Investigating the effect of Transparency and Data Control on Trust, Privacy Concerns and the Perceived Quality of Deep Movie Recommender Systems

Master Thesis

Mikkel Ulstrup (23670) & Thomas Wingsted (108513)

School: Copenhagen Business School

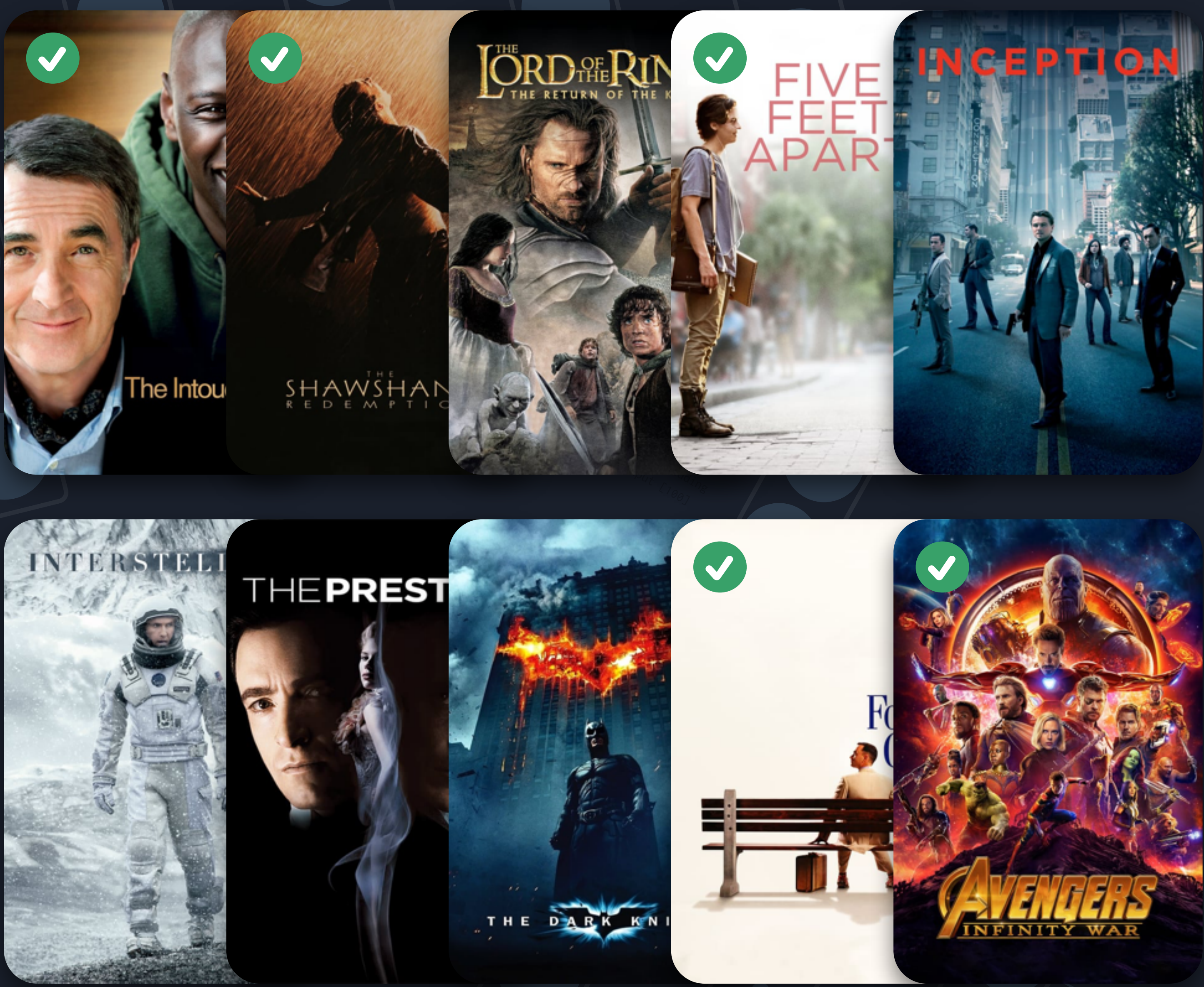
Study: MSc in Business Administration and E-business

Date of Submission: 16.11.2020

Supervisor: Thomas Frick

Number of Characters / Pages: 100,287 / 50

Investigating the effect of Transparency and Data Control on Trust, Privacy Concerns and the Perceived Quality of Deep Movie Recommender Systems



Abstract

Consumers are presented with a sea of different products and services, and the consumer decision-making process becomes overwhelming as a result. Recommender systems have been invented to solve this issue, alleviating the consumer from the stress that is navigating in and filtering the endless amount of information. Personalized recommendations have proven to create significant value for consumers and companies, but they pose a threat to consumers' privacy. A scenario-based experiment was conducted to uncover some of the underlying factors influencing the perceived quality of recommendations. The experiment finds no evidence to support a significant influence of transparency nor data control on the perceived recommendation quality. Further, this paper proposes that, in a non-privacy sensitive domain such as movie recommender, trust and privacy does not matter in the specific context.

1. Introduction

It has never been easier to gather information about products and services as it is today. However, the amount of information available has shown to be overwhelming and challenging for consumers to filter, and decision making becomes more difficult as a result. When investing in a product or service, may it be with money or time, we want to make the best decision possible so we do not waste money or time. Information overload has made that choice difficult, as there are simply too many options to choose from (Isinkaye et al., 2015). Recommendation systems have been invented to solve this issue as the recommendation agent does the filtering for the consumer, leaving the consumer with fewer and better options to choose from, while alleviating the consumer from the stressful process of filtering the information.

Much research has gotten into increasing the accuracy of recommendation agents, and advancements in the field of deep learning have proven to be able to solve this task. Deep learning has caught many researchers' interest in the field of computer vision and natural language processing, where it has shown great results. Further, deep learning excels at learning feature representation, which makes it very suitable for recommendation systems (S. Zhang et al., 2019).

Personalized recommendations rely on personal information, which consumers, in some cases, have concerns about providing (Kokolakis, 2017). This term has been coined the personalization privacy paradox, where consumers want personalized services that fit their preferences better, but at the same time, their privacy concerns increase as they have to expose personal information (Sutanto et al., 2013). Past research has suggested that it is not solely the recommendation accuracy that matters to consumers when evaluating a recommender system (Nilashi et al., 2016).

This paper investigated the effect of transparency and data control on privacy concerns, trust, and the perceived recommendation quality. First we built a recommender agent using deep learning, then we built a front-end Javascript application and a back-end API in Python and integrated the recommender agent in order to complete the recommender system. In order to assess the effect of transparency and data control we conducted an experiment centered around our recommender system. We manipulated our two independent variables, transparency and data control by making a 2x2 scenario based experiment where the user was presented with data control, transparency, both or neither.

We chose to work with deep learning, as it has seen much growth in interest, research, and impressive results (S. Zhang et al., 2019). Deep recommender systems are used at scale in some of the largest internet companies that serve the most massive recommender systems known to man, such as at Google (Cheng et al., 2016). This thesis paper aimed to look at privacy and recommender systems in a setting that is very close to real life and is using deep learning for improved model performance. This is a challenging task, building a complete web service with an advanced deep recommender model with integrated custom surveys, website design, and server hosting. As far as we know, there is no prior research that has this extensive and as-close-to-real-life experiment style.

With 247 participants in the experiment distributed randomly in four different scenarios, we analyzed the effect of the before mentioned variables on trust, privacy concern, and perceived recommendation quality. This paper finds no evidence to suggest that transparency or data control influences the perceived recommendation quality in the context of movie recommendations. This paper suggests that the domain's impact can be a deciding factor for privacy concerns, and sensitivity is essential. Furthermore, investigating privacy concerns can be difficult due to privacy bias, the experimental design is an important factor in trying to mitigate the privacy bias.

We present this paper's Research Question: *How do recommendation agent transparency and availability of data control functions influence consumers' perception of recommendation quality?*

2. Theoretical background and related work

The following chapter will outline the foundations of recommender systems, purpose, origins, and recommendation techniques. Next, it will dive into deep recommender systems and outline why that particular field is gaining an increased attention level. Further, it will look at previous research regarding privacy, transparency, and data control and outline results from past surveys and experiments in the field. It will look into the term privacy paradox and researchers and engineers have dealt with privacy in recommendation systems.

2.1. Recommender Systems

Recommender systems (RS) are essential systems for the modern web and businesses across the world. Effective personalized recommender systems filter massive amounts of data and present the user with personalized data-items that fit their respective interest and needs at a given time, making their search and use of information easier (Nagarnaik & Thomas, 2015). The RS tries to predict what a given user might like, such as a new not-seen-before product when shopping online, a new song when streaming music, or a new movie when streaming movies. What the system recommends is generally termed an "item," i.e., a movie is the "item" in a movie recommender system (Amatriain & Basilico, 2015).

Recommender systems are relatively new compared to other information system tools, such as databases. Research in RS as an independent research area started in the mid-1990s (Amatriain & Basilico, 2015). Since then, much focus has been given to the area, since it has proven valuable to internet businesses, such as Netflix, Amazon, Spotify, Facebook, and IMDb. Given the vast application field, recommender systems can help improve many business metrics, which is why many companies focus on building better and more accurate recommender systems. For e-commerce, RS might help increase sales, diversify sales, and increase user satisfaction since the RS can recommend more items the user did not know she wanted or did not think to look for. The user might become more loyal, as they feel like the e-commerce site is more personalized towards their preferences, which helps increase retention and sales (Ricci et al., 2015).

For streaming services, such as a movie streaming service, the RS can recommend new movies that a user might not have seen and fitted their movie preferences. Research from Netflix shows that their members, on average, lose interest after 60 to 90 seconds of browsing movies on their site. In that timeframe, the user generally reviews 10 to 20 movies on one or two screens. If the user finds something interesting, that is great. However, if the user finds nothing of interest, they risk the user abandoning their service altogether. Recommender systems prove very important, as stated: *"But, most important, when produced and used* *correctly, recommendations lead to meaningful increases in overall engagement with the product (e.g., streaming hours) and lower subscription cancellations rates." (Gomez-Uribe & Hunt, 2016, p. 13:7).*

There is no doubt that recommender systems provide a great deal of business value if implemented correctly. Gomez-Uribe et al. (2016) estimate that their combined effect of personalization and recommendations is valued at $1B per year in increased lifetime value and reduced churn. They proved that they reduced churn by improving their recommender algorithm, which saved them a large sum of money each year.

Further cementing the importance of recommender systems is the Netflix Prize (*Netflix Prize*, 2009), where Netflix presented a dataset that asked researchers to improve their recommendation algorithm with a 10% root mean squared error (RMSE) (Amatriain & Basilico, 2015). The prize was $1m, which gives testimony to how important Netflix thinks recommender systems are. Most interesting from the competition was the blog post from Simon Funk, who introduced a new way to compute matrix factorization, known as SVD (singular value decomposition). This new method is now a benchmark algorithm for recommender systems (Funk, 2006).

In general, there are three approaches for building recommender systems: Collaborative Filtering, Content-Based, or a Hybrid of the two. Content-Based models are created by comparing items by attributes and grouping them. The model can predict a new item given previously activated items and user attributes by creating user and item attribute representations. This way, a movie recommender content-based model could group similar movies and present the user with movies whose attributes match the movies already seen by the user, e.g., if the user likes Batman Begins, she probably also would like to see the Dark Knight or the Dark Knight Rises. Likewise, if the system were based on user attributes, it could match the user with other movies seen by users whose attribute profile matches the particular users, e.g., teenage girls from Denmark probably have somewhat similar movie taste (Isinkaye et al., 2015; Ricci et al., 2015; Rocca, 2019). Collaborative Filtering, on the other hand, is based on the past interactions of a user. The items the user interacts with are stored in a large matrix, with all other users and their interactions, in a large "user-item interactions matrix." The model then tries to find similar user/item interactions vectors to find new items for users based on the "nearest" interaction vectors from other users. If user X likes movie A, B, C, and D, it is likely that user Y, who likes movie A, B, and C, also likes movie D. Collaborative filtering models are usually sub-grouped into either memory-based (large sparse vectors) or model-based approaches (small dense vectors) (Isinkaye et al., 2015; Ricci et al., 2015; Rocca, 2019).

The most significant problem with collaborative filtering models is that they suffer from the "cold-start" problem. Whenever a new user is created, that particular user has no items selected/rated. Therefore, their user-item interactions matrix is empty, and the model cannot present the user with any new recommendations. Therefore, many models are based on a hybrid approach where items are presented based on the available information like the user's personal attributes (Ning et al., 2015). Other strategies include presenting random (or most popular) items, testing various combinations, and over time improving the presented “cold” items. Then, after the user has interacted with the system, the model can recommend items (Rocca, 2019).

2.1.1. Personalization in Recommender Systems

Personalization is defined as the task of proactively customizing products and service recommendations to fit the individual's personal preferences and tastes (Chellappa & Sin, 2005). Besides helping users in the decision-making process and saving them time and effort, the personalization feature is ultimately for pure business reasons. Personalization is great for online companies in multiple ways. It can help them predict demand, build loyalty from the customers, and increase the possibility of cross-sales if applicable (B. Zhang et al., 2014).

A recommender system’s task is to help the user navigate the endless possibilities and choices online by narrowing it down to a few choices to make the decision process more straightforward and convenient. Every individual is unique, and personalization of recommendations is one of the core features of a sound recommender system (B. Zhang et al., 2014).

2.1.2. Deep Recommender Systems

Since the popularization of Deep Learning, a field within Machine Learning and Big Data Analytics, Deep Recommender Systems have seen a rise in research and business interest. Deep learning has provided massive improvements in areas such as computer vision, speech recognition, and predictive forecasting (S. Zhang et al., 2019). Therefore, it's natural that researchers, and businesses, look for other application areas, as it can solve immensely complex tasks with the most impressive results (Xizhao Wang et al., 2020).

The reason that one might use deep learning, as described by S. Zhang et al. (2019), can be broken down into four parts; non-linearity, representations, sequence modeling, and flexibility. A lot of previous recommender models are based on linearity, such as matrix factorization (MF). The way MF works is by linearly combining user/item interactions latent factors. Linearity is often the reason for models oversimplifying their output, and therefore limiting their effectiveness. Deep learning is very well known for capturing intricate interaction patterns and, therefore, more effectively showing items that match a user's preferences. Deep learning models are very efficient in learning the underlying factors. It learns many factors that matter for a given model and makes it easier to use. There is no need for intensive feature engineering with deep learning, as the model itself captures the features. Deep learning models are very well suited for sequential modeling tasks and are highly flexible in using it (with many open-source frameworks such as Tensorflow[1] and Keras[2]). Deep learning is very good at creating models that can capture a wide variety of variables. Unlike the non-deep-learning models,

[1] https://www.tensorflow.org/
[2] https://keras.io/

that can mostly only capture a single thing at a time, deep recommender systems can be built to capture everything from the collaborative filtering data (user-item interaction), content data about the item and the user, and even the sequence of events (most notably from recurrent neural networks (RNN)) (S. Zhang et al., 2019).

In the field of deep learning, different models and techniques exist. The choice of model depends on the task at hand. Therefore, there are no good nor bad models as such - it depends on the context. The following section will outline some of the different deep learning techniques. Most deep recommender system techniques have specific models that focus on the different kinds of recommender systems types; content, collaborative, or hybrid (S. Zhang et al., 2019).

Multilayer Perceptron (MLP) is a neural network with one or more hidden layers. It is the essence of deep learning and the visualization people often refers to when speaking about deep neural networks. It consists of at least three layers, an input layer, an output layer, and one or more hidden layers in between (Yang, 2019). The number of hidden layers is a design choice. The more hidden layers the network has, the more complex the network gets, thereby allowing the network to complete more complex tasks (FPF (Future of Privacy Forum), 2018). Cheng et al. (2016) present a "wide and deep" model used by Google in their Google Play store to recommend apps on a massive scale. Extending the "wide and deep" learning models, proposed by Google, Guo et al. (2017) extends this approach with a factorization machine. He et al. (2017) presents a multilayer perceptron model called "Neural Collaborative Filtering", built using two latent vectors as input that go through several hidden layers before reaching the output layer. This approach captures the linearity of matrix factorization and non-linearity of multilayer perceptron models to create an enhanced recommendation quality. Therefore, we chose this model as inspiration for one of our hybrid models, as it showed promising results and was more comfortable for us to implement.

