0.5433	0.5317	0.5076	0.4855
TensorFlow Neural Network Classifier	0.5957	0.5831	0.6312	0.6282	0.5941	0.5012	0.4913	0.5114	0.5165	0.5028
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.6452	0.6111	0.6289	0.6593	0.6504	0.4927	0.5346	0.5097	0.4958	0.5010
RandomForest Classifier (Gini)	0.6547	0.6180	0.6920	0.7773	0.7048	0.4820	0.5169	0.5002	0.5189	0.4880

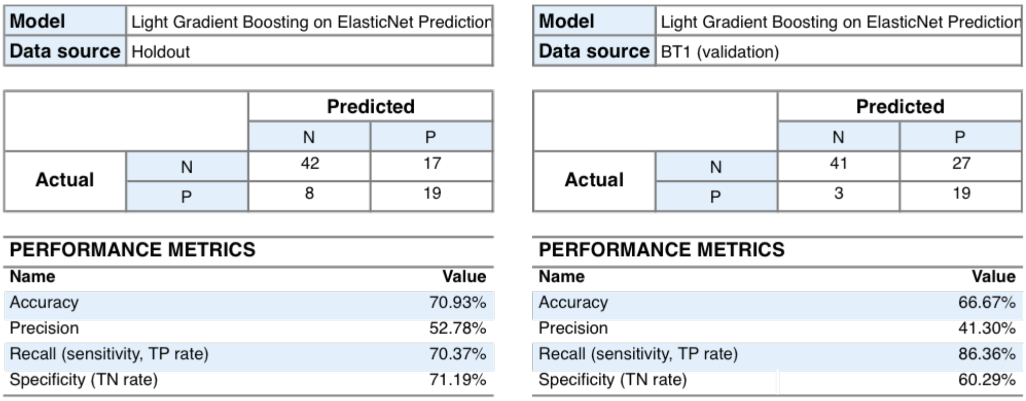
Appendix Table IV: Random sample test



HOLDOUT PERFORMANCE (ACCURACY)			
Model type	MCD*	Accuracy	Difference
Light Gradient Boosting on ElasticNet Predictions	68.60%	70.93%	2.33%
Elastic-Net Classifier (L2 / Binomial Deviance)	68.60%	70.93%	2.33%
eXtreme Gradient Boosted Trees Classifier	68.60%	66.28%	-2.32%
Light Gradient Boosted Trees Classifier with Early Stopping	68.60%	54.65%	-13.95%
Vowpal Wabbit Classifier	68.60%	69.77%	1.17%
TensorFlow Neural Network Classifier	68.60%	62.79%	-5.81%
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	68.60%	62.79%	-5.81%
RandomForest Classifier (Gini)	68.60%	63.95%	-4.65%

*) Majority class distribution

Appendix Table V: Holdout performance



Model	Light Gradient Boosting on ElasticNet Predictior
Data source	Holdout

		Predicted	
		N	P
Actual	N	42	17
	P	8	19

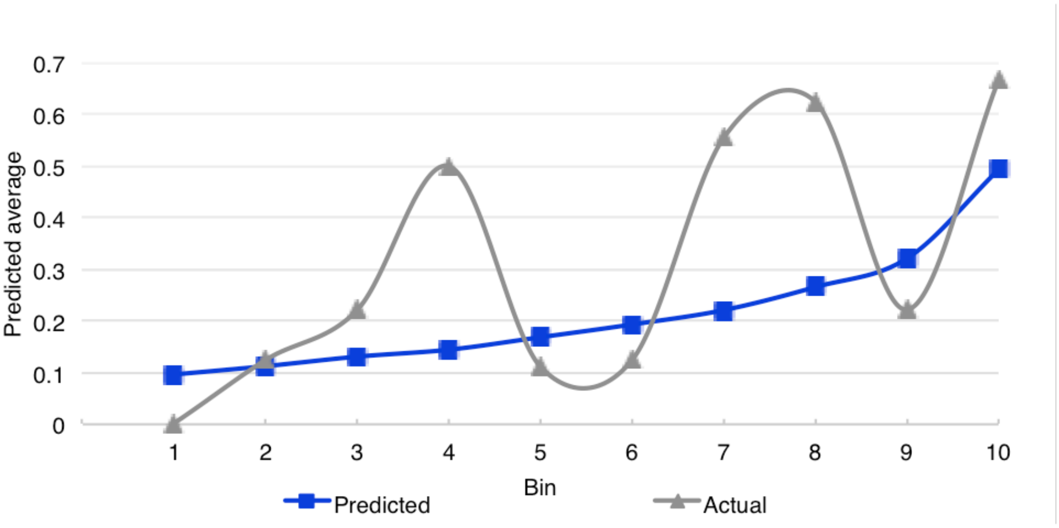
PERFORMANCE METRICS	
Name	Value
Accuracy	70.93%
Precision	52.78%
Recall (sensitivity, TP rate)	70.37%
Specificity (TN rate)	71.19%

Model	Light Gradient Boosting on ElasticNet Predictior
Data source	BT1 (validation)

		Predicted	
		N	P
Actual	N	41	27
	P	3	19

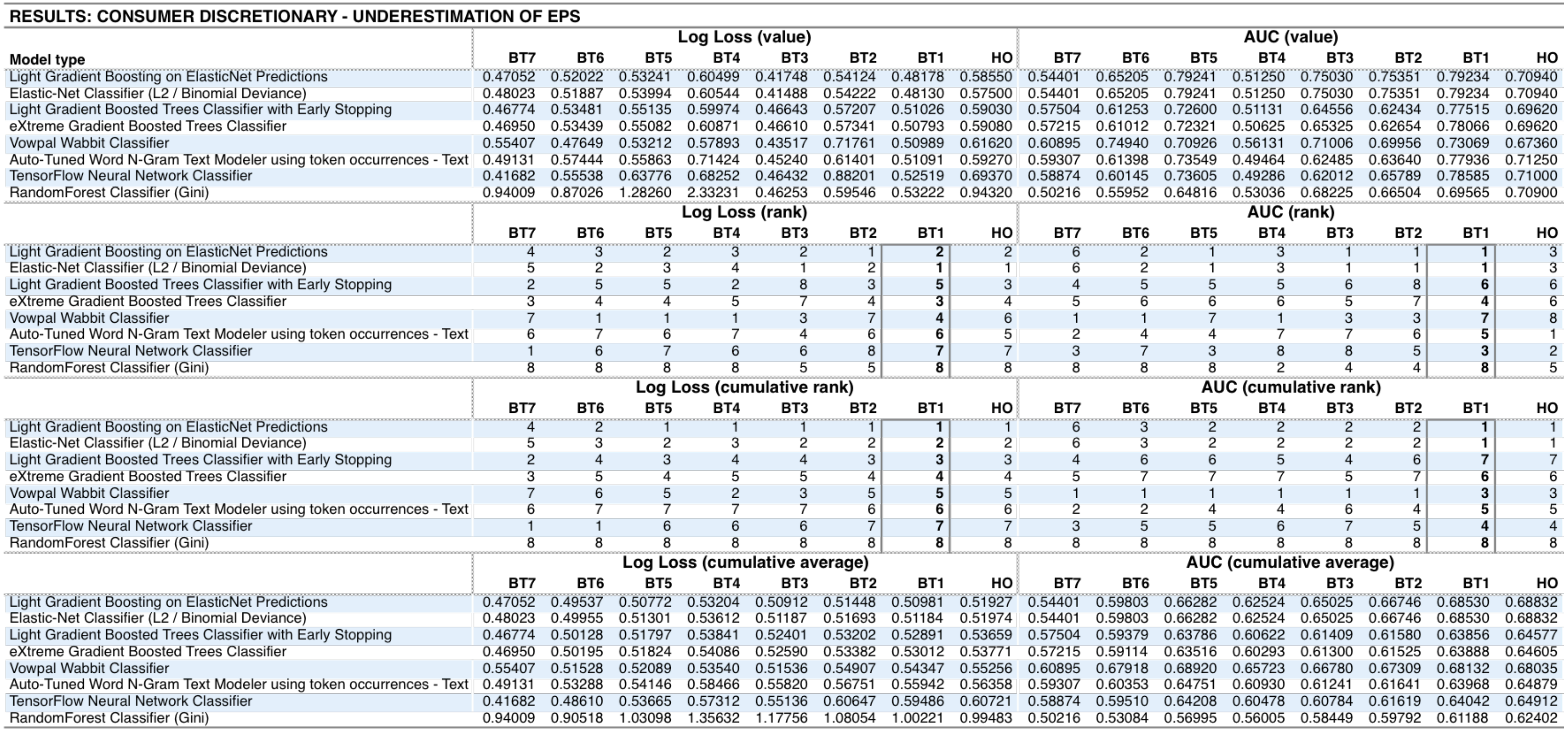
PERFORMANCE METRICS	
Name	Value
Accuracy	66.67%
Precision	41.30%
Recall (sensitivity, TP rate)	86.36%
Specificity (TN rate)	60.29%

Appendix Table VI: Confusion matrices – Holdout and validation



Appendix Table VII: Lift chart

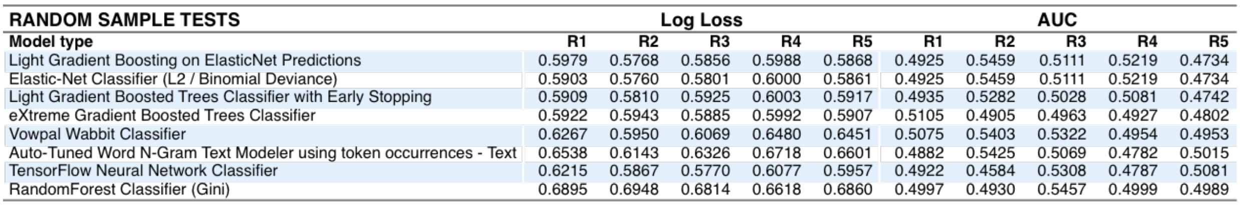
Appendix XIV. Consumer Discretionary (Underestimation of EPS)



RESULTS: CONSUMER DISCRETIONARY - UNDERESTIMATION OF EPS

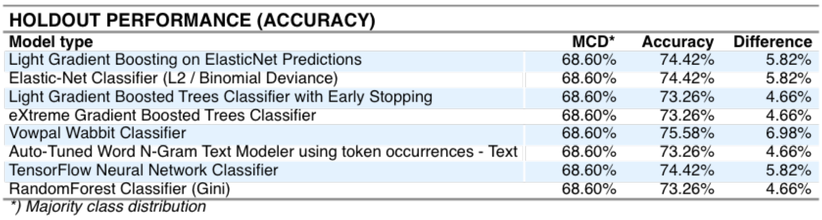
	Log Loss (value)								AUC (value)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
Light Gradient Boosting on ElasticNet Predictions	0.47052	0.52022	0.53241	0.60499	0.41748	0.54124	0.48178	0.58550	0.54401	0.65205	0.79241	0.51250	0.75030	0.75351	0.79234	0.70940
Elastic-Net Classifier (L2 / Binomial Deviance)	0.48023	0.51887	0.53994	0.60544	0.41488	0.54222	0.48130	0.57500	0.54401	0.65205	0.79241	0.51250	0.75030	0.75351	0.79234	0.70940
Light Gradient Boosted Trees Classifier with Early Stopping	0.46774	0.53481	0.55135	0.59974	0.46643	0.57207	0.51026	0.59030	0.57504	0.61253	0.72600	0.51131	0.64556	0.62434	0.77515	0.69620
eXtreme Gradient Boosted Trees Classifier	0.46950	0.53439	0.55082	0.60871	0.46610	0.57341	0.50793	0.59080	0.57215	0.61012	0.72321	0.50625	0.65325	0.62654	0.78066	0.69620
Vowpal Wabbit Classifier	0.55407	0.47649	0.53212	0.57893	0.43517	0.71761	0.50989	0.61620	0.60895	0.74940	0.70926	0.56131	0.71006	0.69956	0.73069	0.67360
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.49131	0.57444	0.55863	0.71424	0.45240	0.61401	0.51091	0.59270	0.59307	0.61398	0.73549	0.49464	0.62485	0.63640	0.77936	0.71250
TensorFlow Neural Network Classifier	0.41682	0.55538	0.63776	0.68252	0.46432	0.88201	0.52519	0.69370	0.58874	0.60145	0.73605	0.49286	0.62012	0.65789	0.78585	0.71000
RandomForest Classifier (Gini)	0.94009	0.87026	1.28260	2.33231	0.46253	0.59546	0.53222	0.94320	0.50216	0.55952	0.64816	0.53036	0.68225	0.66504	0.69565	0.70900
	Log Loss (rank)								AUC (rank)							
	BT7	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
Light Gradient Boosting on ElasticNet Predictions	4	3	2	3	2	1	**2**	2	6	2	1	3	1	1	**1**	3
Elastic-Net Classifier (L2 / Binomial Deviance)	5	2	3	4	1	2	**1**	1	6	2	1	3	1	1	**1**	3
Light Gradient Boosted Trees Classifier with Early Stopping	2	5	5	2	8	3	**5**	3	4	5	5	5	6	8	**6**	6
eXtreme Gradient Boosted Trees Classifier	3	4	4	5	7	4	**3**	4	5	6	6	6	5	7	**4**	6
Vowpal Wabbit Classifier	7	1	1	1	3	7	**4**	6	1	1	7	1	3	3	**7**	8
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	6	7	6	7	4	6	**6**	5	2	4	4	7	7	6	**5**	1
TensorFlow Neural Network Classifier	1	6	7	6	6	8	**7**	7	3	7	3	8	8	5	**3**	2
RandomForest Classifier (Gini)	8	8	8	8	5	5	**8**	8	8	8	8	2	4	4	**8**	5
	Log Loss (cumulative rank)								AUC (cumulative rank)							
	BT7	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
Light Gradient Boosting on ElasticNet Predictions	4	2	1	1	1	1	**1**	1	6	3	2	2	2	2	**1**	1
Elastic-Net Classifier (L2 / Binomial Deviance)	5	3	2	3	2	2	**2**	2	6	3	2	2	2	2	**1**	1
Light Gradient Boosted Trees Classifier with Early Stopping	2	4	3	4	4	3	**3**	3	4	6	6	5	4	6	**7**	7
eXtreme Gradient Boosted Trees Classifier	3	5	4	5	5	4	**4**	4	5	7	7	7	5	7	**6**	6
Vowpal Wabbit Classifier	7	6	5	2	3	5	**5**	5	1	1	1	1	1	1	**3**	3
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	6	7	7	7	7	6	**6**	6	2	2	4	4	6	4	**5**	5
TensorFlow Neural Network Classifier	1	1	6	6	6	7	**7**	7	3	5	5	6	7	5	**4**	4
RandomForest Classifier (Gini)	8	8	8	8	8	8	**8**	8	8	8	8	8	8	8	**8**	8
	Log Loss (cumulative average)								AUC (cumulative average)							
	BT7	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
Light Gradient Boosting on ElasticNet Predictions	0.47052	0.49537	0.50772	0.53204	0.50912	0.51448	0.50981	0.51927	0.54401	0.59803	0.66282	0.62524	0.65025	0.66746	0.68530	0.68832
Elastic-Net Classifier (L2 / Binomial Deviance)	0.48023	0.49955	0.51301	0.53612	0.51187	0.51693	0.51184	0.51974	0.54401	0.59803	0.66282	0.62524	0.65025	0.66746	0.68530	0.68832
Light Gradient Boosted Trees Classifier with Early Stopping	0.46774	0.50128	0.51797	0.53841	0.52401	0.53202	0.52891	0.53659	0.57504	0.59379	0.63786	0.60622	0.61409	0.61580	0.63856	0.64577
eXtreme Gradient Boosted Trees Classifier	0.46950	0.50195	0.51824	0.54086	0.52590	0.53382	0.53012	0.53771	0.57215	0.59114	0.63516	0.60293	0.61300	0.61525	0.63888	0.64605
Vowpal Wabbit Classifier	0.55407	0.51528	0.52089	0.53540	0.51536	0.54907	0.54347	0.55256	0.60895	0.67918	0.68920	0.65723	0.66780	0.67309	0.68132	0.68035
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.49131	0.53288	0.54146	0.58466	0.55820	0.56751	0.55942	0.56358	0.59307	0.60353	0.64751	0.60930	0.61241	0.61641	0.63968	0.64879
TensorFlow Neural Network Classifier	0.41682	0.48610	0.53665	0.57312	0.55136	0.60647	0.59486	0.60721	0.58874	0.59510	0.64208	0.60478	0.60784	0.61619	0.64042	0.64912
RandomForest Classifier (Gini)	0.94009	0.90518	1.03098	1.35632	1.17756	1.08054	1.00221	0.99483	0.50216	0.53084	0.56995	0.56005	0.58449	0.59792	0.61188	0.62402

Appendix Table VIII: Log Loss and AUC (Values and rank)



RANDOM SAMPLE TESTS	Log Loss					AUC				
Model type	R1	R2	R3	R4	R5	R1	R2	R3	R4	R5
Light Gradient Boosting on ElasticNet Predictions	0.5979	0.5768	0.5856	0.5988	0.5868	0.4925	0.5459	0.5111	0.5219	0.4734
Elastic-Net Classifier (L2 / Binomial Deviance)	0.5903	0.5760	0.5801	0.6000	0.5861	0.4925	0.5459	0.5111	0.5219	0.4734
Light Gradient Boosted Trees Classifier with Early Stopping	0.5909	0.5810	0.5925	0.6003	0.5917	0.4935	0.5282	0.5028	0.5081	0.4742
eXtreme Gradient Boosted Trees Classifier	0.5922	0.5943	0.5885	0.5992	0.5907	0.5105	0.4905	0.4963	0.4927	0.4802
Vowpal Wabbit Classifier	0.6267	0.5950	0.6069	0.6480	0.6451	0.5075	0.5403	0.5322	0.4954	0.4953
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.6538	0.6143	0.6326	0.6718	0.6601	0.4882	0.5425	0.5069	0.4782	0.5015
TensorFlow Neural Network Classifier	0.6215	0.5867	0.5770	0.6077	0.5957	0.4922	0.4584	0.5308	0.4787	0.5081
RandomForest Classifier (Gini)	0.6895	0.6948	0.6814	0.6618	0.6860	0.4997	0.4930	0.5457	0.4999	0.4989

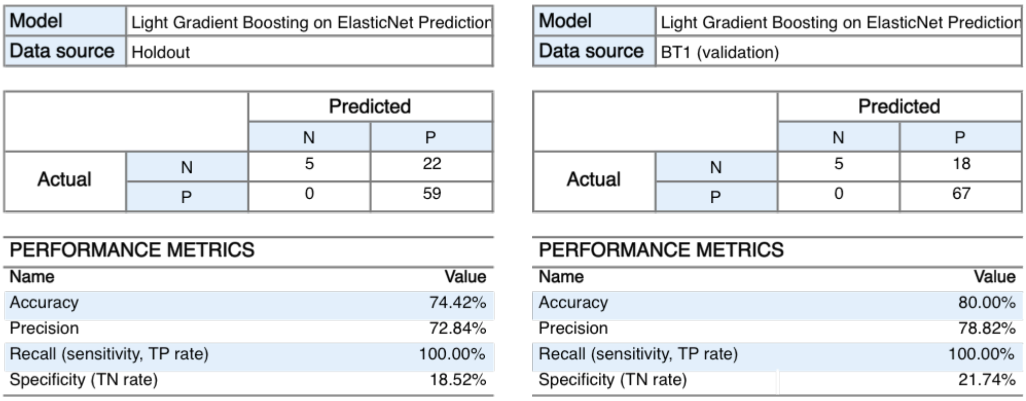
Appendix Table IX: Random sample test



HOLDOUT PERFORMANCE (ACCURACY)			
Model type	MCD*	Accuracy	Difference
Light Gradient Boosting on ElasticNet Predictions	68.60%	74.42%	5.82%
Elastic-Net Classifier (L2 / Binomial Deviance)	68.60%	74.42%	5.82%
Light Gradient Boosted Trees Classifier with Early Stopping	68.60%	73.26%	4.66%
eXtreme Gradient Boosted Trees Classifier	68.60%	73.26%	4.66%
Vowpal Wabbit Classifier	68.60%	75.58%	6.98%
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	68.60%	73.26%	4.66%
TensorFlow Neural Network Classifier	68.60%	74.42%	5.82%
RandomForest Classifier (Gini)	68.60%	73.26%	4.66%

*) Majority class distribution

Appendix Table X: Holdout performance



Model	Light Gradient Boosting on ElasticNet Predictior
Data source	Holdout

		Predicted	
		N	P
Actual	N	5	22
	P	0	59

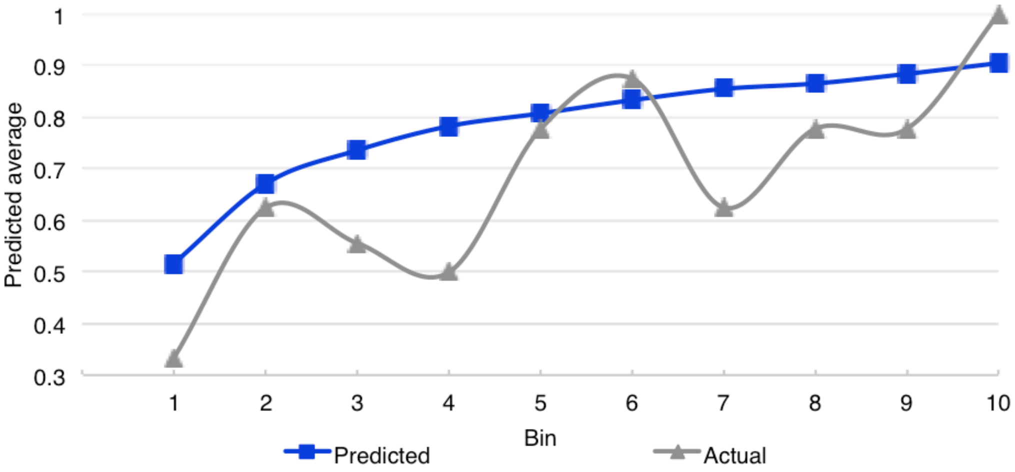
PERFORMANCE METRICS	
Name	Value
Accuracy	74.42%
Precision	72.84%
Recall (sensitivity, TP rate)	100.00%
Specificity (TN rate)	18.52%

Model	Light Gradient Boosting on ElasticNet Predictior
Data source	BT1 (validation)

		Predicted	
		N	P
Actual	N	5	18
	P	0	67

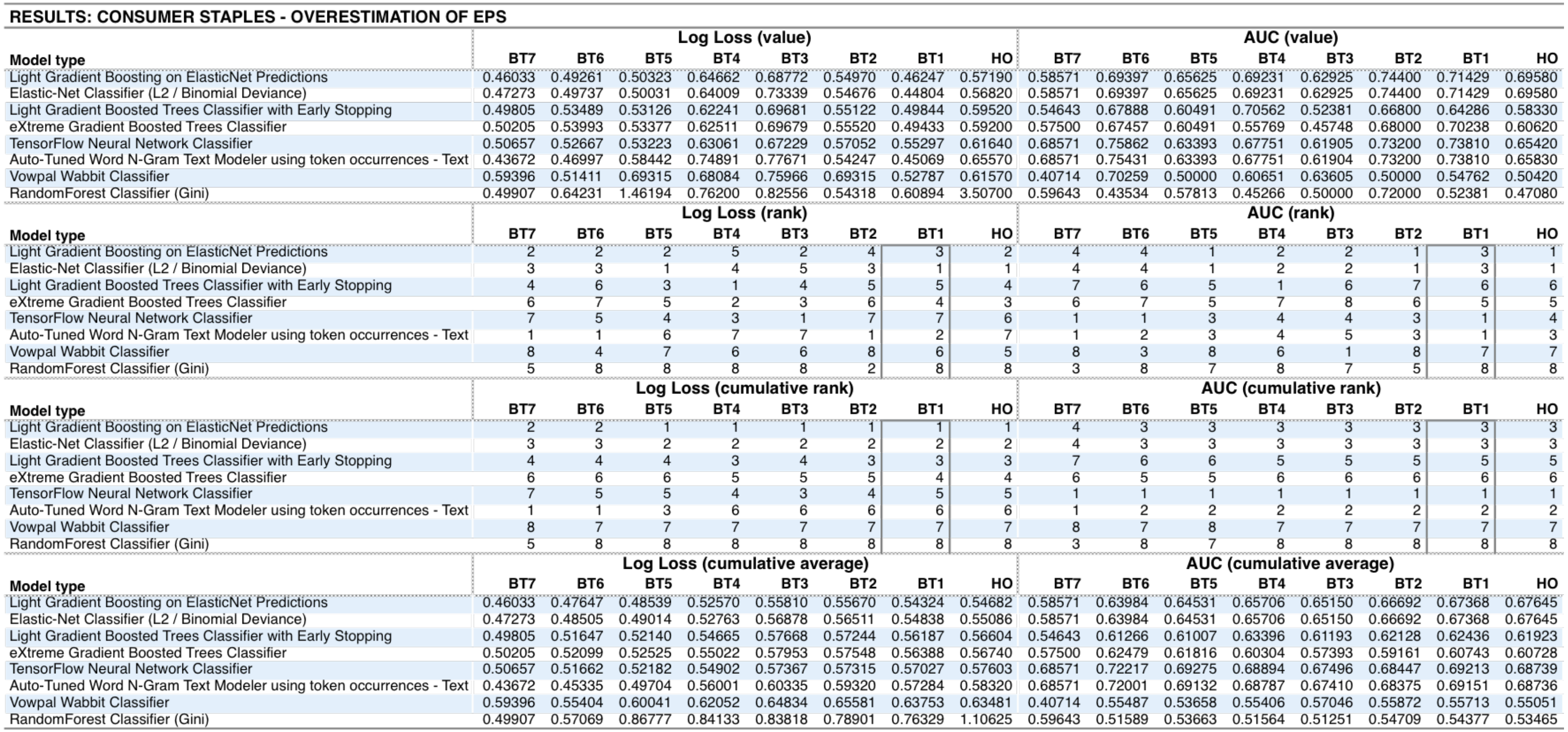
PERFORMANCE METRICS	
Name	Value
Accuracy	80.00%
Precision	78.82%
Recall (sensitivity, TP rate)	100.00%
Specificity (TN rate)	21.74%

Appendix Table XI: Confusion matrices – Holdout and validation



Appendix Table XII: Lift chart

Appendix XV. Consumer Staples (Overestimation of EPS)



