Master Thesis
MSc in Information Management, spring 2023

Utilizing Machine Learning to Predict the Perceived Helpfulness of E-commerce Product Reviews

---

"Roland Rezső Gimesi" 202100185

Table of Contents

Abstract

The objective of the dissertation is to explore the concept of helpfulness in online product reviews and to create a machine learning classifier for predicting the perceived helpfulness of these reviews. E-commerce customers rely more and more heavily on customer-generated online product reviews, which benefit both customers and e-commerce businesses. However, the information quality of such reviews is highly inconsistent, resulting in less helpful reviews and consequently an information overload. It is in favor of e-commerce businesses to facilitate the creation of more helpful reviews since it creates both customer and business value. The dissertation can be split into (not exclusively) three main parts: literature review, theoretical foundations, and methods. The literature review gains insights into reviews and the specific components of reviews that drive the perception of helpfulness and result in helpfulness votes when the customer evaluates the review. The theoretical foundations describe the different natural language processing and machine learning techniques and concepts utilized in the dissertation to extract meaningful insights from reviews and eventually predict helpfulness. The methods chapter contains the process of building a machine learning model that is capable of predicting the perceived usefulness of Amazon book reviews. The dissertation identifies the elements of value in online product reviews, identifies the components that drive the perception helpfulness in online product reviews and provides supporting evidence that it is possible to utilize machine learning algorithms to predict the perceived helpfulness of online product reviews. Furthermore, the dissertation proposes that it is relevant for e-commerce businesses to complement their review systems with a feature that assesses the reviews of customers during writing, evaluates whether the review will be perceived as helpful, and makes recommendations to the reviewer accordingly. The findings of the dissertation can be used in the design process of such a feature and as a basis of future research focusing on improving the accuracy of the predictions. Naturally, the dissertation has certain limitations, the majority of them being connected to the fact that the dissertation utilized a limited number of alternatives both in natural language processing and supervised machine learning, due to the dissertation’s inherent constraints such as extent and scope.

1. Introduction

Applying machine learning models to identify underlying patterns in customer-generated product reviews is not something uncommon in the Information Systems and Marketing literature (Liu et al. 2021). This dissertation seeks to utilize different machine learning methods with the aim of predicting the perceived helpfulness of Amazon online product reviews.

E-commerce is certainly a booming industry, especially since the outburst of the Covid-19 pandemic. With the wide variety of products offered on e-commerce websites, customers rely more and more heavily on customer-generated online product reviews, to gain additional information on the products and make better purchasing decisions. However, not all reviews have the same high quality, and in order for customers to be able to differentiate between reviews, many e-commerce websites introduced “helpfulness votes” as a measure of quality for it’s reviews. The purpose of this dissertation is to uncover whether it is possible to utilize machine learning to predict the perceived helpfulness of such reviews.

To achieve this purpose, during the dissertation I will investigate online product reviews, to get a deeper understanding of the concept of online product reviews and what is the value of reviews both for customers and businesses. Furthermore, I will seek to identify the components of helpful reviews that can be used as predictor features in the process of predicting perceived helpfulness. Last but not least, I will utilize natural language processing for extracting these components from the reviews and supervised machine learning to build a classifier model for the predictions.

The dissertation will address the problem above by seeking answers to the following research questions:

- *RQ 1: What is the value of online product reviews and what are the concrete components that generate value?*
- *RQ 2: What are the challenges of online product review objectivity?*
- *RQ 3: What are the variables that influence the perceived helpfulness of online product reviews?*
- *RQ 4: Is it possible to predict the perceived helpfulness of Amazon book reviews using supervised machine learning?*

The structure of the dissertation will be the following: first, the *Background* chapter will introduce the framework of the dissertation, to help better analyze the problem. Then the *Literature review* chapter will summarize the findings of the literature review process and identify potential features for the predictive model. Afterward, the *Theoretical foundations* chapter will introduce the different concepts and methods used during the model-building process, while the *Methods* chapter will carry the reader through each stage of the model-building. The results of the model-building process will be summarized in the dedicated *Results* chapter. Lastly, the *Discussion* chapter will assess whether the research questions of the dissertation managed to be answered, and discuss the practical implications of the findings and possible future research, alongside the limitations of the dissertation.

2. Background

This chapter serves as an introduction to the background of this dissertation, using the IS success model of Delone and Mclean (1992, 2003). Using a model provides us with a framework to analyze a problem in a systematic way by examining it from multiple directions. Here, these directions are the different dimensions identified by Delone and Mclean (1992, 2003) that influence the success of information systems. The IS success model had been chosen because it captures how the success of any information system is dependent on the realized benefits of both customers and businesses. This dissertation is focusing particularly on two of the dimensions: Information Quality and Net Benefits. Through these two dimensions, I intend to show the relevancy of review systems (and thus reviews) for both businesses and customers through the dimension of Net Benefits, and highlight the inconsistent information quality of reviews through the dimension of Information Quality.

The Information System success model (Delone and Mclean, 1992, 2003) is a framework that evaluates the success of information systems. The revised model identifies 6 interdependent variables that contribute to IS success, namely: *System Quality, Information Quality, Service Quality, Intention to Use, User Satisfaction, and Net Benefits* (Delone and Mclean, 2003). Delone and Mclain (1992, 2003) argue that the success of information systems is determined by how well a system performs across these 6 dimensions.

In the original model, Delone and Mclean (1992) distinguished between *Individual impact* and *Organisational Impact* as dimensions of information system success, which they later on grouped together as *Net benefits* in their 2003 revision, including additional levels of benefits (eg.: group, societal). As the original IS success model distinguished the effect an information system has on different levels, I find it just as important to make a clear distinction when we talk about value - a frequently recurring term throughout the dissertation. In order to avoid confusion and controversy, we must differentiate *customer value* and *business value* when we talk about the value created by information systems, based on who perceives the value created. Similarly, the original IS success model differentiated *Individual impact* and *Organisational impact* (Delone and Mclean, 1992). These distinctions help to keep an eye on both parties that are impacted by review systems (individual customers and e-commerce organizations) and what value is created for them (customer value and business value), as they are both essential for the IS's success.

Delone and Mclean (1992, 2003) described Net Benefits as the most important measure of IS success. By exploring the *Organisational impact* of review systems, we can see that the Information Systems literature agrees that e-commerce businesses generally benefit from review systems as they have an influential effect on additional sales and improve the perception of the usefulness of the website (Delone and Mclean, 2003; Kwark et al. 2014; Mudambi and Schuff, 2010), so e-commerce businesses should welcome and encourage customers to write relevant reviews (Kwark et al. 2014). Besides improving sales, reviews can mean an additional revenue source for e-commerce businesses. Certain firms are purchasing product reviews from e-commerce websites (eg.: Amazon) to display them on their own websites providing additional information to their own customers (Mudambi and Schuff, 2010) making it highly relevant for e-commerce businesses to facilitate high-quality reviews as much as possible.