Autoencoder (AE) is a neural network used for unsupervised machine learning, where the output remains as close as possible to the input. The model consists of four parts: Encoder, Bottleneck, Decoder, and Reconstruction Loss. Essentially an AE decomposes the input data and reconstructs it as the output. This kind of neural network is often used in anomaly detection tasks (Badr, 2019). Sedhain et al. (2015) presented one of the earliest versions of AE to be used in recommender systems. Their "AutoRec" model was built to tackle the collaborative filtering problem. Based on the "AutoRec" model, Kuchaiev and Ginsburg (2017) created a much deeper model with a much higher dropout rate. Most autoencoder models are used for collaborative filtering (S. Zhang et al., 2019). However, as stated, we were not interested in a pure collaborative filtering approach, as we could not overcome the cold-start problem associated with these models. Therefore, we choose to build our autoencoder, with inspiration from the research in autoencoder collaborative filtering models, but as content-based models. However, an exciting research development is to use autoencoders to fill in the "blanks" in the collaborative filtering model, to fix the cold start problem. This is beyond the scope of this paper, but an appealing research area.

Convolutional Neural Network (CNN) is a neural network identified by the convolution layers and pooling layers. CNN's are great for tasks involving images as data input, such as image recognition and image classification. The convolution and pooling layers' task is to reduce the image into a form that does not require as much to process (Saha, 2018). For recommender systems, an interesting approach is to use the CNN in collaboration with another model, e.g., in a movie recommender setting, where the movie poster (or sequence of images, i.e., a video) would be used as a feature in the model because visuals might compel users more than we think. Let et al. (2016) presents a similar approach to this fascinating deep recommender research area.

Recurrent Neural Network (RNN) is well suited for modeling sequential data as this network contains loops and memories of previous computations (S. Zhang et al., 2019), meaning that they remember things learned from previous inputs. Therefore, the same input can produce different outputs in an RNN depending on previous inputs in the series (Venkatachalam, 2019). This approach is interesting, as most people do not have static preferences, and this type of model can learn the preferences progression over time, i.e., the sequence of user inputs over time.

2.2. Privacy & the Personalization Privacy Paradox

With the advances in technology, it is now possible to conduct business entirely online, which is why many companies sell goods and services online. Technology enables the possibility to collect data in vast amounts, and data has become one of the most valuable resources in companies, as the data can be analyzed and insights on consumers derived. The domain of personalized recommendation systems relies heavily on personal user data in order to make recommendations. These user data can be divided into different groups. The groups include Behavioral information, Contextual information, Domain knowledge, Item metadata, and Purchase or consumption history (Naeem et al., 2013). The amount of consumer data being collected can pose a threat to individuals' privacy, which, when talking about privacy in recommender systems, is referred to as information privacy. Information privacy, in its essence, is about keeping the information within the intended scope. As soon as the information is moved outside the intended scope, a breach of privacy has happened (Naeem et al., 2013).

Even though surveys indicate that people are concerned about their privacy, they are still willing to give up personal information to receive better service. This difference in perception and the consumer's actual behavior is being referred to as the personalization privacy paradox (Sutanto et al., 2013). There has been much research in this area trying to understand the personalization privacy paradox.

Awad and Krishnan (2006) found that the more information transparency the consumer wanted, the less willing to share information, which also highlights the essence of the personalization privacy paradox, that consumers want the best service but are not willing to give out the personal information needed in order to receive that service.

In one of our previous research papers investigating privacy as a competitive advantage in the search engine domain, we found that privacy does affect the choice of a search engine when the search query is sensitive (Wingsted & Ulstrup, 2019).

Different studies show that privacy behavior is highly contextual on the environment and domains, and we can not expect people to behave the same way in different contexts (Kokolakis, 2017). Acquisti (2004) found that it is unlikely that individuals can act rationally in the economic aspect when dealing with privacy-sensitive decisions. According to Kokolakis (2017), we should not expect respondents in an experiment to act as they would in real life; even if we provided false information to make the respondents believe it is not related to privacy, they still would not behave the same way.

Barth et al. (2019) investigated the effect of technological know-how, privacy awareness, and financial resources on the privacy paradox. Their results indicate no influence from these measures to explain the behavior, as technical and financially independent individuals risked their privacy even if they knew about the potential privacy risks.

2.2.1. Dealing with Privacy in Recommender Systems

As mentioned, much different user data is relied on for showing excellent results in personalized recommender systems. Although this paper does not deal with the technical way of considering privacy in the recommender system, it is worth mentioning that much research has gone into exploring how to cope with privacy without compromising the recommendation agent's accuracy (Duchi et al., 2014). For instance, an architecture enables users to choose which data can be used and by whom, which is somewhat what we did in our experiment. A data protection mechanism allows users to hide their identities and other ways of lowering users' perceived privacy concerns (B. Zhang et al., 2014). Another way of dealing with privacy in recommender systems is with differential privacy, which has been researched plenty. Differential privacy is a method on how to deal with privacy in different systems, such as recommender systems. It ensures that the same conclusions are derived about whether individuals are part of the data set (Dwork & Roth, 2014). Dwork et al. (2014) use an example of a smoker to explain differential privacy, saying that a study finds out that smoking increases cancer risk. Now, this is true for all smokers, even if they did not participate in the experiment.

2.3. Transparency

In recommender systems, the recommendations' statistical accuracy has been investigated since the rise of recommender systems (Sinha & Swearingen, 2002). Sinha et al. (2002) suggest that little research has investigated the user interface and user perspective on these systems. They conducted a user study to explore the influence of transparency, which, in their case, is the user's understanding of how the recommendation was made. Their study indicated that users felt more confident about the recommendations they received when the system was transparent.

The paper is from 2002 and, therefore, it is quite old compared to how fast technology moves forward. In the last 20 years, much research has examined other factors to influence the perceived quality of recommendations. Cramer (2008) experimented with a content-based art recommender system to investigate the effect of transparency on trust and acceptance of recommendations. Their results suggest transparency (in terms of explaining to the user why the recommendation was made) increased the user's acceptance of the recommendations.

Nilashi (2016) conducted empirical studies to investigate recommendation quality, transparency, and website quality in conjunction. They proposed a new trust model that integrated all of those factors and assessed their relative importance for trust-building in recommender systems. Their findings indicate that transparency is equally crucial as recommendation quality is for trust-building.

While statistical accuracy is a beneficial way of measuring recommendation quality from the business' perspective, Hebrado et al. (2013) identified transparency and feedback as possible means to evaluate the recommendations from the user's perspective. They find that transparency positively affects the perceived trust in the recommendations.

2.4. Data Control

The threat that privacy poses to personalized recommendations is vital for companies to address, as the personalized recommender system hardly works without personal data. According to B. Zhang et al. (2014), the solutions available for dealing with this matter all more or less stems from the fact that the user has some control, or at least perceived control over their data. Jin et al. (2017) explain that user control has been recognized as an essential part of recommender systems. They claim that the more advanced the user control is, the more cognitive efforts are required. They experimented with music as the recommendation output, where they categorized three levels of user control (low, middle, and high). The low user control category allowed the user to sort and rate the recommendations. The middle level of user control allowed the user to specify which of their data could be used in the recommendation engine, whereas the high level of user control allowed the user to tweak the weights of the recommendation model. They found that the higher degree of user control was present, the better the recommendations were. However, only people with deep insights into recommender systems preferred high user control, whereas most people preferred the low or middle level of user control. According to B. Zhang et al. (2014), there are two types of user data input, explicit and implicit, the former being data such as product ratings, and the latter being data such as purchase history and browsing history. Their findings suggest that a data control mechanism effectively reduces privacy concerns in implicit user data, not with explicit user data. They explain this because implicit data is often unsolicited, and users may not be aware that such data are used for making recommendations.

2.5. Privacy Risks in Recommender Systems

Personalized recommender systems are an excellent way for companies to sell more products or make customers renew their subscription to their service. It does, however, pose threats to user's privacy. Suppose one is creating a collaborative filtering recommender system, which is widely used. The recommender system would then be reliant on user feedback in the form of item ratings. These user ratings combined can reveal personal data, such as political orientation, sexual orientation, financial state, and interests (Aggarwal, 2016). As a result of privacy concerns, users might be reluctant to share their personal information, resulting in a lower level of recommendation quality, users not using the service, or ultimately leaving the service (Xiwei Wang & Sztainberg, n.d.). According to Milano et al. (2020), privacy risks happen at different stages. Risks are at the point of data collection, transfer, or storage.

3. Conceptual Framework

The following chapter will present our hypothesis based on previous research and outline our constructs. Second, our conceptual model is presented based on our constructs and hypotheses.

3.1. Hypothesis Development

3.1.1. Perceived Recommendation Quality

Although not everything lies in the recommendation accuracy when evaluating the quality, it does influence the evaluation. Evaluation of recommendation quality can be done in several ways. From a technical perspective, it is worth looking at the algorithm's performance and calculating the model predictions' accuracy score. This can be done by predicting the ratings and comparing them with the actual ratings, or for an E-commerce site to calculate how many recommended products are also being purchased (Isinkaye et al., 2015). The metric to measure would be if the user watched the movie is recommended in the movie recommendations. Depending on the domain and what the algorithm is predicting, the accepted accuracy score can vary. As for recommendation agents, the baseline accuracy is generally low, as we do not expect users to buy or watch everything we recommend to them because there are so many choices and too many unknown variables that influence the decision making. These variables include financial state, mood, and weather. Say, on the other hand; we had built a facial recognition application, then we would not accept a low prediction accuracy, as we would want the application to recognize the face at each attempt with little to no failures.

In the domain of personalized movie recommendations, numerous variables potentially can influence the perceived recommendation quality. Nilashi et al. (2016) found that spending resources alone on the recommendation quality might be insufficient, and companies should also focus on trust-building factors. Tsekouras et al. (2018) found an interaction between the perceived recommendation quality and the amount of effort the user had put into providing data. In order to investigate the impact of user data-control, we hypothesize:

Hypothesis 1 (H1): *When a user's control over their data increases, it positively affects the recommendation agent's perceived quality.*

In other domains such as medical decision making, the importance of transparency and explainability is well known. In recent years it has been highlighted that it also applies to other domains such as recommender systems, and the lack of transparency may also be a reason for less acceptance in specific recommender systems (Mcsherry, 2005).

One way to make the system transparent is by adding explanations for how the recommendations are calculated for each user. Explanations come with other benefits such as justification, user involvement, and user acceptance (Herlocker et al., 2000). The user may understand where the recommendations are coming from, and therefore help them decide on the quality of the recommendations and improve the acceptance since recommendations are justified. Bilgic et al. (2005) present two methods for explaining recommendations that improve the user's estimation of item recommendation quality. The effectiveness of personalized recommendations depends on the user's willingness to share personal data (B. Zhang et al., 2014). With a transparent recommendation system, we believe that users are more willing to share that information. Therefore we hypothesize the following:

Hypothesis 2 (H2): *When transparency increases, it positively affects the recommendation agent's perceived quality.*

3.1.2. Trust

Trust is one of the best ways to effectively lower any uncertainty between the behavior of two parts interacting, may it be social or business-related (Bleier & Eisenbeiss, 2015). In recommendation agents, one of the interacting parts is replaced with software, making it even more crucial to deal with the trust aspect, as humans tend to trust other humans most. Further, the trust might increase the recommender system's perceived quality as the user has no concerns about the system's ability to make recommendations, resulting in users evaluating the recommendation quality without questioning the expertise of the recommender system (Bleier & Eisenbeiss, 2015). We hypothesize:

Hypothesis 3 (H3): *When the user's trust in the service increases, it positively affects the recommendation agent's perceived quality.*

When evaluating a recommender system, it has frequently been centered around the recommendations' accuracy, measured with a measure such as Mean Average Error (MAE). However, it is increasingly being recognized that there might be other factors that influence the recommendations, such as user satisfaction, diversity in recommendations, and trust in the recommender system. Trust is sometimes linked with transparency, and previous research suggests that transparency increases the user's trust in the recommendations (Tintarev & Masthoff, 2007).

Evaluating trust is not straightforward, as trust is a personal feeling. Tintarev et al. (2007) propose that trust can be measured by asking users through a survey or measuring it indirectly by looking at the bi-products of trust, such as user loyalty and increased sales. According to Chen (2005), trust has, for a long time, been a critical factor influencing users' decision-making process. Their findings suggest that users are more likely to return to recommender systems that they find trustworthy. Therefore, we hypothesize that a transparent recommender system will lead to a more trusted recommendation system. We hypothesize:

Hypothesis 4 (H4): *When transparency increases, it positively affects trust in the service.*

3.1.3. Privacy Concerns

Privacy concerns are inevitable in today's society, with business striving after collection and usage of personal data to understand and target consumers. Technology is the main reason for this, as businesses can now collect and analyze vast amounts of data with seamless effort. Entire business models are created around personal consumer data and are being used for personalized services. Consumers in general desire personalized services as they are built around their preferences, but at the same time, they express a higher degree of privacy concerns as the personalized services are reliant on personal information (Sutanto et al., 2013). As the personalized recommendation agents are reliant on personal data, but consumers express privacy concerns providing the personal data, we hypothesize:

Hypothesis 5 (H5): *User's privacy concerns have a negative effect on user's perceived quality of recommendations.*

One way to address this personalized privacy paradox in recommender systems is by giving the user control of their data regarding if and how the service can use it. Giving the user that kind of data-control will presumably lower their privacy concerns towards the service in general. Therefore we hypothesize:

Hypothesis 6 (H6): *The presence of data control in recommendation agents decreases users' privacy concerns.*

Personalized recommender systems rely on user data to make recommendations. The user data can be categorized into explicit and implicit data, where explicit is data that the user provides with full knowledge such as name, movie preferences, age, and gender. Implicit data, however, is the type that the user might not be aware of when providing it, which includes browsing history, purchases, or time spent on a site (B. Zhang et al., 2014). While explicit data poses a threat to the user's privacy concerns, the implicit data might be more intrusive as the user can be tracked without knowing about it (B. Zhang et al., 2014). However, Carrascal et al. (2013) experimented to figure out how much consumers value their personal information. They found that in terms of implicit data, e.g., browsing history, consumers thought it was 7 euros worth them. On the contrary, explicit data, e.g., age and gender, was thought to be valued quite a bit higher at 25 euros. In light of those findings, our experiment only deals with explicit user data.

Many different types of research have been made to deal with privacy concerns in recommendation systems across the online domain. The majority of the solutions are about giving the user the ability to control their privacy preferences, and thus privacy can be defined as the degree of control towards the service (B. Zhang et al., 2014). We hypothesize:

Hypothesis 7 (H7): *When a user's control over their data increases, it positively affects the trust in the service.*

In online interaction, trust is closely related to privacy concerns (B. Zhang et al., 2014). Wakefield (2013) also showed evidence that trust is an essential factor when users are asked to provide their personal information to the service. We hypothesize that the same is true the other way around. The trust in the service is, to some degree, determined by the degree of privacy concerns towards the service.

Hypothesis 8 (H8): *When the user's privacy concerns decrease, it positively affects trust in the service.*

Making personalized recommendations includes collecting personal data that affect users' privacy concerns. These privacy concerns impact the user's evaluation of the recommender system (B. Zhang et al., 2014).

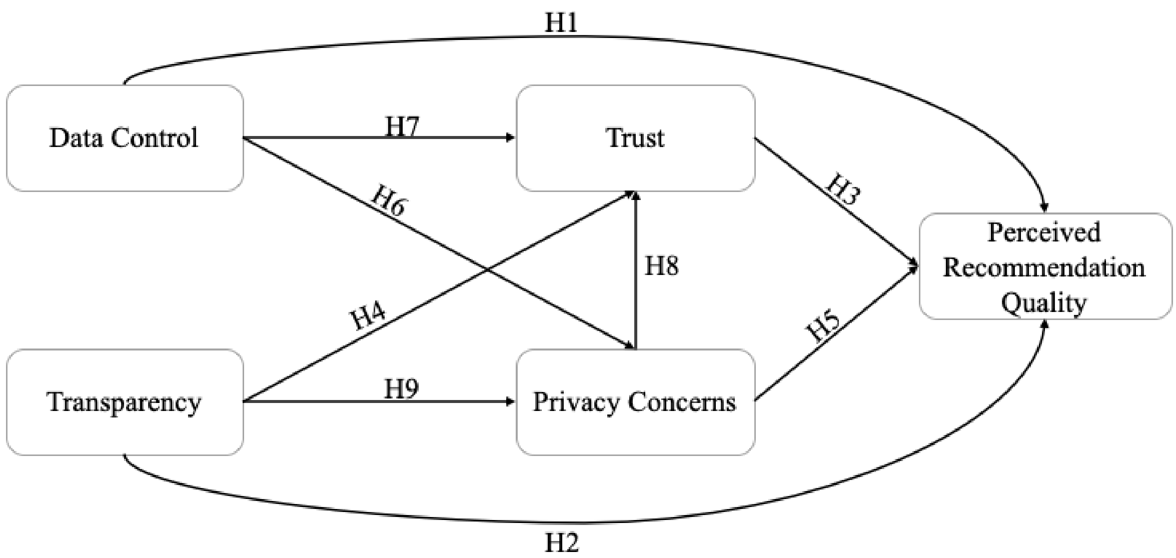
Recommender systems tend to work as a black box from the user's perspective, and when making decisions, logic dictates that the more information the decision is based on, the better. Therefore, removing the black box appearance from the recommender system might increase perceived recommendation quality from the users' perspective.