RESULTS: CONSUMER STAPLES - OVERESTIMATION OF EPS

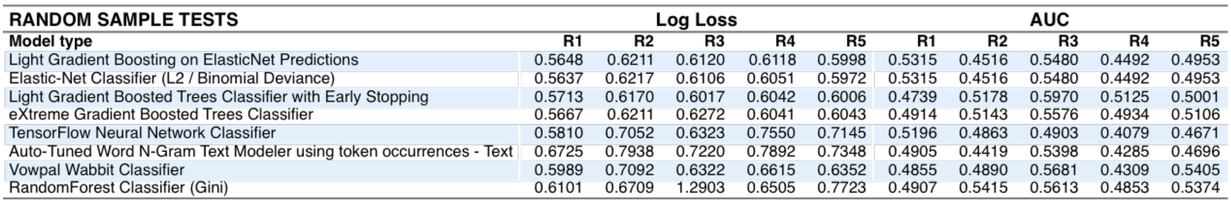
	Log Loss (value)								AUC (value)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
Light Gradient Boosting on ElasticNet Predictions	0.46033	0.49261	0.50323	0.64662	0.68772	0.54970	0.46247	0.57190	0.58571	0.69397	0.65625	0.69231	0.62925	0.74400	0.71429	0.69580
Elastic-Net Classifier (L2 / Binomial Deviance)	0.47273	0.49737	0.50031	0.64009	0.73339	0.54676	0.44804	0.56820	0.58571	0.69397	0.65625	0.69231	0.62925	0.74400	0.71429	0.69580
Light Gradient Boosted Trees Classifier with Early Stopping	0.49805	0.53489	0.53126	0.62241	0.69681	0.55122	0.49844	0.59520	0.54643	0.67888	0.60491	0.70562	0.52381	0.66800	0.64286	0.58330
eXtreme Gradient Boosted Trees Classifier	0.50205	0.53993	0.53377	0.62511	0.69679	0.55520	0.49433	0.59200	0.57500	0.67457	0.60491	0.55769	0.45748	0.68000	0.70238	0.60620
TensorFlow Neural Network Classifier	0.50657	0.52667	0.53223	0.63061	0.67229	0.57052	0.55297	0.61640	0.68571	0.75862	0.63393	0.67751	0.61905	0.73200	0.73810	0.65420
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.43672	0.46997	0.58442	0.74891	0.77671	0.54247	0.45069	0.65570	0.68571	0.75431	0.63393	0.67751	0.61904	0.73200	0.73810	0.65830
Vowpal Wabbit Classifier	0.59396	0.51411	0.69315	0.68084	0.75966	0.69315	0.52787	0.61570	0.40714	0.70259	0.50000	0.60651	0.63605	0.50000	0.54762	0.50420
RandomForest Classifier (Gini)	0.49907	0.64231	1.46194	0.76200	0.82556	0.54318	0.60894	3.50700	0.59643	0.43534	0.57813	0.45266	0.50000	0.72000	0.52381	0.47080

	Log Loss (rank)								AUC (rank)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
Light Gradient Boosting on ElasticNet Predictions	2	2	2	5	2	4	3	2	4	4	1	2	2	1	3	1
Elastic-Net Classifier (L2 / Binomial Deviance)	3	3	1	4	5	3	1	1	4	4	1	2	2	1	3	1
Light Gradient Boosted Trees Classifier with Early Stopping	4	6	3	1	4	5	5	4	7	6	5	1	6	7	6	6
eXtreme Gradient Boosted Trees Classifier	6	7	5	2	3	6	4	3	6	7	5	7	8	6	5	5
TensorFlow Neural Network Classifier	7	5	4	3	1	7	7	6	1	1	3	4	4	3	1	4
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	1	1	6	7	7	1	2	7	1	2	3	4	5	3	1	3
Vowpal Wabbit Classifier	8	4	7	6	6	8	6	5	8	3	8	6	1	8	7	7
RandomForest Classifier (Gini)	5	8	8	8	8	2	8	8	3	8	7	8	7	5	8	8

	Log Loss (cumulative rank)								AUC (cumulative rank)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
Light Gradient Boosting on ElasticNet Predictions	2	2	1	1	1	1	1	1	4	3	3	3	3	3	3	3
Elastic-Net Classifier (L2 / Binomial Deviance)	3	3	2	2	2	2	2	2	4	3	3	3	3	3	3	3
Light Gradient Boosted Trees Classifier with Early Stopping	4	4	4	3	4	3	3	3	7	6	6	5	5	5	5	5
eXtreme Gradient Boosted Trees Classifier	6	6	6	5	5	5	4	4	6	5	5	6	6	6	6	6
TensorFlow Neural Network Classifier	7	5	5	4	3	4	5	5	1	1	1	1	1	1	1	1
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	1	1	3	6	6	6	6	6	1	2	2	2	2	2	2	2
Vowpal Wabbit Classifier	8	7	7	7	7	7	7	7	8	7	8	7	7	7	7	7
RandomForest Classifier (Gini)	5	8	8	8	8	8	8	8	3	8	7	8	8	8	8	8

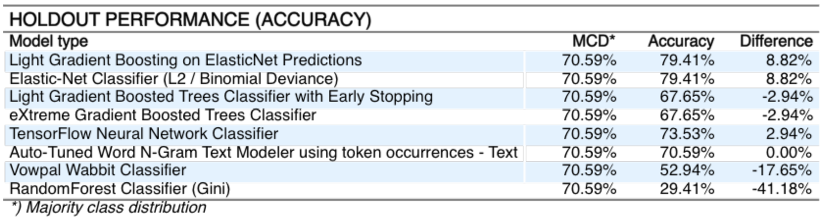
	Log Loss (cumulative average)								AUC (cumulative average)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
Light Gradient Boosting on ElasticNet Predictions	0.46033	0.47647	0.48539	0.52570	0.55810	0.55670	0.54324	0.54682	0.58571	0.63984	0.64531	0.65706	0.65150	0.66692	0.67368	0.67645
Elastic-Net Classifier (L2 / Binomial Deviance)	0.47273	0.48505	0.49014	0.52763	0.56878	0.56511	0.54838	0.55086	0.58571	0.63984	0.64531	0.65706	0.65150	0.66692	0.67368	0.67645
Light Gradient Boosted Trees Classifier with Early Stopping	0.49805	0.51647	0.52140	0.54665	0.57668	0.57244	0.56187	0.56604	0.54643	0.61266	0.61007	0.63396	0.61193	0.62128	0.62436	0.61923
eXtreme Gradient Boosted Trees Classifier	0.50205	0.52099	0.52525	0.55022	0.57953	0.57548	0.56388	0.56740	0.57500	0.62479	0.61816	0.60304	0.57393	0.59161	0.60743	0.60728
TensorFlow Neural Network Classifier	0.50657	0.51662	0.52182	0.54902	0.57367	0.57315	0.57027	0.57603	0.68571	0.72217	0.69275	0.68894	0.67496	0.68447	0.69213	0.68739
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.43672	0.45335	0.49704	0.56001	0.60335	0.59320	0.57284	0.58320	0.68571	0.72001	0.69132	0.68787	0.67410	0.68375	0.69151	0.68736
Vowpal Wabbit Classifier	0.59396	0.55404	0.60041	0.62052	0.64834	0.65581	0.63753	0.63481	0.40714	0.55487	0.53658	0.55406	0.57046	0.55872	0.55713	0.55051
RandomForest Classifier (Gini)	0.49907	0.57069	0.86777	0.84133	0.83818	0.78901	0.76329	1.10625	0.59643	0.51589	0.53663	0.51564	0.51251	0.54709	0.54377	0.53465

Appendix Table XIII: Log Loss and AUC (Values and rank)



RANDOM SAMPLE TESTS	Log Loss					AUC				
Model type	**R1**	**R2**	**R3**	**R4**	**R5**	**R1**	**R2**	**R3**	**R4**	**R5**
Light Gradient Boosting on ElasticNet Predictions	0.5648	0.6211	0.6120	0.6118	0.5998	0.5315	0.4516	0.5480	0.4492	0.4953
Elastic-Net Classifier (L2 / Binomial Deviance)	0.5637	0.6217	0.6106	0.6051	0.5972	0.5315	0.4516	0.5480	0.4492	0.4953
Light Gradient Boosted Trees Classifier with Early Stopping	0.5713	0.6170	0.6017	0.6042	0.6006	0.4739	0.5178	0.5970	0.5125	0.5001
eXtreme Gradient Boosted Trees Classifier	0.5667	0.6211	0.6272	0.6041	0.6043	0.4914	0.5143	0.5576	0.4934	0.5106
TensorFlow Neural Network Classifier	0.5810	0.7052	0.6323	0.7550	0.7145	0.5196	0.4863	0.4903	0.4079	0.4671
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.6725	0.7938	0.7220	0.7892	0.7348	0.4905	0.4419	0.5398	0.4285	0.4696
Vowpal Wabbit Classifier	0.5989	0.7092	0.6322	0.6615	0.6352	0.4855	0.4890	0.5681	0.4309	0.5405
RandomForest Classifier (Gini)	0.6101	0.6709	1.2903	0.6505	0.7723	0.4907	0.5415	0.5613	0.4853	0.5374

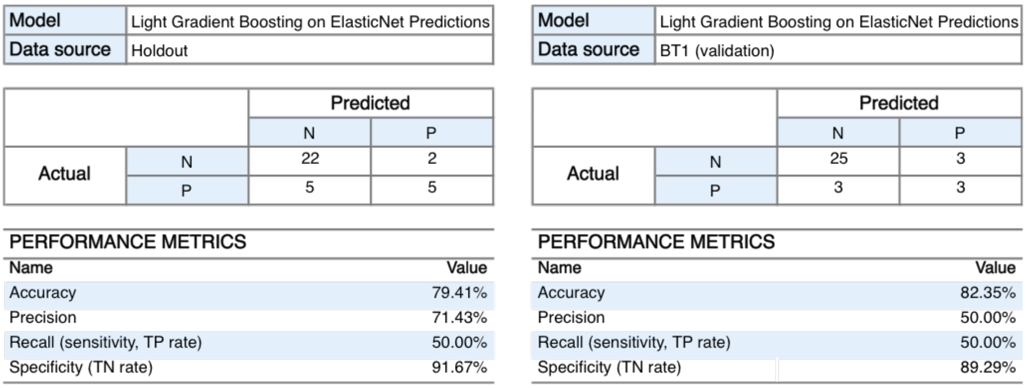
Appendix Table XIV: Random sample test



HOLDOUT PERFORMANCE (ACCURACY)			
Model type	MCD*	Accuracy	Difference
Light Gradient Boosting on ElasticNet Predictions	70.59%	79.41%	8.82%
Elastic-Net Classifier (L2 / Binomial Deviance)	70.59%	79.41%	8.82%
Light Gradient Boosted Trees Classifier with Early Stopping	70.59%	67.65%	-2.94%
eXtreme Gradient Boosted Trees Classifier	70.59%	67.65%	-2.94%
TensorFlow Neural Network Classifier	70.59%	73.53%	2.94%
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	70.59%	70.59%	0.00%
Vowpal Wabbit Classifier	70.59%	52.94%	-17.65%
RandomForest Classifier (Gini)	70.59%	29.41%	-41.18%

*) Majority class distribution

Appendix Table XV: Holdout performance



Model	Light Gradient Boosting on ElasticNet Predictions
Data source	Holdout

		Predicted	
		N	P
Actual	N	22	2
	P	5	5

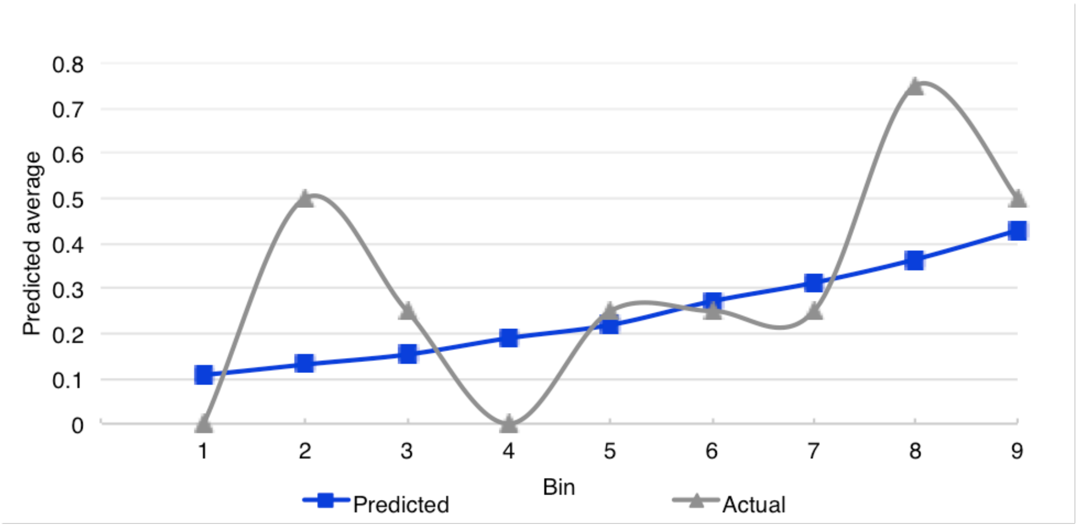
PERFORMANCE METRICS	
Name	Value
Accuracy	79.41%
Precision	71.43%
Recall (sensitivity, TP rate)	50.00%
Specificity (TN rate)	91.67%

Model	Light Gradient Boosting on ElasticNet Predictions
Data source	BT1 (validation)

		Predicted	
		N	P
Actual	N	25	3
	P	3	3

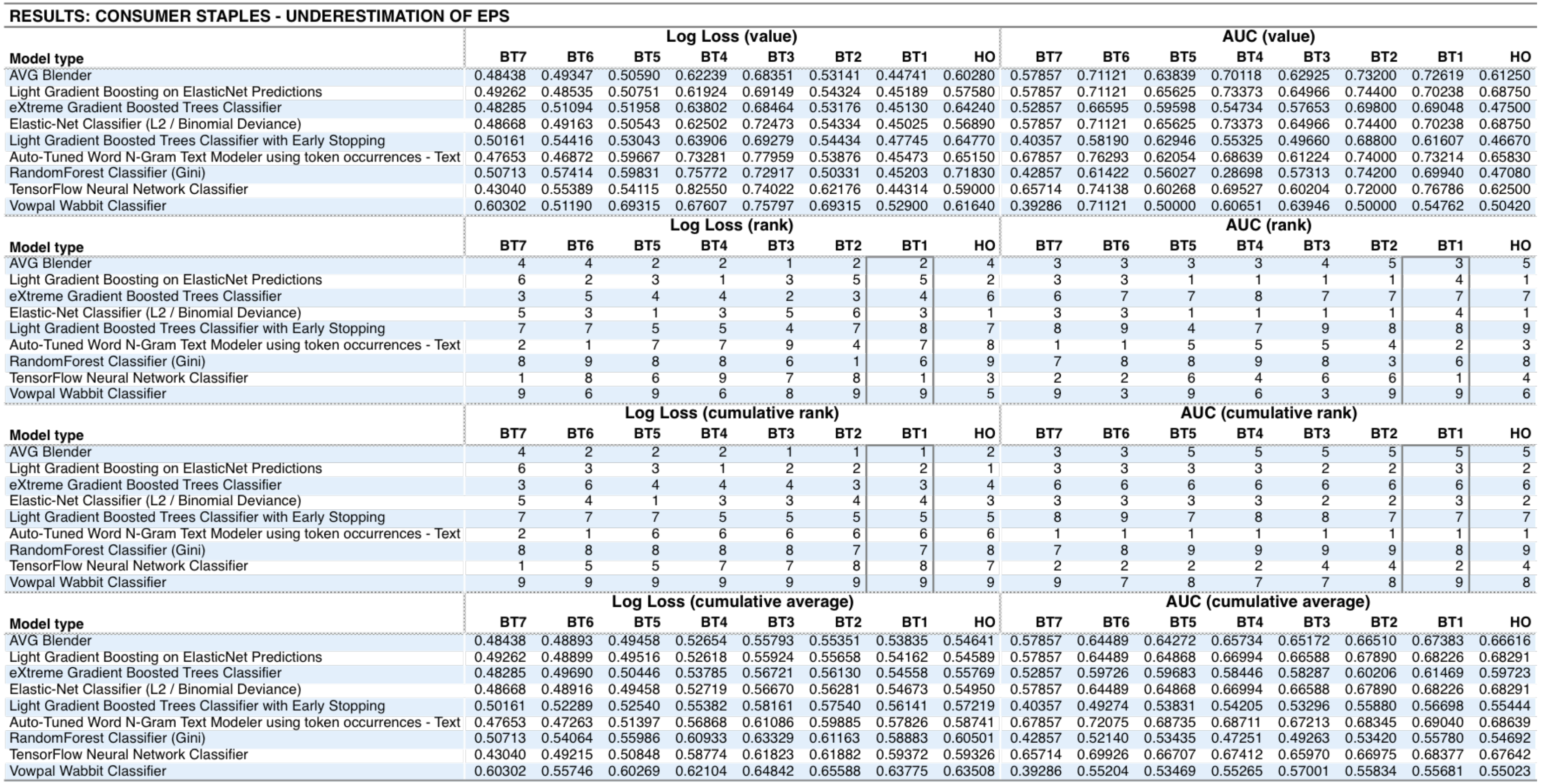
PERFORMANCE METRICS	
Name	Value
Accuracy	82.35%
Precision	50.00%
Recall (sensitivity, TP rate)	50.00%
Specificity (TN rate)	89.29%

Appendix Table XVI: Confusion matrices – Holdout and validation



Appendix Table XVII: Lift chart

Appendix XVI. Consumer Staples (Underestimation of EPS)



RESULTS: CONSUMER STAPLES - UNDERESTIMATION OF EPS

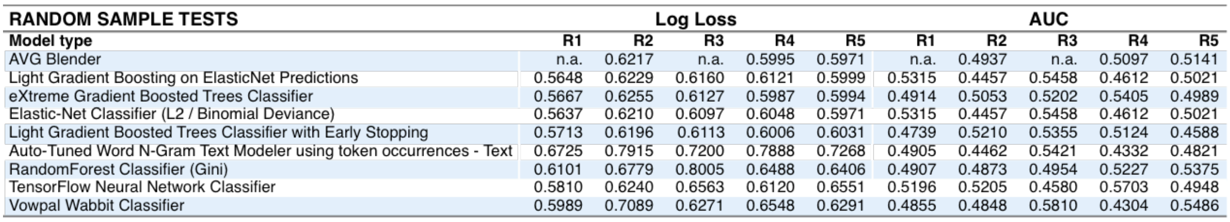
	Log Loss (value)								AUC (value)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
AVG Blender	0.48438	0.49347	0.50590	0.62239	0.68351	0.53141	0.44741	0.60280	0.57857	0.71121	0.63839	0.70118	0.62925	0.73200	0.72619	0.61250
Light Gradient Boosting on ElasticNet Predictions	0.49262	0.48535	0.50751	0.61924	0.69149	0.54324	0.45189	0.57580	0.57857	0.71121	0.65625	0.73373	0.64966	0.74400	0.70238	0.68750
eXtreme Gradient Boosted Trees Classifier	0.48285	0.51094	0.51958	0.63802	0.68464	0.53176	0.45130	0.64240	0.52857	0.66595	0.59598	0.54734	0.57653	0.69800	0.69048	0.47500
Elastic-Net Classifier (L2 / Binomial Deviance)	0.48668	0.49163	0.50543	0.62502	0.72473	0.54334	0.45025	0.56890	0.57857	0.71121	0.65625	0.73373	0.64966	0.74400	0.70238	0.68750
Light Gradient Boosted Trees Classifier with Early Stopping	0.50161	0.54416	0.53043	0.63906	0.69279	0.54434	0.47745	0.64770	0.40357	0.58190	0.62946	0.55325	0.49660	0.68800	0.61607	0.46670
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.47653	0.46872	0.59667	0.73281	0.77959	0.53876	0.45473	0.65150	0.67857	0.76293	0.62054	0.68639	0.61224	0.74000	0.73214	0.65830
RandomForest Classifier (Gini)	0.50713	0.57414	0.59831	0.75772	0.72917	0.50331	0.45203	0.71830	0.42857	0.61422	0.56027	0.28698	0.57313	0.74200	0.69940	0.47080
TensorFlow Neural Network Classifier	0.43040	0.55389	0.54115	0.82550	0.74022	0.62176	0.44314	0.59000	0.65714	0.74138	0.60268	0.69527	0.60204	0.72000	0.76786	0.62500
Vowpal Wabbit Classifier	0.60302	0.51190	0.69315	0.67607	0.75797	0.69315	0.52900	0.61640	0.39286	0.71121	0.50000	0.60651	0.63946	0.50000	0.54762	0.50420

	Log Loss (rank)								AUC (rank)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
AVG Blender	4	4	2	2	1	2	2	4	3	3	3	3	4	5	3	5
Light Gradient Boosting on ElasticNet Predictions	6	2	3	1	3	5	5	2	3	3	1	1	1	1	4	1
eXtreme Gradient Boosted Trees Classifier	3	5	4	4	2	3	4	6	6	7	7	8	7	7	7	7
Elastic-Net Classifier (L2 / Binomial Deviance)	5	3	1	3	5	6	3	1	3	3	1	1	1	1	4	1
Light Gradient Boosted Trees Classifier with Early Stopping	7	7	5	5	4	7	8	7	8	9	4	7	9	8	8	9
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	2	1	7	7	9	4	7	8	1	1	5	5	5	4	2	3
RandomForest Classifier (Gini)	8	9	8	8	6	1	6	9	7	8	8	9	8	3	6	8
TensorFlow Neural Network Classifier	1	8	6	9	7	8	1	3	2	2	6	4	6	6	1	4
Vowpal Wabbit Classifier	9	6	9	6	8	9	9	5	9	3	9	6	3	9	9	6

	Log Loss (cumulative rank)								AUC (cumulative rank)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
AVG Blender	4	2	2	2	1	1	1	2	3	3	5	5	5	5	5	5
Light Gradient Boosting on ElasticNet Predictions	6	3	3	1	2	2	2	1	3	3	3	3	2	2	3	2
eXtreme Gradient Boosted Trees Classifier	3	6	4	4	4	3	3	4	6	6	6	6	6	6	6	6
Elastic-Net Classifier (L2 / Binomial Deviance)	5	4	1	3	3	4	4	3	3	3	3	3	2	2	3	2
Light Gradient Boosted Trees Classifier with Early Stopping	7	7	7	5	5	5	5	5	8	9	7	8	8	7	7	7
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	2	1	6	6	6	6	6	6	1	1	1	1	1	1	1	1
RandomForest Classifier (Gini)	8	8	8	8	8	7	7	8	7	8	9	9	9	9	8	9
TensorFlow Neural Network Classifier	1	5	5	7	7	8	8	7	2	2	2	2	4	4	2	4
Vowpal Wabbit Classifier	9	9	9	9	9	9	9	9	9	7	8	7	7	8	9	8

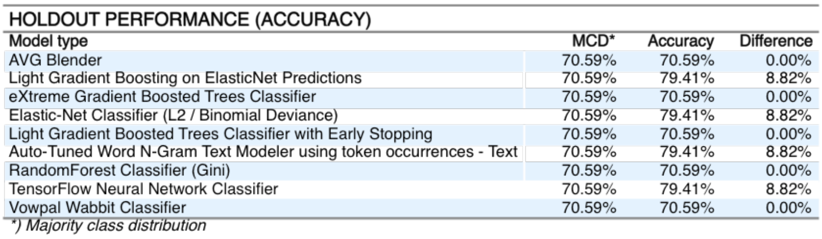
	Log Loss (cumulative average)								AUC (cumulative average)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
AVG Blender	0.48438	0.48893	0.49458	0.52654	0.55793	0.55351	0.53835	0.54641	0.57857	0.64489	0.64272	0.65734	0.65172	0.66510	0.67383	0.66616
Light Gradient Boosting on ElasticNet Predictions	0.49262	0.48899	0.49516	0.52618	0.55924	0.55658	0.54162	0.54589	0.57857	0.64489	0.64868	0.66994	0.66588	0.67890	0.68226	0.68291
eXtreme Gradient Boosted Trees Classifier	0.48285	0.49690	0.50446	0.53785	0.56721	0.56130	0.54558	0.55769	0.52857	0.59726	0.59683	0.58446	0.58287	0.60206	0.61469	0.59723
Elastic-Net Classifier (L2 / Binomial Deviance)	0.48668	0.48916	0.49458	0.52719	0.56670	0.56281	0.54673	0.54950	0.57857	0.64489	0.64868	0.66994	0.66588	0.67890	0.68226	0.68291
Light Gradient Boosted Trees Classifier with Early Stopping	0.50161	0.52289	0.52540	0.55382	0.58161	0.57540	0.56141	0.57219	0.40357	0.49274	0.53831	0.54205	0.53296	0.55880	0.56698	0.55444
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.47653	0.47263	0.51397	0.56868	0.61086	0.59885	0.57826	0.58741	0.67857	0.72075	0.68735	0.68711	0.67213	0.68345	0.69040	0.68639
RandomForest Classifier (Gini)	0.50713	0.54064	0.55986	0.60933	0.63329	0.61163	0.58883	0.60501	0.42857	0.52140	0.53435	0.47251	0.49263	0.53420	0.55780	0.54692
TensorFlow Neural Network Classifier	0.43040	0.49215	0.50848	0.58774	0.61823	0.61882	0.59372	0.59326	0.65714	0.69926	0.66707	0.67412	0.65970	0.66975	0.68377	0.67642
Vowpal Wabbit Classifier	0.60302	0.55746	0.60269	0.62104	0.64842	0.65588	0.63775	0.63508	0.39286	0.55204	0.53469	0.55265	0.57001	0.55834	0.55681	0.55023

Appendix Table XVIII: Log Loss and AUC (Values and rank)



RANDOM SAMPLE TESTS	Log Loss					AUC				
Model type	R1	R2	R3	R4	R5	R1	R2	R3	R4	R5
AVG Blender	n.a.	0.6217	n.a.	0.5995	0.5971	n.a.	0.4937	n.a.	0.5097	0.5141
Light Gradient Boosting on ElasticNet Predictions	0.5648	0.6229	0.6160	0.6121	0.5999	0.5315	0.4457	0.5458	0.4612	0.5021
eXtreme Gradient Boosted Trees Classifier	0.5667	0.6255	0.6127	0.5987	0.5994	0.4914	0.5053	0.5202	0.5405	0.4989
Elastic-Net Classifier (L2 / Binomial Deviance)	0.5637	0.6210	0.6097	0.6048	0.5971	0.5315	0.4457	0.5458	0.4612	0.5021
Light Gradient Boosted Trees Classifier with Early Stopping	0.5713	0.6196	0.6113	0.6006	0.6031	0.4739	0.5210	0.5355	0.5124	0.4588
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.6725	0.7915	0.7200	0.7888	0.7268	0.4905	0.4462	0.5421	0.4332	0.4821
RandomForest Classifier (Gini)	0.6101	0.6779	0.8005	0.6488	0.6406	0.4907	0.4873	0.4954	0.5227	0.5375
TensorFlow Neural Network Classifier	0.5810	0.6240	0.6563	0.6120	0.6551	0.5196	0.5205	0.4580	0.5703	0.4948
Vowpal Wabbit Classifier	0.5989	0.7089	0.6271	0.6548	0.6291	0.4855	0.4848	0.5810	0.4304	0.5486