By analyzing review systems from the perspective of *Individual impact,* we can see that the main benefit of such systems is the additional information a customer receives about the products that might not be explicit in the product description. The additional information can be whatever information a particular customer is missing from the original product description that would make them make a better decision. These bits of extra information play a significant role in the customers' decision-making

process (Liu et al. 2021, Shen et al, 2015). Moreover, review systems provide a means for customers to express their post-purchase satisfaction or dissatisfaction with a certain product.

The core purpose of every information system is to generate high-quality information for its users, which is measured by accuracy, precision, and the ability of the system to assist its users in making decisions (Petter et al, 2013). Correspondingly, the aim of review systems is to provide additional information to customers that help them to make better decisions. Research shows that even though customers rely heavily on reviews as an information source for making better purchasing decisions, review quality as well as the quality of the information provided by the review is highly inconsistent, despite the vast amount of reviews available (Liu et al. 2021). The indicator of information quality in certain review systems is helpfulness which can be expressed through helpfulness votes given by customers if they found that a particular review contained valuable information that helped them in making the purchasing decision. This dissertation proposes that information quality can be improved by facilitating the creation of more helpful reviews through the use of natural language processing and machine learning algorithms. In the later chapters, I will assess why using these methods is highly necessary for identifying helpful reviews in advance.

3. Literature review

It is difficult to identify in advance which new products or services will represent great value to an organization as well as its customers. Therefore, the next chapters will first introduce and explain the concept of Digital offerings and show what are the pre-requirements of identifying a potentially successful digital offering for an organization, that are worth investing in. Then I am going to summarize the existing Information Systems literature on reviews to gain valuable insights on what are the variables that influence the perceived helpfulness of reviews. Acquiring proper theoretical foundations about the topic can help to identify predictors and ensure generalizability (Liu et al. 2021), thus these insights are expected to provide a suitable direction for choosing the variables for the machine learning model.

The review of the literature has been done in an iterative way using 3 iterations (Appendix 1.1, 1.2, 1.3, 1.4) from two main sources. As a first source, literature has been collected with a search string from Scopus including the 'basket of eight' Information Systems Management journals (Appendix 2.) using a collection of relevant keywords and had been filtered first on the title, then abstract and lastly the whole content. As a second source, papers had been collected from the curriculum of the Information Management program to ensure the connection to the author's respective studyline, and had also been assessed in the same iterative way. Throughout the literature review process, 22 papers were chosen out of the 808 papers that were assessed in total. The exact inclusion and exclusion criteria used for filtering the literature can be found in Appendix 1.5.

3.1. Digital offerings

The literature on Information Systems is based on the core assumption that information and computer technologies play a crucial role in promoting service innovation. With the widespread adoption of large-scale information digitalization and digital infrastructures, it has become possible to collect, process, and utilize vast amounts of data, which in turn enables the creation of new products and services. In addition to facilitating the creation of new products and services, digital innovation also helps create value through the delivery of services (Barret et al. 2015)

The number of potential new digital products and services (hereafter: digital offerings) is nearly unlimited which makes it hard to identify viable digital offerings that are worth investing in. Ross et al. (2019) suggested that in order to identify successful digital offerings, one must look at the intersection of what an organization is capable of creating by utilizing digital technologies and what the customers will value enough to use or pay for. Even though some potential products would match the portfolio and capabilities of an organization, they might deliver small to no value to their customers. On the other hand, even if certain products or services are highly desired by customers, if an organization doesn't possess the right resources and capabilities that are necessary for proper delivery, it is simply doesn't worth the investment. Therefore, it requires continuous experimentation to test what offerings can a company develop and what the customers will find valuable, to find the right digital offerings. (Ross et al, 2019) In the next subchapters, I will explore the technological opportunities in the form of Artificial intelligence and machine learning and the specific elements of value that customers are looking for in products and services.

3.1.2. Technological opportunities - AI and Machine Learning

Artificial intelligence and machine learning are two terms that are often used either interchangeably or comparatively, therefore I've found it important to clarify the meaning of these concepts at the beginning of this chapter. Artificial Intelligence (or AI) refers to the broader spectrum of creating machines that are able to perform tasks that would generally require human intelligence. Meanwhile, machine learning – as a subfield of AI – focuses on training machines to learn from data to solve different classification or predictive tasks and automatically improve themselves without being explicitly programmed. Both fields are utilizing the higher speed and capacity of computers – compared to humans.

A research study done by Ransbotham et al. (2017) revealed that more than 60% of the consumer sector expected that AI will play a large impact on their offerings by 2022, compared to 20% in 2017. Even though we cannot compare it to actual numbers today, it is a great indication of the trend that organizations move towards utilizing Artificial Intelligence in their business offerings. Obtaining business value from AI is clearly on the agenda of the majority of organizations. The requirements of this value creation include effective algorithm training and a vast amount of meaningful data that the algorithm can learn from. (Ranshbotham et al., 2017)

Certain tasks are more suitable for AI algorithms than others, nevertheless, today's AI is a widely applicable and efficient technology, that supports innovation in a wide range of fields including processing images, text, and speech (Fügener et al, 2022). Classification-related tasks focusing on these domains are expected to become more precise and autonomous in the future through the utilization of AI, as machine learning algorithms tend to outperform humans in such tasks regarding accuracy. However, as Fügener et al. (2022) demonstrated, the highest performance can be achieved through effective collaboration, when humans and AI are working together, with AI being the one delegating work. This is due to the fact that certain tasks are still better suited for humans, such as writing "human-like" text, which is still difficult to achieve for most AI algorithms by themselves (Scientist, 2017).

This dissertation proposes that human-AI collaboration could be adopted to improve the helpfulness of online product reviews when machine learning algorithms are used to identify patterns in review characteristics that indicate helpful content but customers formulate the actual sentences accordingly. The topic of the dissertation, which is to analyze reviews for extracting features that can be used for predicting helpfulness, should be the first step of this collaboration.

3.1.2. What customers value

As outlined in Chapter 3.1. in order to identify successful digital offerings, organizations must have a thorough understanding of customer value. Almquist et al. (2016) identified 30 elements of customer value that address four types of needs: functional, emotional, life-changing, and social impact. They argue that the more elements a product or service offers to its customers, the higher customer loyalty it can achieve, and the greater the organization's sustained revenue growth will be in general. Almquist et al. (2016) also identified *quality* as the element of value that affects customers more than any other, and no other element can compensate for the lack of it effectively.

Helpfulness votes of review systems indicate perceived usefulness (helpfulness and usefulness are terms often used interchangeably in the product review literature), which has been an important measure of information systems. It is the extent people believe a system will help them to perform better. Interpreting it in

the context of product reviews would be the extent to which customers believe a review will help them make better purchasing decisions.

Analyzing reviews based on the 30 elements of value, we can see that helpful reviews contain 3 elements of value, and all of these 3 are on the functional level. The other 3 levels (emotional, life-changing, and social impact) and their value elements cannot be directly connected to helpful reviews, therefore I won't be focusing on those. The 3 value elements of helpful reviews are *quality,* which as we have seen must be the basis of every offering; *informing,* as a helpful review must provide viable and accurate information to potential customers regarding product attributes and usage; and *reducing risk* which is the overall aim of online product reviews, to reduce uncertainty by providing additional, quality information about the product in question, so that potential customers can improve their purchasing decision.