Hypothesis 9 (H9): *When transparency increases, it decreases the user's privacy concerns*

3.2. Conceptual Model

Figure 1 presents the conceptual model of this paper. The model consists of 2 independent variables; *Data Control* and *Transparency* on the left-hand side. In the middle of the model, we have our two mediating variables; *Trust* and *Privacy Concerns*, and on the right-hand side of the model, we have our dependent variable, *Perceived Recommendation Quality*. Between the variables, we have the corresponding hypothesis.

Figure 1: Conceptual Model



4. Methodology

The following chapter will outline the deep recommender model used for our experiment and the comparing baseline models. Next, our application's front- and backend will be explained, and the experimental design and procedure. Further, this chapter will outline our pre-test and sample. Finally, this chapter highlights our measures and the manipulation checks used in the experiment.

We wanted to present the user with a real-life-like system for our research, so our experiment would be as close to reality as possible. For this research, we wanted to create a scalable and accurate movie recommender system to test our hypotheses in a setting that was as close to movie streaming services users would be familiar with, i.e., Netflix or HBO.

We first developed our own custom recommender system, and next, we conducted our experiment using our developed system. First, we developed our recommender model based on the latest research within the recommender systems field, aiming to build a model that could score higher or as high as benchmark models. Afterward, we developed a backend (API) that would handle the data processing, applying our model to real-time user data and data storage for post hoc analysis. Lastly, we developed a web-app frontend that could talk to our API, serve our data, and experiment in an easy to understand way.

We conducted a scenario-based online experiment for our experiment, where our participants used the service we had developed. Our API would use a custom sorting algorithm, so participants were randomly allocated to the different scenarios. We manipulated whether users were exposed to a service that was (1) very transparent in how the algorithm works, and (2) whether they had control over their data. At the end of the experiment, we required the participants to fill in a survey where results were sent to our API for processing - everything using our developed systems.

We invite our reader to try our recommender system, web application, and experiment by visiting:

https://moviethesis.com

4.1. Recommender Model

For our thesis, we wanted to build a working recommender model that had a high accuracy score. The field of recommender systems is ever-evolving, and research into new models and methods are published frequently. We wanted to take some of the latest developments and build an algorithm we could use for our experiment, with confidence that users would have an experience close to a regular streaming service, like Netflix.

4.1.1. Big Data Analytics Methodology

Before starting our model development, we wanted to have a sharp methodology for our process to be as effective as possible. Big data analytics is complicated, with many steps and iterations until the final model has been developed, tested, and deployed. Therefore, process models have been developed, so data scientists have a foundation for the process from initial Exploratory Data Analysis (EDA) to model deployment. Wirth et al. (2000) present the CRISP-DM (Cross Industry Standard Process for Data Mining) model, which we deploy as our foundational process model for this data analysis and model development.

For this data analysis, we used Python with Jupyter Notebooks, which are commonly used among data scientists. We used standard data science Python libraries, such as Scikit-Learn, Pandas, and Numpy, for basic data manipulation, model building, and exploration. For our advanced deep learning models, we used Keras with a Tensorflow backend.

Due to the nature of deep learning, we used a custom deep learning machine to speed up model training. We built our deep learning server using mostly computer gaming parts, mainly Graphics Processing Units (GPU). However, as deep learning training requires matrix multiplication and convolution, the same as computer gaming graphics, GPUs are very well suited for deep learning tasks (Schmidhuber, 2014). The machine had two Nvidia Titan X, each with 3,584 CUDA cores at 1.5GHz, providing 11 TFLOP. We also included 64GB of RAM to handle the massive datasets, so we did not need to split our data-in-memory into batch memory loading. The machine was running Ubuntu 18.04, and the environment was easily created using a Docker container with a custom Docker Image we created (so that our programming environment had all the tools needed). The Docker image also enabled us to use the GPU powered Tensorflow version, which is significantly faster than the CPU version. Sundar et al. (2018) found the training time to be 82x and 102x faster on commonly used deep learning models, using the Titan X vs. a CPU.

We invite our reader to audit our code and recommender algorithm by visiting:

https://github.com/moviethesis/recommender-algorithm

4.1.1.1. Cold Start Problem

We wanted to build a full-scale real-time recommender system that took the user's movie preferences into account right away. However, due to the nature of how collaborative filtering models work, it would require us to recalculate the entire model each time we added a new user. This was not feasible or doable in our project, so we had to work around the cold start problem. In a regular streaming service, the user will most likely return over time, why they can recalculate their models over time, and their recommendations using collaborative filtering. We could not do this, as our "users" were not returning after the first use. Therefore, we decided to create a hybrid recommender system that took a selected movie and found the most similar movies based on the "content" (i.e., movie features - tags and genres in our case) and the movies for which the user's movie ratings profile were the closest (collaborative filtering like model). Thus, we could feed our final model a sequence of selected movies (from our user) and calculate the similarity score of all movies to each selected movie. For each movie, we could then find the top 10 most similar movies.

4.1.2. Dataset

For our recommender model, we used a publicly available dataset for movies and reviews called MovieLens (Harper & Konstan, 2015). We used two of their publicly available datasets: MovieLens 25M and MovieLens Latest Small. The 25M dataset was released on 12/2019, and the small was last updated on 9/2018. Each dataset consists of movie ratings performed by multiple users. Each rating is represented by an associated user ID, movie ID, and their given rating. A set with all the movies is given with their respective movie ID, title, and genre.

Furthermore, each movie is linked to an IMDb ID and TMDb ID for easy lookup, which was very useful when using TMDb API to retrieve movie poster images for our experiment website. Finally, a set of tags is given, which is user-generated metadata for a given movie. Each tag is either a word or short phrase and is given with a user ID and movie ID to be joined with the other data. Table 1 presents each file used with the available attributes.

Table 1: Dataset Files and Attributes

Filename	Row Attributes	Count
movies.csv	movieId, title, genres	62,423
links.csv	movieId, imdbId, tmdbId	62,423
ratings.csv	userId, movieId, rating, timestamp	25,000,095
tags.csv	userId, movieId, tag, timestamp	1,093,360

We used the small dataset for quickly testing our models and iterating fast. The small dataset consists of 100,000 movie ratings, 3,600 tags, 9,000 movies, and 600 users. This way, we were able to test our model hypotheses without waiting for the full dataset to be trained (which took hours and, for some, days). The 25M dataset consists of 25 million movie ratings, 1 million tags, 62,000 movies, and 162,000 users. This dataset was used for our final model training to get the most accurate and applicable recommender system.

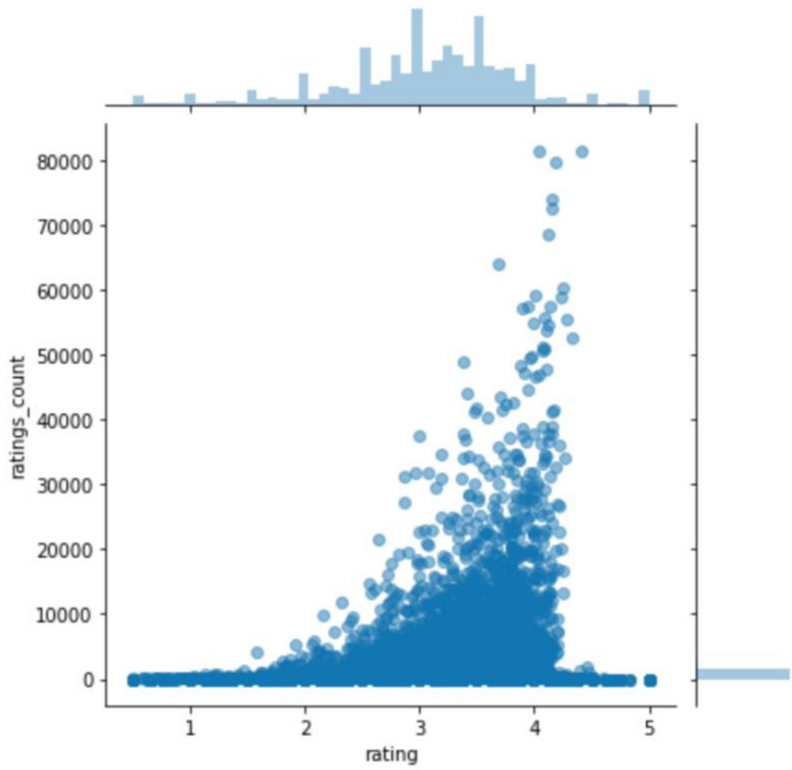
4.1.3. Baseline Models

Before we began our data analysis process, we wanted to find baseline models for comparison. We found a Python library called "surprise" that implements some of the most popular and used recommender models for easy use. From their baselines, we could see that, as we have explained, that the SVD model performs best on their test on the MovieLens 100K and 1M datasets with a respective: 0.934 RMSE / 0.737 MAE and 0.873 RMSE / 0.686 MAE (Hug, 2020). When running the 25M dataset through the matrix factorization SVD model, we scored an RMSE of 0.7811. The better score is most likely explained by the fact that the 25M dataset is much larger and gives a better result. These benchmarks were for comparing our collaborative filtering models, as we found no useful content model baselines. Furthermore, we found that the best collaborative filtering model, we could develop, was a model that used binary cross-entropy as the loss function and sigmoid as the activation function, making comparison impossible, as it became a binary optimization problem. Therefore, we looked subjectively at the recommendation results that our models produced, based on movies we liked and took note of our perceived quality of the recommender models.

4.1.4. Exploratory Data Analysis

We learned more about our dataset from our initial EDA and how we should tackle our model-building challenge. In our movie dataset, we found 62,423 unique movies. Likewise, we found 59,047 and 45,251 unique movies in the ratings and tags sets. We found 162,541 unique users, and by grouping the ratings per movie, we found that each movie on average got 423 ratings, where 1 were the least amount of ratings and 81,491 ratings was the highest ($M_{ratings\ count} = 423$; $SD = 2{,}478$; $min = 1$; $max = 81{,}491$). We found that 48,721 movies had fewer than 100 ratings, and 34,717 have fewer than 10. These movies are mostly unknown indie movies that we expect most people have never heard of. When looking at the users, we see that the number of ratings a user gave on average was 153 ($M_{ratings\ count} = 153$; $SD = 268$; $min = 20$; $max = 32{,}202$). We find it highly unlikely that a single user can give 32,202 movie ratings. The average rating score among users was 3,679. Figure 2 presents all movies by their rating, and the rating counts. We see a relationship between the number of counts and the rating.

Figure 2: Movies By Rating and Rating Count



4.1.5. Data Cleaning

Before we could develop and train our models, we needed to clean the data and format it. First, we created a data frame for each dataset file to manipulate and work with the data. Then we dropped the timestamps, as they were not needed in the analysis. For our data frames, we only needed to include movies that also had associated ratings. Therefore, we removed movies that did not have any ratings. As we saw in our EDA, there were more unique movies in our movies.csv file than our ratings.csv file. After that, we cleaned the tags file that we were going to use for our content based model and removed all tags that did not have ratings associated with the movie ID. We did this as we could not create a hybrid model if the movie did not have any ratings to be used in the collaborative filtering model, i.e., the content-based model would only work, and that would make the model ensemble fail.

After making sure our data frames all contained the equal number of unique movies represented in all data frames, we created specific data frames for our two model approaches; content-based and collaborative filtering. Due to the nature of embedding matrices, we needed to map our data frames into an index based continuous sequence of integers starting at 0. Finally, we completed our content based model data frame by removing missing data and combining the genre and tags into one large text corpus that our autoencoder could be trained on. The average rating for each movie and the total rating count were also calculated for the content-based model data frame. Table 2 presents a content data row sample.

Table 2: Content Data Frame Row Sample

movieId	title	genres	tag	corpus	rating	ratings_count
245	Batman Begins (2005)	Action Crime IMAX	action batman billionaire Christian Bale comic...	Action Crime IMAX action batman billionaire Ch...	3.93	30,684

4.1.6. Autoencoder Content Based Model

As we tried to find movies for which they had similar attributes, we used the "corpus" we created by combining the genres and user-generated tags. Finding similar movies based on a textual corpus is a natural language processing (NLP) problem. To handle textual data problems, a common way is to use Term Frequency - Inverse Document Frequency (TF-IDF) vectorization, which tokenizes the unstructured text into a numeric matrix, that can be handled by machine learning models. TF-IDF is the most frequent scheme used in recommender systems (Beel et al., 2016). We used the scikit-learn[3] TfidfVectorizer on our text corpus and was given a numerical feature matrix of size $(59{,}047\ x\ 10{,}146)$.

Our goal was to find a movie embedding that allowed us to calculate the cosine similarity. Embeddings are features mapped to a latent space (continuous vector space with a lesser dimension) representing complex relations. The most famous example is the word2vec (Mikolov et al., 2013) that can show words that are similar to be grouped closely in the lower dimensional vector space, such as the analogy: *"king is to queen as man is to woman" (Mikolov et al., 2013)*. When we had our embeddings, we could calculate the cosine similarity, as the embeddings for which the vectors were pointing in the same direction would have the highest similarity scores:

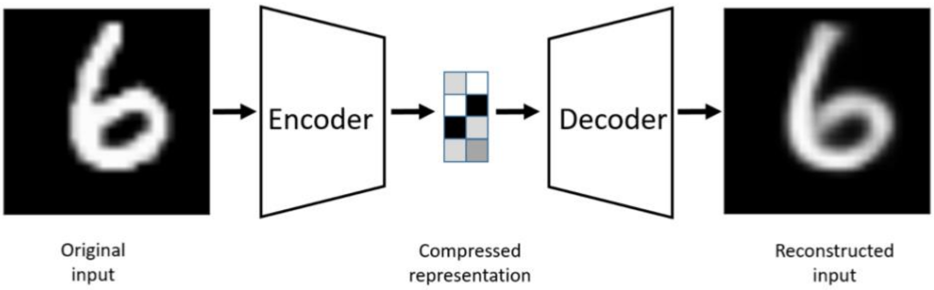
$$similarity = cos(\theta) = \frac{\boldsymbol{A} \cdot \boldsymbol{B}}{||\boldsymbol{A}||||\boldsymbol{B}||}$$

Due to the cold-start issue and data sparsity constraints, many recommender models have trouble showcasing recommendations to new users. Therefore, we looked to autoencoders for learning our movie embedding layer. The autoencoder is an unsupervised model that learns complex and informative representations of the data input. Autoencoders use two data transformations; encoder $encode(x): R^n \to R^d$ and decoder $decoder(z): R^d \to R^n$. The goal is then to minimize the distance between x and $f(x) = decode(encode(x))$

[3] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

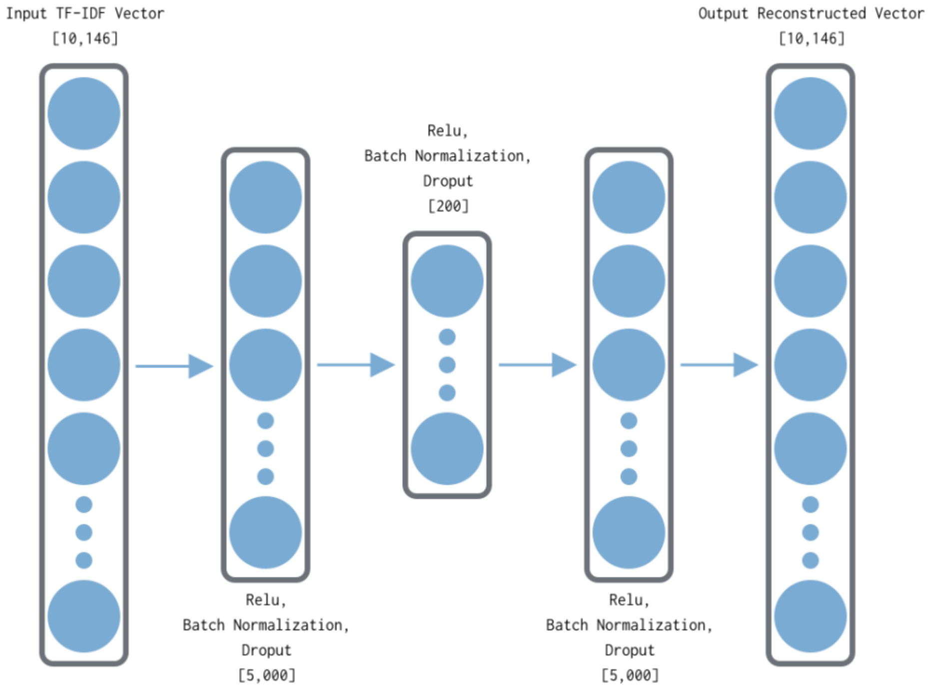
- like a principal component analysis (PCA) (Kuchaiev & Ginsburg, 2017). Figure 3 from (Bank et al., 2020) shows what the autoencoder looks like. The input is encoded to a compressed data representation and then decoded.