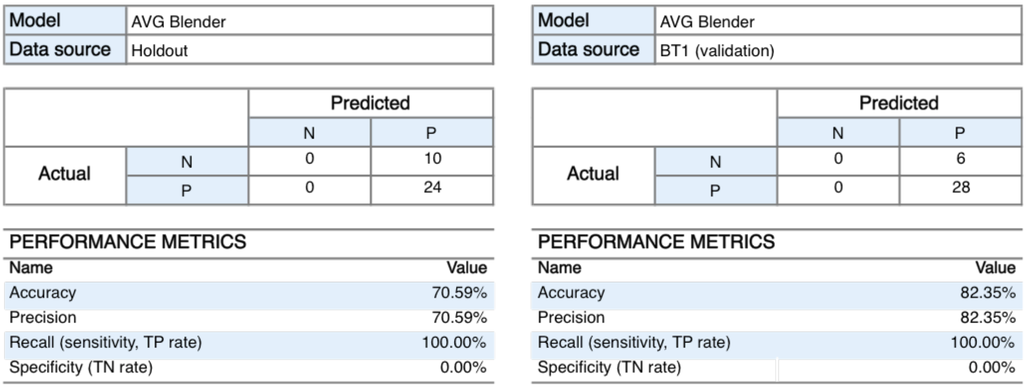
Appendix Table XIX: Random sample test



HOLDOUT PERFORMANCE (ACCURACY)			
Model type	MCD*	Accuracy	Difference
AVG Blender	70.59%	70.59%	0.00%
Light Gradient Boosting on ElasticNet Predictions	70.59%	79.41%	8.82%
eXtreme Gradient Boosted Trees Classifier	70.59%	70.59%	0.00%
Elastic-Net Classifier (L2 / Binomial Deviance)	70.59%	79.41%	8.82%
Light Gradient Boosted Trees Classifier with Early Stopping	70.59%	70.59%	0.00%
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	70.59%	79.41%	8.82%
RandomForest Classifier (Gini)	70.59%	70.59%	0.00%
TensorFlow Neural Network Classifier	70.59%	79.41%	8.82%
Vowpal Wabbit Classifier	70.59%	70.59%	0.00%

*) Majority class distribution

Appendix Table XX: Holdout performance



Model	AVG Blender
Data source	Holdout

		Predicted	
		N	P
Actual	N	0	10
	P	0	24

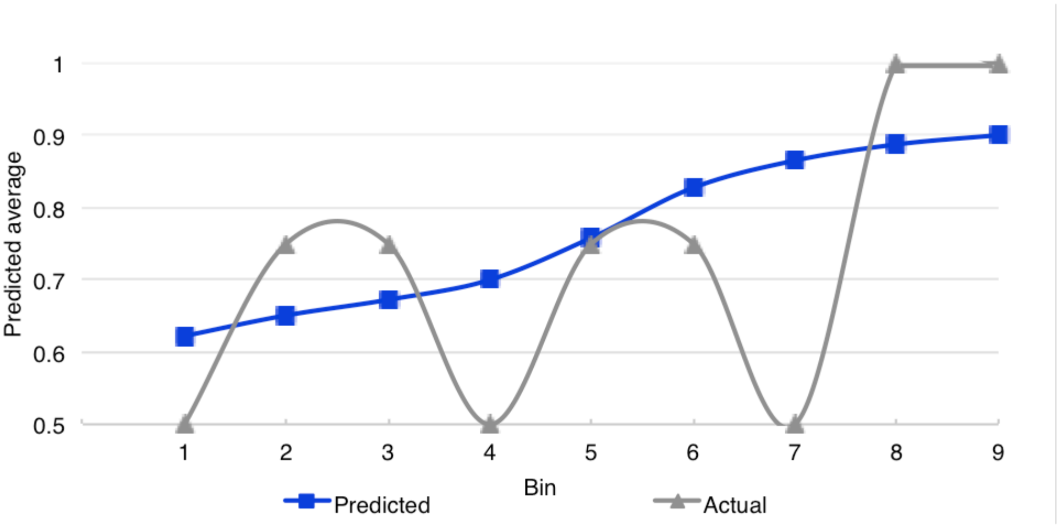
PERFORMANCE METRICS	
Name	Value
Accuracy	70.59%
Precision	70.59%
Recall (sensitivity, TP rate)	100.00%
Specificity (TN rate)	0.00%

Model	AVG Blender
Data source	BT1 (validation)

		Predicted	
		N	P
Actual	N	0	6
	P	0	28

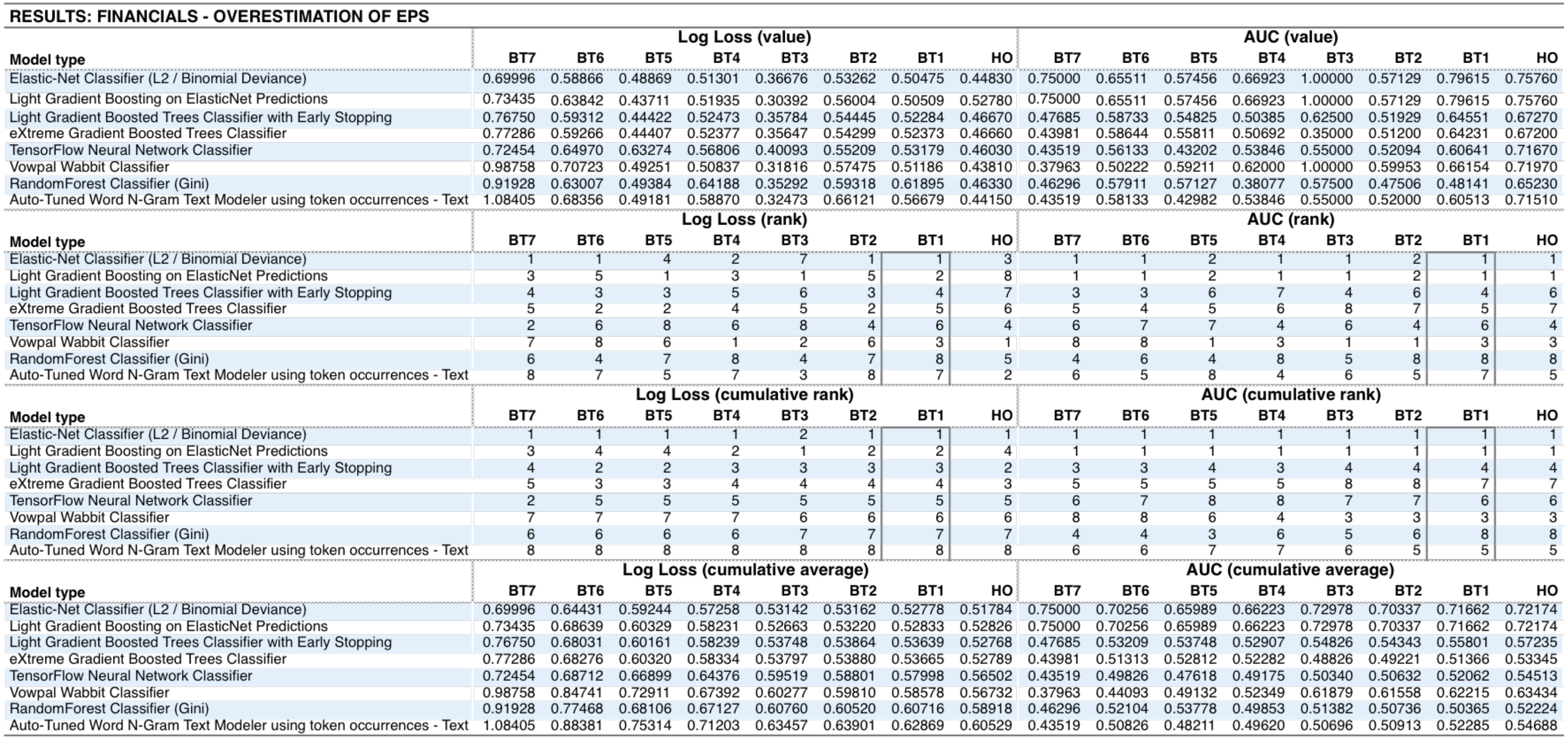
PERFORMANCE METRICS	
Name	Value
Accuracy	82.35%
Precision	82.35%
Recall (sensitivity, TP rate)	100.00%
Specificity (TN rate)	0.00%

Appendix Table XXI: Confusion matrices – Holdout and validation



Appendix Table XXII: Lift chart

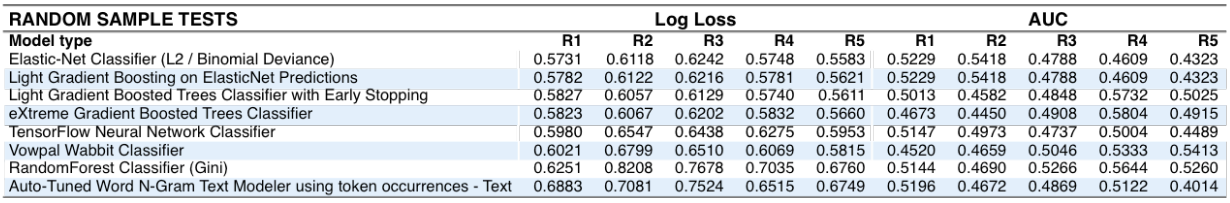
Appendix XVII. Financials (Overestimation of EPS)



RESULTS: FINANCIALS - OVERESTIMATION OF EPS

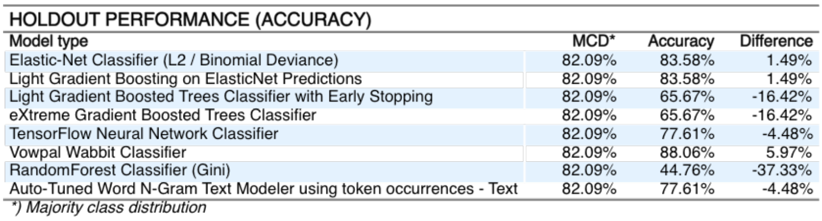
	Log Loss (value)								AUC (value)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
Elastic-Net Classifier (L2 / Binomial Deviance)	0.69996	0.58866	0.48869	0.51301	0.36676	0.53262	0.50475	0.44830	0.75000	0.65511	0.57456	0.66923	1.00000	0.57129	0.79615	0.75760
Light Gradient Boosting on ElasticNet Predictions	0.73435	0.63842	0.43711	0.51935	0.30392	0.56004	0.50509	0.52780	0.75000	0.65511	0.57456	0.66923	1.00000	0.57129	0.79615	0.75760
Light Gradient Boosted Trees Classifier with Early Stopping	0.76750	0.59312	0.44422	0.52473	0.35784	0.54445	0.52284	0.46670	0.47685	0.58733	0.54825	0.50385	0.62500	0.51929	0.64551	0.67270
eXtreme Gradient Boosted Trees Classifier	0.77286	0.59266	0.44407	0.52377	0.35647	0.54299	0.52373	0.46660	0.43981	0.58644	0.55811	0.50692	0.35000	0.51200	0.64231	0.67200
TensorFlow Neural Network Classifier	0.72454	0.64970	0.63274	0.56806	0.40093	0.55209	0.53179	0.46030	0.43519	0.56133	0.43202	0.53846	0.55000	0.52094	0.60641	0.71670
Vowpal Wabbit Classifier	0.98758	0.70723	0.49251	0.50837	0.31816	0.57475	0.51186	0.43810	0.37963	0.50222	0.59211	0.62000	1.00000	0.59953	0.66154	0.71970
RandomForest Classifier (Gini)	0.91928	0.63007	0.49384	0.64188	0.35292	0.59318	0.61895	0.46330	0.46296	0.57911	0.57127	0.38077	0.57500	0.47506	0.48141	0.65230
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	1.08405	0.68356	0.49181	0.58870	0.32473	0.66121	0.56679	0.44150	0.43519	0.58133	0.42982	0.53846	0.55000	0.52000	0.60513	0.71510
	Log Loss (rank)								**AUC (rank)**							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
Elastic-Net Classifier (L2 / Binomial Deviance)	1	1	4	2	7	1	1	3	1	1	2	1	1	2	1	1
Light Gradient Boosting on ElasticNet Predictions	3	5	1	3	1	5	2	8	1	1	2	1	1	2	1	1
Light Gradient Boosted Trees Classifier with Early Stopping	4	3	3	5	6	3	4	7	3	3	6	7	4	6	4	6
eXtreme Gradient Boosted Trees Classifier	5	2	2	4	5	2	5	6	5	4	5	6	8	7	5	7
TensorFlow Neural Network Classifier	2	6	8	6	8	4	6	4	6	7	7	4	6	4	6	4
Vowpal Wabbit Classifier	7	8	6	1	2	6	3	1	8	8	1	3	1	1	3	3
RandomForest Classifier (Gini)	6	4	7	8	4	7	8	5	4	6	4	8	5	8	8	8
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	8	7	5	7	3	8	7	2	6	5	8	4	6	5	7	5
	Log Loss (cumulative rank)								**AUC (cumulative rank)**							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
Elastic-Net Classifier (L2 / Binomial Deviance)	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1
Light Gradient Boosting on ElasticNet Predictions	3	4	4	2	1	2	2	4	1	1	1	1	1	1	1	1
Light Gradient Boosted Trees Classifier with Early Stopping	4	2	2	3	3	3	3	2	3	3	4	3	4	4	4	4
eXtreme Gradient Boosted Trees Classifier	5	3	3	4	4	4	4	3	5	5	5	5	8	8	7	7
TensorFlow Neural Network Classifier	2	5	5	5	5	5	5	5	6	7	8	8	7	7	6	6
Vowpal Wabbit Classifier	7	7	7	7	6	6	6	6	8	8	6	4	3	3	3	3
RandomForest Classifier (Gini)	6	6	6	6	7	7	7	7	4	4	3	6	5	6	8	8
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	8	8	8	8	8	8	8	8	6	6	7	7	6	5	5	5
	Log Loss (cumulative average)								**AUC (cumulative average)**							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
Elastic-Net Classifier (L2 / Binomial Deviance)	0.69996	0.64431	0.59244	0.57258	0.53142	0.53162	0.52778	0.51784	0.75000	0.70256	0.65989	0.66223	0.72978	0.70337	0.71662	0.72174
Light Gradient Boosting on ElasticNet Predictions	0.73435	0.68639	0.60329	0.58231	0.52663	0.53220	0.52833	0.52826	0.75000	0.70256	0.65989	0.66223	0.72978	0.70337	0.71662	0.72174
Light Gradient Boosted Trees Classifier with Early Stopping	0.76750	0.68031	0.60161	0.58239	0.53748	0.53864	0.53639	0.52768	0.47685	0.53209	0.53748	0.52907	0.54826	0.54343	0.55801	0.57235
eXtreme Gradient Boosted Trees Classifier	0.77286	0.68276	0.60320	0.58334	0.53797	0.53880	0.53665	0.52789	0.43981	0.51313	0.52812	0.52282	0.48826	0.49221	0.51366	0.53345
TensorFlow Neural Network Classifier	0.72454	0.68712	0.66899	0.64376	0.59519	0.58801	0.57998	0.56502	0.43519	0.49826	0.47618	0.49175	0.50340	0.50632	0.52062	0.54513
Vowpal Wabbit Classifier	0.98758	0.84741	0.72911	0.67392	0.60277	0.59810	0.58578	0.56732	0.37963	0.44093	0.49132	0.52349	0.61879	0.61558	0.62215	0.63434
RandomForest Classifier (Gini)	0.91928	0.77468	0.68106	0.67127	0.60760	0.60520	0.60716	0.58918	0.46296	0.52104	0.53778	0.49853	0.51382	0.50736	0.50365	0.52224
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	1.08405	0.88381	0.75314	0.71203	0.63457	0.63901	0.62869	0.60529	0.43519	0.50826	0.48211	0.49620	0.50696	0.50913	0.52285	0.54688

Appendix Table XXIII: Log Loss and AUC (Values and rank)



RANDOM SAMPLE TESTS	Log Loss					AUC				
Model type	**R1**	**R2**	**R3**	**R4**	**R5**	**R1**	**R2**	**R3**	**R4**	**R5**
Elastic-Net Classifier (L2 / Binomial Deviance)	0.5731	0.6118	0.6242	0.5748	0.5583	0.5229	0.5418	0.4788	0.4609	0.4323
Light Gradient Boosting on ElasticNet Predictions	0.5782	0.6122	0.6216	0.5781	0.5621	0.5229	0.5418	0.4788	0.4609	0.4323
Light Gradient Boosted Trees Classifier with Early Stopping	0.5827	0.6057	0.6129	0.5740	0.5611	0.5013	0.4582	0.4848	0.5732	0.5025
eXtreme Gradient Boosted Trees Classifier	0.5823	0.6067	0.6202	0.5832	0.5660	0.4673	0.4450	0.4908	0.5804	0.4915
TensorFlow Neural Network Classifier	0.5980	0.6547	0.6438	0.6275	0.5953	0.5147	0.4973	0.4737	0.5004	0.4489
Vowpal Wabbit Classifier	0.6021	0.6799	0.6510	0.6069	0.5815	0.4520	0.4659	0.5046	0.5333	0.5413
RandomForest Classifier (Gini)	0.6251	0.8208	0.7678	0.7035	0.6760	0.5144	0.4690	0.5266	0.5644	0.5260
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.6883	0.7081	0.7524	0.6515	0.6749	0.5196	0.4672	0.4869	0.5122	0.4014

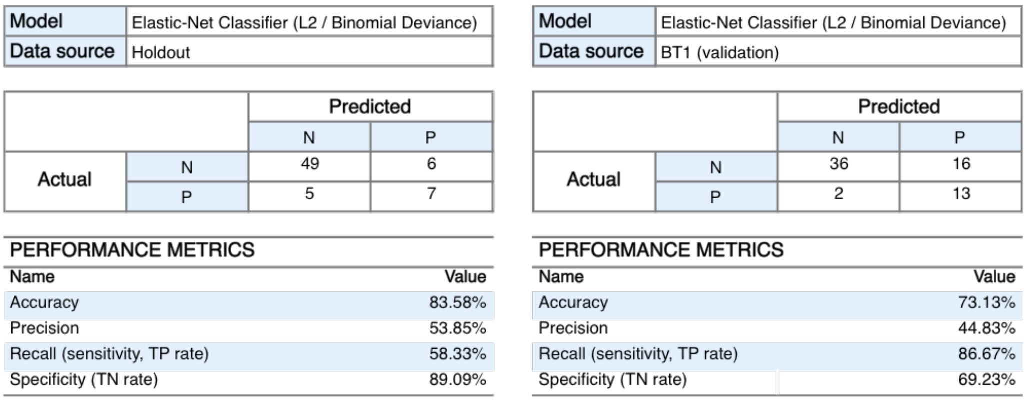
Appendix Table XXIV: Random sample test



HOLDOUT PERFORMANCE (ACCURACY)			
Model type	MCD*	Accuracy	Difference
Elastic-Net Classifier (L2 / Binomial Deviance)	82.09%	83.58%	1.49%
Light Gradient Boosting on ElasticNet Predictions	82.09%	83.58%	1.49%
Light Gradient Boosted Trees Classifier with Early Stopping	82.09%	65.67%	-16.42%
eXtreme Gradient Boosted Trees Classifier	82.09%	65.67%	-16.42%
TensorFlow Neural Network Classifier	82.09%	77.61%	-4.48%
Vowpal Wabbit Classifier	82.09%	88.06%	5.97%
RandomForest Classifier (Gini)	82.09%	44.76%	-37.33%
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	82.09%	77.61%	-4.48%

*) Majority class distribution

Appendix Table XXV: Holdout performance



Model	Elastic-Net Classifier (L2 / Binomial Deviance)
Data source	Holdout

		Predicted	
		N	P
Actual	N	49	6
	P	5	7

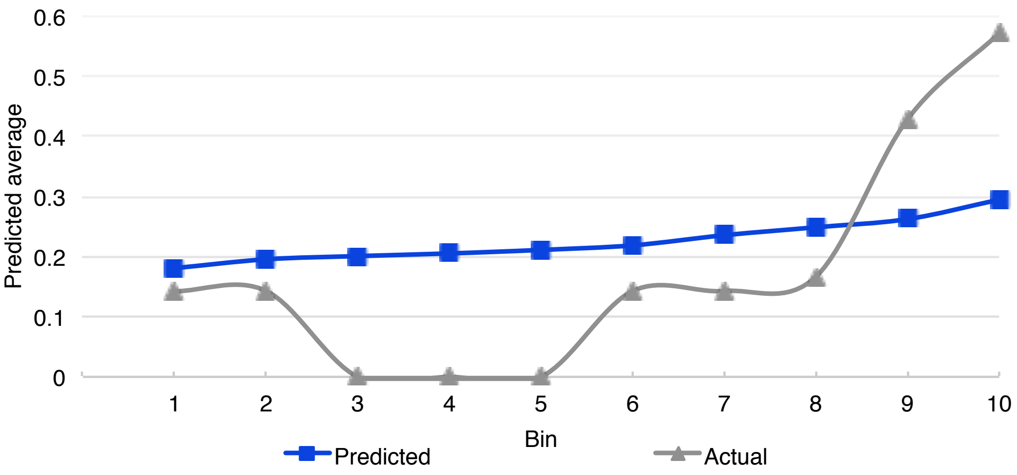
PERFORMANCE METRICS	
Name	Value
Accuracy	83.58%
Precision	53.85%
Recall (sensitivity, TP rate)	58.33%
Specificity (TN rate)	89.09%

Model	Elastic-Net Classifier (L2 / Binomial Deviance)
Data source	BT1 (validation)

		Predicted	
		N	P
Actual	N	36	16
	P	2	13

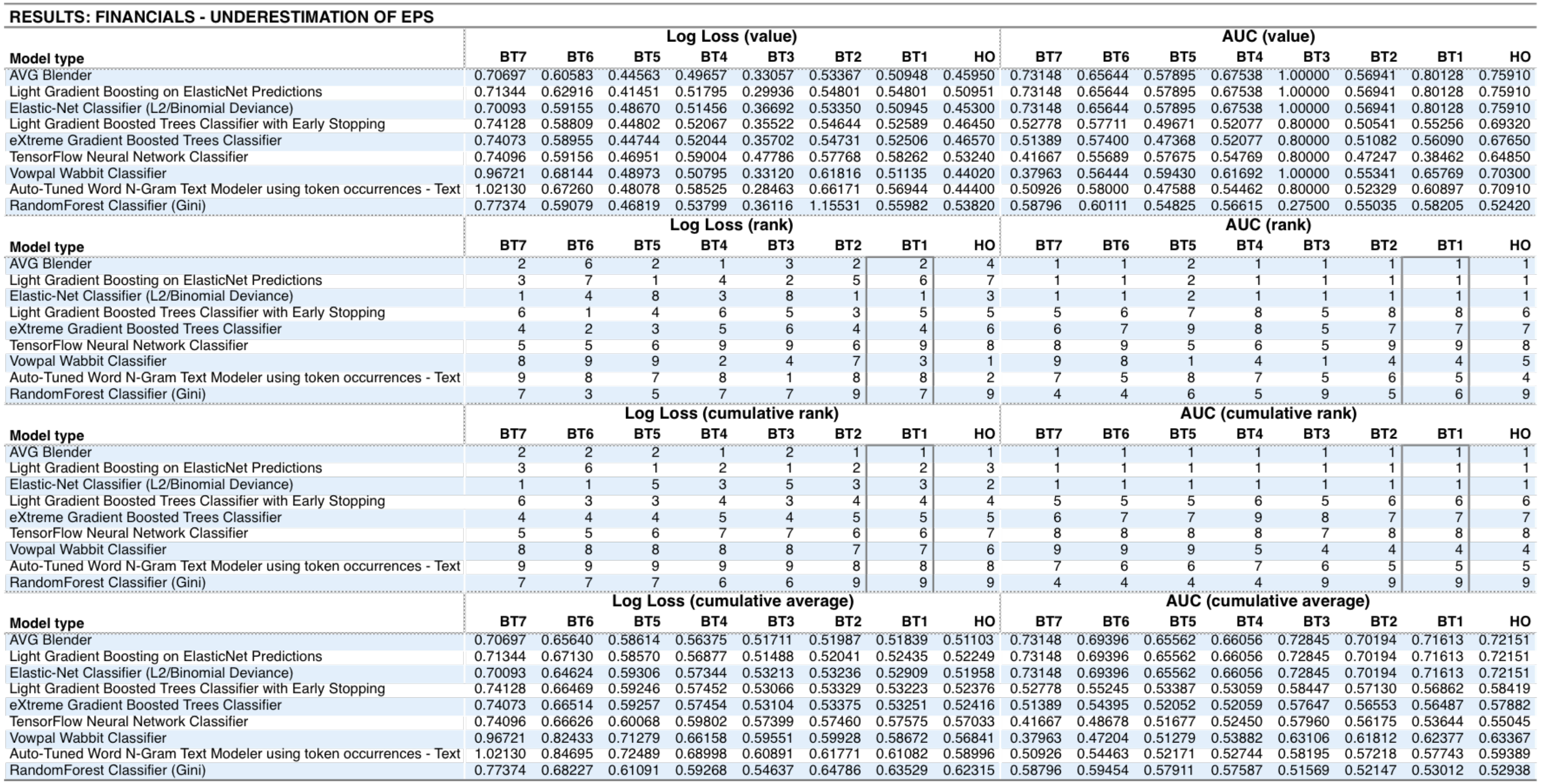
PERFORMANCE METRICS	
Name	Value
Accuracy	73.13%
Precision	44.83%
Recall (sensitivity, TP rate)	86.67%
Specificity (TN rate)	69.23%

Appendix Table XXVI: Confusion matrices – Holdout and validation



Appendix Table XXVII: Lift chart

Appendix XVIII. Financials (Underestimation of EPS)