Customers increasingly expect to find reviews on e-commerce websites. According to Shen et al. (2015), 62 percent of customers read reviews prior to purchasing and 82 percent of them report that they have been influenced by the reviews in their purchasing decision. However, not all product reviews are helpful which compromises the quality of review systems. Therefore, besides maintaining a review system and understanding the incentives of online product reviewers (Shen et al. 2015), it is in favor of e-commerce businesses to facilitate the creation of more helpful reviews to provide higher value to their customers (Mudambi and Schuff, 2010).

3.2. Reviews

Online product reviews are "peer-generated product evaluations posted on a company or third party website" (Mudambi and Schuff, 2010, p.186), including either or both rating scales and open-ended comments. Humans have a selfless desire to benefit others, even at their own expense, and since online product review contributions are dependent on voluntary effort (Qiao et al. 2020), it is seen as the primary motivation behind such contributions. In fact, financial incentives can even have a negative impact on these prosocial contributions (Qiao et al. 2020).

Their aim is to provide future customers with relevant attribute- or usage-oriented information and warn against purchasing low-quality products (Qiao et al. 2020). Online product reviews have grown in importance in the past years as a source of additional information for customers, helping to reduce their level of uncertainty

regarding the quality of a product and its suitability to their varying needs (Kwark et al. 2014).

The effect of product reviews on manufacturers and e-commerce websites also has been extensively studied by scholars and has a wide range of literature. Besides providing additional information for customers, review systems have been found to improve the perception and the perceived usefulness of the website (Mudambi and Schuff, 2010; Kumar and Benbasat, 2006). Such community features also create value for the e-commerce website through increased revenue (Delone and Mclean, 2003; Mudambi and Schuff, 2010). Even though positive ratings have a visibly positive effect on sales (Mudambi and Schuff, 2010), it has been found that negative reviews can have an even higher negative impact (Yin et al. 2016).

Although online product reviews are seen as an important source of information (Kwark et al. 2014), it is important to mention that reviews are not entirely objective. Based on the assessment of the literature the factors affecting review objectivity and thus overall review quality can be organized into 2 groups: biases and reviewer strategies, which are going to be outlined in the rest of this chapter.

Biases

Since review contribution happens on a voluntary basis, not all relevant customers write reviews, in fact, approximately one out of a thousand customers contribute a review about their personal experiences after purchasing a product on Amazon (Hu et al. 2017). This is due to the time and effort needed to write and submit a comprehensive review, which leads to the emergence of self-selection biases (HU et al. 2017). One of the most frequent self-selection biases influencing review contribution is the underreporting bias when customers with extreme experiences (either positive or negative) are more likely to take the time and effort to write a review than those with more or less neutral feelings, which leads to the asymmetric, positively skewed and bimodal distribution of online product reviews (Hu et al. 2017).

Another factor influencing self-selection is the disconfirmation effect, which is the difference between the customer's expected and experienced assessment of the purchased product (Ho et al. 2017). A reviewer is more likely to contribute with a review if the disconfirmation is larger, moreover, the left rating might not be identical to the customer's post-purchase evaluation but is heavily influenced by the extent of

disconfirmation (Ho et al. 2017). It is also noteworthy that customers who only write reviews on an occasional basis are more sensitive to disconfirmation than those who write reviews more frequently (Ho et al. 2017). Hu et al. (2017) demonstrated that customers are aware of the presence of self-selection biases and that they attempt to compensate for them by using additional review parameters, however, they cannot completely account for them since they are not fully rational.

Lastly, those who already purchased a product perceive the product as having a higher value than those who did not purchase the product yet - which is called the endowment effect (Hu et al. 2017). Therefore, just by making the purchase the customer is already biased toward the product, affecting their ability to assess the product objectively.

Reviewer strategies

The product review literature showed that besides biases, reviewer strategies also have a huge impact on review quality. When there are no monetary incentives involved, online reputation and attention from other customers have an influential effect on reviewers' contributions (Shen et al. 2015). For example on e-commerce websites with a review ranking system, such as Amazon - which takes into account the number of reviews a reviewer has produced, their helpfulness rate, and other review quality factors -, reviewers tend to avoid crowded review segments as they are sensitive to competition, and they also tend to provide more differentiated ratings which have an influence on their overall rank (Shen et al. 2015). Shen et al. (2015) also outlined that review contributors with different reputations tend to provide different ratings and use different tones in their writings for strategic purposes. Such strategic maneuvers lead to the trend that reviewers with higher reputation costs tend to provide reviews that conforms to the overall opinion of the given segment (Ho et al. 2017).

3.2.1. Review helpfulness

Yin et al. (2021) highlight the significance of understanding the components of helpful reviews for e-commerce websites to promote informative content and to provide guidelines for reviewers to encourage helpful review creation hence ultimately increasing customer satisfaction. This chapter will examine the concept of helpfulness

votes, the benefits of helpful reviews, and the different factors influencing the perception of helpfulness in customer-generated product reviews.

Helpfulness votes

Even though there is a vast amount of additional information available through the increasing number of product reviews, it can lead to the problem of information overload, therefore it is in favor of the customers to receive fewer, but more helpful reviews (Yin et al. 2014). The concept of *helpfulness* in our context refers to providing reviews that contain information that is considered helpful by potential customers in their decision-making process (Mudambi and Schuff, 2010). However, in research helpfulness, and the level of helpfulness are identified through the use of *helpfulness votes* which can be given by customers to reviews (Qiao et al. 2020). Besides indicating helpful content to customers, helpfulness votes are frequently used by reviewers as an indicator of the attention they have gained through their reviews (Shen et al. 2015).

Benefits of helpful reviews

After reviewing the available literature, it became clear that facilitating the creation of more helpful reviews would be in favor of customers, e-commerce websites, and reviewers. Customers benefit from more helpful reviews by receiving meaningful and high-quality information for the decision-making process and having decreased exposure to information overload (Yin et al, 2014). E-commerce businesses benefit from more helpful reviews as helpful reviews have a higher influence on the purchasing decision (Yin et al, 2014), positively influence sales (Mudambi and Schuff, 2010), and provide a means for the website to offer a potentially higher value to its customers (Mudambi and Schuff, 2010). Last but not least, reviewers are also incentivized to produce more helpful reviews as top reviewers can monetize the attention and reputation they earned through writing helpful reviews (Shen et al. 2015).

Variables influencing perceived helpfulness

One of the aims of this literature review is to gain meaningful insights about reviews and review helpfulness that can be used to identify variables for our machine learning model. In the upcoming paragraphs, I would like to summarize the variables

that appeared throughout the literature review and have an impact on the perception of review helpfulness. I will present these variables by organizing them into 4 categories: *rating extremity and length, prior beliefs, sentiment and emotions,* and *identity disclosure*.