Figure 3: Autoencoder Example (Bank et al., 2020)



Based on Kuchaiev & Ginsburg (2017) and Sedhain et al. (2015), we created a deep autoencoder with 103,541,346 trainable parameters. Figure 4 presents our autoencoder model network. We trained our model using an MSE loss function, $MSE = \frac{1}{2}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$, and an Adam optimizer (Kingma & Ba, 2014).

Figure 4: Autoencoder Model Network



Our model trained quickly, and afterward, we could extract the encoder model that we could use to encode our TF-IDF vector into an embedding, we could use to calculate cosine similarities. Table 3 presents the top 5 movies for Batman Begins (2005) using these embeddings and their cosine similarity scores. While our results for this model presented similar movies that all related closely to our selected movie, we would have liked to see the sequels to Batman Begins (The Dark Knight 2008 and The Dark Knight Rises 2012) in the related trilogy - our content based model did not capture this relation.

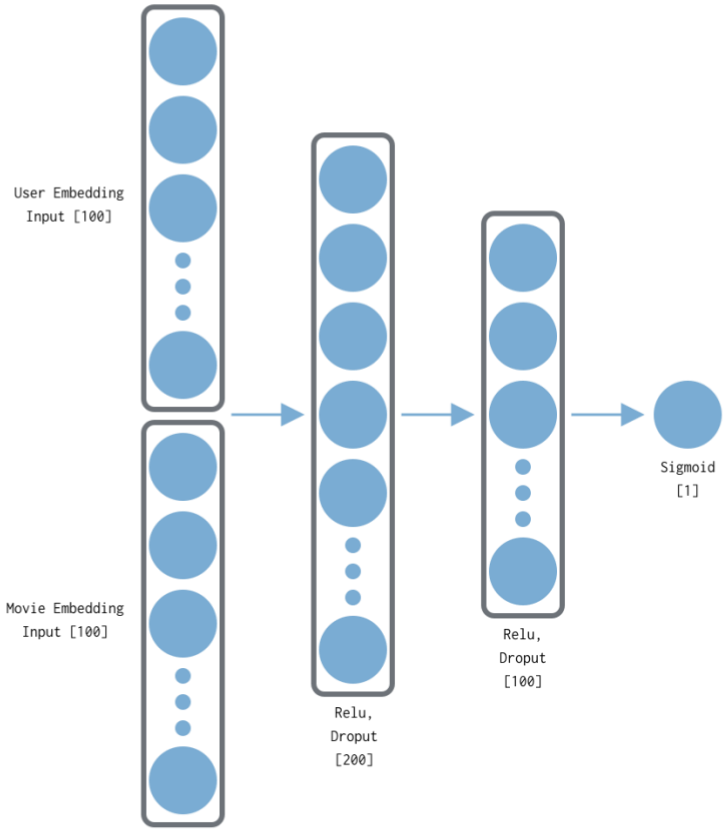
Table 3: Autoencoder Results for Batman Begins (2005)

Title	Similarity score
Batman Begins (2005)	1.000
Superman (1978)	0.928
Batman Returns (1992)	0.923
Batman (1989)	0.919
Spawn (1997)	0.914

4.1.7. Collaborative Filtering MLP Model

To capture more relations in our model, we wanted to add a collaborative filtering content variant using a deep neural network. Based on He et al. (2017), we created two input embeddings; a user latent vector and a movie latent vector. These large sparse matrices are like the user-item interaction matrices we have seen with matrix factorization algorithms. However, here they are flattened into a dimensionality of 100. Hereafter, we did a concatenation between the embedding layers as input in a regular MLP network. Finally, our output was a sigmoid activation function that allowed us to use the binary cross-entropy loss function. Figure 5 presents the model network, and table 4 presents the top 5 results for Batman Begins (2005) to compare the two models standalone.

Figure 5: Collaborative Filtering MLP Model Network



The model captures the trilogy relation, as we hoped for, and a lot of random recommendations. It seemed as the collaborative filtering part weighted lesser-known movies higher.

Table 4: MLP Results for Batman Begins (2005)

Title	Similarity score
Batman Begins (2005)	1.000
Dark Knight, The (2008)	0.845
Dark Knight Rises, The (2012)	0.800
Detective Conan: The Last Wizard of the Centur...	0.793
Love is God (2003)	0.777

4.1.8. Ensemble Hybrid Model and Recommender Engine

With our two models trained, we wanted to ensemble them, to get the best from both models. We took the results from both models and calculated an average score from the two models. We excluded movies with less than 50 votes, and that was released before 1990 in this process. Table 5 presents the final ensemble results for Batman Begins. We were delighted with the results and decided to use this model for our experiment.

Table 5: Ensemble Results for Batman Begins (2005)

Title	Similarity score
Batman Begins (2005)	1.000
Dark Knight, The (2008)	0.837
Dark Knight Rises, The (2012)	0.802
Spider-Man (2002)	0.621
Batman vs. Robin (2015)	0.592

4.2. Moviethesis API

To deploy our movie recommendation algorithm, we created a simple backend that would act as our API[4]. The Moviethesis API was built with Python and the Flask web[5] framework. The API served and received HTTP Get and POST calls for different endpoints, which had different logic. The API was also responsible for handling user data creation, storing survey information, and presenting user-specific and generated data.

The API was hosted on a single and free Google Cloud App Engine instance[6] connected to a Google Cloud Datastore database[7] (a NoSQL database). This allowed us to quickly build and deploy our API while spending zero funds and not handling maintenance, scaling, and server deployment.

Whenever a request was made to an endpoint, the API would look at the request headers to find a user ID that our frontend stored in the browser and sent off with each request. If no ID were found, the system created a new user entity, stored it in our database, and returned it to the frontend. Whenever a new user was created, we used a custom algorithm to assign users to our four manipulation groups randomly.

At various points along our experiment, like a new user was created or a user completed the experiment, the API stored different data points about the user's actions, selections, timestamps, and aggregated summary statistics about the usage of our service. We used this data for quality assurance, data integrity control, user experience (such as having the data stored if the user accidentally closed their browser), and our analysis.

[4] Application Programming Interface
[5] https://palletsprojects.com/p/flask/
[6] https://cloud.google.com/appengine
[7] https://cloud.google.com/datastore

The most advanced part of the API was the recommendation endpoint. Whenever a user had completed their movie selection, the output was sent for processing. However, due to constraints of the cold start problem and computational resource constraints (and thus monetary constraints), we had to preload our recommendations instead of calculating them in real-time. Therefore, we created our top movie list that our users were able to select from beforehand, and then, our deep learning machine looped through all possible movie selections and calculated the recommendations for each. This way, we could store all top ten recommendations for each movie on our top list as JSON files and load them whenever needed. Because of this, we were able to create a very fast and efficient endpoint that did not require our system to do any heavy calculations in real-time - all heavy computations were done beforehand on our deep learning machine.

However, the recommendation endpoint was not without calculations. The system took all the pre-calculated recommendations for the movies selected by the user and found; if there were movies recommended more than once, their average similarity score, and what selected movies was the reason for a particular recommendation. After that, we used a weighted average Bayesian estimate formula (*IMDb*, n.d.), to calculate the highest scores and sort them by descending scores. Finally, the results were formatted in a way that was easy for the client to handle and sent as a JSON response.

We invite our reader to audit our code and API by visiting:

https://github.com/moviethesis/backend

4.3. Moviethesis Website

To test our hypotheses, we built a web application called Moviethesis that allowed our users to try our recommendation algorithm, participate in our experiment, and answer our survey. Moviethesis was a mock movie streaming service. The Moviethesis website was the front-end/client that talked and interacted with our API, ensuring the data was represented in an easy to understand and great-looking way. We focused on making our UI/UX as good-looking as possible, and to look as close to other streaming services, e.g., Netflix, as possible. Figure 6 presents how our movie selection screen looked like. The web application is built with the Javascript framework Vue.js[8] using the Nuxt.js[9] framework. The application is built as a single-page app and is statically hosted at Netlify[10] for free.

As seen in figure 6, we presented the users with a top list of movies from the MovieLens dataset from which they could select their favorites. The web application fetched this list from our API and showed it to the user.

[8] https://vuejs.org/
[9] https://nuxtjs.org/
[10] https://www.netlify.com/

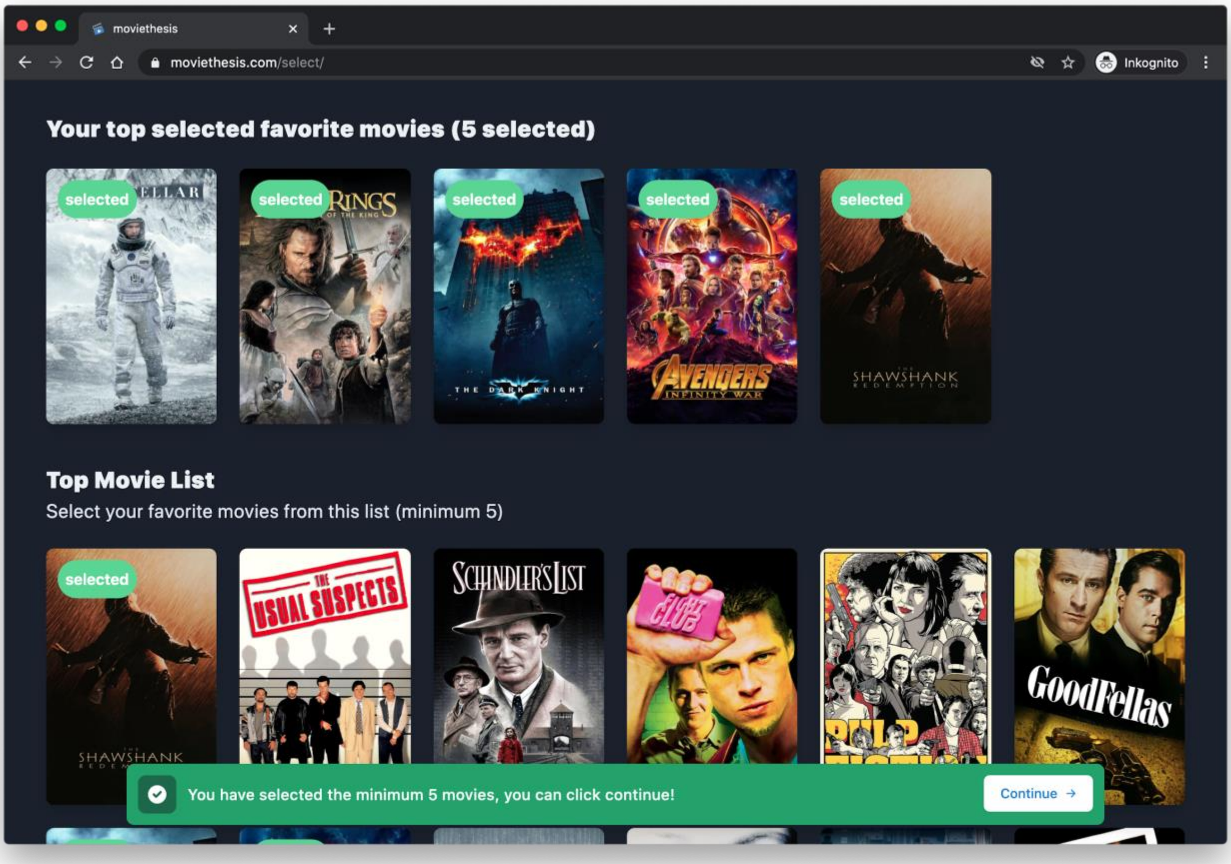
The list was based on the 250 most popular movies from the MovieLens dataset. The top list was calculated from user ratings by a weighted average Bayesian estimate formula (*IMDb*, n.d.). After that, we removed all non-english speaking movies and movies older than 1990. We choose this approach to make the list as generic as possible. The list had 132 movies after we cleaned it. The user clicked on the movies they liked, and the web application stored this selection until the user clicked “Continue.” After that, the web application sent the selection to the API for processing, and on response, redirected the user to a page that showed their new movie recommendations.

In our experiment, the user went through the entire flow. We made sure to do proper error handling, show loading states, and communicate clearly with colors, messages, and design decisions. As seen in figure 6, we presented the action bar as green when the user had completed the tasks they were presented with. The site was designed to work responsively on all platforms, from desktop browser to mobile browser.

We invite our reader to audit our code and web application by visiting:

https://github.com/moviethesis/frontend

Figure 6: Movie Selection Screen



4.4. Experimental Design and Procedure

After we had built our recommender model, backend and frontend, and tested that our data collection pipeline was working, we began implementing the experiment. To test our hypotheses, we used a scenario-based online experiment, a common method of testing (Bleier & Eisenbeiss, 2015; Fisher & Dubé, 2005; Frick, 2018; Mitra & Lynch, 1995). The experiment was created to test our hypotheses. We created the website so that we could easily manipulate our treatment variables and record the results. The experiment is a simple way to learn more about how transparency and data control could affect our trust, privacy, and recommender quality constructs.

4.4.1. Experiment Flow

The experiment consisted of five pages (or sections) on our website that each user had to go through. We aimed for a logical experiment flow where the user got the least amount of friction. We also made it as simple as possible to avoid that our participants lost interest, dropped out, or did not complete it with the care needed.

1. First, The user was presented with our homepage, which explained what the experiment was all about, who created the experiment, and necessary legal information.

2. Secondly, after the user had actively clicked "Let's get started," the user was presented with a scenario text, based on the group they had been randomly assigned, and, if the group allowed it, a data control box.

3. Thirdly, the user was asked to select their favorite movies based on a top list that we had curated beforehand, as seen in figure 6.

4. Fourthly, the user was presented with our recommender system results and could look through these. The user was either presented with "plain" results or "transparent" results.

5. Finally, the user was presented with a short survey that contained our measurements, manipulation checks, and simple demographics data collection. Appendix 3 presents the survey questions shown after a participant had looked through their movie recommendations.

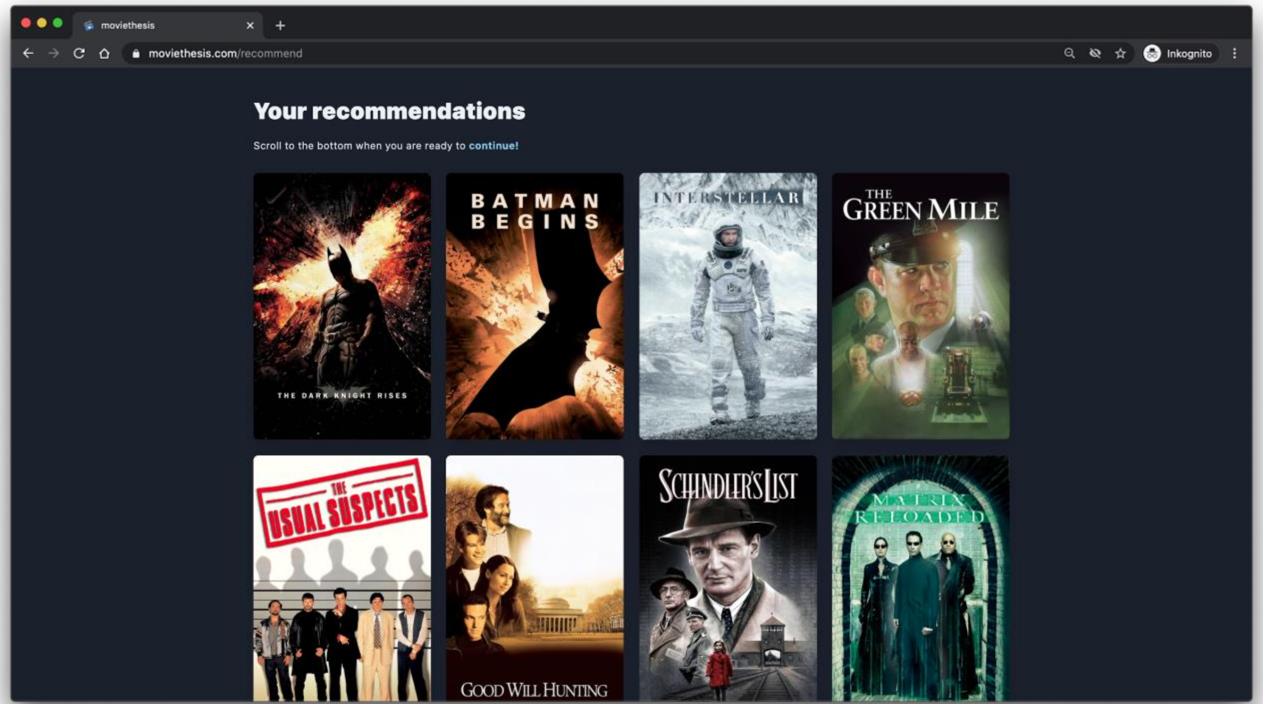
We pre-tested our experiment and learned that it did not work as expected, and therefore we tweaked the experiment, especially the scenario descriptions, so our manipulations would be more precise.