RESULTS: FINANCIALS - UNDERESTIMATION OF EPS

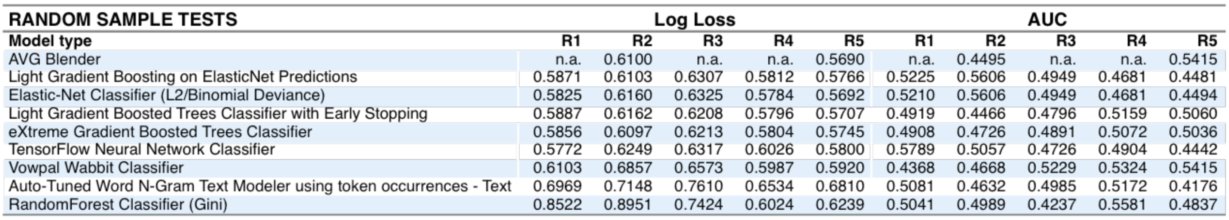
	Log Loss (value)								AUC (value)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
AVG Blender	0.70697	0.60583	0.44563	0.49657	0.33057	0.53367	0.50948	0.45950	0.73148	0.65644	0.57895	0.67538	1.00000	0.56941	0.80128	0.75910
Light Gradient Boosting on ElasticNet Predictions	0.71344	0.62916	0.41451	0.51795	0.29936	0.54801	0.54801	0.50951	0.73148	0.65644	0.57895	0.67538	1.00000	0.56941	0.80128	0.75910
Elastic-Net Classifier (L2/Binomial Deviance)	0.70093	0.59155	0.48670	0.51456	0.36692	0.53350	0.50945	0.45300	0.73148	0.65644	0.57895	0.67538	1.00000	0.56941	0.80128	0.75910
Light Gradient Boosted Trees Classifier with Early Stopping	0.74128	0.58809	0.44802	0.52067	0.35522	0.54644	0.52589	0.46450	0.52778	0.57711	0.49671	0.52077	0.80000	0.50541	0.55256	0.69320
eXtreme Gradient Boosted Trees Classifier	0.74073	0.58955	0.44744	0.52044	0.35702	0.54731	0.52506	0.46570	0.51389	0.57400	0.47368	0.52077	0.80000	0.51082	0.56090	0.67650
TensorFlow Neural Network Classifier	0.74096	0.59156	0.46951	0.59004	0.47786	0.57768	0.58262	0.53240	0.41667	0.55689	0.57675	0.54769	0.80000	0.47247	0.38462	0.64850
Vowpal Wabbit Classifier	0.96721	0.68144	0.48973	0.50795	0.33120	0.61816	0.51135	0.44020	0.37963	0.56444	0.59430	0.61692	1.00000	0.55341	0.65769	0.70300
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	1.02130	0.67260	0.48078	0.58525	0.28463	0.66171	0.56944	0.44400	0.50926	0.58000	0.47588	0.54462	0.80000	0.52329	0.60897	0.70910
RandomForest Classifier (Gini)	0.77374	0.59079	0.46819	0.53799	0.36116	1.15531	0.55982	0.53820	0.58796	0.60111	0.54825	0.56615	0.27500	0.55035	0.58205	0.52420

	Log Loss (rank)								AUC (rank)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
AVG Blender	2	6	2	1	3	2	2	4	1	1	2	1	1	1	1	1
Light Gradient Boosting on ElasticNet Predictions	3	7	1	4	2	5	6	7	1	1	2	1	1	1	1	1
Elastic-Net Classifier (L2/Binomial Deviance)	1	4	8	3	8	1	1	3	1	1	2	1	1	1	1	1
Light Gradient Boosted Trees Classifier with Early Stopping	6	1	4	6	5	3	5	5	5	6	7	8	5	8	8	6
eXtreme Gradient Boosted Trees Classifier	4	2	3	5	6	4	4	6	6	7	9	8	5	7	7	7
TensorFlow Neural Network Classifier	5	5	6	9	9	6	9	8	8	9	5	6	5	9	9	8
Vowpal Wabbit Classifier	8	9	9	2	4	7	3	1	9	8	1	4	1	4	4	5
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	9	8	7	8	1	8	8	2	7	5	8	7	5	6	5	4
RandomForest Classifier (Gini)	7	3	5	7	7	9	7	9	4	4	6	5	9	5	6	9

	Log Loss (cumulative rank)								AUC (cumulative rank)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
AVG Blender	2	2	2	1	2	1	1	1	1	1	1	1	1	1	1	1
Light Gradient Boosting on ElasticNet Predictions	3	6	1	2	1	2	2	3	1	1	1	1	1	1	1	1
Elastic-Net Classifier (L2/Binomial Deviance)	1	1	5	3	5	3	3	2	1	1	1	1	1	1	1	1
Light Gradient Boosted Trees Classifier with Early Stopping	6	3	3	4	3	4	4	4	5	5	5	6	5	6	6	6
eXtreme Gradient Boosted Trees Classifier	4	4	4	5	4	5	5	5	6	7	7	9	8	7	7	7
TensorFlow Neural Network Classifier	5	5	6	7	7	6	6	7	8	8	8	8	7	8	8	8
Vowpal Wabbit Classifier	8	8	8	8	8	7	7	6	9	9	9	5	4	4	4	4
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	9	9	9	9	9	8	8	8	7	6	6	7	6	5	5	5
RandomForest Classifier (Gini)	7	7	7	6	6	9	9	9	4	4	4	4	9	9	9	9

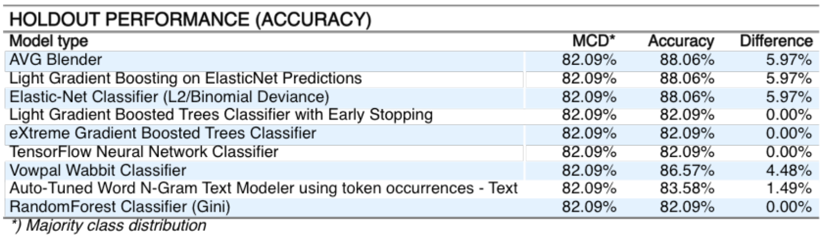
	Log Loss (cumulative average)								AUC (cumulative average)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
AVG Blender	0.70697	0.65640	0.58614	0.56375	0.51711	0.51987	0.51839	0.51103	0.73148	0.69396	0.65562	0.66056	0.72845	0.70194	0.71613	0.72151
Light Gradient Boosting on ElasticNet Predictions	0.71344	0.67130	0.58570	0.56877	0.51488	0.52041	0.52435	0.52249	0.73148	0.69396	0.65562	0.66056	0.72845	0.70194	0.71613	0.72151
Elastic-Net Classifier (L2/Binomial Deviance)	0.70093	0.64624	0.59306	0.57344	0.53213	0.53236	0.52909	0.51958	0.73148	0.69396	0.65562	0.66056	0.72845	0.70194	0.71613	0.72151
Light Gradient Boosted Trees Classifier with Early Stopping	0.74128	0.66469	0.59246	0.57452	0.53066	0.53329	0.53223	0.52376	0.52778	0.55245	0.53387	0.53059	0.58447	0.57130	0.56862	0.58419
eXtreme Gradient Boosted Trees Classifier	0.74073	0.66514	0.59257	0.57454	0.53104	0.53375	0.53251	0.52416	0.51389	0.54395	0.52052	0.52059	0.57647	0.56553	0.56487	0.57882
TensorFlow Neural Network Classifier	0.74096	0.66626	0.60068	0.59802	0.57399	0.57460	0.57575	0.57033	0.41667	0.48678	0.51677	0.52450	0.57960	0.56175	0.53644	0.55045
Vowpal Wabbit Classifier	0.96721	0.82433	0.71279	0.66158	0.59551	0.59928	0.58672	0.56841	0.37963	0.47204	0.51279	0.53882	0.63106	0.61812	0.62377	0.63367
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	1.02130	0.84695	0.72489	0.68998	0.60891	0.61771	0.61082	0.58996	0.50926	0.54463	0.52171	0.52744	0.58195	0.57218	0.57743	0.59389
RandomForest Classifier (Gini)	0.77374	0.68227	0.61091	0.59268	0.54637	0.64786	0.63529	0.62315	0.58796	0.59454	0.57911	0.57587	0.51569	0.52147	0.53012	0.52938

Appendix Table XXVIII: Log Loss and AUC (Values and rank)



RANDOM SAMPLE TESTS	Log Loss					AUC				
Model type	R1	R2	R3	R4	R5	R1	R2	R3	R4	R5
AVG Blender	n.a.	0.6100	n.a.	n.a.	0.5690	n.a.	0.4495	n.a.	n.a.	0.5415
Light Gradient Boosting on ElasticNet Predictions	0.5871	0.6103	0.6307	0.5812	0.5766	0.5225	0.5606	0.4949	0.4681	0.4481
Elastic-Net Classifier (L2/Binomial Deviance)	0.5825	0.6160	0.6325	0.5784	0.5692	0.5210	0.5606	0.4949	0.4681	0.4494
Light Gradient Boosted Trees Classifier with Early Stopping	0.5887	0.6162	0.6208	0.5796	0.5707	0.4919	0.4466	0.4796	0.5159	0.5060
eXtreme Gradient Boosted Trees Classifier	0.5856	0.6097	0.6213	0.5804	0.5745	0.4908	0.4726	0.4891	0.5072	0.5036
TensorFlow Neural Network Classifier	0.5772	0.6249	0.6317	0.6026	0.5800	0.5789	0.5057	0.4726	0.4904	0.4442
Vowpal Wabbit Classifier	0.6103	0.6857	0.6573	0.5987	0.5920	0.4368	0.4668	0.5229	0.5324	0.5415
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.6969	0.7148	0.7610	0.6534	0.6810	0.5081	0.4632	0.4985	0.5172	0.4176
RandomForest Classifier (Gini)	0.8522	0.8951	0.7424	0.6024	0.6239	0.5041	0.4989	0.4237	0.5581	0.4837

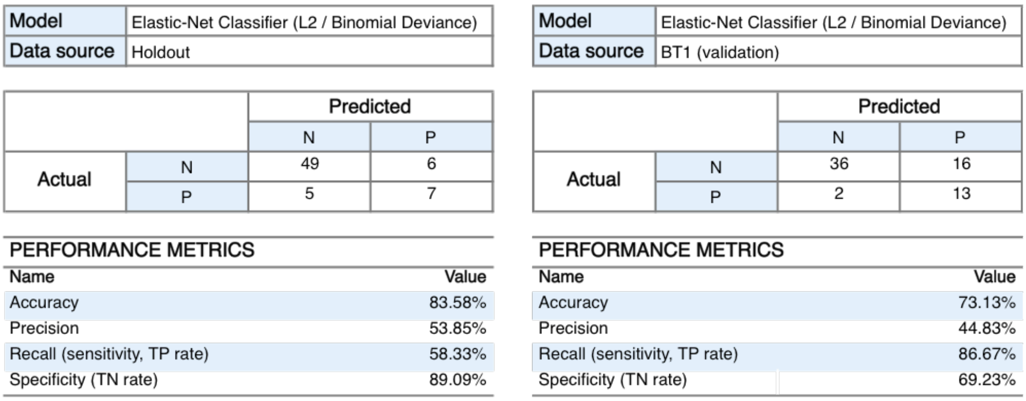
Appendix Table XXIX: Random sample test



HOLDOUT PERFORMANCE (ACCURACY)			
Model type	MCD*	Accuracy	Difference
AVG Blender	82.09%	88.06%	5.97%
Light Gradient Boosting on ElasticNet Predictions	82.09%	88.06%	5.97%
Elastic-Net Classifier (L2/Binomial Deviance)	82.09%	88.06%	5.97%
Light Gradient Boosted Trees Classifier with Early Stopping	82.09%	82.09%	0.00%
eXtreme Gradient Boosted Trees Classifier	82.09%	82.09%	0.00%
TensorFlow Neural Network Classifier	82.09%	82.09%	0.00%
Vowpal Wabbit Classifier	82.09%	86.57%	4.48%
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	82.09%	83.58%	1.49%
RandomForest Classifier (Gini)	82.09%	82.09%	0.00%

**) Majority class distribution*

Appendix Table XXX: Holdout performance



Model	Elastic-Net Classifier (L2 / Binomial Deviance)
Data source	Holdout

		Predicted	
		N	P
Actual	N	49	6
	P	5	7

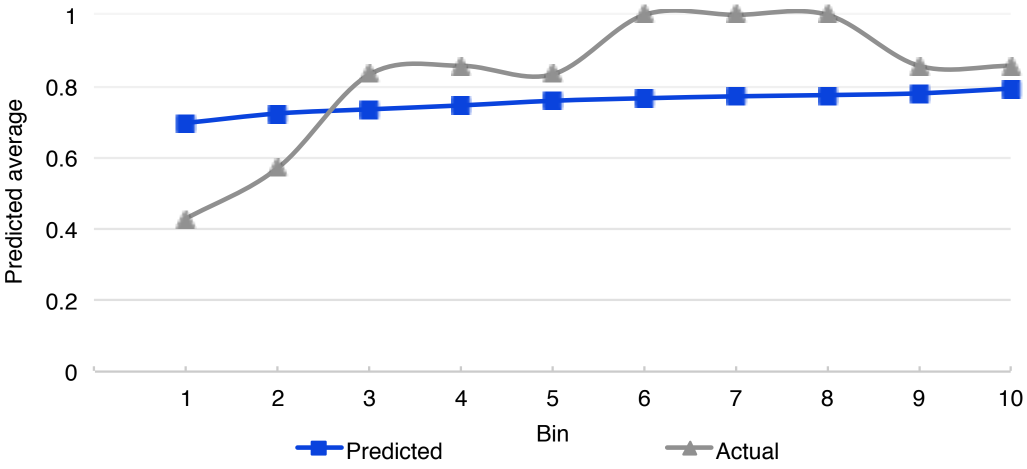
PERFORMANCE METRICS	
Name	Value
Accuracy	83.58%
Precision	53.85%
Recall (sensitivity, TP rate)	58.33%
Specificity (TN rate)	89.09%

Model	Elastic-Net Classifier (L2 / Binomial Deviance)
Data source	BT1 (validation)

		Predicted	
		N	P
Actual	N	36	16
	P	2	13

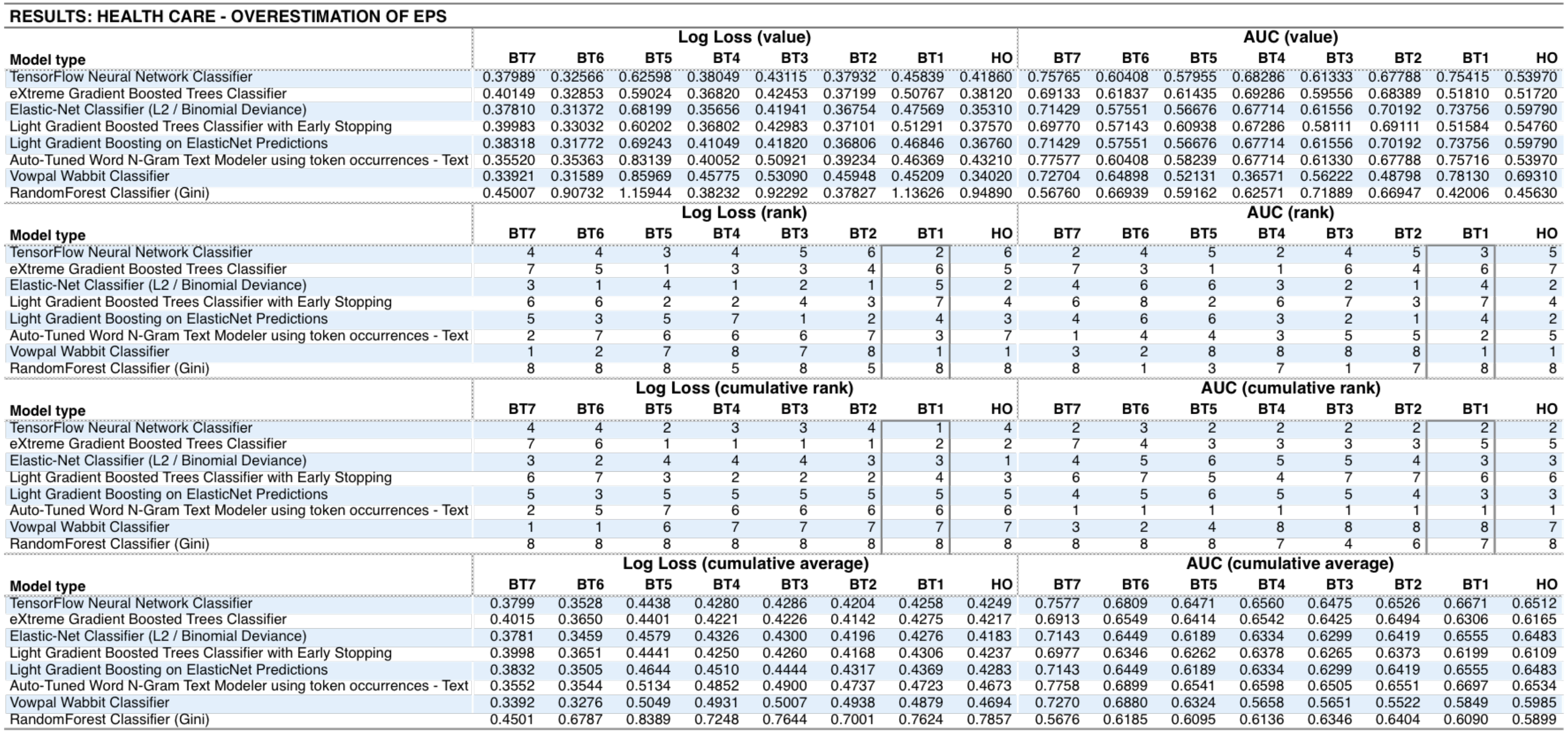
PERFORMANCE METRICS	
Name	Value
Accuracy	73.13%
Precision	44.83%
Recall (sensitivity, TP rate)	86.67%
Specificity (TN rate)	69.23%

Appendix Table XXXI: Confusion matrices – Holdout and validation



Appendix Table XXXII: Lift chart

Appendix XIX. Health Care (Overestimation of EPS)



RESULTS: HEALTH CARE - OVERESTIMATION OF EPS

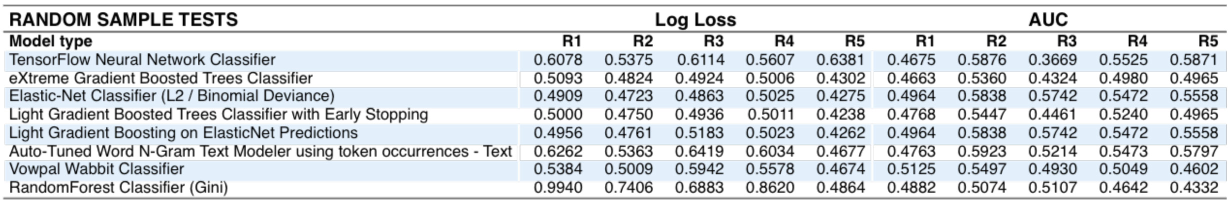
	Log Loss (value)								AUC (value)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
TensorFlow Neural Network Classifier	0.37989	0.32566	0.62598	0.38049	0.43115	0.37932	0.45839	0.41860	0.75765	0.60408	0.57955	0.68286	0.61333	0.67788	0.75415	0.53970
eXtreme Gradient Boosted Trees Classifier	0.40149	0.32853	0.59024	0.36820	0.42453	0.37199	0.50767	0.38120	0.69133	0.61837	0.61435	0.69286	0.59556	0.68389	0.51810	0.51720
Elastic-Net Classifier (L2 / Binomial Deviance)	0.37810	0.31372	0.68199	0.35656	0.41941	0.36754	0.47569	0.35310	0.71429	0.57551	0.56676	0.67714	0.61556	0.70192	0.73756	0.59790
Light Gradient Boosted Trees Classifier with Early Stopping	0.39983	0.33032	0.60202	0.36802	0.42983	0.37101	0.51291	0.37570	0.69770	0.57143	0.60938	0.67286	0.58111	0.69111	0.51584	0.54760
Light Gradient Boosting on ElasticNet Predictions	0.38318	0.31772	0.69243	0.41049	0.41820	0.36806	0.46846	0.36760	0.71429	0.57551	0.56676	0.67714	0.61556	0.70192	0.73756	0.59790
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.35520	0.35363	0.83139	0.40052	0.50921	0.39234	0.46369	0.43210	0.77577	0.60408	0.58239	0.67714	0.61330	0.67788	0.75716	0.53970
Vowpal Wabbit Classifier	0.33921	0.31589	0.85969	0.45775	0.53090	0.45948	0.45209	0.34020	0.72704	0.64898	0.52131	0.36571	0.56222	0.48798	0.78130	0.69310
RandomForest Classifier (Gini)	0.45007	0.90732	1.15944	0.38232	0.92292	0.37827	1.13626	0.94890	0.56760	0.66939	0.59162	0.62571	0.71889	0.66947	0.42006	0.45630

	Log Loss (rank)								AUC (rank)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
TensorFlow Neural Network Classifier	4	4	3	4	5	6	2	6	2	4	5	2	4	5	3	5
eXtreme Gradient Boosted Trees Classifier	7	5	1	3	3	4	6	5	7	3	1	1	6	4	6	7
Elastic-Net Classifier (L2 / Binomial Deviance)	3	1	4	1	2	1	5	2	4	6	6	3	2	1	4	2
Light Gradient Boosted Trees Classifier with Early Stopping	6	6	2	2	4	3	7	4	6	8	2	6	7	3	7	4
Light Gradient Boosting on ElasticNet Predictions	5	3	5	7	1	2	4	3	4	6	6	3	2	1	4	2
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	2	7	6	6	6	7	3	7	1	4	4	3	5	5	2	5
Vowpal Wabbit Classifier	1	2	7	8	7	8	1	1	3	2	8	8	8	8	1	1
RandomForest Classifier (Gini)	8	8	8	5	8	5	8	8	8	1	3	7	1	7	8	8

	Log Loss (cumulative rank)								AUC (cumulative rank)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
TensorFlow Neural Network Classifier	4	4	2	3	3	4	1	4	2	3	2	2	2	2	2	2
eXtreme Gradient Boosted Trees Classifier	7	6	1	1	1	1	2	2	7	4	3	3	3	3	5	5
Elastic-Net Classifier (L2 / Binomial Deviance)	3	2	4	4	4	3	3	1	4	5	6	5	5	4	3	3
Light Gradient Boosted Trees Classifier with Early Stopping	6	7	3	2	2	2	4	3	6	7	5	4	7	7	6	6
Light Gradient Boosting on ElasticNet Predictions	5	3	5	5	5	5	5	5	4	5	6	5	5	4	3	3
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	2	5	7	6	6	6	6	6	1	1	1	1	1	1	1	1
Vowpal Wabbit Classifier	1	1	6	7	7	7	7	7	3	2	4	8	8	8	8	7
RandomForest Classifier (Gini)	8	8	8	8	8	8	8	8	8	8	8	7	4	6	7	8

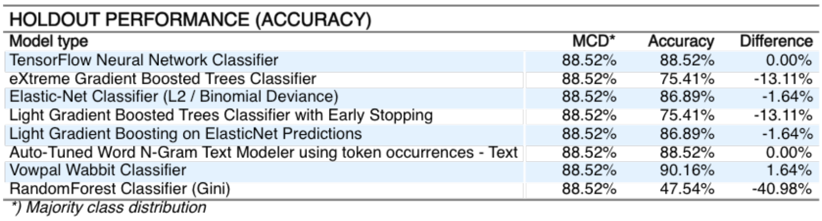
	Log Loss (cumulative average)								AUC (cumulative average)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
TensorFlow Neural Network Classifier	0.3799	0.3528	0.4438	0.4280	0.4286	0.4204	0.4258	0.4249	0.7577	0.6809	0.6471	0.6560	0.6475	0.6526	0.6671	0.6512
eXtreme Gradient Boosted Trees Classifier	0.4015	0.3650	0.4401	0.4221	0.4226	0.4142	0.4275	0.4217	0.6913	0.6549	0.6414	0.6542	0.6425	0.6494	0.6306	0.6165
Elastic-Net Classifier (L2 / Binomial Deviance)	0.3781	0.3459	0.4579	0.4326	0.4300	0.4196	0.4276	0.4183	0.7143	0.6449	0.6189	0.6334	0.6299	0.6419	0.6555	0.6483
Light Gradient Boosted Trees Classifier with Early Stopping	0.3998	0.3651	0.4441	0.4250	0.4260	0.4168	0.4306	0.4237	0.6977	0.6346	0.6262	0.6378	0.6265	0.6373	0.6199	0.6109
Light Gradient Boosting on ElasticNet Predictions	0.3832	0.3505	0.4644	0.4510	0.4444	0.4317	0.4369	0.4283	0.7143	0.6449	0.6189	0.6334	0.6299	0.6419	0.6555	0.6483
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.3552	0.3544	0.5134	0.4852	0.4900	0.4737	0.4723	0.4673	0.7758	0.6899	0.6541	0.6598	0.6505	0.6551	0.6697	0.6534
Vowpal Wabbit Classifier	0.3392	0.3276	0.5049	0.4931	0.5007	0.4938	0.4879	0.4694	0.7270	0.6880	0.6324	0.5658	0.5651	0.5522	0.5849	0.5985
RandomForest Classifier (Gini)	0.4501	0.6787	0.8389	0.7248	0.7644	0.7001	0.7624	0.7857	0.5676	0.6185	0.6095	0.6136	0.6346	0.6404	0.6090	0.5899

Appendix Table XXXIII: Log Loss and AUC (Values and rank)



RANDOM SAMPLE TESTS	Log Loss					AUC				
Model type	**R1**	**R2**	**R3**	**R4**	**R5**	**R1**	**R2**	**R3**	**R4**	**R5**
TensorFlow Neural Network Classifier	0.6078	0.5375	0.6114	0.5607	0.6381	0.4675	0.5876	0.3669	0.5525	0.5871
eXtreme Gradient Boosted Trees Classifier	0.5093	0.4824	0.4924	0.5006	0.4302	0.4663	0.5360	0.4324	0.4980	0.4965
Elastic-Net Classifier (L2 / Binomial Deviance)	0.4909	0.4723	0.4863	0.5025	0.4275	0.4964	0.5838	0.5742	0.5472	0.5558
Light Gradient Boosted Trees Classifier with Early Stopping	0.5000	0.4750	0.4936	0.5011	0.4238	0.4768	0.5447	0.4461	0.5240	0.4965
Light Gradient Boosting on ElasticNet Predictions	0.4956	0.4761	0.5183	0.5023	0.4262	0.4964	0.5838	0.5742	0.5472	0.5558
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.6262	0.5363	0.6419	0.6034	0.4677	0.4763	0.5923	0.5214	0.5473	0.5797
Vowpal Wabbit Classifier	0.5384	0.5009	0.5942	0.5578	0.4674	0.5125	0.5497	0.4930	0.5049	0.4602
RandomForest Classifier (Gini)	0.9940	0.7406	0.6883	0.8620	0.4864	0.4882	0.5074	0.5107	0.4642	0.4332

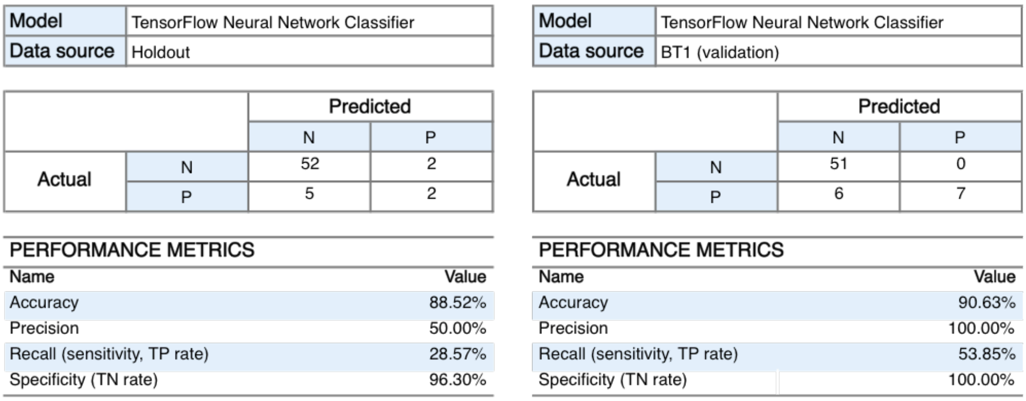
Appendix Table XXXIV: Random sample test