I grouped *rating extremity and length* into one category, for two reasons: 1) they are the most conspicuous features of a review, 2) and the different effects they have on search goods and experience goods. Even though customers with extreme experiences are more likely to contribute to reviews (Hu et al. 2017), there are controversial findings in research regarding whether extreme or moderate reviews are perceived as more helpful. Mudambi and Schuff (2010) demonstrated that this controversy might be due to the difference between search goods and experience goods. Their findings suggest that moderate ratings are perceived as more helpful in general in the case of experience goods, where the assessment of the product is more subjective, however for search goods, moderate reviews proved to be less helpful than extreme (either positive or negative) reviews (Mudambi and Schuff, 2010). Moreover, the length of the review usually increases the perception of helpfulness in search goods, but this effect is smaller for experience goods (Mudambi and Schuff, 2010), presumably because these reviews contain more subjective content in general.

While some studies suggest that negative reviews tend to be perceived as more helpful, other studies found the opposite. Since consumers are influenced by their *prior beliefs* when processing information, this contradiction can be due to confirmation bias - when the reader prefers a review that confirms his/her initial beliefs (Yin et al. 2016). Yin et al. (2016) suggest that these prior beliefs are formed based on the product's average rating, which is the first indicator of the product overall, that the customer encounters. Therefore if the customer comes across a review that is in contradiction with their initial beliefs formed based on the average rating of the product, the customer will perceive that rating as less helpful. In summary, these findings suggest that the helpfulness rating of a review will be dependent on the average rating of the product (Yin et al. 2016).

Emotions can substantially influence how reviews are perceived by the readers as they can perceive and distinguish between emotions embedded in the text, which are processed faster and perceived automatically (Yin et al, 2014). Yin et al (2014) propose that beyond simple *sentiment* (positive – neutral – negative tone of the text)

different embedded emotions might have a different impact on the perceived helpfulness of the review, regardless of their positive or negative nature. For example, embedded anxiety is suggested to be perceived as more helpful compared to embedded anger – even though both of them are negative emotions (Yin et al, 2014). This might be due to the fact that anxious reviewers are expected to have invested a higher cognitive effort during the creation of the review, compared to angry reviewers (Yin et al, 2014). Furthermore, angry reviewers -and people in general- are perceived as communicating less comprehensively, and less carefully, with difficulty seeing the outside picture of their current situation (Yin et al, 2021). They are also perceived as unable to regulate their feelings which compromises their ability of reasoning thus negatively influencing the value of the review (Yin et al. 2021).

Forman et al. (2008) demonstrated that social identity plays a major role in how customers respond to information disclosed by the reviewer when processing information. They found that identity descriptive information revealed by the reviewer is often used by customers to supplement or replace information about the product when evaluating review helpfulness and that *identity disclosure* in reviews is positively correlated with perceived helpfulness (Forman et al. 2008). They also found that identity disclosure is a better indicator of helpfulness when the reviews are ambiguous (Forman et al. 2008).

3.3. Summarizing the challenges of identifying and predicting helpfulness

This chapter highlighted that identifying who will write a helpful review is quite challenging (Liu et al. 2021). There are many aspects that could potentially influence review objectivity and review quality such as self-selection biases (underreporting bias and disconfirmation effect) and reviewer strategies for gaining higher attention and a better online reputation. Waiting to see which reviews turn out to be more helpful takes time, as many readers have to read and evaluate them, however, natural language processing and machine learning can provide ways to assess what will be considered helpful based on previous reviews (Liu et al. 2021). However, a machine learning model needs high-quality features to make accurate predictions. During this chapter, several variables were identified as having an influential effect on perceived

helpfulness, and these variables will serve as the basis for the choices of feature selection. These features will be the following:

- Rating
- Text length
- Prior beliefs
- Sentiment polarity
- Emotions
- Review text

Rating and *text length* are easily extractable characteristics of a review, ratings are usually included in most datasets, while text length can be extracted with a simple character count. *Prior beliefs* are much harder to include, as our dataset does not contain the average rating of the product that was being reviewed. It includes however the ratings count of every product reviewed, which can be used as an indicator to the extent of how much average ratings could have influenced the readers - that the higher the ratings count, the higher impact the average rating has on the readers' prior beliefs. *Sentiment polarity* and *emotions* will be extracted with the use of natural language processing, to identify the overall positive-neutral-negative tone of the text, as well as the underlying distinct emotions. Last but not least, *review text* will be also included as a predictor feature. Since machine learning models cannot interpret raw textual data, I will utilize Topic Modeling as a way of identifying the main underlying topics best representing each review and the dataset as a whole, and use the probability distribution of the topics as features.

Identity descriptive information was also found to be an influential variable for review helpfulness, however, the extraction of such information is out of the scope of the dissertation, as it requires more extensive text analysis.

The next chapter will outline the theoretical foundations of building the predictive model.

4. Theoretical foundations

The dissertation is considered qualitative research. Throughout the whole dissertation, I am gathering insights on the concept of review helpfulness in order to better understand what are the specific components that make reviews perceived to be helpful. This dissertation takes a qualitative approach to both natural language processing and supervised machine learning as well. Natural language processing methods are utilized to extract the underlying topics best representing the review dataset and the emotions embedded in the reviews. At the same time, the supervised machine learning part of the dissertation doesn't seek to predict a continuous variable, but a set of fixed classes indicating whether a review is perceived as helpful or not.

4.1. Natural Language Processing

Natural language processing is a wide range term, used to cover any manipulation and analysis done to everyday human language (Bird et al. 2009). In this dissertation, I will utilize some of the most common techniques in order to convert the review text into numerical features that are interpretable by machine learning algorithms and extract valuable information from reviews that were identified during the literature review process. The natural language processing part of this dissertation relies heavily on 2 main frameworks: spaCy and Gensim. Both of these open-source frameworks have been developed for natural language processing tasks, with spaCy excelling on large-scale text processing and information extraction, while Gensim is a library primarily for topic modeling.

Before any type of analysis, it is important to prepare the data first. In the case of natural language processing, we have to convert the raw text into analyzable bits of information that are interpretable by the models. During this preprocessing, we extract additional information from the data, filter out parts that wouldn't add anything meaningful to the results but otherwise would increase complexity, and make sure to keep the format of the data consistent across the whole dataset.

SpaCy provides a good solution for the pre-processing tasks necessary before text analysis. The core objects of spaCy are tokens and Docs. Tokenization is one of the foundations of many natural language processing tasks (such as topic modeling and sentiment analysis) and involves segmenting the text into smaller parts, such as

words, punctuations, etc. (Honnibal and Montain, 2017). The segmentation is done with spaCy's pre-trained libraries, which are trained to recognize many useful attributes about the specific tokens, for example, part-of-speech tags (noun, verb, adverb, etc), named entities (real-world objects eg.: famous people, organizations, geographical locations, etc.), syntactic dependencies, and the lemmatized "root" form of the token (eg.: the lemmatized form of both "surprises" and "surprising" is *surprise)*. The segmented text is then stored in separate tokens including their obtained attributes, and the tokens are stored in the Doc objects, which keep the tokens organized and let the users access them in a structured manner (Honnibal and Montain, 2017).

During the natural language processing tasks carried out in this dissertation, I will utilize spaCy's pre-trained English libraries during pre-processing, to split the reviews into tokens and obtain some of the tokens' attributes (specifically: part-of-speech, lemmatized form, and sentiment polarity).