4.4.2. Experiment Treatment Groups

We wanted to explore what effect trust and privacy had on the perceived recommendation quality by looking at two possible factors that could explain these constructs; (1) system transparency and (2) personal data control. We created a 2 x 2 experimental study based on our hypotheses, where users were randomly allocated a group in which we manipulated the factors.

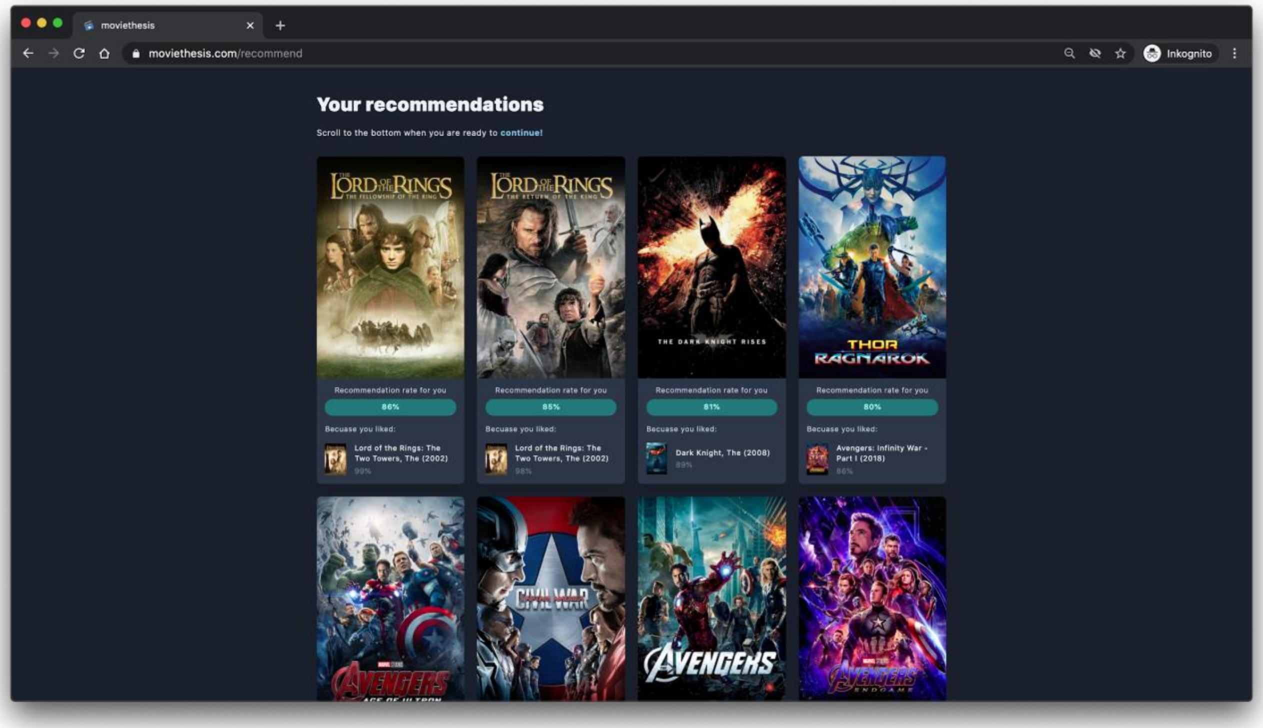
In terms of (1) system transparency, users were either shown a scenario explanation that explained precisely how the recommendation algorithm was built and worked or one that told that there was no information on how the service worked behind the scenes. Scenario texts for each scenario can be found in appendix 2. Users in the non-transparent group were not shown how the algorithm calculated their results; their movie recommendations were presented "as is" - see figure 7.

Figure 7: Screenshot of Non-Transparent Movie Recommendations



Users in the transparent group were presented with their movie recommendations together with a "recommendation rate" for each movie that indicated how much that particular movie matched the movie selection profile that the user had just completed. Furthermore, each showed what selection of movies gave the recommendation and how much that movie matched the user's movie selection profile. Figure 8 presents the "transparent" movie recommendations.

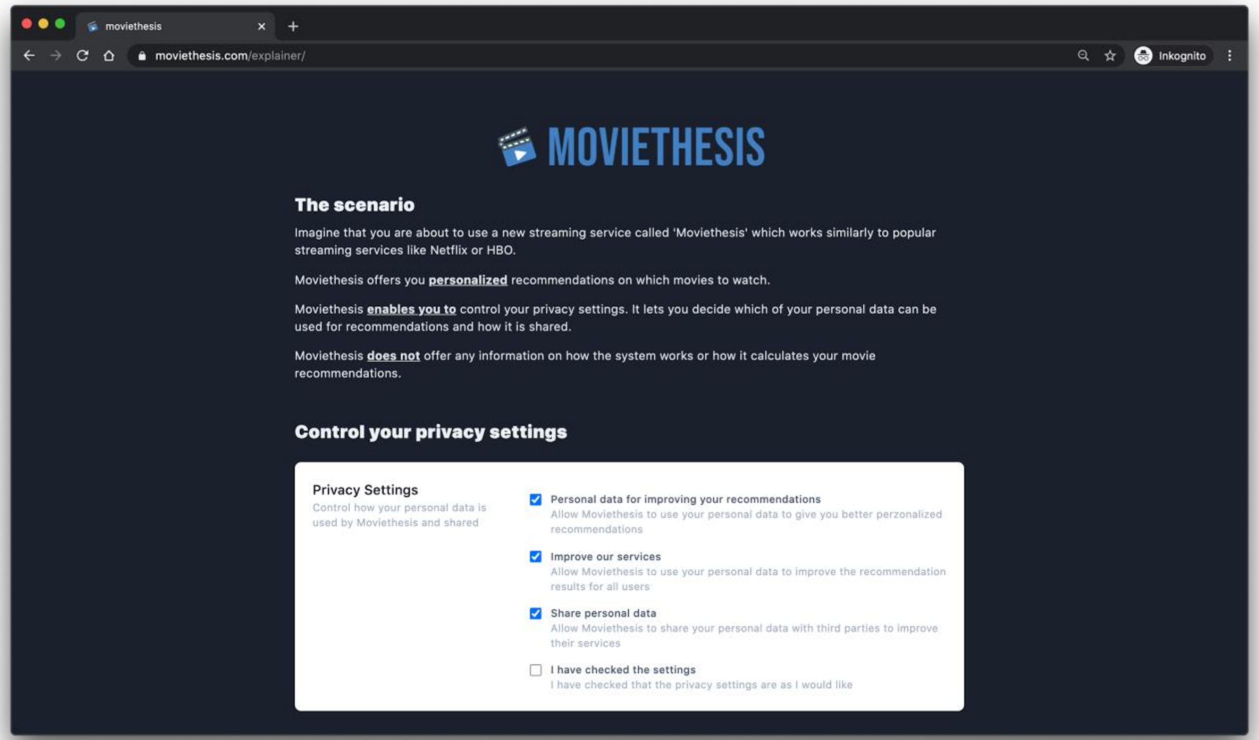
Figure 8: Screenshot of Transparent Movie Recommendations



For the (2) data control groups, in our scenario explanation, we explained that users either had full control over how their data was used and shared or had no control or visibility into how their data was used or shared. Furthermore, the participants in the group that had data control were presented with a “Privacy Settings” box that allowed them to click off their personal data preferences, see figure 9. The data control box also included an attention check that was required before they could proceed.

We choose to use opt-out instead of opt-in, as we could then see who had privacy concerns by looking at the people who actively opted out. However, if the system were opt-in, we would not be able to look if people had considered the options the same way. People generally use the default answer due to inattention, cognitive, and physical laziness (Samuelson & Zeckhauser, 1988). How questions are framed and written, have a significant influence on how the user answers, especially when they must answer it “on the spot” (Bellman et al., 2001). Therefore, we acknowledge that some users might not care to check the settings simply due to laziness - especially when users do not feel like they are risking anything (Naeem et al., 2013, Chapter 3). However, we tried to meet this by introducing the attention check, as described above.

Figure 9: Privacy Settings Data Control Box Page



For all participants, their movie recommendations were calculated by the same recommender model to ensure that the relative quality was equal across the groups. Therefore, we could measure the perceived recommendation quality, not based on a different version of the model, but from the manipulations. We wanted to measure the manipulated factors and not the model's accuracy, and therefore, the model was the same for all users.

4.4.3. Pre-test

We ran our experiment as a pre-test ($n = 46, M_{age} = 33.2, 56.5\%\ female$) among family, friends, and fellow students to verify that our experimental setup was working. Unfortunately, we found that our manipulations were not perceived as we expected. While the (2) data control manipulation worked ($\Delta M = 1.1742, t = 3.3207, p = 0.0009$), the manipulation check for (1) transparency failed ($\Delta M = 0.5217, t = 1.5140, p = 0.0686$), and users did not perceive the experiment as we intended. Therefore, we invited some friends to test our experiment in person while watching how they progressed through our site. We found that they could not tell the difference between transparency and data control. They thought that the transparency was referring to personal data transparency and not recommendation system transparency (how the system works and calculates its recommendations, i.e., *why* the user was presented with the results).

After examining the participants in real-time, we analyzed the pre-test dataset and found that the data confirms our observation. The groups who had data control were also the groups that scored the highest in perceived transparency. While our manipulation check for the (1) transparency group failed, if we tested it with the data group, we could see a statistically significant difference in the perceived transparency among the has data control and not groups ($\Delta M = 0.9470, t = 2.9261, p = 0.0027$). Therefore, we could conclude that users perceived the fact that they either had data control or not, as if the system was transparent or not - which was unintended.

We learned what was necessary for the descriptions to differentiate between system transparency and personal data transparency, i.e., how and why the recommender system works vs. how much the user knows about the service's use and handling of their personal data. First, we looked at the manipulation question and found it was framed wrong: "*I felt like Moviethesis was very transparent about how it got its results and its use of my personal data.*" We removed the wordings about personal data and were more explicit in our new question: "*I felt like Moviethesis was transparent about how the system works and why it gave me my results.*" Next, we changed our scenario descriptions to more clearly state whether a user had data control and either explain how the system worked or that there was no information offered. Appendix 1 presents the pre-test scenario descriptions, while appendix 2 presents the scenario descriptions used in our final experiment. Finally, we changed wordings in the data control to be more straightforward and named it "*Privacy Settings*," further emphasizing our system as something users are likely to be familiar with.

It was vital for us to learn and further emphasize the difference between data control and transparency in our experiment. We acknowledge that for some users, transparency is mostly related to personal data and privacy. However, we had the privacy factor as part of our data control group, which we also made a more significant point in our new scenario descriptions. After our new implementations, we found that users perceived our experiment as we expected. The participants from the pre-test were not eligible to participate in the main experiment.

4.4.4. Sample

The final experiment was conducted by family, friends, and participants acquired on the Prolific platform[11]. 268 people completed the experiment. However, for some participants, data could not be accepted. Therefore, before we started to analyze the data and to guarantee the data quality, we discarded a total of 21 participants: 11 participants did not pass our attention check, that required the participant to tick a specific box ("STRONGLY AGREE") on a Likert scale, and 10 were discarded because it was apparent they did not complete the experiment with the required attention, e.g., completing the survey faster than it should be

[11] https: //www.prolific.co/, a service for purchasing survey participants

possible, or only selecting maximum/minimum survey choices. Using these methods to ensure data quality for the analysis is frequently used in survey research (Huang et al., 2012).

After data cleaning, our final sample consists of 247 participants ($M_{age} = 32.5, 44.5\%\ female$), with the remaining demographics as shown in table 6.

Table 6: Experiment Demographics Statistics

Variable	Levels	Frequency	Percentage
Education	Elementary School	1	0.4%
	High School	68	27.5%
	Trade/Technical School	2	0.8%
	Associates Degree	26	10.5%
	Bachelor's Degree	88	35.6%
	Master's Degree	54	21.8%
	Ph.D. or Higher	8	3.2%
Work Status	Unemployed (looking for a job)	15	6.1%
	Unemployed (not looking for a job)	11	4.4%
	Unable to work	5	2.0%
	Retired	5	2.0%
	Student	37	14.9%
	Self-employed	15	6.0%
	Employed part-time	21	8.5%
	Employed full time	138	55.8%
Technical Knowledge	No knowledge	15	6.0%
	Vague knowledge	78	31.5%
	Good knowledge	80	32.3%
	Very knowledgeable	54	21.8%
	Expert knowledge	20	8.0%

As we randomly allocated the participants into our four different scenarios, we had the treatment distribution as shown in table 7.

Table 7: Experiment Scenario Distribution

	Has Data Control	No Data Control	Total
Transparent	64	64	128
Non-Transparent	60	59	119
Total	124	123	247

4.4.5. Manipulation Checks

At the end of our experiment, when users were asked to fill out a survey, we made use of manipulation checks to confirm that our recommender service was perceived as either (1) transparent about the inner workings of the algorithm or that users perceived to (2) have control over their personal data when using the service. Since we manipulated our test subjects with different versions of the recommender service, we needed to make sure our experiment worked as expected. As described in our pre-test study, our manipulation checks initially failed. We learned that we needed to be more explicit and not confuse our two constructs with each other. The manipulation checks were created for our specific purpose; however, it was inspired and adapted by the manipulation checks from (Bleier & Eisenbeiss, 2015; Frick, 2018) (2015) and Frick (2018).

For the Transparency manipulation, we asked our participants on a 5-point Likert scale from 1 (*"STRONGLY DISAGREE"*) to 5 (*"STRONGLY AGREE"*): "I felt like Moviethesis was transparent about how the system works and why it gave me my results." We find that users in the transparent group felt the system was more transparent than the non-transparent group did ($\Delta M = 0.4644, t = 2.9322, p = 0.0018$).

For the Data Control manipulation, we asked our participants on a 5-point Likert scale from 1 (*"STRONGLY DISAGREE"*) to 5 (*"STRONGLY AGREE"*): "I could control for which purpose Moviethesis can use my personal data." We find that users in the data control group felt they had more control over their personal data than the opposite group ($\Delta M = 1.2704, t = 8.9857, p < 0.0001$).

4.4.6. Measures

We measured the perceived *Recommender Quality* (*RQ*) by combining four items from (Tsekouras et al., 2018) (2018): "*Moviethesis provides valuable recommendations to me*," "*Moviethesis provides relevant recommendations to me*," "*The recommendations from Moviethesis are very trustworthy*," and "*Moviethesis saves me time*." The questions, as all measures were in this experiment, were based on a 5-point Likert scale from 1 (*"STRONGLY DISAGREE"*) to 5 (*"STRONGLY AGREE"*). The alpha reliability was 0.80. However, we did deem this measure unreliable and inconsistent, as we discussed the experiment's findings after it was completed. Even though the reliability score was above 0.7, we changed our *RQ* measure to only consist of the first question: "*Moviethesis provides valuable recommendations to me.*" Another problem we found was that the question, "*The recommendations from Moviethesis are very trustworthy*," was too close to our *trust* construct and would not give us a reliable *RQ*/*trust* result.

The *Trust* construct was measured with six items based on and adapted from Bleier & Eisenbeiss (2015) and Dabholkar & Sheng (2012). The participants were asked how much they agreed with different statements, that allows us to measure their trust in the service. All the items are found in appendix 3. The alpha reliability was 0.87.

Finally, we measured the *Privacy Concerns* (*PC*) of the respondents by four items from Bleier & Eisenbeiss (2015) and Leon et al. (2015): "*It bothers me that Moviethesis is able to track information about me,*" "*I am concerned that Moviethesis has too much information about me,*" "*It bothers me that Moviethesis is able to access information about me,*" and "*I am concerned that my information could be used in ways I could not foresee.*" The alpha reliability was 0.94.

Apart from *RQ*, which is represented as a single item why reliability estimates cannot be calculated, the other two constructs (*Trust* and *PC*) have alpha reliability over the recommended threshold of 0.7 (Gliem & Gliem, 2003).