HOLDOUT PERFORMANCE (ACCURACY)			
Model type	**MCD***	**Accuracy**	**Difference**
TensorFlow Neural Network Classifier	88.52%	88.52%	0.00%
eXtreme Gradient Boosted Trees Classifier	88.52%	75.41%	-13.11%
Elastic-Net Classifier (L2 / Binomial Deviance)	88.52%	86.89%	-1.64%
Light Gradient Boosted Trees Classifier with Early Stopping	88.52%	75.41%	-13.11%
Light Gradient Boosting on ElasticNet Predictions	88.52%	86.89%	-1.64%
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	88.52%	88.52%	0.00%
Vowpal Wabbit Classifier	88.52%	90.16%	1.64%
RandomForest Classifier (Gini)	88.52%	47.54%	-40.98%

**) Majority class distribution*

Appendix Table XXXV: Holdout performance



Model	TensorFlow Neural Network Classifier
Data source	Holdout

		Predicted	
		N	P
Actual	N	52	2
	P	5	2

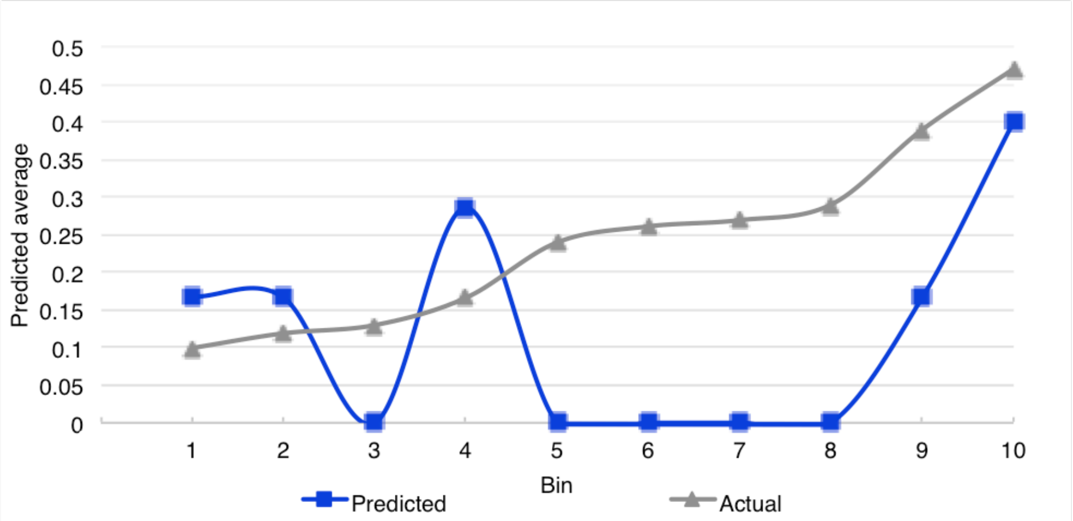
PERFORMANCE METRICS	
Name	**Value**
Accuracy	88.52%
Precision	50.00%
Recall (sensitivity, TP rate)	28.57%
Specificity (TN rate)	96.30%

Model	TensorFlow Neural Network Classifier
Data source	BT1 (validation)

		Predicted	
		N	P
Actual	N	51	0
	P	6	7

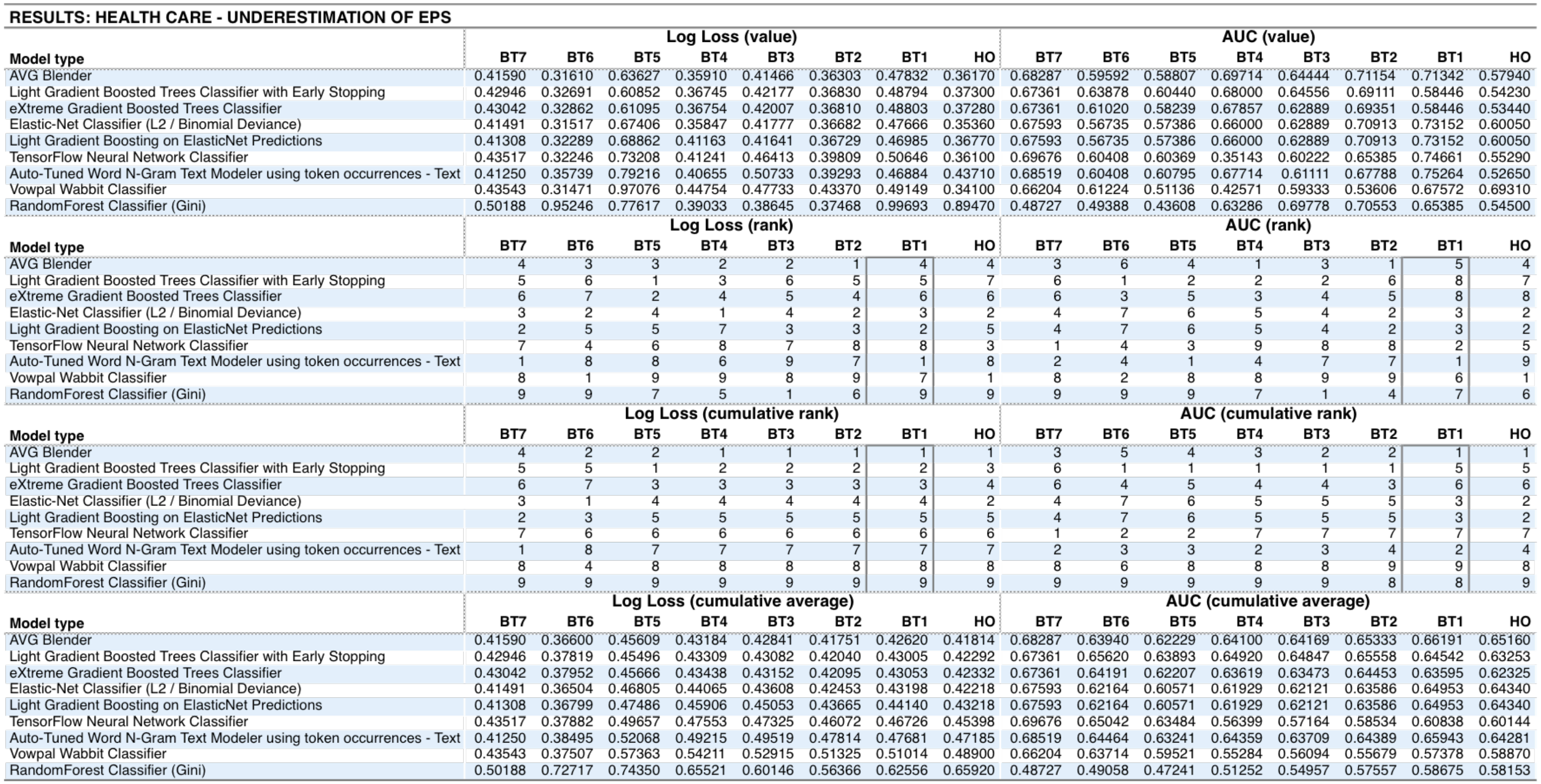
PERFORMANCE METRICS	
Name	**Value**
Accuracy	90.63%
Precision	100.00%
Recall (sensitivity, TP rate)	53.85%
Specificity (TN rate)	100.00%

Appendix Table XXXVI: Confusion matrices



Appendix Table XXXVII: Lift chart

Appendix XX. Health Care (Underestimation of EPS)



RESULTS: HEALTH CARE - UNDERESTIMATION OF EPS

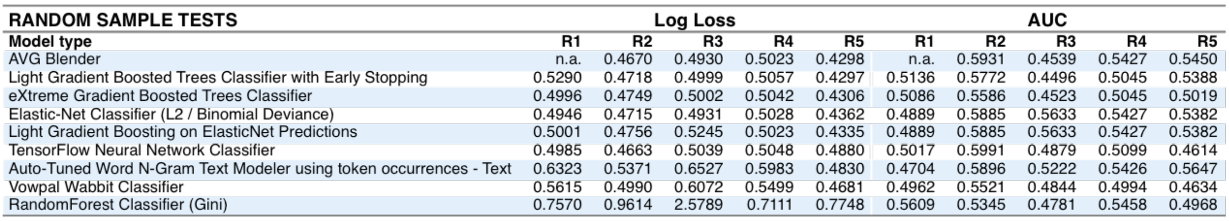
	Log Loss (value)								AUC (value)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
AVG Blender	0.41590	0.31610	0.63627	0.35910	0.41466	0.36303	0.47832	0.36170	0.68287	0.59592	0.58807	0.69714	0.64444	0.71154	0.71342	0.57940
Light Gradient Boosted Trees Classifier with Early Stopping	0.42946	0.32691	0.60852	0.36745	0.42177	0.36830	0.48794	0.37300	0.67361	0.63878	0.60440	0.68000	0.64556	0.69111	0.58446	0.54230
eXtreme Gradient Boosted Trees Classifier	0.43042	0.32862	0.61095	0.36754	0.42007	0.36810	0.48803	0.37280	0.67361	0.61020	0.58239	0.67857	0.62889	0.69351	0.58446	0.53440
Elastic-Net Classifier (L2 / Binomial Deviance)	0.41491	0.31517	0.67406	0.35847	0.41777	0.36682	0.47666	0.35360	0.67593	0.56735	0.57386	0.66000	0.62889	0.70913	0.73152	0.60050
Light Gradient Boosting on ElasticNet Predictions	0.41308	0.32289	0.68862	0.41163	0.41641	0.36729	0.46985	0.36770	0.67593	0.56735	0.57386	0.66000	0.62889	0.70913	0.73152	0.60050
TensorFlow Neural Network Classifier	0.43517	0.32246	0.73208	0.41241	0.46413	0.39809	0.50646	0.36100	0.69676	0.60408	0.60369	0.35143	0.60222	0.65385	0.74661	0.55290
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.41250	0.35739	0.79216	0.40655	0.50733	0.39293	0.46884	0.43710	0.68519	0.60408	0.60795	0.67714	0.61111	0.67788	0.75264	0.52650
Vowpal Wabbit Classifier	0.43543	0.31471	0.97076	0.44754	0.47733	0.43370	0.49149	0.34100	0.66204	0.61224	0.51136	0.42571	0.59333	0.53606	0.67572	0.69310
RandomForest Classifier (Gini)	0.50188	0.95246	0.77617	0.39033	0.38645	0.37468	0.99693	0.89470	0.48727	0.49388	0.43608	0.63286	0.69778	0.70553	0.65385	0.54500

	Log Loss (rank)								AUC (rank)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
AVG Blender	4	3	3	2	2	1	4	4	3	6	4	1	3	1	5	4
Light Gradient Boosted Trees Classifier with Early Stopping	5	6	1	3	6	5	5	7	6	1	2	2	2	6	8	7
eXtreme Gradient Boosted Trees Classifier	6	7	2	4	5	4	6	6	6	3	5	3	4	5	8	8
Elastic-Net Classifier (L2 / Binomial Deviance)	3	2	4	1	4	2	3	2	4	7	6	5	4	2	3	2
Light Gradient Boosting on ElasticNet Predictions	2	5	5	7	3	3	2	5	4	7	6	5	4	2	3	2
TensorFlow Neural Network Classifier	7	4	6	8	7	8	8	3	1	4	3	9	8	8	2	5
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	1	8	8	6	9	7	1	8	2	4	1	4	7	7	1	9
Vowpal Wabbit Classifier	8	1	9	9	8	9	7	1	8	2	8	8	9	9	6	1
RandomForest Classifier (Gini)	9	9	7	5	1	6	9	9	9	9	9	7	1	4	7	6

	Log Loss (cumulative rank)								AUC (cumulative rank)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
AVG Blender	4	2	2	1	1	1	1	1	3	5	4	3	2	2	1	1
Light Gradient Boosted Trees Classifier with Early Stopping	5	5	1	2	2	2	2	3	6	1	1	1	1	1	5	5
eXtreme Gradient Boosted Trees Classifier	6	7	3	3	3	3	3	4	6	4	5	4	4	3	6	6
Elastic-Net Classifier (L2 / Binomial Deviance)	3	1	4	4	4	4	4	2	4	7	6	5	5	5	3	2
Light Gradient Boosting on ElasticNet Predictions	2	3	5	5	5	5	5	5	4	7	6	5	5	5	3	2
TensorFlow Neural Network Classifier	7	6	6	6	6	6	6	6	1	2	2	7	7	7	7	7
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	1	8	7	7	7	7	7	7	2	3	3	2	3	4	2	4
Vowpal Wabbit Classifier	8	4	8	8	8	8	8	8	8	6	8	8	8	9	9	8
RandomForest Classifier (Gini)	9	9	9	9	9	9	9	9	9	9	9	9	9	8	8	9

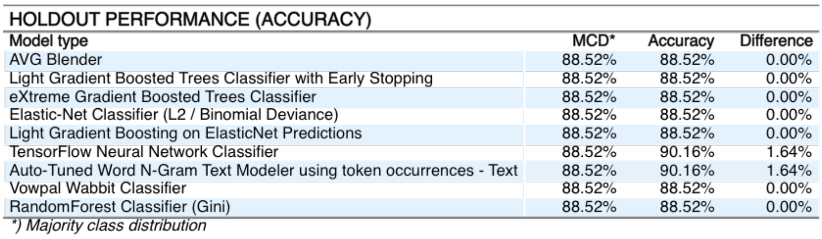
	Log Loss (cumulative average)								AUC (cumulative average)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
AVG Blender	0.41590	0.36600	0.45609	0.43184	0.42841	0.41751	0.42620	0.41814	0.68287	0.63940	0.62229	0.64100	0.64169	0.65333	0.66191	0.65160
Light Gradient Boosted Trees Classifier with Early Stopping	0.42946	0.37819	0.45496	0.43309	0.43082	0.42040	0.43005	0.42292	0.67361	0.65620	0.63893	0.64920	0.64847	0.65558	0.64542	0.63253
eXtreme Gradient Boosted Trees Classifier	0.43042	0.37952	0.45666	0.43438	0.43152	0.42095	0.43053	0.42332	0.67361	0.64191	0.62207	0.63619	0.63473	0.64453	0.63595	0.62325
Elastic-Net Classifier (L2 / Binomial Deviance)	0.41491	0.36504	0.46805	0.44065	0.43608	0.42453	0.43198	0.42218	0.67593	0.62164	0.60571	0.61929	0.62121	0.63586	0.64953	0.64340
Light Gradient Boosting on ElasticNet Predictions	0.41308	0.36799	0.47486	0.45906	0.45053	0.43665	0.44140	0.43218	0.67593	0.62164	0.60571	0.61929	0.62121	0.63586	0.64953	0.64340
TensorFlow Neural Network Classifier	0.43517	0.37882	0.49657	0.47553	0.47325	0.46072	0.46726	0.45398	0.69676	0.65042	0.63484	0.56399	0.57164	0.58534	0.60838	0.60144
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.41250	0.38495	0.52068	0.49215	0.49519	0.47814	0.47681	0.47185	0.68519	0.64464	0.63241	0.64359	0.63709	0.64389	0.65943	0.64281
Vowpal Wabbit Classifier	0.43543	0.37507	0.57363	0.54211	0.52915	0.51325	0.51014	0.48900	0.66204	0.63714	0.59521	0.55284	0.56094	0.55679	0.57378	0.58870
RandomForest Classifier (Gini)	0.50188	0.72717	0.74350	0.65521	0.60146	0.56366	0.62556	0.65920	0.48727	0.49058	0.47241	0.51252	0.54957	0.57557	0.58675	0.58153

Appendix Table XXXVIII: Log Loss and AUC (Values and rank)



RANDOM SAMPLE TESTS	Log Loss					AUC				
Model type	R1	R2	R3	R4	R5	R1	R2	R3	R4	R5
AVG Blender	n.a.	0.4670	0.4930	0.5023	0.4298	n.a.	0.5931	0.4539	0.5427	0.5450
Light Gradient Boosted Trees Classifier with Early Stopping	0.5290	0.4718	0.4999	0.5057	0.4297	0.5136	0.5772	0.4496	0.5045	0.5388
eXtreme Gradient Boosted Trees Classifier	0.4996	0.4749	0.5002	0.5042	0.4306	0.5086	0.5586	0.4523	0.5045	0.5019
Elastic-Net Classifier (L2 / Binomial Deviance)	0.4946	0.4715	0.4931	0.5028	0.4362	0.4889	0.5885	0.5633	0.5427	0.5382
Light Gradient Boosting on ElasticNet Predictions	0.5001	0.4756	0.5245	0.5023	0.4335	0.4889	0.5885	0.5633	0.5427	0.5382
TensorFlow Neural Network Classifier	0.4985	0.4663	0.5039	0.5048	0.4880	0.5017	0.5991	0.4879	0.5099	0.4614
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.6323	0.5371	0.6527	0.5983	0.4830	0.4704	0.5896	0.5222	0.5426	0.5647
Vowpal Wabbit Classifier	0.5615	0.4990	0.6072	0.5499	0.4681	0.4962	0.5521	0.4844	0.4994	0.4634
RandomForest Classifier (Gini)	0.7570	0.9614	2.5789	0.7111	0.7748	0.5609	0.5345	0.4781	0.5458	0.4968

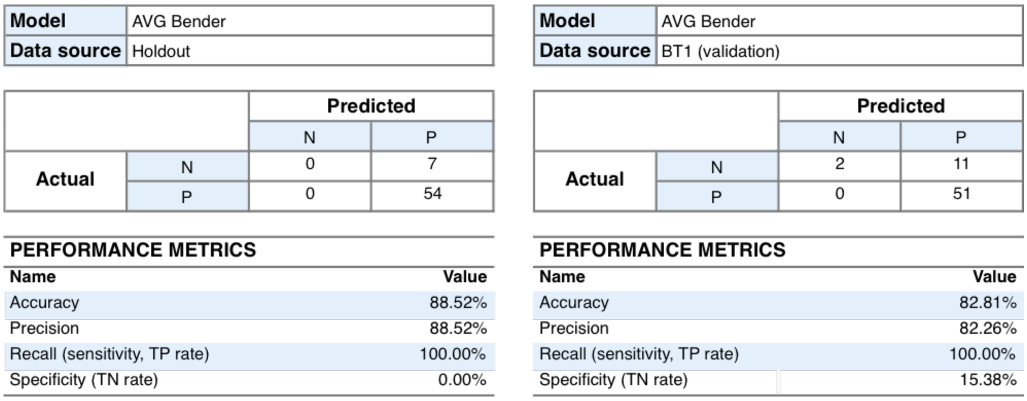
Appendix Table XXXIX: Random sample test



HOLDOUT PERFORMANCE (ACCURACY)			
Model type	MCD*	Accuracy	Difference
AVG Blender	88.52%	88.52%	0.00%
Light Gradient Boosted Trees Classifier with Early Stopping	88.52%	88.52%	0.00%
eXtreme Gradient Boosted Trees Classifier	88.52%	88.52%	0.00%
Elastic-Net Classifier (L2 / Binomial Deviance)	88.52%	88.52%	0.00%
Light Gradient Boosting on ElasticNet Predictions	88.52%	88.52%	0.00%
TensorFlow Neural Network Classifier	88.52%	90.16%	1.64%
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	88.52%	90.16%	1.64%
Vowpal Wabbit Classifier	88.52%	88.52%	0.00%
RandomForest Classifier (Gini)	88.52%	88.52%	0.00%

*) Majority class distribution

Appendix Table XL: Holdout performance



Model	AVG Bender
Data source	Holdout

		Predicted	
		N	P
Actual	N	0	7
	P	0	54

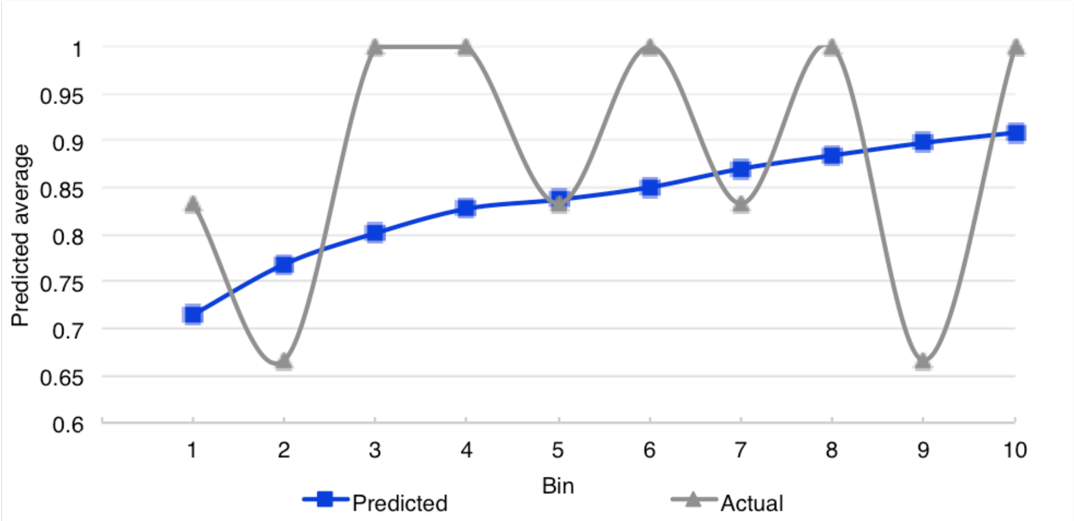
PERFORMANCE METRICS	
Name	Value
Accuracy	88.52%
Precision	88.52%
Recall (sensitivity, TP rate)	100.00%
Specificity (TN rate)	0.00%

Model	AVG Bender
Data source	BT1 (validation)

		Predicted	
		N	P
Actual	N	2	11
	P	0	51

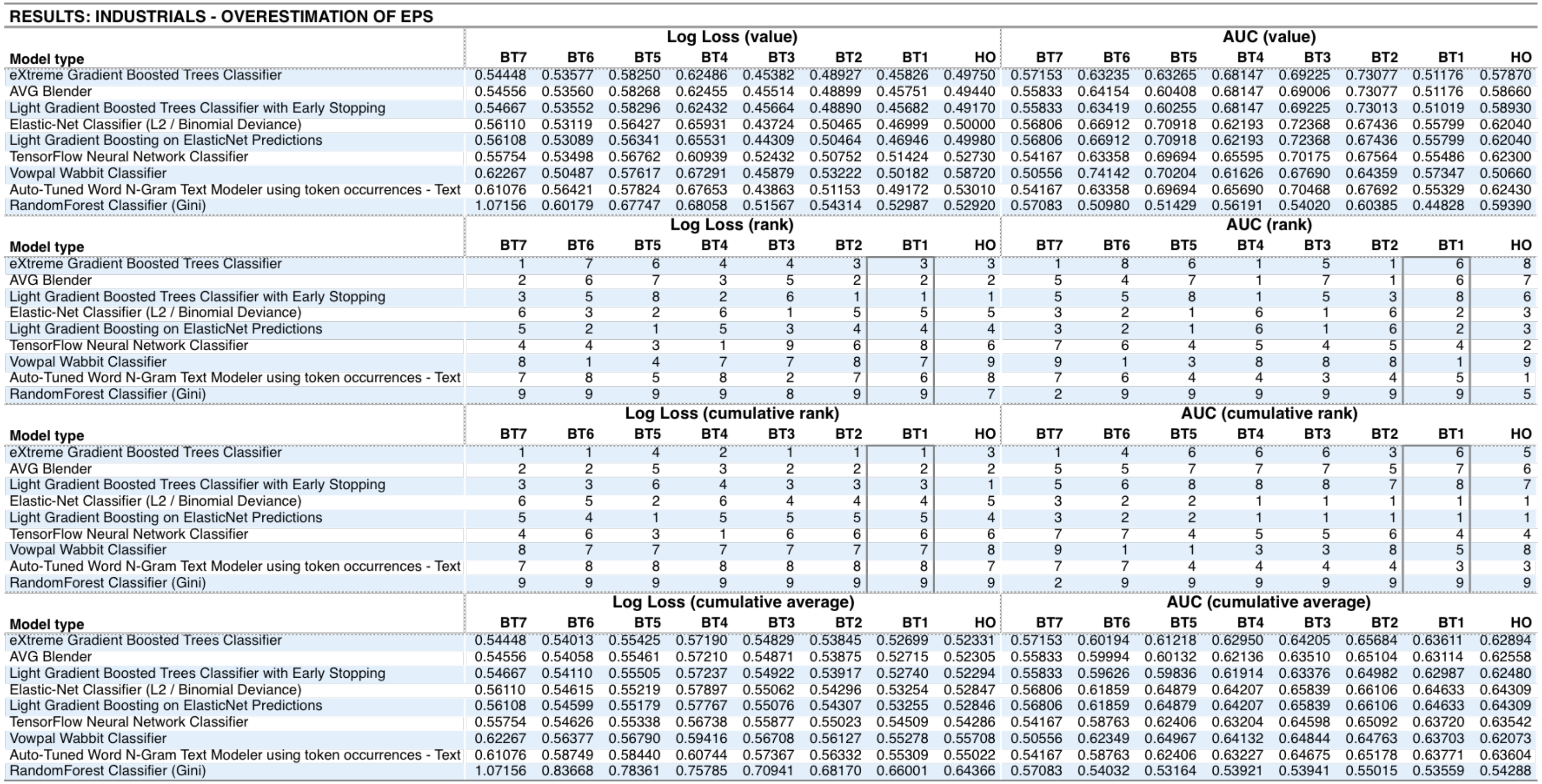
PERFORMANCE METRICS	
Name	Value
Accuracy	82.81%
Precision	82.26%
Recall (sensitivity, TP rate)	100.00%
Specificity (TN rate)	15.38%

Appendix Table XLI: Confusion matrices – Holdout and validation



Appendix Table XLII: Lift chart

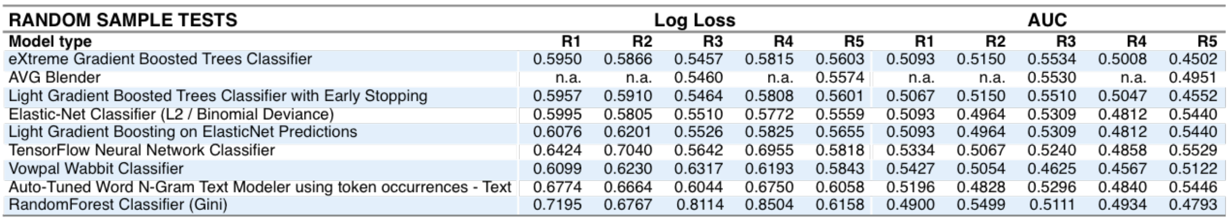
Appendix XXI. Industrials (Overestimation of EPS)