Gensim on the other hand provides a framework for building large-scale topic models. It is built around a few core concepts: *vector, dictionary, corpus,* and *model*. As mentioned earlier, it is crucial to convert our text-containing documents in order to provide a representation of our data that can be mathematically manipulated. This numerical representativeness is achieved through the *vectorization* of the words. In this case, a numerical vector is assigned to each word, with the same words having the same vector across the whole dataset. This approach allows the model to approximate the meaning of the words (as similar words have similar vectors) and represent the word in a reduced dimensional space.

The dictionary is the collection of all the tokens appearing in the corpus after the pre-processing. It contains every word only once, assigning a unique ID for all of them. Meanwhile, the corpus is the collection of documents (or reviews), which is used as the input for the model to look for common themes and topics. The corpus contains the unique ID for every token of the documents, and their frequency. This is called the bag-of-words approach, which is a vectorization technique that only looks at the occurrence frequency of the tokens in the documents and completely ignores their order. Last but not least, a model refers to the transformation of the vectorized corpus. It is an algorithm trained to transform the documents represented as vectors to another vector space based on different approaches (Rehurek and Sojka, 2011). One example

of a model is the Latent Dirichlet Allocation, that will be discussed in the following chapter.

4.1.1. Topic Modelling

Topic modeling is a natural language processing technique to reveal, discover and annotate the underlying themes in a large number of texts (Kherwa and Bansal, 2019). The foundation of topic modeling is the Vector Space Model which extracts semantic structures from word usage, utilizing a term-document matrix (Kherwa and Bansal, 2019). It is based on the bag-of-words approach, in which one ignores the order of the words in the text and focuses on the occurrence frequency of the words in the given documents, to reveal the underlying structures (Rehurek and Sojka, 2011). The output of topic modeling is a given number of topics that represent the underlying themes in the collection of the documents the best, each topic containing the probability distribution over terms where the given term belongs to that given topic. It is important to mention that there are other approaches to vectorization as well that are order-sensitive, compared to bag-of-words, for example Long Short-Term Memory (LSTM) networks. However, since they are not utilized in this dissertation, they fall out of the scope of the dissertation and won't be discussed further.

Latent Dirichlet Allocation (LDA)

There are multiple approaches for conducting topic modeling. The aim of this dissertation is not to compare different topic modeling methods, and choose the best possible option, but to provide a viable example to the reader of how to extract certain features from the reviews utilizing topic modeling. I chose to use Latent Dirichlet Allocation (Blei et al, 2003) as it is one of the most common algorithm choices for topic modeling (Silge and Robinson, 2017) and has an interactive visualization tool, which makes it easier to interpret the results of the model. The two main principles of LDA are that each document is a mixture of topics and each topic is a mixture of terms, which are being estimated simultaneously. Using these two principles, LDA determines the mixtures of terms that represent the given number of topics the best, while also identifying the mixture of topics that identify the given documents. It allows documents to overlap regarding content, instead of being separated (Silge and Robinson, 2017). It is important to note here, that LDA topic modeling does not give us the exact topic names that appear throughout the documents. Rather it outputs a

collection of terms that are most likely to appear together, these terms being the best representation of the n number of topics, and the topics being the best representation of the whole corpus.

Due to the qualitative nature of the dissertation, the final measure of the topic model will be how the model is able to extract meaningful topics from the reviews that can be used as input for the predictive model. However, when it comes to the evaluation of topic models, we can also utilize a more quantitative approach to get an understanding in advance (before building the predictive model) about how good our model is. There are multiple metrics one can choose from, two of the most widespread being *perplexity* and *topic coherence*. I chose to focus on topic coherence over perplexity because higher perplexity does not necessarily result in human interpretability, while topic coherence tries to model human judgment by measuring the semantic similarity between terms in a topic, thus trying to differentiate interpretable topics from topics created by statistical inference (Kapadia, 2019).

There are multiple ways of measuring topic coherence from C_v, C_umass, C_npmi, etc. For this dissertation, I chose to use the C_v coherence score as the main evaluation metric, as this is the default metric provided by Gensim. The C_v measure is based on a sliding window, one-set segmentation of top words, and an indirect confirmation measure using normalized pointwise mutual information (NPMI) to identify dependencies between words, and cosine similarity to check the semantic similarity between the words (Kapadia, 2019). Basically, what it does is it identifies the top words within each topic and examines the words that appear close to each other in the text by checking how frequently they appear together compared to appearing alone, and how similar and related the vector of the words are to each other.

There was no clear-cut answer found for the range of coherence score, or how high it should be, however, the rule of thumb is generally to maximize it as a higher coherence score indicates higher semantic similarity. Nevertheless, topic coherence should not be the only metric to rely on from the many available approaches. Higher coherence might indicate higher semantic similarity, however, it does not necessarily result in more meaningful topics that can indicate helpfulness. The results of the topic model also should be evaluated based on how meaningful topics it produced for predicting perceived helpfulness. See feature selection later.

In this dissertation, topic modeling had been used in order to transform the textual content of the reviews into numerical features. This has been done in the form of extracting the probability distributions of the topics being present in each of the reviews, which can be fed to the machine learning model as numerical features (one for every topic) in order to make predictions.

4.1.2. Simple Sentiment Analysis

Sentiment analysis or opinion mining focuses on trying to capture the underlying sentiment the writer expresses towards a subject (Yi et al, 2003). It analyzes a text based on the contained words, the words having assigned a specific sentiment polarity score, and assesses whether the underlying tone in the text is positive, negative, or neutral. Simple sentiment analysis only assesses the overall direction of the opinion or feelings inside the text, it is less complex than distinct emotional states, which I will cover in the next chapter. In this dissertation, the overall sentiment of the reviews is being used as a feature, that ranges between -1 (negative), 0 (neutral), and 1 (positive). The sentiment polarity score will be calculated for the reviews with spaCy's sentiment analysis extension.

4.1.3. Emotion Detection

Emotion detection is a subpart of natural language processing, and a branch of sentiment analysis (Acheampong et al, 2021). It is more detailed than simple sentiment analysis, focusing on trying to extract the underlying distinct emotional states from text or speech. Detecting emotions is not an easy task, and this part of natural language processing lacks research focus because there is an unavailability of context extraction methods for texts, and some texts include multiple emotional expressions leading to the problem of classification ambiguity - just to mention a few reasons (Acheampong et al, 2021).

According to Acheampong et al, (2021) despite the lack of focus, the utilization of pre-trained transformer-based models achieved good results in the recent years. These models are utilizing a model architecture proposed by Vaswani et al (2017), that allows for significantly more parallelization and can extract relational context better than previous models (Vaswani et al, 2017; Acheampong et al, 2021).

The choice of the model for extracting the underlying emotions from the reviews was EmoRoBERTa (Kamath et al, 2022), a BERT (Bidirectional Encoder

Representations from Transformers) based open-source, pre-trained model for text classification. The foundation of EmoRoBERTa, BERT is a model by Devlin et al. (2019) that is used to pre-train natural language processing models for various tasks, such as text summarization and sentiment analysis. BERT is first pre-trained to understand the text through Masked Language Modeling (when certain tokens are being hidden in the text and the model is trained to recognize the hidden tokens based on the context) and Next Sentence Prediction (when a model is trained to identify whether 2 sentences are logically connected or coherent), then it is optimized for the specific task it is intended to solve (Acheampong et al, 2021). To be more precise, EmoRoBERTa is based on a Robustly Optimized BERT model which is a fine-tuned version of BERT, having a larger pre-training dataset (160 GB compared to the 16 GB of the original BERT).