5. Analysis

We present our analysis results, where we analyze the results of our experiment and whether we can find statistical evidence that suggest our hypotheses can be accepted or rejected. Our main goal is to find evidence as to what factors might influence a user's perception of recommendation engine quality. We present our intercorrelations of variables in table 8, and our scenario RQ means and standard deviation in table 9.

Table 8: Intercorrelations

#	Variable	1	2	3	4	5	6	7	8	9
1	Age									
2	Education	0.23***								
3	Male	-0.15*	-0.10							
4	Technical Knowhow	-0.14*	-0.12	0.38***						
5	Work Status	0.04	0.32***	0.03	0.02					
6	Data Control +	0.02	-0.01	-0.05	-0.04	0.02				
7	Transparency +	0.03	-0.07	0.10	0.06	-0.04	0.00			
8	Trust	-0.12	0.10	0.13*	0.25***	0.16*	0.03	0.00		
9	PC	-0.01	-0.11	0.02	-0.08	-0.11	-0.05	0.05	-0.26***	
10	RQ	-0.09	0.12	0.16*	0.21***	0.08	0.02	-0.07	0.71***	-0.17**

$*** p \leq 0.001$; $** p \leq 0.01$; $* p \leq 0.05$; +Binary Treatment Variables

From table 9, we initially notice no significant difference in means across our scenario groups. However, we start our analysis by going through each of our hypotheses, as established by our Conceptual Framework section.

Table 9: Experiment Scenario RQ Means and Std. Dev.

	Has Data Control / Transparent	Has Data Control / Non Transparent	No Data Control / Transparent	No Data Control / Non Transparent	Total
M_{RQ}	3.8125	4.1333	3.9843	3.8813	3.9514
SD_{RQ}	0.9574	0.5030	0.7661	0.6455	0.7473

First, we analyze the experimental treatments' influence on the participants overall perception of *recommendation quality*. Between all participants, we find that they on average rate quality of the service at a very high level ($M_{RQ} = 3.9514$). For this model, we use the single item from our survey, as mentioned in measurements. The experimental treatment variables are binary variables, that are 1 if the treatment is active or 0 otherwise. The perceived *recommendation quality* is modeled through an ordinary least squares regression, where RQ_i denotes how the estimated perceived *recommendation quality* of participant *i* is influenced by *data control*, *transparency*, and their interaction. Therefore, we estimate the model **M1**:

$$\textbf{M1}: RQ_i = \beta_0 + \beta_1 data\ control_i + \beta_2 transparency_i + \beta_3 data\ control_i \times transparency_i + \varepsilon_i$$

Model **M1** are represented by; β_0 the constant term, β_1 and β_2 the experimental treatment variables, β_3 the interaction effect between our experimental treatment variables, and ε_i the error term. Table 10 presents the model results.

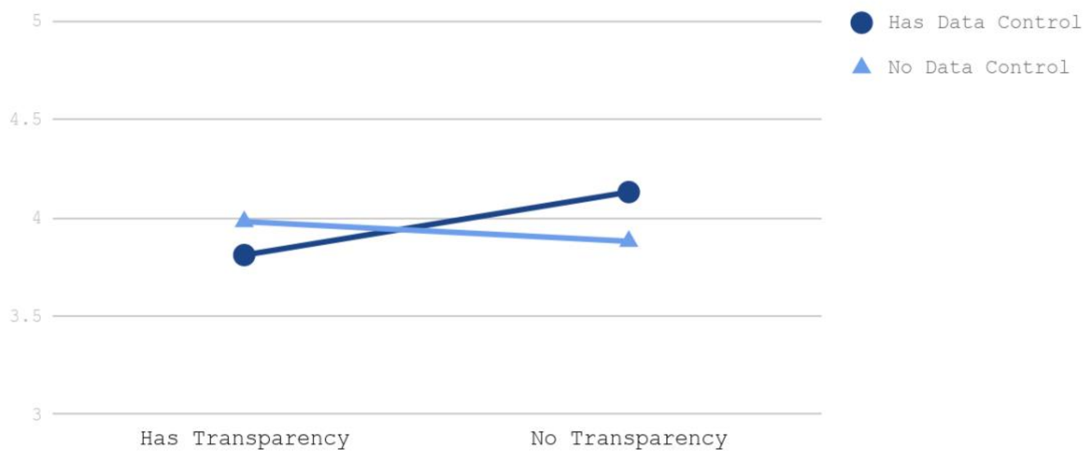
Table 10: **M1** Results

DV: RQ	(1) OLS	(2) OLS	(3) OLS
Data Control	0.0328 [0.0952]	0.0323 [0.0952]	0.0323 [0.0944]
Transparency		-0.1098 [0.0952]	-0.1097 [0.0945]
Interaction			-0.4238 [0.1890] *
Constant	3.9349 [0.0675]	3.9921 [0.0837]	3.9916 [0.0830]
Observations	247	247	247
R-squared	0.0004	0.0058	0.0260

$*** p \leq 0.001$; $** p \leq 0.01$; $* p \leq 0.05$; SD in brackets

We find no evidence that neither of our treatment variables has any significant influence on *RQ*. Therefore, our analysis does not support that neither *data control* nor *transparency* influences the user's *RQ*; thus, we find no evidence to support **H1** or **H2**. However, we see that the interaction between the two variables significantly affects the perceived *recommendation quality*. Even though we did not hypothesize this interaction, we find it exciting and puzzling, as the main effects (*transparency* and *data control*) have no significant effects. Therefore, we plot the Least Squares Means (*LSM*) for the interaction to see if we can learn anything and present the results in figure 10.

Figure 10: LSM Plot



We see that there is indeed an interaction effect. Let us look at the LSM Differences Student's t. We see that the significant difference is within the group that has data control, but differs on transparency, as presented in table 11. We find that users who have *data control* are more affected by the interaction from the *transparency* treatment, meaning that transparency has a significant influence over the *"has data control"* group.

Table 11: LSM Differences Student's t

Scenario	-Scenario	Difference	p-Value
Has Data Control / Non Transparent	Has Data Control / Transparent	0.32	0.0169
Has Data Control / Non Transparent	No Data Control / Non Transparent	0.25	0.0653
No Data Control / Transparent	Has Data Control / Transparent	0.17	0.1914
Has Data Control / Non Transparent	No Data Control / Transparent	0.14	0.2651
No Data Control / Transparent	No Data Control / Non Transparent	0.10	0.4426
No Data Control / Non Transparent	Has Data Control / Transparent	0.06	0.6077

Next, we analyze the experimental treatments' influence on the participants overall perception of *trust*. Between all participant's, we find that they, on average, trust the service at a high level ($M_{trust} = 3.7368$). Like the previous model, we operationalize *trust* as the average across our survey items as described in measurements. The perceived trust is modeled through an ordinary least squares regression, where $trust_i$ denotes how the estimated perceived *trust* of participant *i* is influenced by *data control* and *transparency*. Therefore, we estimate the model (like **M1**, but with a different dependent variable; *trust*) **M2**. Table 12 presents the model results.

$$\textbf{M2}: trust_i = \beta_0 + \beta_1 data\ control_i + \beta_2 transparency_i + \varepsilon_i$$

Table 12: **M2** Results

DV: trust	(1) OLS	(2) OLS
Data Control	0.0453 [0.0828]	0.0453 [0.0830]
Transparency		-0.0022 [0.0831]
Constant	3.7140 [0.0586]	3.7153 [0.0731]
Observations	247	247
R-squared	0.0012	0.0012

$***p \leq 0.001$; $**p \leq 0.01$; $*p \leq 0.05$; SD in brackets

We find no evidence that neither of our treatment variables has any significant influence on *trust*. Therefore, our analysis does not support that neither *data control* nor *transparency* influences the user's *trust*; thus, we find no evidence to support **H4** or **H7**.

Next, we analyze the impact of *transparency* and *data control*, the experimental treatments, on the participant's overall privacy concerns. We see that users generally rate their *privacy concerns* at a middle value[12] ($M_{privacy} = 3.1427$), meaning they are neither much nor little concerned about their privacy. Our next model tries to see how our experimental treatments affect the perceived *privacy concerns*. The higher the privacy score, the higher the participants' *privacy concern* is (operationalized as the average of our four privacy survey items, as described in measures). The perceived *privacy concern* is modeled through an ordinary least squares (*OLS*) regression, where PC_i denotes how the estimated perceived *privacy concern* of participant *i* is influenced by *data control* and *transparency*. Therefore, we estimate the model **M3**. Table 13 presents the model results.

$$\textbf{M3}: PC_i = \beta_0 + \beta_1 data\ control_i + \beta_2 transparency_i + \varepsilon_i$$

Table 13: **M3** Results

DV: PC	(1) OLS	(2) OLS
Data Control	-0.1185 [0.1390]	-0.1180 [0.1391]
Transparency		0.1146 [0.1392]
Constant	3.2032 [0.0985]	3.1435 [0.1223]
Observations	247	247
R-squared	0.0029	0.0057

$***p \leq 0.001$; $**p \leq 0.01$; $*p \leq 0.05$; SD in brackets

[12] The value of 3 corresponds to *"NEITHER AGREE OR DISAGREE"* in our 5-point likert scale

We find no evidence that neither of our treatment variables has any significant influence on *PC*. Therefore, our analysis does not support that neither *data control* nor *transparency* has any influence on the user's *privacy concerns*, hence we find no evidence to support **H6** or **H9**.

Even though we find no significant evidence of our main effects explaining our constructs, we continue our analysis to see if we find any significant relations. Next, we analyze the relationship between the *trust* and *PC* variables with our dependent variable, *RQ*. We continue as before, and, isolated, look at the *trust* and *PC* as variables to explain the perceived *recommendation quality*. We estimate our model (as previously) **M4.** Table 14 presents the model results.

$$\textbf{M4}: RQ_i = \beta_0 + \beta_1 trust_i + \beta_2 PC_i + \varepsilon_i$$

Table 14: **M4** Results

DV: RQ	(1) OLS	(2) OLS
Trust	0.8145 [0.0518] ***	0.8189 [0.0537] ***
PC		0.0099 [0.0320]
Constant	0.9074 [0.1966]	0.8600 [0.2493]
Observations	247	247
R-squared	0.5019	0.5021

$*** p \leq 0.001$; $** p \leq 0.01$; $* p \leq 0.05$; SD in brackets

We find that *trust* has a significant impact on *recommendation quality* as hypothesized by **H3** ($\beta_{trust} = 0.8145, p < 0.0001$). We find no evidence to support that *PC* might have an effect on *RQ*, and therefore cannot support **H5**.

Next, we look at our *trust* and *privacy concerns* variables. We look at the privacy concern as a factor that can affect trust. We estimate a simple linear regression model that helps us look at the relationship. We estimate the model **M5**. Table 15 presents the model results.

$$\textbf{M5}: trust_i = \beta_0 + \beta_1\ PC_i$$

Table 15: **M5** Results

DV: Trust	
PC	-0.1548 [0.0367] ***
Constant	4.2235 [0.1221]
Observations	247
R-squared	0.0676

$*** p \leq 0.001$; $** p \leq 0.01$; $* p \leq 0.05$; SD in brackets

We find significant evidence that *trust* and *privacy* do affect each other. We see that the more concerned a user is with privacy, the less they trust the service. Therefore, we find evidence to support **H8**.

We find no evidence that neither of our independent binary treatment variables can explain our dependent variable nor our mediating variables. We find a significant effect from *trust* to *recommendation quality*. However, this might be because the items are very much alike, and users did not see them as different constructs. *Trust* cannot explain the entirety of the *recommendation quality* construct, but some of it. As we find no significant relationships from our independent variables, we choose not to continue with a serial mediation model. The mediators are not associated with our independent variables, and therefore, by nature, cannot possibly mediate (Hayes, 2009).

6. Discussion

The following chapter will discuss our experimental design, the constructs used, and our findings, including why the outcome is not expected. Further, this chapter will outline the limitations of the thesis and the practical implications and theoretical contributions. Finally, future research and acknowledgments will round out this thesis paper.

6.1. The Impact of Domain

The findings and results from our experiment were different from the expected and hypothesized outcome. We presented previous research that showed transparency and user control would have a significant effect on the privacy concerns and perceived recommendation quality. However, the results did not show any significant relationship between our variables. We started to reflect on why the results were not as expected and how we thought it would be.

From a previous research study paper we did (Wingsted & Ulstrup, 2019), we saw that the inquiry's sensitivity was a determinant factor when assessing the privacy concerns. Therefore, we hypothesize that our domain was too insensitive. In this thesis, there are no sensitive inquiries to affect the privacy concerns towards the service. The only way that would happen in a movie recommender is if the preferences were all in the adult movies genre, for instance, which is unlikely to occur in a movie streaming service, and not a possibility in our experiment as none of such movies was present to the user. This could be a reason why we do not find evidence to support our hypotheses. People, in general, may not consider the movie domain to be privacy sensitive enough.

Beresford et al. (2010) showed a field experiment where participants were asked to buy a DVD from one of two almost identical stores. They asked for different privacy-related information at purchase, but one was significantly more sensitive than the other. Even though the store asked for more sensitive data, participants chose that store because it was 1€ cheaper. 75% said they had a strong interest in data protection, and 95% indicated that they cared about protecting their data. This shows that, even though sensitivity was at play, people do not necessarily rank privacy as their most profound concern in the purchase situation - even if they say they care deeply about privacy. This might counter our sensitivity hypothesis. However, another outcome of this is that people might not, in a close-to-real-world-setting, think about their privacy and data protection when confronted with basic every-day tasks as finding a movie to see or purchasing a DVD. Even though Beresford et al. (2010) could not find any significant evidence of sensitivity as an explaining factor in their experiment, Mothersbaugh et al. (2012) suggest that sensitivity might be a critical factor, especially when researching the privacy paradox. The privacy concerns, such as social threats, organizational threats, marketing issues, spying, and the likes (Krasnova et al., 2009), are present when the inquiry might be sensitive enough. One can imagine that a search for "how to get an abortion" in a country where that is highly illegal, or a sensitive political expression, could give rise to many privacy concerns. We hypothesize that this is not the case when streaming movies and looking at movie recommendations. It is not sensitive enough for people to think about the privacy concerns that might arise.

As B. Zhang et al. (2014) found, a control mechanism might alleviate the user from privacy concerns about implicit data, such as browsing and purchase history. These data types are not measured or dealt with within our experiment, as we focus on the explicit data. This is partly because there is nothing to purchase in our experiment, and registering the browsing history makes no sense in our experiment. It might be that when users type in explicit data, they are already accepting the risk of their privacy and therefore accepting it, which in that case, a control mechanism will not do anything. What people find sensitive information is also subjective. In our experiment, users only type in demographic information, which might not be perceived as sensitive. For instance, there is no such information as credit card numbers that can be leaked, and therefore the potential cost of providing the information is too low to affect privacy concerns.

6.2. Quality of the Model

Our participants generally were satisfied with our recommender system. On a scale of 1-5, they scored 3.95, which is relatively high. For this reason, we believe that we have created a relatively robust recommender system that people thought worked well. However, it is worth considering if the quality of the algorithm was "too" good. As we see it, there are two possible explanations for this. Either the recommender system we built is a good one, or the fact that we are only using a top list of movies in the experiment results in a lot of popular and high rated movies, which in general is movies that people like. Therefore, evaluating the recommendations might be misinterpreted since the user is more unlikely to be recommended a lousy movie by the recommendation engine.

One thing is the user's movie preferences in general, but movies, as well as songs, to draw parallels to another domain, are dependent on the user's mood, which is a near-impossible factor to predict. It might be that a user likes the comedy genre, but something could have happened that one day the user wanted to see a movie from a completely different and non-related genre, in which case the recommendations might not be relevant. Nonetheless, users generally rated the quality of the recommendation engine very high. Drawing on the research on the privacy paradox, it might be that the users weighted the benefits of the recommendations higher than the potential cost associated, in which case it makes sense that our variables had no significant impact on the perceived recommendation quality, if it is perceived high at its core.

6.3. Privacy

With increased online activities, individuals' privacy is under pressure as companies leverage personal information. When obtaining this information, companies' challenge is how to turn them into a competitive advantage, as individuals' privacy preferences are subjective. This means that it is difficult for companies to classify individuals into target groups as it is hard to label types of privacy groups (Preibusch, 2010). In the field of recommender systems, however, where personalization is vital, the goal is to build individual user profiles that fit the user's preferences, rather than a target group. Investigating and researching behavior in terms of privacy decisions is inherently tricky, however. Kokolakis (2017) proposes that it is impossible to capture the actual behavior of human decision making with privacy in play. The experimental factor will affect the results, as individuals will not behave the same as the real world would, no matter how much the experiment is manipulated. Even though we tried to make our experiment as close to real life as possible, we can never replicate a real-world scenario because once you try to replicate, it is already not real. According to Preibusch (2010), it is very challenging to explore the motivations behind the irrational consumer behavior in privacy decisions. To investigate it correctly, he suggests designing the experiment 3-fold by making two recommender engines. A third option of not using either is justified because a stand-alone privacy design is generally more positive than when a slightly better option is available. If we were to create two different, or at least perceived different, recommendation engines for respondents to try out, the experiment would take too long, and we assumed that too many would not complete the experiment. Further, we wanted to create an experiment as close to a real-life scenario as possible.