RESULTS: INDUSTRIALS - OVERESTIMATION OF EPS

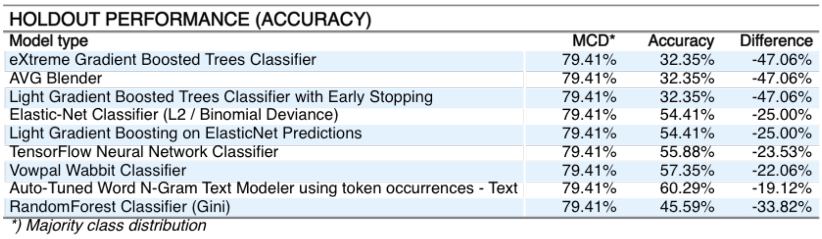
	Log Loss (value)								AUC (value)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
eXtreme Gradient Boosted Trees Classifier	0.54448	0.53577	0.58250	0.62486	0.45382	0.48927	0.45826	0.49750	0.57153	0.63235	0.63265	0.68147	0.69225	0.73077	0.51176	0.57870
AVG Blender	0.54556	0.53560	0.58268	0.62455	0.45514	0.48899	0.45751	0.49440	0.55833	0.64154	0.60408	0.68147	0.69006	0.73077	0.51176	0.58660
Light Gradient Boosted Trees Classifier with Early Stopping	0.54667	0.53552	0.58296	0.62432	0.45664	0.48890	0.45682	0.49170	0.55833	0.63419	0.60255	0.68147	0.69225	0.73013	0.51019	0.58930
Elastic-Net Classifier (L2 / Binomial Deviance)	0.56110	0.53119	0.56427	0.65931	0.43724	0.50465	0.46999	0.50000	0.56806	0.66912	0.70918	0.62193	0.72368	0.67436	0.55799	0.62040
Light Gradient Boosting on ElasticNet Predictions	0.56108	0.53089	0.56341	0.65531	0.44309	0.50464	0.46946	0.49980	0.56806	0.66912	0.70918	0.62193	0.72368	0.67436	0.55799	0.62040
TensorFlow Neural Network Classifier	0.55754	0.53498	0.56762	0.60939	0.52432	0.50752	0.51424	0.52730	0.54167	0.63358	0.69694	0.65595	0.70175	0.67564	0.55486	0.62300
Vowpal Wabbit Classifier	0.62267	0.50487	0.57617	0.67291	0.45879	0.53222	0.50182	0.58720	0.50556	0.74142	0.70204	0.61626	0.67690	0.64359	0.57347	0.50660
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.61076	0.56421	0.57824	0.67653	0.43863	0.51153	0.49172	0.53010	0.54167	0.63358	0.69694	0.65690	0.70468	0.67692	0.55329	0.62430
RandomForest Classifier (Gini)	1.07156	0.60179	0.67747	0.68058	0.51567	0.54314	0.52987	0.52920	0.57083	0.50980	0.51429	0.56191	0.54020	0.60385	0.44828	0.59390
	Log Loss (rank)								**AUC (rank)**							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
eXtreme Gradient Boosted Trees Classifier	1	7	6	4	4	3	3	3	1	8	6	1	5	1	6	8
AVG Blender	2	6	7	3	5	2	2	2	5	4	7	1	7	1	6	7
Light Gradient Boosted Trees Classifier with Early Stopping	3	5	8	2	6	1	1	1	5	5	8	1	5	3	8	6
Elastic-Net Classifier (L2 / Binomial Deviance)	6	3	2	6	1	5	5	5	3	2	1	6	1	6	2	3
Light Gradient Boosting on ElasticNet Predictions	5	2	1	5	3	4	4	4	3	2	1	6	1	6	2	3
TensorFlow Neural Network Classifier	4	4	3	1	9	6	8	6	7	6	4	5	4	5	4	2
Vowpal Wabbit Classifier	8	1	4	7	7	8	7	9	9	1	3	8	8	8	1	9
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	7	8	5	8	2	7	6	8	7	6	4	4	3	4	5	1
RandomForest Classifier (Gini)	9	9	9	9	8	9	9	7	2	9	9	9	9	9	9	5
	Log Loss (cumulative rank)								**AUC (cumulative rank)**							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
eXtreme Gradient Boosted Trees Classifier	1	1	4	2	1	1	1	3	1	4	6	6	6	3	6	5
AVG Blender	2	2	5	3	2	2	2	2	5	5	7	7	7	5	7	6
Light Gradient Boosted Trees Classifier with Early Stopping	3	3	6	4	3	3	3	1	5	6	8	8	8	7	8	7
Elastic-Net Classifier (L2 / Binomial Deviance)	6	5	2	6	4	4	4	5	3	2	2	1	1	1	1	1
Light Gradient Boosting on ElasticNet Predictions	5	4	1	5	5	5	5	4	3	2	2	1	1	1	1	1
TensorFlow Neural Network Classifier	4	6	3	1	6	6	6	6	7	7	4	5	5	6	4	4
Vowpal Wabbit Classifier	8	7	7	7	7	7	7	8	9	1	1	3	3	8	5	8
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	7	8	8	8	8	8	8	7	7	7	4	4	4	4	3	3
RandomForest Classifier (Gini)	9	9	9	9	9	9	9	9	2	9	9	9	9	9	9	9
	Log Loss (cumulative average)								**AUC (cumulative average)**							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
eXtreme Gradient Boosted Trees Classifier	0.54448	0.54013	0.55425	0.57190	0.54829	0.53845	0.52699	0.52331	0.57153	0.60194	0.61218	0.62950	0.64205	0.65684	0.63611	0.62894
AVG Blender	0.54556	0.54058	0.55461	0.57210	0.54871	0.53875	0.52715	0.52305	0.55833	0.59994	0.60132	0.62136	0.63510	0.65104	0.63114	0.62558
Light Gradient Boosted Trees Classifier with Early Stopping	0.54667	0.54110	0.55505	0.57237	0.54922	0.53917	0.52740	0.52294	0.55833	0.59626	0.59836	0.61914	0.63376	0.64982	0.62987	0.62480
Elastic-Net Classifier (L2 / Binomial Deviance)	0.56110	0.54615	0.55219	0.57897	0.55062	0.54296	0.53254	0.52847	0.56806	0.61859	0.64879	0.64207	0.65839	0.66106	0.64633	0.64309
Light Gradient Boosting on ElasticNet Predictions	0.56108	0.54599	0.55179	0.57767	0.55076	0.54307	0.53255	0.52846	0.56806	0.61859	0.64879	0.64207	0.65839	0.66106	0.64633	0.64309
TensorFlow Neural Network Classifier	0.55754	0.54626	0.55338	0.56738	0.55877	0.55023	0.54509	0.54286	0.54167	0.58763	0.62406	0.63204	0.64598	0.65092	0.63720	0.63542
Vowpal Wabbit Classifier	0.62267	0.56377	0.56790	0.59416	0.56708	0.56127	0.55278	0.55708	0.50556	0.62349	0.64967	0.64132	0.64844	0.64763	0.63703	0.62073
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.61076	0.58749	0.58440	0.60744	0.57367	0.56332	0.55309	0.55022	0.54167	0.58763	0.62406	0.63227	0.64675	0.65178	0.63771	0.63604
RandomForest Classifier (Gini)	1.07156	0.83668	0.78361	0.75785	0.70941	0.68170	0.66001	0.64366	0.57083	0.54032	0.53164	0.53921	0.53941	0.55015	0.53559	0.54288

Appendix Table XLIII: Log Loss and AUC (Values and rank)



RANDOM SAMPLE TESTS	Log Loss					AUC				
Model type	R1	R2	R3	R4	R5	R1	R2	R3	R4	R5
eXtreme Gradient Boosted Trees Classifier	0.5950	0.5866	0.5457	0.5815	0.5603	0.5093	0.5150	0.5534	0.5008	0.4502
AVG Blender	n.a.	n.a.	0.5460	n.a.	0.5574	n.a.	n.a.	0.5530	n.a.	0.4951
Light Gradient Boosted Trees Classifier with Early Stopping	0.5957	0.5910	0.5464	0.5808	0.5601	0.5067	0.5150	0.5510	0.5047	0.4552
Elastic-Net Classifier (L2 / Binomial Deviance)	0.5995	0.5805	0.5510	0.5772	0.5559	0.5093	0.4964	0.5309	0.4812	0.5440
Light Gradient Boosting on ElasticNet Predictions	0.6076	0.6201	0.5526	0.5825	0.5655	0.5093	0.4964	0.5309	0.4812	0.5440
TensorFlow Neural Network Classifier	0.6424	0.7040	0.5642	0.6955	0.5818	0.5334	0.5067	0.5240	0.4858	0.5529
Vowpal Wabbit Classifier	0.6099	0.6230	0.6317	0.6193	0.5843	0.5427	0.5054	0.4625	0.4567	0.5122
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.6774	0.6664	0.6044	0.6750	0.6058	0.5196	0.4828	0.5296	0.4840	0.5446
RandomForest Classifier (Gini)	0.7195	0.6767	0.8114	0.8504	0.6158	0.4900	0.5499	0.5111	0.4934	0.4793

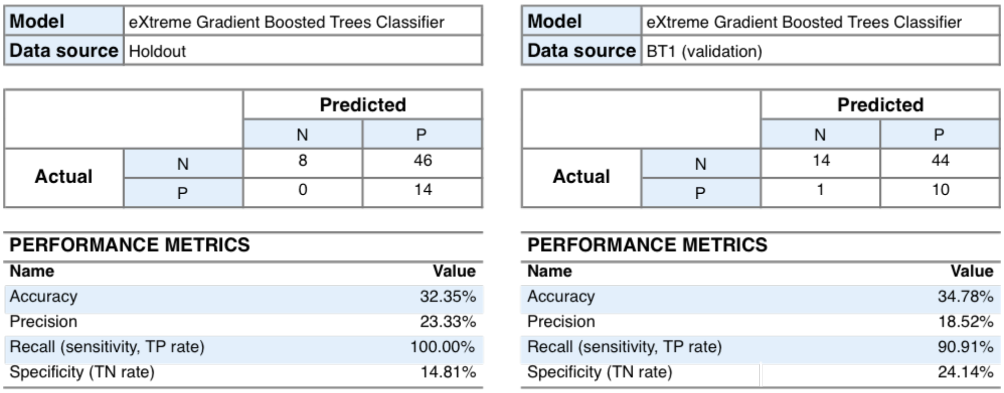
Appendix Table XLIV: Random sample test



HOLDOUT PERFORMANCE (ACCURACY)			
Model type	MCD*	Accuracy	Difference
eXtreme Gradient Boosted Trees Classifier	79.41%	32.35%	-47.06%
AVG Blender	79.41%	32.35%	-47.06%
Light Gradient Boosted Trees Classifier with Early Stopping	79.41%	32.35%	-47.06%
Elastic-Net Classifier (L2 / Binomial Deviance)	79.41%	54.41%	-25.00%
Light Gradient Boosting on ElasticNet Predictions	79.41%	54.41%	-25.00%
TensorFlow Neural Network Classifier	79.41%	55.88%	-23.53%
Vowpal Wabbit Classifier	79.41%	57.35%	-22.06%
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	79.41%	60.29%	-19.12%
RandomForest Classifier (Gini)	79.41%	45.59%	-33.82%

**) Majority class distribution*

Appendix Table XLV: Holdout performance



Model	eXtreme Gradient Boosted Trees Classifier
Data source	Holdout

		Predicted	
		N	P
Actual	N	8	46
	P	0	14

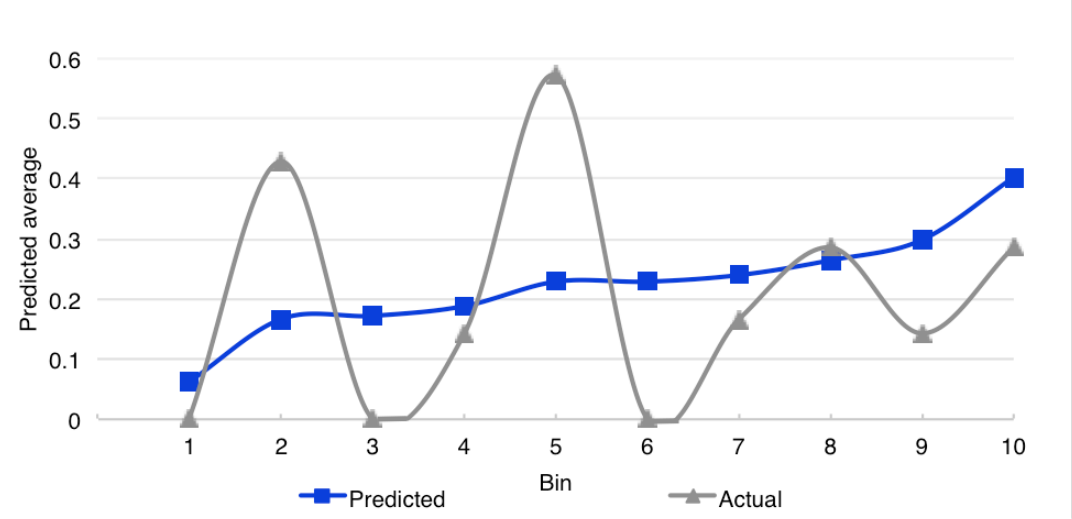
PERFORMANCE METRICS	
Name	Value
Accuracy	32.35%
Precision	23.33%
Recall (sensitivity, TP rate)	100.00%
Specificity (TN rate)	14.81%

Model	eXtreme Gradient Boosted Trees Classifier
Data source	BT1 (validation)

		Predicted	
		N	P
Actual	N	14	44
	P	1	10

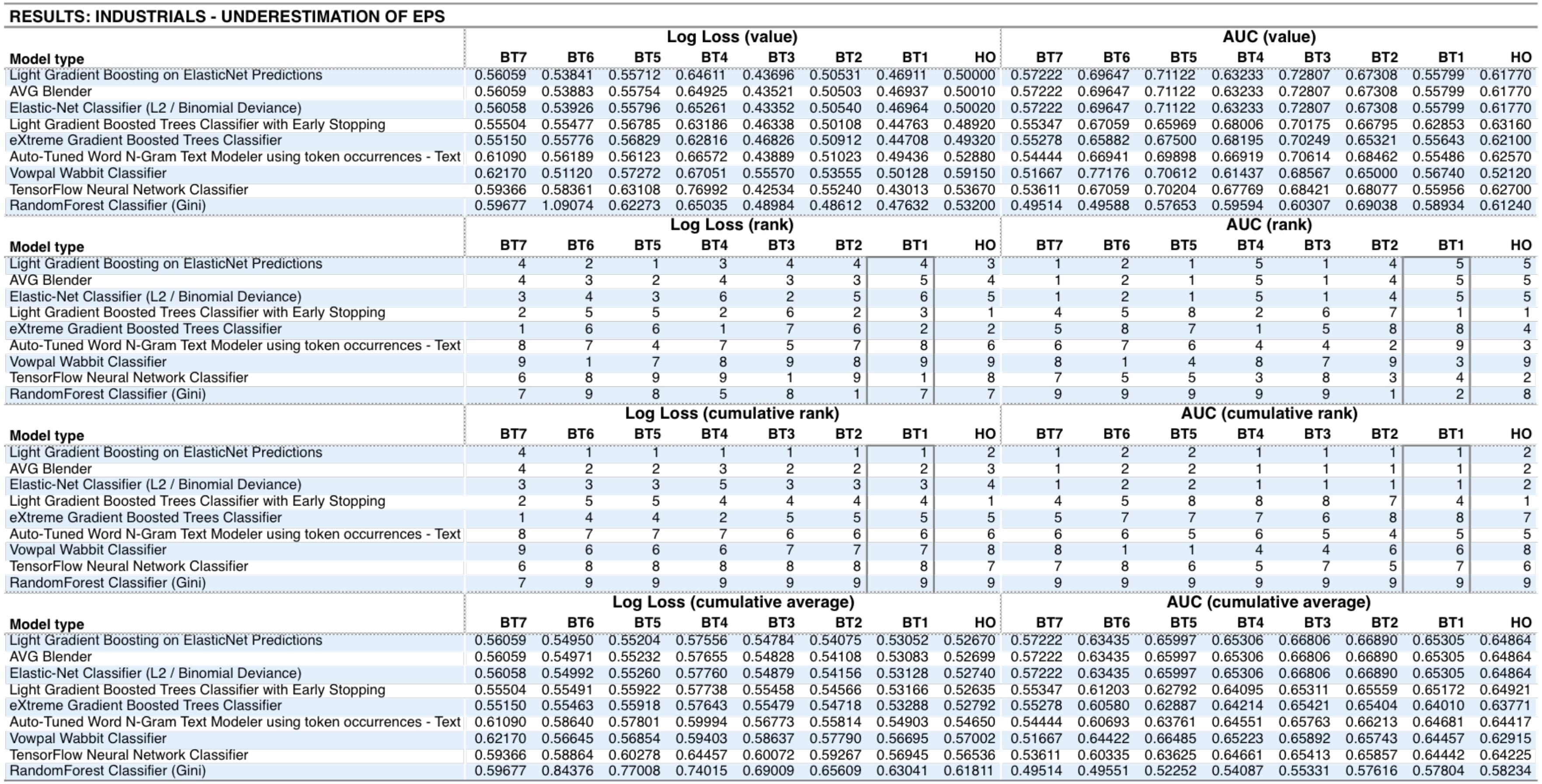
PERFORMANCE METRICS	
Name	Value
Accuracy	34.78%
Precision	18.52%
Recall (sensitivity, TP rate)	90.91%
Specificity (TN rate)	24.14%

Appendix Table XLVI: Confusion matrices – Holdout and validation



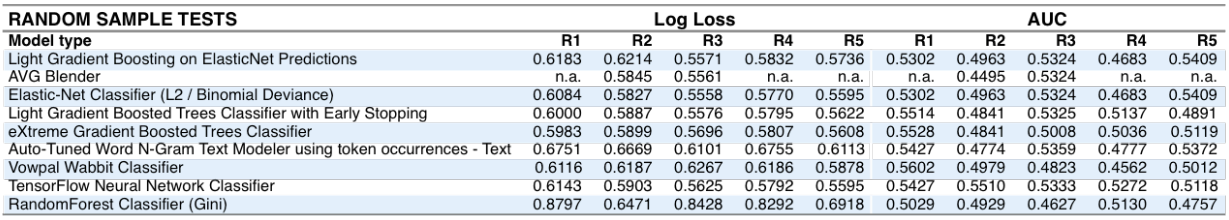
Appendix Table XLVII: Lift chart

Appendix XXII. Industrials (Underestimation of EPS)



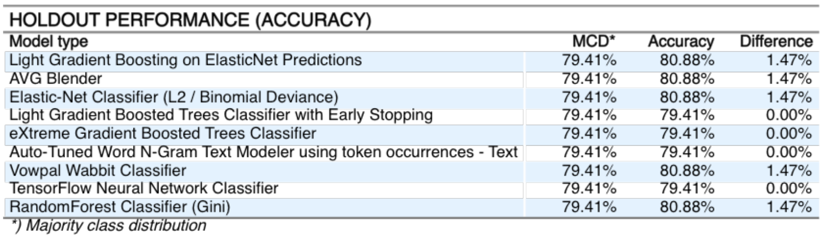
RESULTS: INDUSTRIALS - UNDERESTIMATION OF EPS																
	Log Loss (value)								AUC (value)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
Light Gradient Boosting on ElasticNet Predictions	0.56059	0.53841	0.55712	0.64611	0.43696	0.50531	0.46911	0.50000	0.57222	0.69647	0.71122	0.63233	0.72807	0.67308	0.55799	0.61770
AVG Blender	0.56059	0.53883	0.55754	0.64925	0.43521	0.50503	0.46937	0.50010	0.57222	0.69647	0.71122	0.63233	0.72807	0.67308	0.55799	0.61770
Elastic-Net Classifier (L2 / Binomial Deviance)	0.56058	0.53926	0.55796	0.65261	0.43352	0.50540	0.46964	0.50020	0.57222	0.69647	0.71122	0.63233	0.72807	0.67308	0.55799	0.61770
Light Gradient Boosted Trees Classifier with Early Stopping	0.55504	0.55477	0.56785	0.63186	0.46338	0.50108	0.44763	0.48920	0.55347	0.67059	0.65969	0.68006	0.70175	0.66795	0.62853	0.63160
eXtreme Gradient Boosted Trees Classifier	0.55150	0.55776	0.56829	0.62816	0.46826	0.50912	0.44708	0.49320	0.55278	0.65882	0.67500	0.68195	0.70249	0.65321	0.55643	0.62100
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.61090	0.56189	0.56123	0.66572	0.43889	0.51023	0.49436	0.52880	0.54444	0.66941	0.69898	0.66919	0.70614	0.68462	0.55486	0.62570
Vowpal Wabbit Classifier	0.62170	0.51120	0.57272	0.67051	0.55570	0.53555	0.50128	0.59150	0.51667	0.77176	0.70612	0.61437	0.68567	0.65000	0.56740	0.52120
TensorFlow Neural Network Classifier	0.59366	0.58361	0.63108	0.76992	0.42534	0.55240	0.43013	0.53670	0.53611	0.67059	0.70204	0.67769	0.68421	0.68077	0.55956	0.62700
RandomForest Classifier (Gini)	0.59677	1.09074	0.62273	0.65035	0.48984	0.48612	0.47632	0.53200	0.49514	0.49588	0.57653	0.59594	0.60307	0.69038	0.58934	0.61240
	Log Loss (rank)								AUC (rank)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
Light Gradient Boosting on ElasticNet Predictions	4	2	1	3	4	4	4	3	1	2	1	5	1	4	5	5
AVG Blender	4	3	2	4	3	3	5	4	1	2	1	5	1	4	5	5
Elastic-Net Classifier (L2 / Binomial Deviance)	3	4	3	6	2	5	6	5	1	2	1	5	1	4	5	5
Light Gradient Boosted Trees Classifier with Early Stopping	2	5	5	2	6	2	3	1	4	5	8	2	6	7	1	1
eXtreme Gradient Boosted Trees Classifier	1	6	6	1	7	6	2	2	5	8	7	1	5	8	8	4
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	8	7	4	7	5	7	8	6	6	7	6	4	4	2	9	3
Vowpal Wabbit Classifier	9	1	7	8	9	8	9	9	8	1	4	8	7	9	3	9
TensorFlow Neural Network Classifier	6	8	9	9	1	9	1	8	7	5	5	3	8	3	4	2
RandomForest Classifier (Gini)	7	9	8	5	8	1	7	7	9	9	9	9	9	1	2	8
	Log Loss (cumulative rank)								AUC (cumulative rank)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
Light Gradient Boosting on ElasticNet Predictions	4	1	1	1	1	1	1	2	1	2	2	1	1	1	1	2
AVG Blender	4	2	2	3	2	2	2	3	1	2	2	1	1	1	1	2
Elastic-Net Classifier (L2 / Binomial Deviance)	3	3	3	5	3	3	3	4	1	2	2	1	1	1	1	2
Light Gradient Boosted Trees Classifier with Early Stopping	2	5	5	4	4	4	4	1	4	5	8	8	8	7	4	1
eXtreme Gradient Boosted Trees Classifier	1	4	4	2	5	5	5	5	5	7	7	7	6	8	8	7
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	8	7	7	7	6	6	6	6	6	6	5	6	5	4	5	5
Vowpal Wabbit Classifier	9	6	6	6	7	7	7	8	8	1	1	4	4	6	6	8
TensorFlow Neural Network Classifier	6	8	8	8	8	8	8	7	7	8	6	5	7	5	7	6
RandomForest Classifier (Gini)	7	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
	Log Loss (cumulative average)								AUC (cumulative average)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
Light Gradient Boosting on ElasticNet Predictions	0.56059	0.54950	0.55204	0.57556	0.54784	0.54075	0.53052	0.52670	0.57222	0.63435	0.65997	0.65306	0.66806	0.66890	0.65305	0.64864
AVG Blender	0.56059	0.54971	0.55232	0.57655	0.54828	0.54108	0.53083	0.52699	0.57222	0.63435	0.65997	0.65306	0.66806	0.66890	0.65305	0.64864
Elastic-Net Classifier (L2 / Binomial Deviance)	0.56058	0.54992	0.55260	0.57760	0.54879	0.54156	0.53128	0.52740	0.57222	0.63435	0.65997	0.65306	0.66806	0.66890	0.65305	0.64864
Light Gradient Boosted Trees Classifier with Early Stopping	0.55504	0.55491	0.55922	0.57738	0.55458	0.54566	0.53166	0.52635	0.55347	0.61203	0.62792	0.64095	0.65311	0.65559	0.65172	0.64921
eXtreme Gradient Boosted Trees Classifier	0.55150	0.55463	0.55918	0.57643	0.55479	0.54718	0.53288	0.52792	0.55278	0.60580	0.62887	0.64214	0.65421	0.65404	0.64010	0.63771
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.61090	0.58640	0.57801	0.59994	0.56773	0.55814	0.54903	0.54650	0.54444	0.60693	0.63761	0.64551	0.65763	0.66213	0.64681	0.64417
Vowpal Wabbit Classifier	0.62170	0.56645	0.56854	0.59403	0.58637	0.57790	0.56695	0.57002	0.51667	0.64422	0.66485	0.65223	0.65892	0.65743	0.64457	0.62915
TensorFlow Neural Network Classifier	0.59366	0.58864	0.60278	0.64457	0.60072	0.59267	0.56945	0.56536	0.53611	0.60335	0.63625	0.64661	0.65413	0.65857	0.64442	0.64225
RandomForest Classifier (Gini)	0.59677	0.84376	0.77008	0.74015	0.69009	0.65609	0.63041	0.61811	0.49514	0.49551	0.52252	0.54087	0.55331	0.57616	0.57804	0.58234

Appendix Table XLVIII: Log Loss and AUC (Values and rank)



RANDOM SAMPLE TESTS	Log Loss					AUC				
Model type	R1	R2	R3	R4	R5	R1	R2	R3	R4	R5
Light Gradient Boosting on ElasticNet Predictions	0.6183	0.6214	0.5571	0.5832	0.5736	0.5302	0.4963	0.5324	0.4683	0.5409
AVG Blender	n.a.	0.5845	0.5561	n.a.	n.a.	n.a.	0.4495	0.5324	n.a.	n.a.
Elastic-Net Classifier (L2 / Binomial Deviance)	0.6084	0.5827	0.5558	0.5770	0.5595	0.5302	0.4963	0.5324	0.4683	0.5409
Light Gradient Boosted Trees Classifier with Early Stopping	0.6000	0.5887	0.5576	0.5795	0.5622	0.5514	0.4841	0.5325	0.5137	0.4891
eXtreme Gradient Boosted Trees Classifier	0.5983	0.5899	0.5696	0.5807	0.5608	0.5528	0.4841	0.5008	0.5036	0.5119
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.6751	0.6669	0.6101	0.6755	0.6113	0.5427	0.4774	0.5359	0.4777	0.5372
Vowpal Wabbit Classifier	0.6116	0.6187	0.6267	0.6186	0.5878	0.5602	0.4979	0.4823	0.4562	0.5012
TensorFlow Neural Network Classifier	0.6143	0.5903	0.5625	0.5792	0.5595	0.5427	0.5510	0.5333	0.5272	0.5118
RandomForest Classifier (Gini)	0.8797	0.6471	0.8428	0.8292	0.6918	0.5029	0.4929	0.4627	0.5130	0.4757