EmoRoBERTa has been chosen for several reasons. First of all, BERT-based solutions are the most explored transformer-based classifiers for emotion detection (Acheampong et al, 2021). Second, EmoRoBERTa is capable of classifying documents up to 510 tokens into 27 distinct emotions plus one neutral category, therefore it is possible to map out a wider range of emotions than the 6 basic emotions (happiness, sadness, anger, fear, disgust, surprise) identified by Paul Ekman psychologist - as most of the other classifiers do. The pre-trained classifier is capable of identifying the following emotions: *admiration, amusement, anger, annoyance, approval, caring, confusion, curiosity, desire, disappointment, disapproval, disgust, embarrassment, excitement, fear, gratitude, grief, joy, love, nervousness, optimism, pride, realization, relief, remorse, sadness, surprise* + *neutral,* with 47% precision and 66% recall (Kamath et al, 2022), which can be considered as a fair performance giving the 28 available emotions to choose from (more on the evaluation metrics later). During the writing of this dissertation, no peer-reviewed studies have been found about the evaluation of EmoRoBERTa, presumably due to the fact that the Kamath et al paper only got published in 2022. However, EmoRoBERTa has approximately 45 thousand downloads each month, therefore it is a frequently used solution for emotion detection.

4. 2. Supervised Machine Learning

I already presented the definition of machine learning in Chapter 3.1.2. When talking about machine learning, we can differentiate between two main directions, namely supervised learning which focuses on the creation of predictive models, and unsupervised learning which focuses on the building of descriptive models (Boehmke and Greenwell, 2020). Since the aim of this dissertation is to create a model that can be used for predicting the perceived usefulness of reviews, it can be considered a supervised approach. When building a predictive model, we try to create a model to provide accurate predictions of a given output (target) using other variables (or features) (Boehmke and Greenwell, 2020). During the literature review process, I already identified the main features that will be used to predict the target: review helpfulness.

Supervised problems can be categorized into two main directions: *1) regression problems*, when the objective is to predict a numeric outcome, that falls on a continuum (eg.: sales price), and *2) classification problems,* when the aim is to predict a categorical outcome, that is either binary or multinomial (eg.: yes/no or helpful/non-helpful/neutral) (Boehmke and Greenwell, 2020). In this dissertation, the aim is to predict predefined categories as outcomes (whether a review is perceived as helpful, non-helpful, or neutral), therefore in this chapter, I will focus solely on classification problems.

4.2.1. Classification

In a classification problem, we want to predict the probability of an observation belonging to a specific class, and by default, the class with the highest predicted probability will become the predicted class (Boehmke and Greenwell, 2020). As discussed earlier, when we try to predict the outcome of a classification problem, we can distinguish between binary classification problems, when there are only two classes, and multinomial or multiclass classification problems, when there are more than two classes for the model to choose from. Many of the classification algorithms can handle multiclass tasks by default, however, some are specifically tuned for binary tasks by default (such as Logistic Regression), and their strategy has to be a bit fine-tuned in order to be able to handle more than two classes. One way of doing this is to apply the one-vs-rest approach. In this case, we fit a separate classifier for each of the

target classes, when the given class is compared against all the other classes (Pedregosa et al, 2011). For example in our review helpfulness problem, it can be interpreted as predicting the probability of a review belonging to the helpful class against the rest (non-helpful and neutral), to the non-helpful against the rest (helpful and neutral), and so on.

An important concept to be aware of, when building predictive models is the *bias-variance trade-off,* regardless of the classification or regression nature. Bias refers to the difference between the expected predictions of the model and the actual predictions, while variance refers to the difference in performance when we apply the model to a different dataset. Models with high bias are generally less flexible and tend to be underfitting, as they cannot capture the underlying structure of the data. Meanwhile, models with high variance are at the risk of overfitting because they can perform well with the training data, but cannot reproduce the same performance on unseen data. We generally want to achieve low bias and low variance, however, there is a trade-off between the model's ability to minimize both, as low bias generally results in high variance and vice versa. Therefore we have to aim for a balanced amount of bias and variance (Boehmke and Greenwell, 2020).

There are many classification algorithms one can choose from. This dissertation does not aim to find the best method with the highest possible accuracy for the review helpfulness problem, simply to provide a representation to the reader about how machine learning classification works, and how it can be utilized to solve the specific problem of review helpfulness prediction. Since the aim is not primarily the comparison of different classification methods, only a handful will be used in this dissertation to indicate that different approaches fit every problem differently. The chosen methods for this dissertation are Logistic Regression and Neural Networks, which are both widely applied methods for solving classification problems (Boehmke and Greenwell, 2020).

Logistic Regression

Despite the name, logistic regression is a method used to solve classification problems, instead of regression problems. It is a simple and efficient method to use for binary classification problems, however, it can also handle multiclass classification as well, utilizing the above-presented one-vs-rest strategy (Subasi, 2020). It utilizes a non-linear logistic (or sigmoid, because of the S shape) function, that can be

interpreted as the probability of a data point belonging to either class *A* or *B* (Boehmke and Greenwell, 2020). During training it basically checks the weights of the influence of every feature on the likelihood of a data point belonging to either of the classes. Even though it is a relatively simple and efficient solution for classification problems, it is bound by certain assumptions. These assumptions are the *linear relationship* between the features and the target variable, *constant variance* among the error terms, *independent and uncorrelated errors* to prevent bias, *more observations than predictors,* and *no or little multicollinearity* between features, meaning that they should not be closely related to one another. Violation of these assumptions should be avoided as they might lead to flawed predictions (Boehmke and Greenwell, 2020).

Neural Networks

Neural Networks are machine learning models that have a layered structure consisting of artificial neurons, weights, and an activation function (Bonetto and Latzko, 2020). Every neuron contains a set of weights that they receive from the previous layers, and a bias, acting as a threshold. All the inputs are being multiplied by corresponding weights, getting summed up, and thresholded by the bias. Finally, the neuron is getting its activation function applied to output a single numerical value in order to decide if the neuron will be activated in the decision-making. These internal parameters are adjusted during training within every iteration (Bonetto and Latzko, 2020).

The structure of neural networks has at least 3 layers: an input layer, *n* hidden layers, and an output layer, with the more hidden layers the network has, the deeper the neural network (Bonetto and Latzko, 2020). There is no golden rule about how many hidden layers and neurons a neural network should have, however, it is usually assumed that a deeper network with fewer neurons per layer performs better than a wider one, as the network is “forced” to extract useful patterns and optimize itself (Bonetto and Latzko, 2020).