According to these statements by Kokolakis (2017) and Preibusch (2010), however, it could explain why we find no evidence to support our hypotheses regarding privacy concerns. People might be biased. While we assume that people are biased in the way that they will express more privacy concerns in the experiment than they perpetrate, it could also be that the experiment is affecting them not to have privacy concerns towards 'Moviethesis,' as they might know it is purely experimental. In the ideal world of a researcher in the field of behavior in privacy decision making, the experiment would set up individuals' surveillance without them knowing. That way, you could obtain knowledge of their privacy behavior in its purity, which, however, of course, is illegal.

In our experiment, we see that users rate, on average, their privacy concerns at 3.14 on a scale of 1-5. This means that, on average, the users neither agree nor disagree with our privacy concern construct. We think that this goes back to the missing sensitivity aspect. Alternatively, it might be the case that users do not care that much about their online privacy - even though much research says otherwise.

6.4. Theoretical Contributions

This paper investigated the effect of transparency and data control in recommender systems. Although our findings do not suggest any significant influence of transparency and data control, we do, however, extend the literature in terms of transparency and data control in the area of recommender systems. We discussed that the variables are maybe not significant in a non-sensitive domain. Further, we extend the literature in regards to privacy concerns and the personalized privacy paradox and our experiment showcases some of the challenges associated with researching privacy.

6.5. Practical Implications

Many researchers have tried to explain online privacy behavior. Consumers are not acting rationally and not following through on their expressed privacy concerns. Terms like privacy calculus, privacy paradox, and personalized privacy paradox have been invented to explain the behavior. This paper dealt with the personalized privacy paradox, and it suggests that maybe in the domain of movie recommender systems, the role of transparency, data control, and privacy might be negligible.

As we saw in this paper, it might be worth investing in building trust for companies operating in this context, as it affects the perceived quality. However, we cannot present any evidence that data control nor transparency can help in this regard. Generally, we know from previous research (B. Zhang et al., 2014) that using data to personalize makes excellent business sense. Therefore, investing in a great recommender agent can be profitable, as seen with Netflix (Gomez-Uribe & Hunt, 2016).

6.6. Limitations and Future Research

Throughout this paper, the effect of transparency and data control was investigated in the domain of movie recommender systems. We present the domain as the main limitation for this paper, as the non-sensitivity of movie recommender systems might neglect privacy concerns. We propose future research to do the same experiment in a different domain, where both product/service and personal information are more sensitive. It could be an e-commerce shop in the adult category or an online pharmacy recommending different products to cure a sexually transmitted disease. Being more explicit in the experiment about the potential selling of personal information to third parties is worth mentioning as an experimental design change in future research.

Further, the evaluation of the recommender system was, in general, rated high. As discussed, this might be because only top rated movies were used in the experiment, and therefore quite generic. However, the quality of the recommender algorithm was not the focus of this paper and merely used as a tool in the experiment to investigate the influence of other factors on the perceived quality.

6.7. Conclusion

Since the rise of online business, consumers are bombarded with a sea of products and services to choose from. It can be a difficult task to navigate in all the choices. Recommender systems are a great way to help users in the decision making process. Further personalization of recommendations dramatically increases the quality as individuals are different and have different preferences. Personalized recommendations do, however, pose a threat to users' privacy. This paper tried to investigate the influence of data control and transparency on the perceived recommendation quality. This was done via an experimental setup, which did not reveal that transparency and user control affect perceived recommendation quality. Neither did the experiment indicate that privacy concerns play a part in the evaluation of the recommendations. Throughout this paper, it is recognized that investigating privacy behavior is a difficult task, where bias may occur when individuals are placed in an artificial setup, even if it resembles the real world. This paper does not suggest that transparency and user control affect the perceived quality of recommendations, nor does it suggest that it does not.

6.8. Acknowledgments

We want to thank Grouplens (Harper & Konstan, 2015) for their work within data collection and research of recommender algorithms. Without their dataset, our experiment would not be possible.

A huge thanks must go to our supervisor, Thomas Frick. Without his knowledge, inspiration, and guidance, this project would not have reached the potential that we think it did. Thomas was, and still is, a massive inspiration for us, and his work and guidance have shaped how we think critically and work scientifically. We are immensely grateful for this and all the guidance we had before our thesis with him. Thank you.

Finally, a massive thanks must go to our wives, Christina Ulstrup and Camilla Gaffron Hansen. Without their support doing this process, especially helping back home, each with two small kids, we would not have written this thesis. It has been a crazy year that started with Thomas getting his daughter two months early, having to juggle school, work, and two kids, where one was hospitalized with his wife. Then Covid-19 struck, and we had to stay home and take care of our 2-year old kids. Then, in September, Christina gave birth to Mikkel's second daughter, and it has been a busy couple of months, to say the least. We are incredibly grateful that our wifes had the willpower to stand beside us in these challenging times and make sure we could complete our thesis. Thank you.

7. References

Acquisti, A. (2004). *Privacy in Electronic Commerce and the Economics of Immediate Gratification*.

Aggarwal, C. C. (2016). *Recommender Systems*. Springer.

Amatriain, X., & Basilico, J. (2015a). Recommender Systems in Industry: A Netflix Case Study. In F. Ricci, L. Rokach, & B. Shapira (Eds.), *Recommender Systems Handbook* (pp. 385–419). Springer. https://doi.org/10.1007/978-0-387-85820-3_1

Awad, N. F., & Krishnan, M. S. (2006). The Personalization Privacy Paradox: An Empirical Evaluation of Information Transparency and the Willingness to Be Profiled Online for Personalization. *The Mississippi Quarterly*, *30*(1), 13–28. https://doi.org/10.2307/25148715

Badr, W. (2019, April 22). *Auto-Encoder: What Is It? And What Is It Used For? (Part 1)*. Www.towardsdatascience.com. https://towardsdatascience.com/auto-encoder-what-is-it-and-what-is-it-used-for-part-1-3e5c6f017726

Bank, D., Koenigstein, N., & Giryes, R. (2020). Autoencoders. In *arXiv [cs.LG]*. arXiv. http://arxiv.org/abs/2003.05991

Barth, S., de Jong, M. D. T., Junger, M., Hartel, P. H., & Roppelt, J. C. (2019). Putting the privacy paradox to the test: Online privacy and security behaviors among users with technical knowledge, privacy awareness, and financial resources. *Telematics and Informatics*, *41*, 55–69. https://doi.org/10.1016/j.tele.2019.03.003

Beel, J., Gipp, B., Langer, S., & Breitinger, C. (2016). paper recommender systems: a literature survey. *International Journal on Digital Libraries*, *17*(4), 305–338. https://idp.springer.com/authorize/casa?redirect_uri=https://link.springer.com/article/10.1007/s00799-015-0156-0&casa_token=76a203OjWFoAAAAA:wiDvUxhrLpXysuLbg7bFdvjnRJkeyhwkW-M7Ek_yxgeaFP5O3_3CGOqIeekPxmdajivzwBY77wiMXCse

Bellman, S., Johnson, E. J., & Lohse, G. L. (2001). On site: To opt-in or opt-out? It depends on the question. *Communications of the ACM*, *44*(2), 25–27. https://dl.acm.org/doi/fullHtml/10.1145/359205.359241?casa_token=HCyOfCPM0VUAAAAA:_lfa4OHcjqvrHpfndUq9wN_Dt4OY6lAcmpagBw5xMfTSYA35yWI2V46kLAW68II-RWfqQvkjCjpc

Beresford, A. R., Kübler, D., & Preibusch, S. (2010). *Unwillingness to Pay for Privacy: A Field Experiment*.

Bilgic, M., & Mooney, R. J. (2005). *Explaining Recommendations: Satisfaction vs. Promotion*.

Bleier, A., & Eisenbeiss, M. (2015). The Importance of Trust for Personalized Online Advertising. *Journal of Retailing*, *91*(3), 390–409. https://doi.org/10.1016/j.jretai.2015.04.001

Carrascal, J. P., Riederer, C., & Erramilli, V. (2013). *Your Browsing Behavior for a Big Mac: Economics of Personal Information Online*.

Chellappa, R. K., & Sin, R. G. (2005). Personalization versus Privacy: An Empirical Examination of the Online Consumer's Dilemma. *Information Technology and Management*, *6*(2), 181–202.

https://doi.org/10.1007/s10799-005-5879-y
Cheng, H.-T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., Anil, R., Haque, Z., Hong, L., Jain, V., Liu, X., & Shah, H. (2016). Wide & Deep Learning for Recommender Systems. In *arXiv [cs.LG]*. arXiv. http://arxiv.org/abs/1606.07792
Cramer, H., Evers, V., Ramlal, S., van Someren, M., Rutledge, L., Stash, N., Aroyo, L., & Wielinga, B. (2008). The effects of transparency on trust in and acceptance of a content-based art recommender. *User Modeling and User-Adapted Interaction*, *18*(5), 455. https://doi.org/10.1007/s11257-008-9051-3
Dabholkar, P. A., & Sheng, X. (2012). Consumer participation in using online recommendation agents: effects on satisfaction, trust, and purchase intentions. *The Service Industries Journal*, *32*(9), 1433–1449. https://doi.org/10.1080/02642069.2011.624596
Duchi, J. C., Jordan, M. I., & Wainwright, M. J. (2014). Privacy Aware Learning. In *Journal of the ACM* (Vol. 61, Issue 6, pp. 1–57). https://doi.org/10.1145/2666468
Dwork, C., & Roth, A. (2014). The Algorithmic Foundations of Differential Privacy. *Foundations and Trends in Theoretical Computer Science*, *9*(3–4), 211–407. https://doi.org/10.1561/0400000042
Fisher, R. J., & Dubé, L. (2005). Gender Differences in Responses to Emotional Advertising: A Social Desirability Perspective. *The Journal of Consumer Research*, *31*(4), 850–858. https://doi.org/10.1086/426621
FPF (Future of Privacy Forum). (2018). *The Privacy Expert's Guide to Artificial Intelligence and Machine Learning*.
Frick, T. W. (2018). *The Implications of Advertising Personalization for Firms, Consumers, and Ad Platforms*. Erasmus University Rotterdam.
Funk, S. (2006). *Netflix update: Try this at home*. https://sifter.org/~simon/journal/20061211.html
Gliem, J. A., & Gliem, R. R. (2003). *Calculating, interpreting, and reporting Cronbach's alpha reliability coefficient for Likert-type scales*. http://scholarworks.iupui.edu/handle/1805/344
Gomez-Uribe, C. A., & Hunt, N. (2016). The Netflix Recommender System: Algorithms, Business Value, and Innovation. *ACM Trans. Manage. Inf. Syst.*, *6*(4), 1–19. https://doi.org/10.1145/2843948
Guo, H., Tang, R., Ye, Y., Li, Z., & He, X. (2017). DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. In *arXiv [cs.IR]*. arXiv. http://arxiv.org/abs/1703.04247
Harper, F. M., & Konstan, J. A. (2015). The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.*, *5*(4), 1–19. https://doi.org/10.1145/2827872
Hayes, A. F. (2009). Beyond Baron and Kenny: Statistical Mediation Analysis in the New Millennium. *Communication Monographs*, *76*(4), 408–420. https://doi.org/10.1080/03637750903310360
Hebrado, J. L., Lee, H. J., & Choi, J. (2013). Influences of Transparency and Feedback on Customer Intention to Reuse Online Recommender Systems. *The Journal of Society for E-Business Studies*, *18*(2), 279–299. https://doi.org/10.7838/jsebs.2013.18.2.279

Herlocker, J. L., Konstan, J. A., & Riedl, J. (2000). Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work - CSCW '00*. https://doi.org/10.1145/358916.358995

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T.-S. (2017). Neural Collaborative Filtering. *Proceedings of the 26th International Conference on World Wide Web*, 173–182. https://doi.org/10.1145/3038912.3052569

Huang, J. L., Curran, P. G., Keeney, J., Poposki, E. M., & DeShon, R. P. (2012). Detecting and deterring insufficient effort responding to surveys. *Journal of Business and Psychology*, *27*(1), 99–114. https://idp.springer.com/authorize/casa?redirect_uri=https://link.springer.com/article/10.1007%25252Fs10869-011-9231-8&casa_token=9p_P4Sts3ZgAAAAA:VRIRJ9uXSIK3-5XCd9MCTnFDHuq15eAlIGr3oNm4ESmwjkff3kgzdRRS5sqobTI7Ndu8-71saSnuSu-x

Hug, N. (2020). Surprise: A Python library for recommender systems. *Journal of Open Source Software*, *5*(52), 2174. https://doi.org/10.21105/joss.02174

IMDb. (n.d.). Retrieved October 20, 2020, from https://help.imdb.com/article/imdb/track-movies-tv/ratings-faq/G67Y87TFYYP6TWAV

Isinkaye, F. O., Folajimi, Y. O., & Ojokoh, B. A. (2015). Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, *16*(3), 261–273. https://doi.org/10.1016/j.eij.2015.06.005

Jin, Y., Cardoso, B., & Verbert, K. (2017). *How Do Different Levels of User Control Affect Cognitive Load and Acceptance of Recommendations? 1884*, 35–42.

Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. In *arXiv [cs.LG]*. arXiv. http://arxiv.org/abs/1412.6980

Kokolakis, S. (2017). Privacy attitudes and privacy behaviour: A review of current research on the privacy paradox phenomenon. *Computers & Security*, *64*, 122–134. https://doi.org/10.1016/j.cose.2015.07.002

Krasnova, H., Günther, O., Spiekermann, S., & Koroleva, K. (2009). Privacy concerns and identity in online social networks. *Identity in the Information Society*, *2*(1), 39–63. https://doi.org/10.1007/s12394-009-0019-1

Kuchaiev, O., & Ginsburg, B. (2017). Training Deep AutoEncoders for Collaborative Filtering. In *arXiv [stat.ML]*. arXiv. https://doi.org/10.1145/3018661.3018689

Lei, C., Liu, D., Li, W., Zha, Z.-J., & Li, H. (2016). Comparative Deep Learning of Hybrid Representations for Image Recommendations. In *arXiv [cs.CV]*. arXiv. http://arxiv.org/abs/1604.01252

Leon, P. G., Rao, A., Schaub, F., Marsh, A., Cranor, L. F., & Sadeh, N. (2015). *Privacy and Behavioral Advertising: Towards Meeting Users' Preferences*.