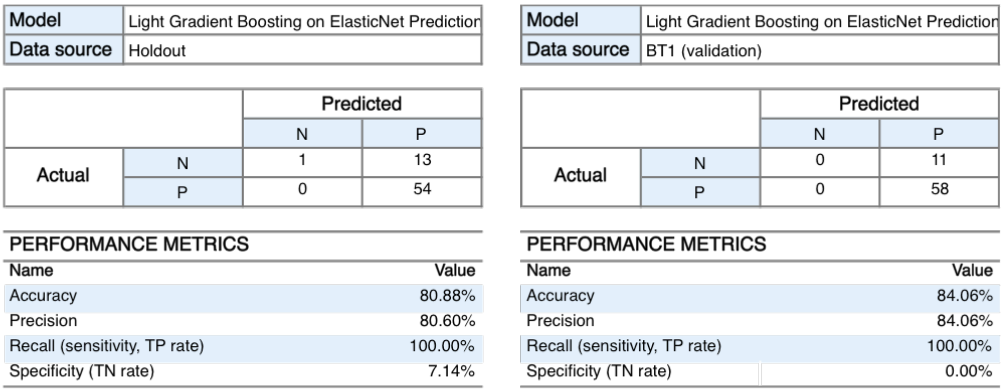
Appendix Table XLIX: Random sample test



HOLDOUT PERFORMANCE (ACCURACY)			
Model type	MCD*	Accuracy	Difference
Light Gradient Boosting on ElasticNet Predictions	79.41%	80.88%	1.47%
AVG Blender	79.41%	80.88%	1.47%
Elastic-Net Classifier (L2 / Binomial Deviance)	79.41%	80.88%	1.47%
Light Gradient Boosted Trees Classifier with Early Stopping	79.41%	79.41%	0.00%
eXtreme Gradient Boosted Trees Classifier	79.41%	79.41%	0.00%
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	79.41%	79.41%	0.00%
Vowpal Wabbit Classifier	79.41%	80.88%	1.47%
TensorFlow Neural Network Classifier	79.41%	79.41%	0.00%
RandomForest Classifier (Gini)	79.41%	80.88%	1.47%

*) Majority class distribution

Appendix Table L: Holdout performance



Model	Light Gradient Boosting on ElasticNet Predictior
Data source	Holdout

		Predicted	
		N	P
Actual	N	1	13
	P	0	54

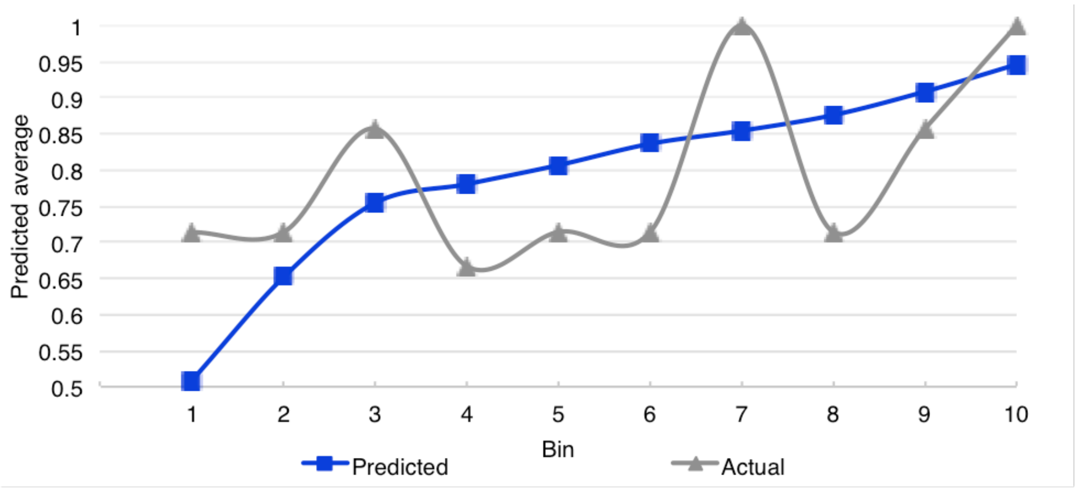
PERFORMANCE METRICS	
Name	Value
Accuracy	80.88%
Precision	80.60%
Recall (sensitivity, TP rate)	100.00%
Specificity (TN rate)	7.14%

Model	Light Gradient Boosting on ElasticNet Predictior
Data source	BT1 (validation)

		Predicted	
		N	P
Actual	N	0	11
	P	0	58

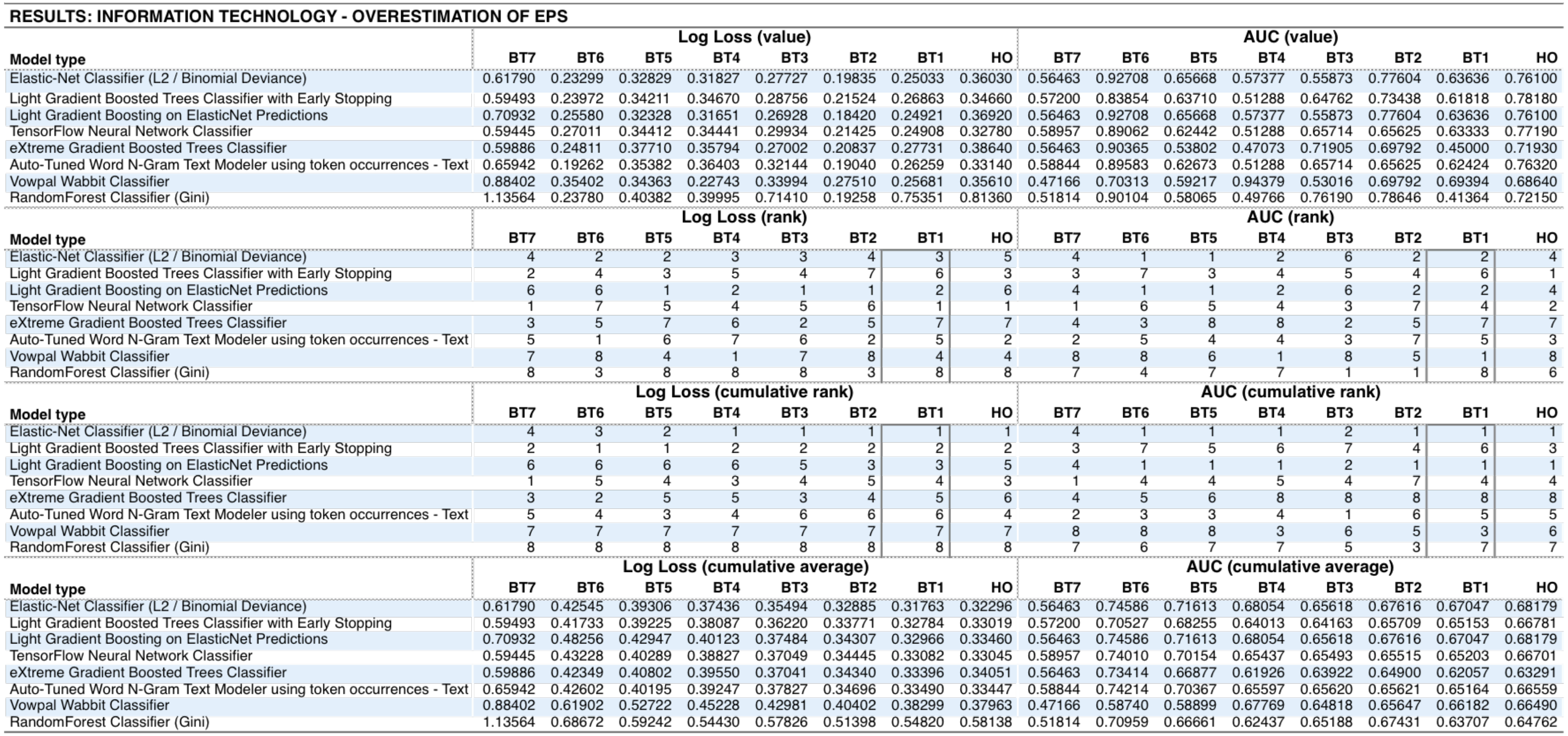
PERFORMANCE METRICS	
Name	Value
Accuracy	84.06%
Precision	84.06%
Recall (sensitivity, TP rate)	100.00%
Specificity (TN rate)	0.00%

Appendix Table LI: Confusion matrices – Holdout and validation



Appendix Table LII: Lift chart

Appendix XXIII. Information Technology (Overestimation of EPS)



RESULTS: INFORMATION TECHNOLOGY - OVERESTIMATION OF EPS

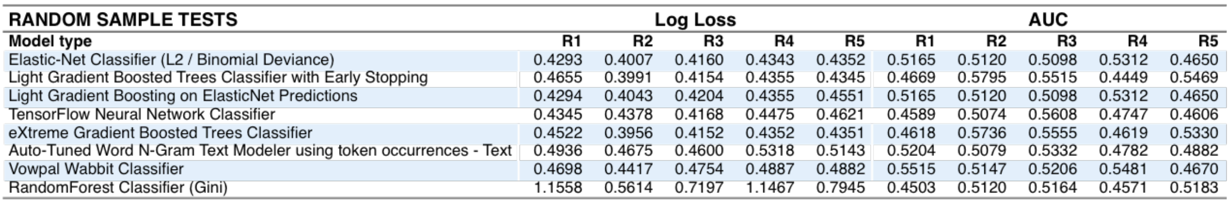
	Log Loss (value)								AUC (value)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
Elastic-Net Classifier (L2 / Binomial Deviance)	0.61790	0.23299	0.32829	0.31827	0.27727	0.19835	0.25033	0.36030	0.56463	0.92708	0.65668	0.57377	0.55873	0.77604	0.63636	0.76100
Light Gradient Boosted Trees Classifier with Early Stopping	0.59493	0.23972	0.34211	0.34670	0.28756	0.21524	0.26863	0.34660	0.57200	0.83854	0.63710	0.51288	0.64762	0.73438	0.61818	0.78180
Light Gradient Boosting on ElasticNet Predictions	0.70932	0.25580	0.32328	0.31651	0.26928	0.18420	0.24921	0.36920	0.56463	0.92708	0.65668	0.57377	0.55873	0.77604	0.63636	0.76100
TensorFlow Neural Network Classifier	0.59445	0.27011	0.34412	0.34441	0.29934	0.21425	0.24908	0.32780	0.58957	0.89062	0.62442	0.51288	0.65714	0.65625	0.63333	0.77190
eXtreme Gradient Boosted Trees Classifier	0.59886	0.24811	0.37710	0.35794	0.27002	0.20837	0.27731	0.38640	0.56463	0.90365	0.53802	0.47073	0.71905	0.69792	0.45000	0.71930
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.65942	0.19262	0.35382	0.36403	0.32144	0.19040	0.26259	0.33140	0.58844	0.89583	0.62673	0.51288	0.65714	0.65625	0.62424	0.76320
Vowpal Wabbit Classifier	0.88402	0.35402	0.34363	0.22743	0.33994	0.27510	0.25681	0.35610	0.47166	0.70313	0.59217	0.94379	0.53016	0.69792	0.69394	0.68640
RandomForest Classifier (Gini)	1.13564	0.23780	0.40382	0.39995	0.71410	0.19258	0.75351	0.81360	0.51814	0.90104	0.58065	0.49766	0.76190	0.78646	0.41364	0.72150

	Log Loss (rank)								AUC (rank)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
Elastic-Net Classifier (L2 / Binomial Deviance)	4	2	2	3	3	4	3	5	4	1	1	2	6	2	2	4
Light Gradient Boosted Trees Classifier with Early Stopping	2	4	3	5	4	7	6	3	3	7	3	4	5	4	6	1
Light Gradient Boosting on ElasticNet Predictions	6	6	1	2	1	1	2	6	4	1	1	2	6	2	2	4
TensorFlow Neural Network Classifier	1	7	5	4	5	6	1	1	1	6	5	4	3	7	4	2
eXtreme Gradient Boosted Trees Classifier	3	5	7	6	2	5	7	7	4	3	8	8	2	5	7	7
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	5	1	6	7	6	2	5	2	2	5	4	4	3	7	5	3
Vowpal Wabbit Classifier	7	8	4	1	7	8	4	4	8	8	6	1	8	5	1	8
RandomForest Classifier (Gini)	8	3	8	8	8	3	8	8	7	4	7	7	1	1	8	6

	Log Loss (cumulative rank)								AUC (cumulative rank)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
Elastic-Net Classifier (L2 / Binomial Deviance)	4	3	2	1	1	1	1	1	4	1	1	1	2	1	1	1
Light Gradient Boosted Trees Classifier with Early Stopping	2	1	1	2	2	2	2	2	3	7	5	6	7	4	6	3
Light Gradient Boosting on ElasticNet Predictions	6	6	6	6	5	3	3	5	4	1	1	1	2	1	1	1
TensorFlow Neural Network Classifier	1	5	4	3	4	5	4	3	1	4	4	5	4	7	4	4
eXtreme Gradient Boosted Trees Classifier	3	2	5	5	3	4	5	6	4	5	6	8	8	8	8	8
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	5	4	3	4	6	6	6	4	2	3	3	4	1	6	5	5
Vowpal Wabbit Classifier	7	7	7	7	7	7	7	7	8	8	8	3	6	5	3	6
RandomForest Classifier (Gini)	8	8	8	8	8	8	8	8	7	6	7	7	5	3	7	7

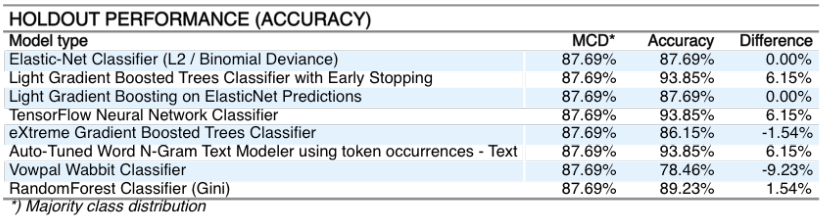
	Log Loss (cumulative average)								AUC (cumulative average)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
Elastic-Net Classifier (L2 / Binomial Deviance)	0.61790	0.42545	0.39306	0.37436	0.35494	0.32885	0.31763	0.32296	0.56463	0.74586	0.71613	0.68054	0.65618	0.67616	0.67047	0.68179
Light Gradient Boosted Trees Classifier with Early Stopping	0.59493	0.41733	0.39225	0.38087	0.36220	0.33771	0.32784	0.33019	0.57200	0.70527	0.68255	0.64013	0.64163	0.65709	0.65153	0.66781
Light Gradient Boosting on ElasticNet Predictions	0.70932	0.48256	0.42947	0.40123	0.37484	0.34307	0.32966	0.33460	0.56463	0.74586	0.71613	0.68054	0.65618	0.67616	0.67047	0.68179
TensorFlow Neural Network Classifier	0.59445	0.43228	0.40289	0.38827	0.37049	0.34445	0.33082	0.33045	0.58957	0.74010	0.70154	0.65437	0.65493	0.65515	0.65203	0.66701
eXtreme Gradient Boosted Trees Classifier	0.59886	0.42349	0.40802	0.39550	0.37041	0.34340	0.33396	0.34051	0.56463	0.73414	0.66877	0.61926	0.63922	0.64900	0.62057	0.63291
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.65942	0.42602	0.40195	0.39247	0.37827	0.34696	0.33490	0.33447	0.58844	0.74214	0.70367	0.65597	0.65620	0.65621	0.65164	0.66559
Vowpal Wabbit Classifier	0.88402	0.61902	0.52722	0.45228	0.42981	0.40402	0.38299	0.37963	0.47166	0.58740	0.58899	0.67769	0.64818	0.65647	0.66182	0.66490
RandomForest Classifier (Gini)	1.13564	0.68672	0.59242	0.54430	0.57826	0.51398	0.54820	0.58138	0.51814	0.70959	0.66661	0.62437	0.65188	0.67431	0.63707	0.64762

Appendix Table LIII: Log Loss and AUC (Values and rank)



RANDOM SAMPLE TESTS	Log Loss					AUC				
Model type	R1	R2	R3	R4	R5	R1	R2	R3	R4	R5
Elastic-Net Classifier (L2 / Binomial Deviance)	0.4293	0.4007	0.4160	0.4343	0.4352	0.5165	0.5120	0.5098	0.5312	0.4650
Light Gradient Boosted Trees Classifier with Early Stopping	0.4655	0.3991	0.4154	0.4355	0.4345	0.4669	0.5795	0.5515	0.4449	0.5469
Light Gradient Boosting on ElasticNet Predictions	0.4294	0.4043	0.4204	0.4355	0.4551	0.5165	0.5120	0.5098	0.5312	0.4650
TensorFlow Neural Network Classifier	0.4345	0.4378	0.4168	0.4475	0.4621	0.4589	0.5074	0.5608	0.4747	0.4606
eXtreme Gradient Boosted Trees Classifier	0.4522	0.3956	0.4152	0.4352	0.4351	0.4618	0.5736	0.5555	0.4619	0.5330
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.4936	0.4675	0.4600	0.5318	0.5143	0.5204	0.5079	0.5332	0.4782	0.4882
Vowpal Wabbit Classifier	0.4698	0.4417	0.4754	0.4887	0.4882	0.5515	0.5147	0.5206	0.5481	0.4670
RandomForest Classifier (Gini)	1.1558	0.5614	0.7197	1.1467	0.7945	0.4503	0.5120	0.5164	0.4571	0.5183

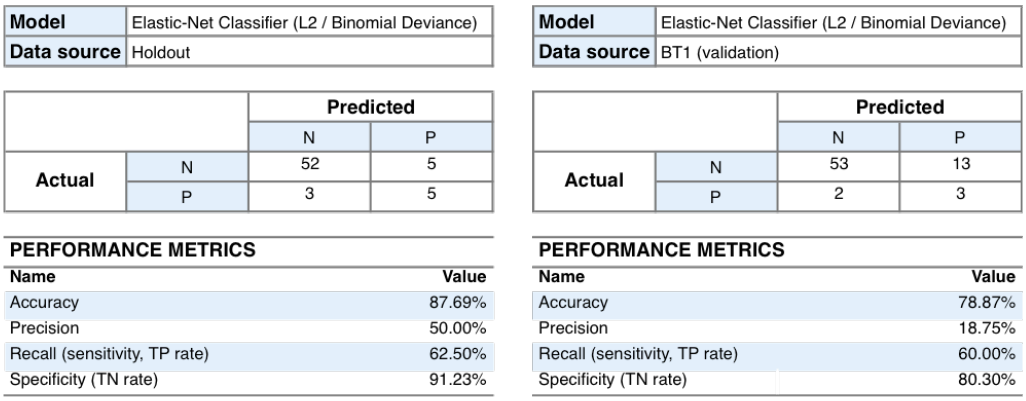
Appendix Table LIV: Random sample test



HOLDOUT PERFORMANCE (ACCURACY)			
Model type	MCD*	Accuracy	Difference
Elastic-Net Classifier (L2 / Binomial Deviance)	87.69%	87.69%	0.00%
Light Gradient Boosted Trees Classifier with Early Stopping	87.69%	93.85%	6.15%
Light Gradient Boosting on ElasticNet Predictions	87.69%	87.69%	0.00%
TensorFlow Neural Network Classifier	87.69%	93.85%	6.15%
eXtreme Gradient Boosted Trees Classifier	87.69%	86.15%	-1.54%
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	87.69%	93.85%	6.15%
Vowpal Wabbit Classifier	87.69%	78.46%	-9.23%
RandomForest Classifier (Gini)	87.69%	89.23%	1.54%

*) Majority class distribution

Appendix Table LV: Holdout performance



Model	Elastic-Net Classifier (L2 / Binomial Deviance)
Data source	Holdout

		Predicted	
		N	P
Actual	N	52	5
	P	3	5

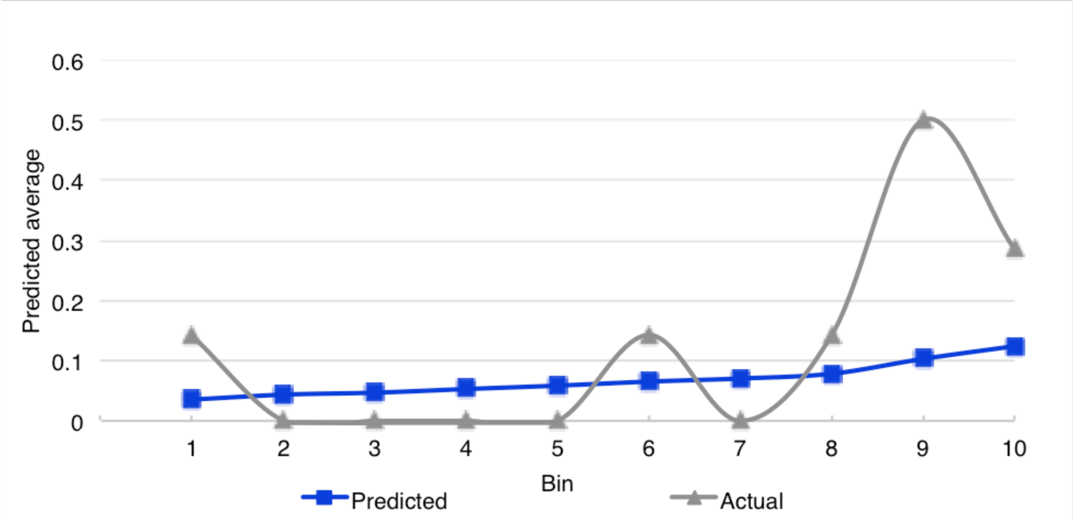
PERFORMANCE METRICS	
Name	Value
Accuracy	87.69%
Precision	50.00%
Recall (sensitivity, TP rate)	62.50%
Specificity (TN rate)	91.23%

Model	Elastic-Net Classifier (L2 / Binomial Deviance)
Data source	BT1 (validation)

		Predicted	
		N	P
Actual	N	53	13
	P	2	3

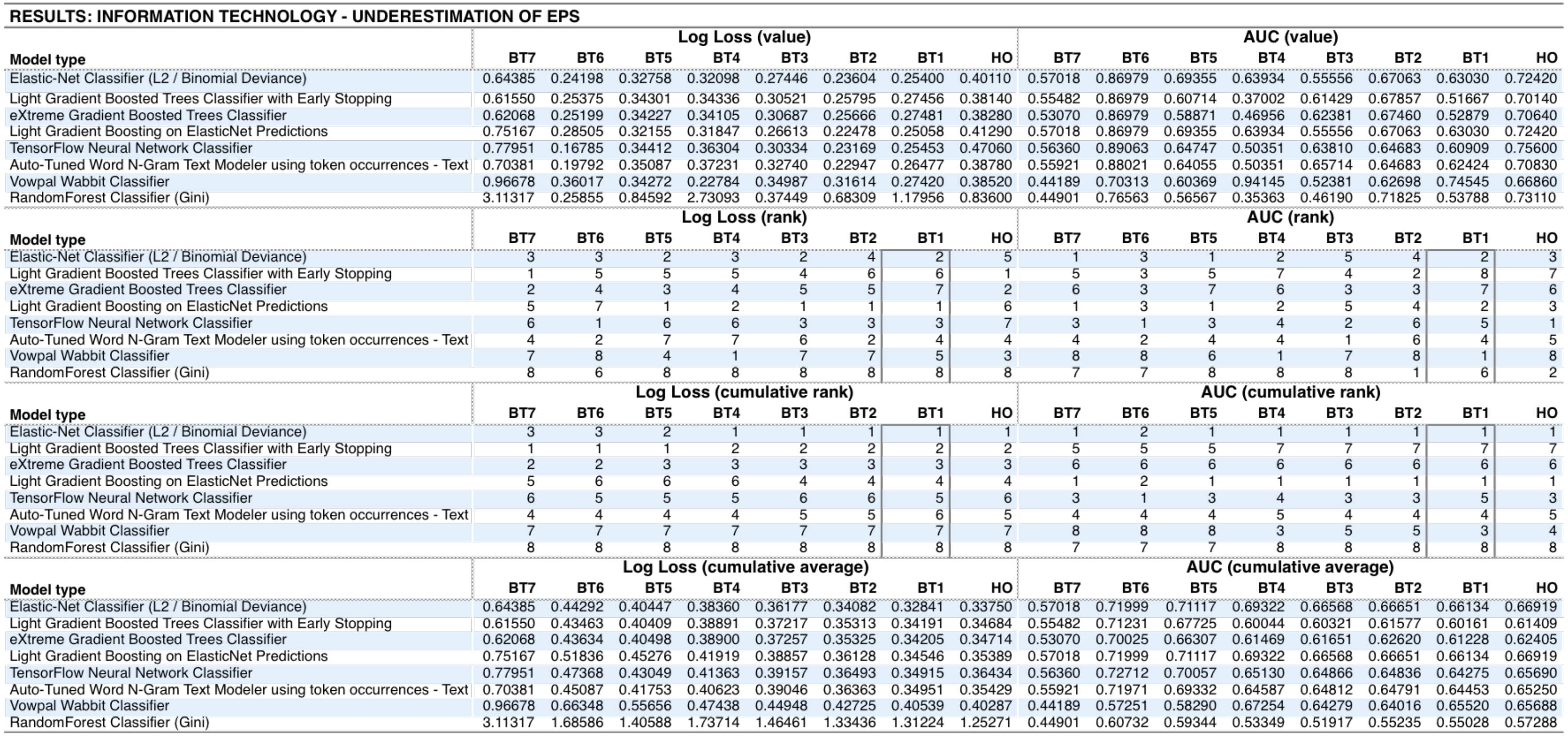
PERFORMANCE METRICS	
Name	Value
Accuracy	78.87%
Precision	18.75%
Recall (sensitivity, TP rate)	60.00%
Specificity (TN rate)	80.30%

Appendix Table LVI: Confusion matrices – Holdout and validation



Appendix Table LVII: Lift chart

Appendix XXIV. Information Technology (Underestimation of EPS)