4.2.2. Evaluation

The predictions of a classifier can be visualized with a *confusion matrix*, that maps out all the real and predicted values in a matrix in the form of True Positive, True Negative, False Positive, and False Negative, and can be seen in Figure 1. In order to interpret the confusion matrix, let's say that we only have a binary target, with helpful

and non-helpful classes, helpful being the positive class (as we want to predict helpful reviews), and non-helpful being the negative class. By looking at the True Positive and True Negative values, we can see how many data points the model managed to predict correctly. Meanwhile, False Negatives indicate the number of reviews the model predicted to be non-helpful, even though it was truly helpful, and False Positives indicate the number of reviews the model predicted to be helpful, why in real life being non-helpful. This can help us to get a better overview of what kind of mistakes the model makes, does it struggle more with predicting the helpful or the non-helpful categories? These measures can give us an insight into how many of the predicted helpful reviews are truly helpful, and how many of the truly helpful reviews did we manage to correctly predict (see precision and recall later in this chapter). The confirmation matrix can be broadened to multiclass classification as well, just switch positive and negative with the actual class names and extend the matrix.

Ground-truth class	Predicted class	
	Positive	Negative
Positive	*TP*	*FN*
Negative	*FP*	*TN*

Figure 1. Confirmation matrix (Zhou, 2021)

After visualizing the results, we can start evaluating the model according to different metrics. One of the most frequently used metrics of classification problems is *accuracy,* which measures how accurately can a classifier predict a given class of a data point. It can be calculated by dividing the number of accurate predictions by the sum of all the predictions. However, it is important to keep in mind that there are other evaluation metrics as well, that provide different evaluation results.

Two other metrics that can be used to evaluate performance from different directions are *precision* and *recall.* Precision shows how many of the positively predicted data points (let's consider 'helpful' reviews as positive) are actually positive (true positive divided by total predicted positive), while recall shows us how many out of all the positives we managed to predict (true positive divided by all the positives). Precision and recall are often contradictory, as they approach the evaluation from different directions (Zhou, 2021). The difference between these two metrics shows

how the quality of a model is relative and depends on the requirements of the specific tasks (Zhou, 2021).

To further visualize the performance of the model, we can use the *Receiver Operating Characteristics* (ROC) curve and the *Area Under the Curve* (AUC). The ROC curve is a probability curve, where the True Positive Rate is plotted against the False Positive Rate of the model. Meanwhile, the AUC measures the area under the ROC curve, capturing how well the model does in separating between classes. The higher the AUC the better the distinguishing ability of the model (Narkhede, 2018). The AUC ranges between 0 and 1. If the AUC is near 0, it means that the model is reciprocating the results, so it takes the positive class as negative and vice versa. If it is near 0.5, it means that it has no ability to distinguish between the 2 classes, which is actually the worst scenario, and if it is near 1, it means that it is able to differentiate between the classes (Narkhede, 2018). To interpret this with actual numbers, if the AUC is 0.75, it means that the model has a 75% chance of differentiating the two classes.

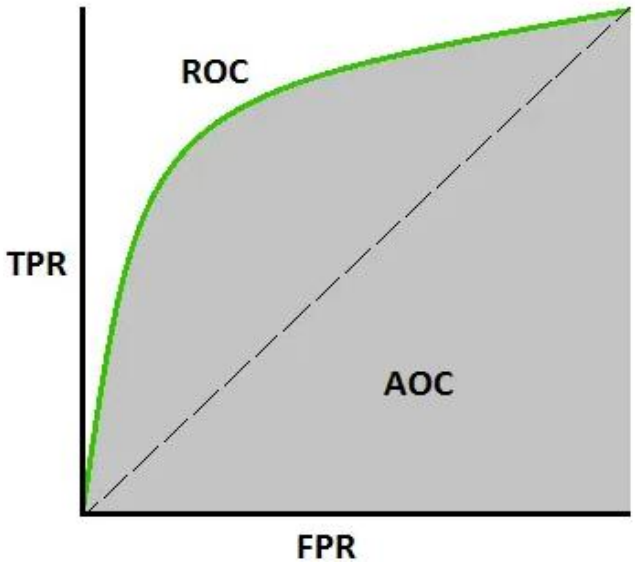


Figure 2. AUC - ROC curve (Narkhede, 2018)

It can be assumed that not all features have the same predictive ability (Zhou, 2021). Accordingly, I should also consider whether all of the topics and emotions extracted through topic modeling and emotion detection can be considered as meaningful predictors for the model. There might be certain topics that contain more valuable information for potential customers, or certain emotions that are just

perceived as more helpful in reviews. Therefore it is important to evaluate and select these features - along with the rest of the predictors - during the fine-tuning of the model, in order to minimize model complexity (so we can avoid overfitting) and maximize predictive ability. There are many approaches one can use for feature selection, such as *variance threshold* and *Recursive Feature Elimination* (RFE). By implementing a variance threshold during the pre-processing of the model, we can filter out the features that explain zero or only an insignificant amount of variance but otherwise would increase model complexity, resulting in a less generalizable model. Meanwhile, in the case of RFE, we can set the number of features we would like to include and recursively reduce the number of the features by iterating over the dataset and removing the least important features within every iteration until we achieve the desired number of features (Pedregosa et al. 2011).

5. Methods

This chapter first will introduce the main steps of the model building that are necessary to answer the 4th research question (*"Is it possible to predict the perceived helpfulness of Amazon book reviews using supervised machine learning?"*), then go into more detail with each step. The results of the steps then will be presented in the Results chapter. The 7 stages of the model building are summarized below.

1. **Data sampling:** In order to be able to effectively analyze the data, samples of smaller sizes need to be created. This reduces the computational burden on the algorithms caused by the huge amount of data, while still keeping the data representative and thus able to give meaningful results. The dataset used for the model building will also be introduced at the beginning of this section.
2. **Data preparation:** Before the analysis, the raw data might need to be transformed in order to be suitable to use as distinct features that can be fed to the machine learning model for making predictions. This step is also known as *feature engineering.* The next steps (steps 3, 4, and 5) also fall under the category of feature engineering, however, they deserve separate sections in this dissertation.
3. **Topic modeling:** Topic modeling is utilized in this dissertation to convert the textual data of reviews into numerical features that capture the content of the reviews. This must be done in order to make them interpretable for the machine learning algorithm to make predictions.
4. **Sentiment analysis:** Simple sentiment is one of the features identified through the literature review as a potential predictor of perceived helpfulness. The analysis yields the sentiment polarity score of the reviews that will be used as a feature to indicate the overall direction of the underlying emotional tone of the review text.
5. **Emotion detection:** The underlying distinct emotions are also predictor features selected through the literature review. The emotions are extracted from the review text using a pre-trained classifier that has been trained to extract the underlying emotions from the text.
6. **Target engineering:** During the target engineering stage, the variable "review/helpfulness" is prepared and transformed into a target variable. This

new variable will have 3 classes (helpful, non-helpful, neutral), which the machine learning model is built to predict.

7. **Classification**: The last step of the model-building process involves the creation of the predictive model itself based on the output of the previous steps. During this stage, a baseline model will be created (to have a basis of comparison), alongside two fine-tuned models to make predictions of review helpfulness.

In the following sections, I will explain each of the steps in depth to give the reader a more detailed insight into what has been done during the predictive model building and why.