Li Chen, P. P. (2005). Trust Building in Recommender Agents. In *Proceedings of the 1st International Workshop on Web Personalisation, Recommender Systems and Intelligent User Interfaces*. https://doi.org/10.5220/0001422901350145

Mcsherry, D. (2005). Explanation in Recommender Systems. *Artificial Intelligence Review*, *24*(2), 179–197.

https://doi.org/10.1007/s10462-005-4612-x
Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. In *arXiv [cs.CL]*. arXiv. http://arxiv.org/abs/1301.3781
Milano, S., Taddeo, M., & Floridi, L. (2020). Recommender systems and their ethical challenges. *AI & Society*. https://doi.org/10.1007/s00146-020-00950-y
Mitra, A., & Lynch, J. G. (1995). Toward a Reconciliation of Market Power and Information Theories of Advertising Effects on Price Elasticity. *The Journal of Consumer Research*, *21*(4), 644–659. https://doi.org/10.1086/209425
Mothersbaugh, D. L., Foxx, W. K., Beatty, S. E., & Wang, S. (2012). Disclosure Antecedents in an Online Service Context: The Role of Sensitivity of Information. *Journal of Service Research*, *15*(1), 76–98. https://doi.org/10.1177/1094670511424924
Naeem, R., van Roelof, Z., Jong-Seok, L., verKai, C., & Xian-Sheng, H. (2013). *Social Media Retrieval*. Springer London Heidelberg New York Dordrecht.
Nagarnaik, P., & Thomas, A. (2015). Survey on recommendation system methods. In *2015 2nd International Conference on Electronics and Communication Systems (ICECS)*. https://doi.org/10.1109/ecs.2015.7124857
Netflix Prize. (2009). https://www.netflixprize.com
Nilashi, M., Jannach, D., Ibrahim, O. B., Esfahani, M. D., & Ahmadi, H. (2016). Recommendation quality, transparency, and website quality for trust-building in recommendation agents. *Electronic Commerce Research and Applications*, *19*, 70–84. https://doi.org/10.1016/j.elerap.2016.09.003
Ning, X., Desrosiers, C., & Karypis, G. (2015). A Comprehensive Survey of Neighborhood-Based Recommendation Methods. In F. Ricci, L. Rokach, & B. Shapira (Eds.), *Recommender Systems Handbook* (pp. 37–76). Springer. https://doi.org/10.1007/978-0-387-85820-3_1
Preibusch, S. (2010). Experiments and formal methods for privacy research. *Privacy and Usability Methods Pow-Wow (PUMP) 2010*.
Ricci, F., Rokach, L., & Shapira, B. (2015). Recommender Systems: Introduction and Challenges. In F. Ricci, L. Rokach, & B. Shapira (Eds.), *Recommender Systems Handbook* (pp. 1–35). Springer. https://doi.org/10.1007/978-0-387-85820-3_1
Rocca, B. (2019, June 2). *Introduction to recommender systems - Towards Data Science*. Towards Data Science. https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada
Saha, S. (2018, December 15). *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*. Www.towardsdatascience.com. https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
Samuelson, W., & Zeckhauser, R. (1988). Status Quo Bias in Decision Making. *Journal of Risk and Uncertainty*, *1*(1), 7–59. http://www.jstor.org/stable/41760530
Schmidhuber, J. (2014). Deep Learning in Neural Networks: An Overview. In *arXiv [cs.NE]*. arXiv.

http://arxiv.org/abs/1404.7828
Sedhain, S., Menon, A. K., Sanner, S., & Xie, L. (2015). AutoRec: Autoencoders Meet Collaborative Filtering. *Proceedings of the 24th International Conference on World Wide Web*, 111–112. https://doi.org/10.1145/2740908.2742726
Sinha, R., & Swearingen, K. (2002). *The Role of Transparency in Recommender Systems*. https://doi.org/10.1145/506443.506619
Sundar, K. V. S., Sai Sundar, K. V., Bonta, L. R., B., A. K. R., Baruah, P. K., & Siva Sankara, S. (2018). Evaluating Training Time of Inception-v3 and Resnet-50,101 Models using TensorFlow across CPU and GPU. In *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*. https://doi.org/10.1109/iceca.2018.8474878
Sutanto, J., Zürich, E. T. H., Palme, E., Tan, C.-H., Phang, C. W., Newscron Ltd., City University of Hong Kong, & Fudan University. (2013). Addressing the Personalization-Privacy Paradox: An Empirical Assessment from a Field Experiment on Smartphone Users. In *MIS Quarterly* (Vol. 37, Issue 4, pp. 1141–1164). https://doi.org/10.25300/misq/2013/37.4.07
Tintarev, N., & Masthoff, J. (2007). *A Survey of Explanations in Recommender Systems*. 801–810. https://doi.org/10.1109/ICDEW.2007.4401070
Tsekouras, D., Li, T., & Benbasat, I. (2018). *Scratch My Back And I'll Scratch Yours: The Impact of The Interaction Between User Effort and Recommendation Agent Effort on Perceived Recommendation Agent Quality*. https://ssrn.com/abstract=3258053
Venkatachalam, M. (2019, March 1). *Recurrent Neural Networks Remembering what's important*. Www.towardsdatascience.com. https://towardsdatascience.com/recurrent-neural-networks-d4642c9bc7ce
Wakefield, R. (2013). The influence of user affect in online information disclosure. *The Journal of Strategic Information Systems*, *22*(2), 157–174. https://doi.org/10.1016/j.jsis.2013.01.003
Wang, X., & Sztainberg, M. (n.d.). *Guiding Us Throughout a Sea of Data - A Survey on Recommender Systems and Its Privacy Challenges*.
Wang, X., Zhao, Y., & Pourpanah, F. (2020). Recent advances in deep learning. *International Journal of Machine Learning and Cybernetics*, *11*(4), 747–750. https://doi.org/10.1007/s13042-020-01096-5
Wingsted, B. T., & Ulstrup, M. (2019). *Is it possible to challenge Googles search market dominance with privacy?* Copenhagen Business School.
Wirth, R., & Hipp, J. (2000). CRISP-DM: Towards a Standard Process Model for Data Mining. *Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining*.
Yang, X.-S. (2019). *Introduction to Algorithms for Data Mining and Machine Learning*.
Zhang, B., Wang, N., & Jin, H. (2014). *Privacy Concerns in Online Recommender Systems: Influences of Control and User Data Input*.
Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep Learning Based Recommender System: A Survey and

New Perspectives. *ACM Comput. Surv.*, *52*(1), 1–38. https://doi.org/10.1145/3285029

8. Appendices

Appendix 1: Pre-test Scenario Descriptions

Scenario non-transparent and no user control:

Moviethesis is an online streaming service just like Netflix, HBO etc. The platform offers recommendations on which movies to watch. How exactly the recommendations are made is not public known, and neither is which data they collect and use to make the recommendations.

Scenario non-transparent and user control:

Moviethesis is an online streaming service just like Netflix, HBO etc. The platform offers recommendations on which movies to watch. How exactly the recommendations are made is not public known. They do, however, let you decide for yourself which of your data will be used to make the recommendations as well as if the recommendation engine may use your data for recommendations to other people, your own recommendations, neither or both. You do so by checking the boxes below.

Scenario transparent and no user control:

Moviethesis is an online streaming service just like Netflix, HBO etc. The platform offers recommendations on which movies to watch. The recommendation engine is an algorithm which is trained on a large dataset with approx 20.000.000 users. These users have rated, commented and attached tags to the movies they have watched. Based on this data the algorithm find similarities between users and movies, and it will use your selection of movies to create a list of movies other people with the same movie preferences also like.

It is not known which type of data they collect from you and use to make the recommendations for you.

Scenario transparent and user control:

Moviethesis is an online streaming service just like Netflix, HBO etc. The platform offers recommendations on which movies to watch. The recommendation engine is an algorithm which is trained on a large dataset with approx 20.000.000 users. These users have rated, commented and attached tags to the movies they have watched. Based on this data the algorithm find similarities between users and movies, and it will use your selection of movies to create a list of movies other people with the same movie preferences also like.

Furthermore, they let you decide for yourself which of your data will be used to make the recommendations as well as if the recommendation engine may use your data for recommendations to other people, your own recommendations, neither or both. You do so by checking the boxes below.

Appendix 2: Final Scenario Descriptions

Scenario non-transparent and no user control:

Imagine that you are about to use a new streaming service called 'Moviethesis' which works similarly to popular streaming services like Netflix or HBO.

Moviethesis offers you personalized recommendations on which movies to watch.

Moviethesis does not explain which of your personal data they collect.

They do not let you control your privacy settings and how your personal data is used.

Moviethesis does not offer any information on how the system works or how it calculates your movie recommendations.

Scenario non-transparent and user control:

Imagine that you are about to use a new streaming service called 'Moviethesis' which works similarly to popular streaming services like Netflix or HBO.

Moviethesis offers you personalized recommendations on which movies to watch.

Moviethesis enables you to control your privacy settings. It lets you decide which of your personal data can be used for recommendations and how it is shared.

Moviethesis does not offer any information on how the system works or how it calculates your movie recommendations.

Scenario transparent and no user control:

Imagine that you are about to use a new streaming service called 'Moviethesis' which works similarly to popular streaming services like Netflix or HBO.

Moviethesis offers you personalized recommendations on which movies to watch.

Moviethesis does not explain which of your personal data they collect.

They do not let you control your privacy settings and how your personal data is used.

The movie recommendations offered by Moviethesis are based on 25.000.000 movie reviews created by 160.000 users. Your personalized recommendations are calculated by Moviethesis' recommendation algorithm by using the recommendations created by other people and comparing them with your movie selction and personal data. Moviethesis will match new movies for you to watch based on your movie selection profile, which is calculated based on other similar user profiles.

Scenario transparent and user control:

Imagine that you are about to use a new streaming service called 'Moviethesis' which works similarly to popular streaming services like Netflix or HBO.

Moviethesis offers you personalized recommendations on which movies to watch.

Moviethesis enables you to control your privacy settings. It lets you decide which of your personal data can be used for recommendations and how it is shared.

The movie recommendations offered by Moviethesis are based on 25.000.000 movie reviews created by 160.000 users. Your personalized recommendations are calculated by Moviethesis' recommendation algorithm by using the recommendations created by other people and comparing them with your movie selection and personal data. Moviethesis will match new movies for you to watch based on your movie selection profile, which is calculated based on other similar user profiles.

Appendix 3: All Survey Items

How much do you agree with the following statements?

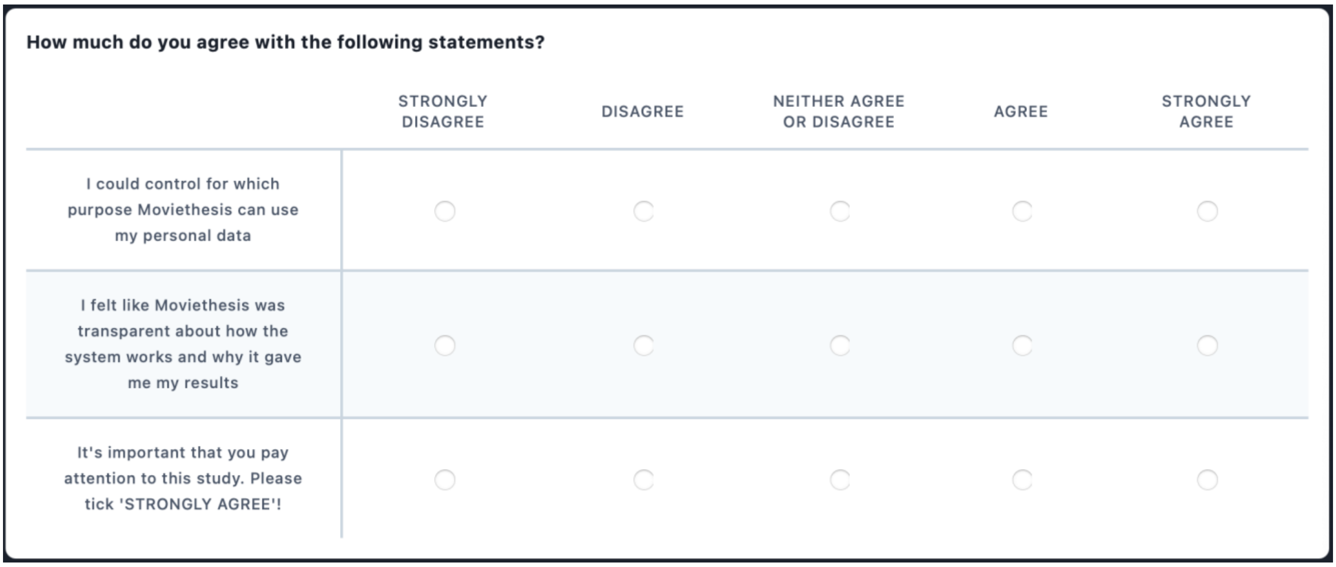
	STRONGLY DISAGREE	DISAGREE	NEITHER AGREE OR DISAGREE	AGREE	STRONGLY AGREE
Moviethesis provides valuable recommendations to me	○	○	○	○	○
Moviethesis provides relevant recommendations to me	○	○	○	○	○
The recommendations from Moviethesis are very trustworthy	○	○	○	○	○
Moviethesis saves me time	○	○	○	○	○

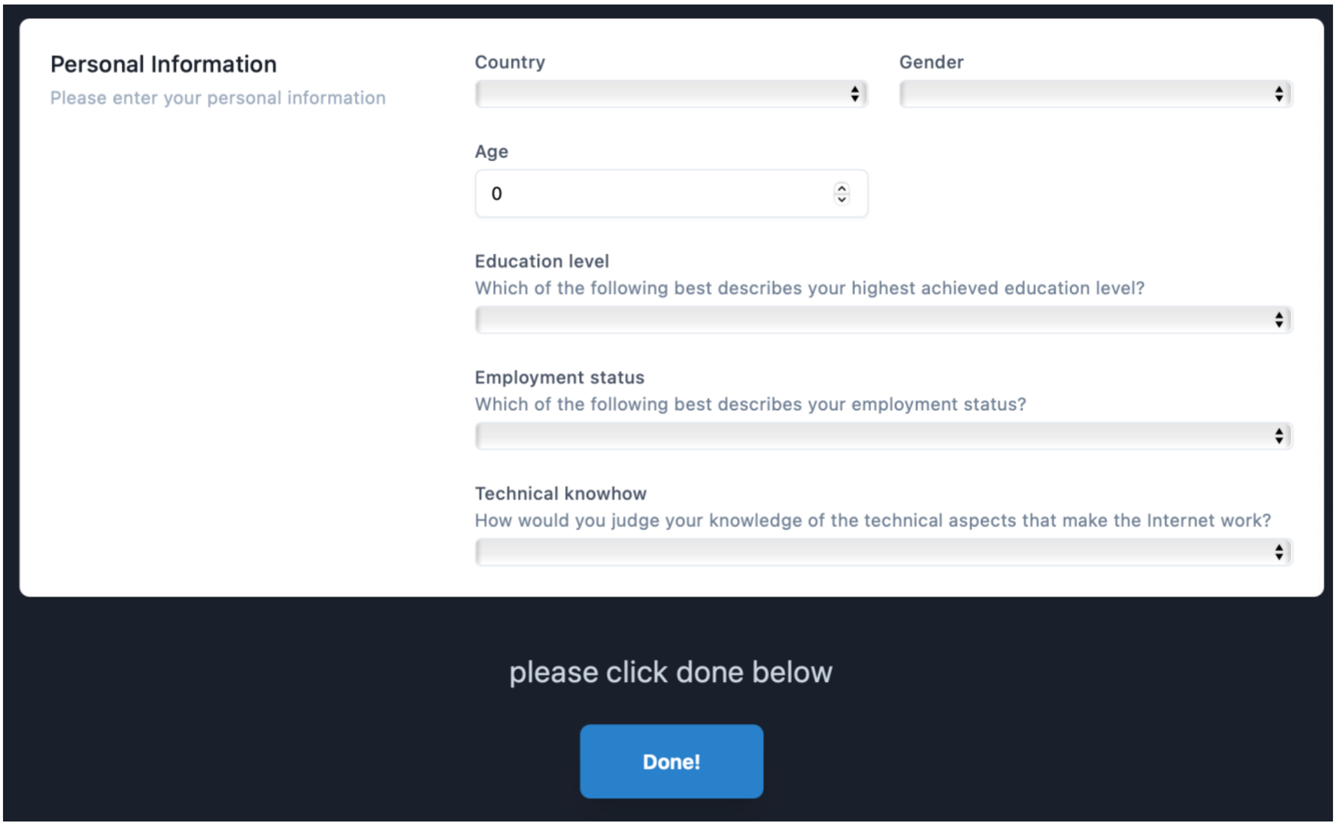
How much do you agree with the following statements?

	STRONGLY DISAGREE	DISAGREE	NEITHER AGREE OR DISAGREE	AGREE	STRONGLY AGREE
Moviethesis can be relied on to keep its promises	○	○	○	○	○
Moviethesis usually keeps the promises that it makes to me	○	○	○	○	○
I can count on Moviethesis to provide a good service	○	○	○	○	○
Moviethesis seems to be very knowledgeable about the movies it shows me	○	○	○	○	○
Moviethesis seems to be able to understand my preferences for this product	○	○	○	○	○
I can rely on Moviethesis for my decision on which movies to watch	○	○	○	○	○

How much do you agree with the following statements?

	STRONGLY DISAGREE	DISAGREE	NEITHER AGREE OR DISAGREE	AGREE	STRONGLY AGREE
It bothers me that Moviethesis is able to track information about me	○	○	○	○	○
I am concerned that Moviethesis has too much information about me	○	○	○	○	○
It bothers me that Moviethesis is able to access information about me	○	○	○	○	○
I am concerned that my information could be used in ways I could not foresee	○	○	○	○	○


How much do you agree with the following statements?
STRONGLY DISAGREE
DISAGREE
NEITHER AGREE OR DISAGREE
AGREE
STRONGLY AGREE
I could control for which purpose Moviethesis can use my personal data
I felt like Moviethesis was transparent about how the system works and why it gave me my results
It's important that you pay attention to this study. Please tick 'STRONGLY AGREE'!


Personal Information
Please enter your personal information
Country
Gender
Age
0
Education level
Which of the following best describes your highest achieved education level?
Employment status
Which of the following best describes your employment status?
Technical knowhow
How would you judge your knowledge of the technical aspects that make the Internet work?
please click done below
Done!