RESULTS: INFORMATION TECHNOLOGY - UNDERESTIMATION OF EPS

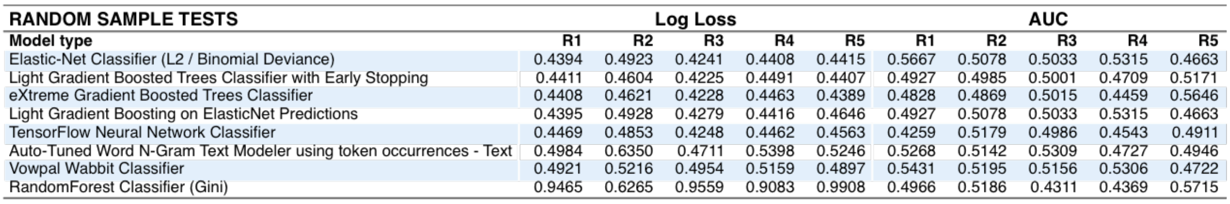
	Log Loss (value)								AUC (value)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
Elastic-Net Classifier (L2 / Binomial Deviance)	0.64385	0.24198	0.32758	0.32098	0.27446	0.23604	0.25400	0.40110	0.57018	0.86979	0.69355	0.63934	0.55556	0.67063	0.63030	0.72420
Light Gradient Boosted Trees Classifier with Early Stopping	0.61550	0.25375	0.34301	0.34336	0.30521	0.25795	0.27456	0.38140	0.55482	0.86979	0.60714	0.37002	0.61429	0.67857	0.51667	0.70140
eXtreme Gradient Boosted Trees Classifier	0.62068	0.25199	0.34227	0.34105	0.30687	0.25666	0.27481	0.38280	0.53070	0.86979	0.58871	0.46956	0.62381	0.67460	0.52879	0.70640
Light Gradient Boosting on ElasticNet Predictions	0.75167	0.28505	0.32155	0.31847	0.26613	0.22478	0.25058	0.41290	0.57018	0.86979	0.69355	0.63934	0.55556	0.67063	0.63030	0.72420
TensorFlow Neural Network Classifier	0.77951	0.16785	0.34412	0.36304	0.30334	0.23169	0.25453	0.47060	0.56360	0.89063	0.64747	0.50351	0.63810	0.64683	0.60909	0.75600
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.70381	0.19792	0.35087	0.37231	0.32740	0.22947	0.26477	0.38780	0.55921	0.88021	0.64055	0.50351	0.65714	0.64683	0.62424	0.70830
Vowpal Wabbit Classifier	0.96678	0.36017	0.34272	0.22784	0.34987	0.31614	0.27420	0.38520	0.44189	0.70313	0.60369	0.94145	0.52381	0.62698	0.74545	0.66860
RandomForest Classifier (Gini)	3.11317	0.25855	0.84592	2.73093	0.37449	0.68309	1.17956	0.83600	0.44901	0.76563	0.56567	0.35363	0.46190	0.71825	0.53788	0.73110

	Log Loss (rank)								AUC (rank)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
Elastic-Net Classifier (L2 / Binomial Deviance)	3	3	2	3	2	4	2	5	1	3	1	2	5	4	2	3
Light Gradient Boosted Trees Classifier with Early Stopping	1	5	5	5	4	6	6	1	5	3	5	7	4	2	8	7
eXtreme Gradient Boosted Trees Classifier	2	4	3	4	5	5	7	2	6	3	7	6	3	3	7	6
Light Gradient Boosting on ElasticNet Predictions	5	7	1	2	1	1	1	6	1	3	1	2	5	4	2	3
TensorFlow Neural Network Classifier	6	1	6	6	3	3	3	7	3	1	3	4	2	6	5	1
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	4	2	7	7	6	2	4	4	4	2	4	4	1	6	4	5
Vowpal Wabbit Classifier	7	8	4	1	7	7	5	3	8	8	6	1	7	8	1	8
RandomForest Classifier (Gini)	8	6	8	8	8	8	8	8	7	7	8	8	8	1	6	2

	Log Loss (cumulative rank)								AUC (cumulative rank)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
Elastic-Net Classifier (L2 / Binomial Deviance)	3	3	2	1	1	1	1	1	1	2	1	1	1	1	1	1
Light Gradient Boosted Trees Classifier with Early Stopping	1	1	1	2	2	2	2	2	5	5	5	7	7	7	7	7
eXtreme Gradient Boosted Trees Classifier	2	2	3	3	3	3	3	3	6	6	6	6	6	6	6	6
Light Gradient Boosting on ElasticNet Predictions	5	6	6	6	4	4	4	4	1	2	1	1	1	1	1	1
TensorFlow Neural Network Classifier	6	5	5	5	6	6	5	6	3	1	3	4	3	3	5	3
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	4	4	4	4	5	5	6	5	4	4	4	5	4	4	4	5
Vowpal Wabbit Classifier	7	7	7	7	7	7	7	7	8	8	8	3	5	5	3	4
RandomForest Classifier (Gini)	8	8	8	8	8	8	8	8	7	7	7	8	8	8	8	8

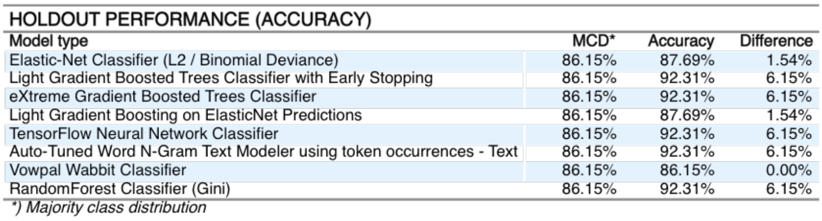
	Log Loss (cumulative average)								AUC (cumulative average)							
Model type	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**	**BT7**	**BT6**	**BT5**	**BT4**	**BT3**	**BT2**	**BT1**	**HO**
Elastic-Net Classifier (L2 / Binomial Deviance)	0.64385	0.44292	0.40447	0.38360	0.36177	0.34082	0.32841	0.33750	0.57018	0.71999	0.71117	0.69322	0.66568	0.66651	0.66134	0.66919
Light Gradient Boosted Trees Classifier with Early Stopping	0.61550	0.43463	0.40409	0.38891	0.37217	0.35313	0.34191	0.34684	0.55482	0.71231	0.67725	0.60044	0.60321	0.61577	0.60161	0.61409
eXtreme Gradient Boosted Trees Classifier	0.62068	0.43634	0.40498	0.38900	0.37257	0.35325	0.34205	0.34714	0.53070	0.70025	0.66307	0.61469	0.61651	0.62620	0.61228	0.62405
Light Gradient Boosting on ElasticNet Predictions	0.75167	0.51836	0.45276	0.41919	0.38857	0.36128	0.34546	0.35389	0.57018	0.71999	0.71117	0.69322	0.66568	0.66651	0.66134	0.66919
TensorFlow Neural Network Classifier	0.77951	0.47368	0.43049	0.41363	0.39157	0.36493	0.34915	0.36434	0.56360	0.72712	0.70057	0.65130	0.64866	0.64836	0.64275	0.65690
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.70381	0.45087	0.41753	0.40623	0.39046	0.36363	0.34951	0.35429	0.55921	0.71971	0.69332	0.64587	0.64812	0.64791	0.64453	0.65250
Vowpal Wabbit Classifier	0.96678	0.66348	0.55656	0.47438	0.44948	0.42725	0.40539	0.40287	0.44189	0.57251	0.58290	0.67254	0.64279	0.64016	0.65520	0.65688
RandomForest Classifier (Gini)	3.11317	1.68586	1.40588	1.73714	1.46461	1.33436	1.31224	1.25271	0.44901	0.60732	0.59344	0.53349	0.51917	0.55235	0.55028	0.57288

Appendix Table LVIII: Log Loss and AUC (Values and rank)



RANDOM SAMPLE TESTS	Log Loss					AUC				
Model type	**R1**	**R2**	**R3**	**R4**	**R5**	**R1**	**R2**	**R3**	**R4**	**R5**
Elastic-Net Classifier (L2 / Binomial Deviance)	0.4394	0.4923	0.4241	0.4408	0.4415	0.5667	0.5078	0.5033	0.5315	0.4663
Light Gradient Boosted Trees Classifier with Early Stopping	0.4411	0.4604	0.4225	0.4491	0.4407	0.4927	0.4985	0.5001	0.4709	0.5171
eXtreme Gradient Boosted Trees Classifier	0.4408	0.4621	0.4228	0.4463	0.4389	0.4828	0.4869	0.5015	0.4459	0.5646
Light Gradient Boosting on ElasticNet Predictions	0.4395	0.4928	0.4279	0.4416	0.4646	0.4927	0.5078	0.5033	0.5315	0.4663
TensorFlow Neural Network Classifier	0.4469	0.4853	0.4248	0.4462	0.4563	0.4259	0.5179	0.4986	0.4543	0.4911
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.4984	0.6350	0.4711	0.5398	0.5246	0.5268	0.5142	0.5309	0.4727	0.4946
Vowpal Wabbit Classifier	0.4921	0.5216	0.4954	0.5159	0.4897	0.5431	0.5195	0.5156	0.5306	0.4722
RandomForest Classifier (Gini)	0.9465	0.6265	0.9559	0.9083	0.9908	0.4966	0.5186	0.4311	0.4369	0.5715

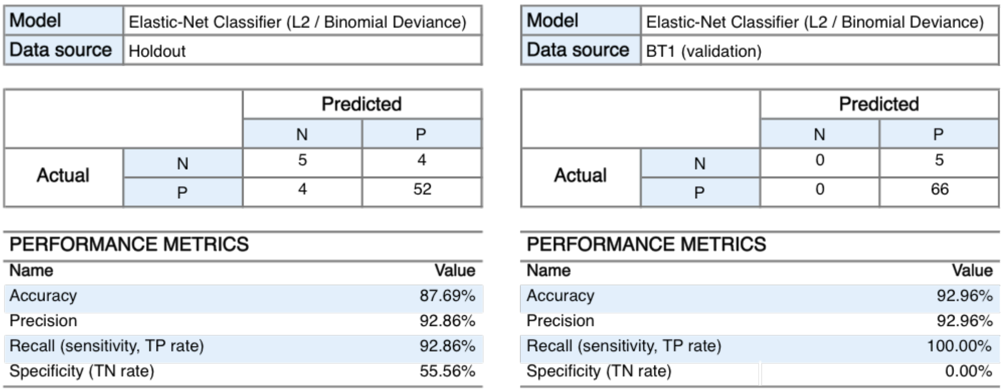
Appendix Table LIX: Random sample test



HOLDOUT PERFORMANCE (ACCURACY)			
Model type	**MCD***	**Accuracy**	**Difference**
Elastic-Net Classifier (L2 / Binomial Deviance)	86.15%	87.69%	1.54%
Light Gradient Boosted Trees Classifier with Early Stopping	86.15%	92.31%	6.15%
eXtreme Gradient Boosted Trees Classifier	86.15%	92.31%	6.15%
Light Gradient Boosting on ElasticNet Predictions	86.15%	87.69%	1.54%
TensorFlow Neural Network Classifier	86.15%	92.31%	6.15%
Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	86.15%	92.31%	6.15%
Vowpal Wabbit Classifier	86.15%	86.15%	0.00%
RandomForest Classifier (Gini)	86.15%	92.31%	6.15%

**) Majority class distribution*

Appendix Table LX: Holdout performance



Model	Elastic-Net Classifier (L2 / Binomial Deviance)
Data source	Holdout

		Predicted	
		N	P
Actual	N	5	4
	P	4	52

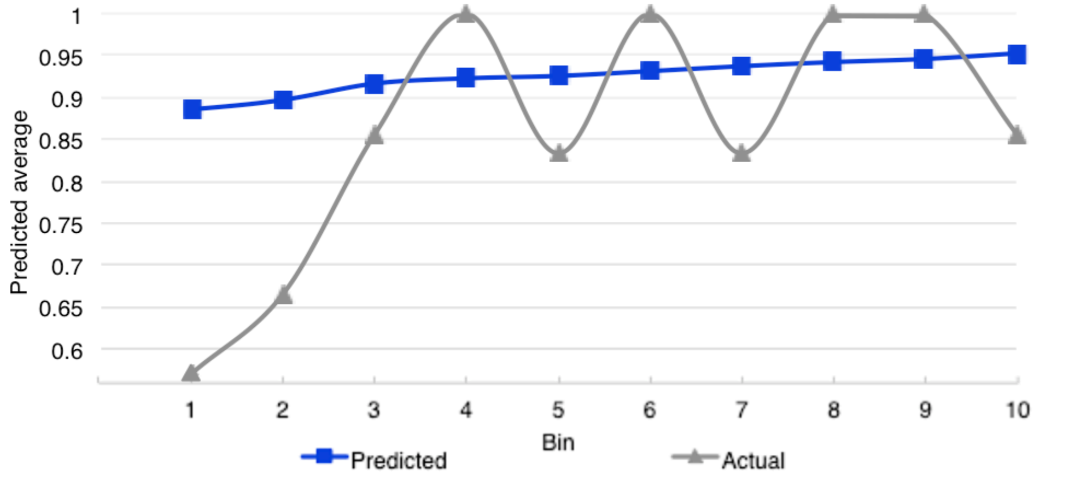
PERFORMANCE METRICS	
Name	**Value**
Accuracy	87.69%
Precision	92.86%
Recall (sensitivity, TP rate)	92.86%
Specificity (TN rate)	55.56%

Model	Elastic-Net Classifier (L2 / Binomial Deviance)
Data source	BT1 (validation)

		Predicted	
		N	P
Actual	N	0	5
	P	0	66

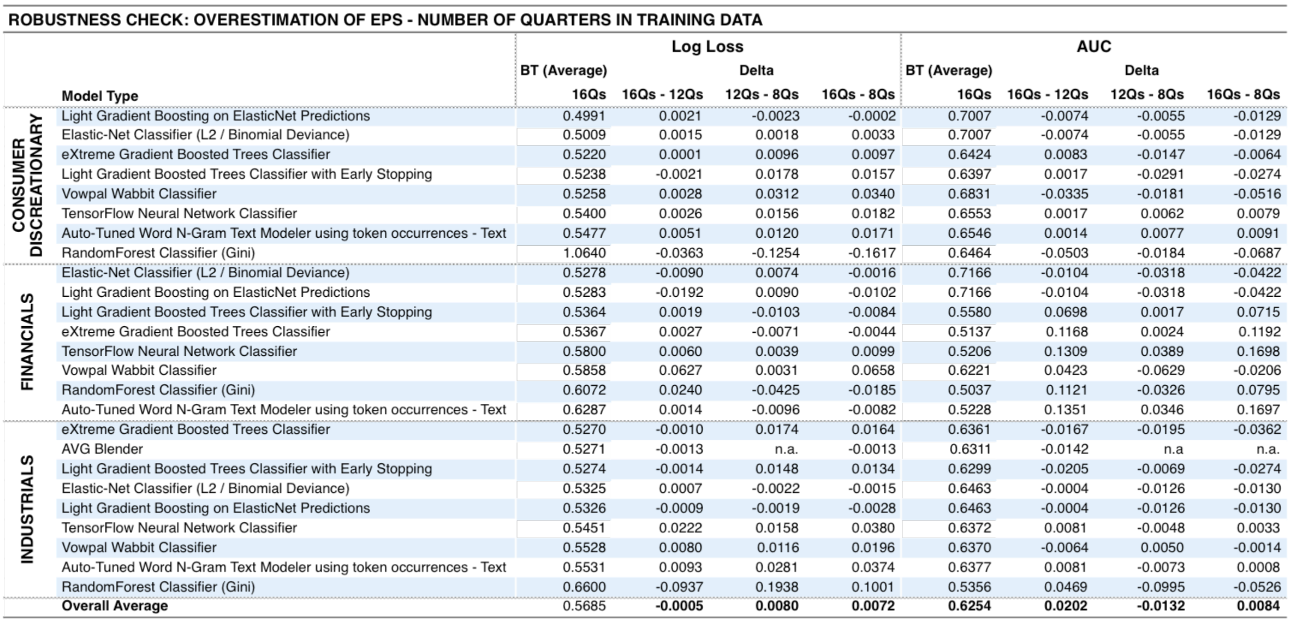
PERFORMANCE METRICS	
Name	**Value**
Accuracy	92.96%
Precision	92.96%
Recall (sensitivity, TP rate)	100.00%
Specificity (TN rate)	0.00%

Appendix Table LXI: Confusion matrices – Holdout and validation



Appendix Table LXII: Lift chart

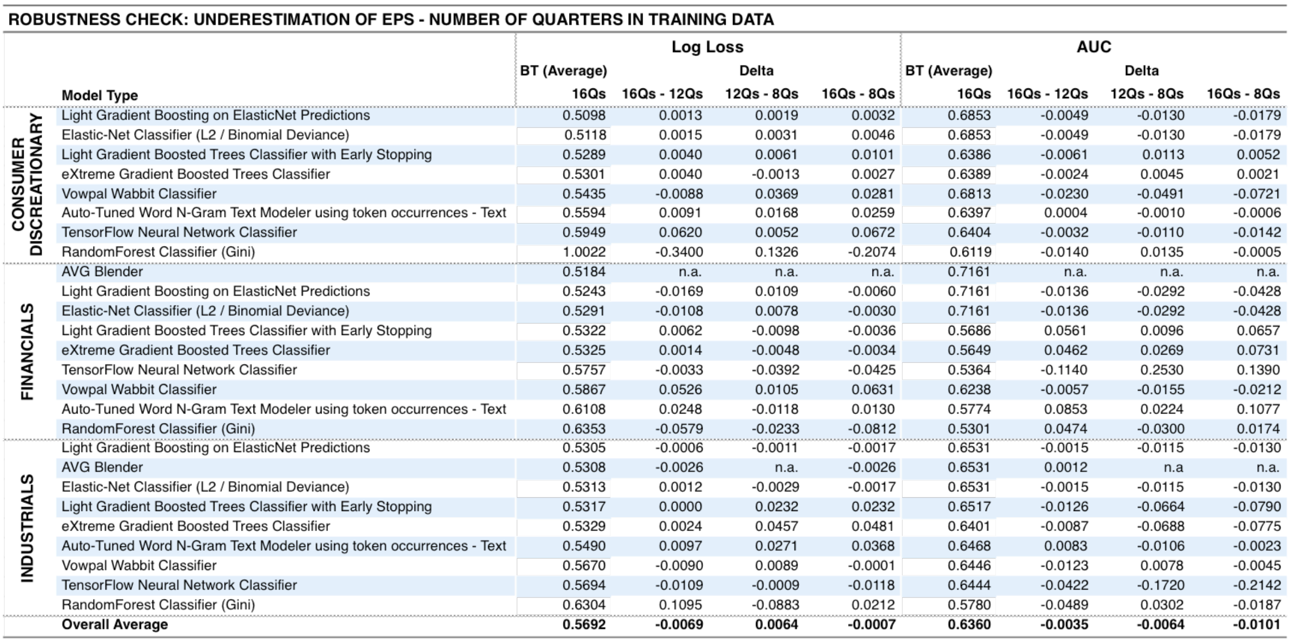
Appendix XXV. Results of Robustness Test on Length of Training Data



ROBUSTNESS CHECK: OVERESTIMATION OF EPS - NUMBER OF QUARTERS IN TRAINING DATA

		Log Loss				AUC			
		BT (Average)	Delta			BT (Average)	Delta		
	Model Type	16Qs	16Qs - 12Qs	12Qs - 8Qs	16Qs - 8Qs	16Qs	16Qs - 12Qs	12Qs - 8Qs	16Qs - 8Qs
CONSUMER DISCREATIONARY	Light Gradient Boosting on ElasticNet Predictions	0.4991	0.0021	-0.0023	-0.0002	0.7007	-0.0074	-0.0055	-0.0129
	Elastic-Net Classifier (L2 / Binomial Deviance)	0.5009	0.0015	0.0018	0.0033	0.7007	-0.0074	-0.0055	-0.0129
	eXtreme Gradient Boosted Trees Classifier	0.5220	0.0001	0.0096	0.0097	0.6424	0.0083	-0.0147	-0.0064
	Light Gradient Boosted Trees Classifier with Early Stopping	0.5238	-0.0021	0.0178	0.0157	0.6397	0.0017	-0.0291	-0.0274
	Vowpal Wabbit Classifier	0.5258	0.0028	0.0312	0.0340	0.6831	-0.0335	-0.0181	-0.0516
	TensorFlow Neural Network Classifier	0.5400	0.0026	0.0156	0.0182	0.6553	0.0017	0.0062	0.0079
	Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.5477	0.0051	0.0120	0.0171	0.6546	0.0014	0.0077	0.0091
	RandomForest Classifier (Gini)	1.0640	-0.0363	-0.1254	-0.1617	0.6464	-0.0503	-0.0184	-0.0687
FINANCIALS	Elastic-Net Classifier (L2 / Binomial Deviance)	0.5278	-0.0090	0.0074	-0.0016	0.7166	-0.0104	-0.0318	-0.0422
	Light Gradient Boosting on ElasticNet Predictions	0.5283	-0.0192	0.0090	-0.0102	0.7166	-0.0104	-0.0318	-0.0422
	Light Gradient Boosted Trees Classifier with Early Stopping	0.5364	0.0019	-0.0103	-0.0084	0.5580	0.0698	0.0017	0.0715
	eXtreme Gradient Boosted Trees Classifier	0.5367	0.0027	-0.0071	-0.0044	0.5137	0.1168	0.0024	0.1192
	TensorFlow Neural Network Classifier	0.5800	0.0060	0.0039	0.0099	0.5206	0.1309	0.0389	0.1698
	Vowpal Wabbit Classifier	0.5858	0.0627	0.0031	0.0658	0.6221	0.0423	-0.0629	-0.0206
	RandomForest Classifier (Gini)	0.6072	0.0240	-0.0425	-0.0185	0.5037	0.1121	-0.0326	0.0795
	Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.6287	0.0014	-0.0096	-0.0082	0.5228	0.1351	0.0346	0.1697
INDUSTRIALS	eXtreme Gradient Boosted Trees Classifier	0.5270	-0.0010	0.0174	0.0164	0.6361	-0.0167	-0.0195	-0.0362
	AVG Blender	0.5271	-0.0013	n.a.	-0.0013	0.6311	-0.0142	n.a	n.a.
	Light Gradient Boosted Trees Classifier with Early Stopping	0.5274	-0.0014	0.0148	0.0134	0.6299	-0.0205	-0.0069	-0.0274
	Elastic-Net Classifier (L2 / Binomial Deviance)	0.5325	0.0007	-0.0022	-0.0015	0.6463	-0.0004	-0.0126	-0.0130
	Light Gradient Boosting on ElasticNet Predictions	0.5326	-0.0009	-0.0019	-0.0028	0.6463	-0.0004	-0.0126	-0.0130
	TensorFlow Neural Network Classifier	0.5451	0.0222	0.0158	0.0380	0.6372	0.0081	-0.0048	0.0033
	Vowpal Wabbit Classifier	0.5528	0.0080	0.0116	0.0196	0.6370	-0.0064	0.0050	-0.0014
	Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.5531	0.0093	0.0281	0.0374	0.6377	0.0081	-0.0073	0.0008
	RandomForest Classifier (Gini)	0.6600	-0.0937	0.1938	0.1001	0.5356	0.0469	-0.0995	-0.0526
	Overall Average	0.5685	**-0.0005**	**0.0080**	**0.0072**	**0.6254**	**0.0202**	**-0.0132**	**0.0084**

Appendix Table LXIII: Results of robustness checks across different lengths of training time (Overestimation)



ROBUSTNESS CHECK: UNDERESTIMATION OF EPS - NUMBER OF QUARTERS IN TRAINING DATA

		Log Loss				AUC			
		BT (Average)	Delta			BT (Average)	Delta		
	Model Type	16Qs	16Qs - 12Qs	12Qs - 8Qs	16Qs - 8Qs	16Qs	16Qs - 12Qs	12Qs - 8Qs	16Qs - 8Qs
CONSUMER DISCREATIONARY	Light Gradient Boosting on ElasticNet Predictions	0.5098	0.0013	0.0019	0.0032	0.6853	-0.0049	-0.0130	-0.0179
	Elastic-Net Classifier (L2 / Binomial Deviance)	0.5118	0.0015	0.0031	0.0046	0.6853	-0.0049	-0.0130	-0.0179
	Light Gradient Boosted Trees Classifier with Early Stopping	0.5289	0.0040	0.0061	0.0101	0.6386	-0.0061	0.0113	0.0052
	eXtreme Gradient Boosted Trees Classifier	0.5301	0.0040	-0.0013	0.0027	0.6389	-0.0024	0.0045	0.0021
	Vowpal Wabbit Classifier	0.5435	-0.0088	0.0369	0.0281	0.6813	-0.0230	-0.0491	-0.0721
	Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.5594	0.0091	0.0168	0.0259	0.6397	0.0004	-0.0010	-0.0006
	TensorFlow Neural Network Classifier	0.5949	0.0620	0.0052	0.0672	0.6404	-0.0032	-0.0110	-0.0142
	RandomForest Classifier (Gini)	1.0022	-0.3400	0.1326	-0.2074	0.6119	-0.0140	0.0135	-0.0005
FINANCIALS	AVG Blender	0.5184	n.a.	n.a.	n.a.	0.7161	n.a.	n.a.	n.a.
	Light Gradient Boosting on ElasticNet Predictions	0.5243	-0.0169	0.0109	-0.0060	0.7161	-0.0136	-0.0292	-0.0428
	Elastic-Net Classifier (L2 / Binomial Deviance)	0.5291	-0.0108	0.0078	-0.0030	0.7161	-0.0136	-0.0292	-0.0428
	Light Gradient Boosted Trees Classifier with Early Stopping	0.5322	0.0062	-0.0098	-0.0036	0.5686	0.0561	0.0096	0.0657
	eXtreme Gradient Boosted Trees Classifier	0.5325	0.0014	-0.0048	-0.0034	0.5649	0.0462	0.0269	0.0731
	TensorFlow Neural Network Classifier	0.5757	-0.0033	-0.0392	-0.0425	0.5364	-0.1140	0.2530	0.1390
	Vowpal Wabbit Classifier	0.5867	0.0526	0.0105	0.0631	0.6238	-0.0057	-0.0155	-0.0212
	Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.6108	0.0248	-0.0118	0.0130	0.5774	0.0853	0.0224	0.1077
	RandomForest Classifier (Gini)	0.6353	-0.0579	-0.0233	-0.0812	0.5301	0.0474	-0.0300	0.0174
INDUSTRIALS	Light Gradient Boosting on ElasticNet Predictions	0.5305	-0.0006	-0.0011	-0.0017	0.6531	-0.0015	-0.0115	-0.0130
	AVG Blender	0.5308	-0.0026	n.a.	-0.0026	0.6531	0.0012	n.a	n.a.
	Elastic-Net Classifier (L2 / Binomial Deviance)	0.5313	0.0012	-0.0029	-0.0017	0.6531	-0.0015	-0.0115	-0.0130
	Light Gradient Boosted Trees Classifier with Early Stopping	0.5317	0.0000	0.0232	0.0232	0.6517	-0.0126	-0.0664	-0.0790
	eXtreme Gradient Boosted Trees Classifier	0.5329	0.0024	0.0457	0.0481	0.6401	-0.0087	-0.0688	-0.0775
	Auto-Tuned Word N-Gram Text Modeler using token occurrences - Text	0.5490	0.0097	0.0271	0.0368	0.6468	0.0083	-0.0106	-0.0023
	Vowpal Wabbit Classifier	0.5670	-0.0090	0.0089	-0.0001	0.6446	-0.0123	0.0078	-0.0045
	TensorFlow Neural Network Classifier	0.5694	-0.0109	-0.0009	-0.0118	0.6444	-0.0422	-0.1720	-0.2142
	RandomForest Classifier (Gini)	0.6304	0.1095	-0.0883	0.0212	0.5780	-0.0489	0.0302	-0.0187
	Overall Average	**0.5692**	**-0.0069**	**0.0064**	**-0.0007**	**0.6360**	**-0.0035**	**-0.0064**	**-0.0101**

Appendix Table LXIV: Results of robustness checks across different lengths of training time (Underestimation)