5.1 Data Sampling

Data

The chosen dataset for the model building was downloaded from kaggle.com (Appendix 7.) and contains 3 million customer reviews for approximately 214 thousand books. Book reviews were chosen, because they are specifically experience goods. The findings of the literature review suggest that the perception of helpfulness in the case of experience good reviews is less influenced by rating extremity and text length while it works the other way for search goods (Mudambi and Schuff, 2010). Using only one category (in this case experience goods) enables the proper comparison of the results with these findings. The dataset contained 10 unique columns (*Id, Title, Price, User_id, profileName, review/helpfulness, review/score, review/time, review/summary, review/text*) and a separate file that contained the details of each book, including the ratings count.

Sampling

The data sampling has been done in two parts, one for the data preparation and the topic modeling, and one for the rest of the text analysis and the model building. This has been done for two reasons. First, the results of topic modeling for each review are dependent on the whole collection of the reviews, therefore it is better to use as large a dataset as possible. Meanwhile, sentiment analysis and emotion detection,

analyze each review individually, without being influenced by the rest. Second, emotion detection in particular is a computationally extensive process - just for comparison, LDA topic modeling on 1 million documents took 3 hours, while emotion detection on 50 thousand took 6.5. Therefore, a large random sample had been created first for the data preparation and the topic modeling, which later on was sampled further for the rest of the model building. The samples mentioned in the upcoming chapters will refer to their respective samples.

Determining the sample size for topic modeling is a trade-off for including as many documents (in our case reviews, hereafter documents) as possible in order to achieve the highest possible representativeness, but without making it computationally too burdensome to analyze (the whole dataset contained approximately 2.45 billion characters). Due to this trade-off, a one-million-size random sample has been created from the original dataset. A sample of this size is representative enough to explore the underlying topics in the review texts while still possible to analyze with 16 GB of RAM and a 4-core CPU.

For the rest of the model building, the one-million-sized sample already containing the results of the topic modeling had been resampled into a smaller, 50-thousand sample. There was no need for stratified sampling in either of the sampling processes, as both random samples kept the original distribution of the target variable.

Further sampling will be included in the classification section of this chapter, as dividing the dataset into training and testing samples is a crucial part of the predictive model building and it requires all the previous steps to be completed.

5.2 Data preparation

During this stage, the raw data needed to be processed and transformed into usable features that can be fed to the machine learning model. The selected features were identified in Chapter 3.3 and included *ratings, text length, prior beliefs, sentiment polarity, distinct emotions,* and *review text.* The exploration and preparation of the first three variables were conducted during this step.

Review rating was included in the original dataset and needed no prior preparation, however, the distribution of the 5 rating classes was checked to get a better overview of the feature.

As text length was not included in the original dataset, it had to be calculated from the review text with a simple character count and appended to the sample. The distribution of the dataset was negatively skewed as expected, and further steps will be taken during the model building in order to normalize this skewness.

Prior beliefs are not a variable that can be easily measured. The original aim was to use the average ratings of the books as the basis of prior beliefs, however, it was not available in the dataset. Instead, it has been decided that the rating count of the distinct books will be used, as an indicator of the *extent* of the role prior beliefs might have played in the perception of helpfulness (rating averages with higher rating counts have a larger influential effect, than those with lower rating count). The books' rating count had to be extracted from the dataset containing the details of every individual book, and attached to the review dataset based on the books' titles, which were present in both the review and the book details dataset. After visualizing the distribution of the ratings count, it was visible that approximately 45% of the reviews lacked a rating count. In a similar situation, one has many options, for example getting rid of the missing values in general, or trying different forms of imputation with zero, mean, median, etc. Unfortunately, the origin of the missing values is not known, whether its due to bad data handling, zero rating count, etc., Therefore in order to keep the cleanliness of the dataset, all the missing values have been deleted, since our dataset was still large enough for making predictions (545 thousand).

There were 2 variables that even though were not selected features during the literature review - in fact, they were not even mentioned in the assessed literature - were included in the original dataset and decided to be kept for experimental purposes. These two variables were the *date* of the reviews and the *price* of the book when the review was made.

The date of the review was indicated with timestamps in the original dataset, this had to be converted into datetime first, then divided into *Year, Month,* and *Day* columns to be interpretable for our model. The dataset included reviews between 1996 and 2014. One must note that using year as a predictor is not necessarily a wise choice for future predictions, because past years will never occur again in the future. The only reason I left the Year variable in the dataset, is because it can be a valuable feature for classifying past events (the more time has passed since the review, the more chance it has a higher helpfulness rating). This impact of the time passed is hoped to

be somewhat balanced with the use of the *helpfulness ratio* as a target, which will be explained in the target engineering section.

During the exploration of the price variable, it was found that approximately 82% of the observations contained only null values in the price column, therefore this column has also been got rid of those that were not included in the selected features.

5.3. Topic modeling

Topic modeling was conducted in order to transform the review text that is not interpretable for machine learning algorithms into numerical features that can be mathematically manipulated. In this chapter, I walk through the necessary tasks that lead up to training the topic model. The pre-processing tasks were conducted utilizing the framework of spaCy that had been presented in Chapter 4.1. These tasks included the tokenization of the reviews, keeping only the lemmatized version of the tokens and filtering out all the unnecessary tokens such as punctuations, stopwords (eg.: a, an, the, for, to, etc., using spaCy's built-in stopword list), number-like tokens (eg.: 10, ten, etc.), symbols, etc. as they only increase the complexity of the model without contributing anything to the quality of the results.

Besides, the documents had been filtered based on part-of-speech tags, to contain only verbs, nouns, adjectives, and adverbs thus increasing the quality of the topics. The inclusion/exclusion of different part-of-speech tags had been previously tested on smaller samples using coherence checks (coherence score for each topic in the range of n topics) and interpretability as the basis of comparison. The Notebook files for these checks can be found in Appendix 3. As a last step, bigrams had been created to include the word pairs that appear at least 1000 times together as one.

After all the pre-processing had been completed, it was time to create the dictionary of the topic model, containing every token appearing through the collection of the reviews assigned to a unique integer ID. Besides, the corpus had been created as a vectorized representation of all the documents, using the bag-of-words approach - containing only the unique token ID and frequency count for each of the documents, ignoring the exact order of the tokens.

Having the dictionary and the corpus ready, it was only left to set the hyperparameters and train the model. The chosen topic modeling method was

Gensim's LDA Multicore model which can be utilized for parallel computing in order to shorten the training time of the model. The number of topics I trained the model to provide was set to 19. This number was identified through the coherence check used for the comparison of samples including different part-of-speech tags. The winning sample had its coherence score peak at 19 topics, which indicated that the semantic similarity between the terms of the topics would be the highest if the model was aiming for finding 19 topics.

After the model was trained, the results were visualized with pyLDAvis (Figure 3) to provide an interactive, straightforward visual representation of the topics to the viewer for easier interpretation. By visualizing the results, it was possible to get an overview of the topic distances in the vector space to compare how distant/overlapping are the topics, the term frequency ratio of the topics compared to the rest of the corpus, and the term-topic exclusivity.

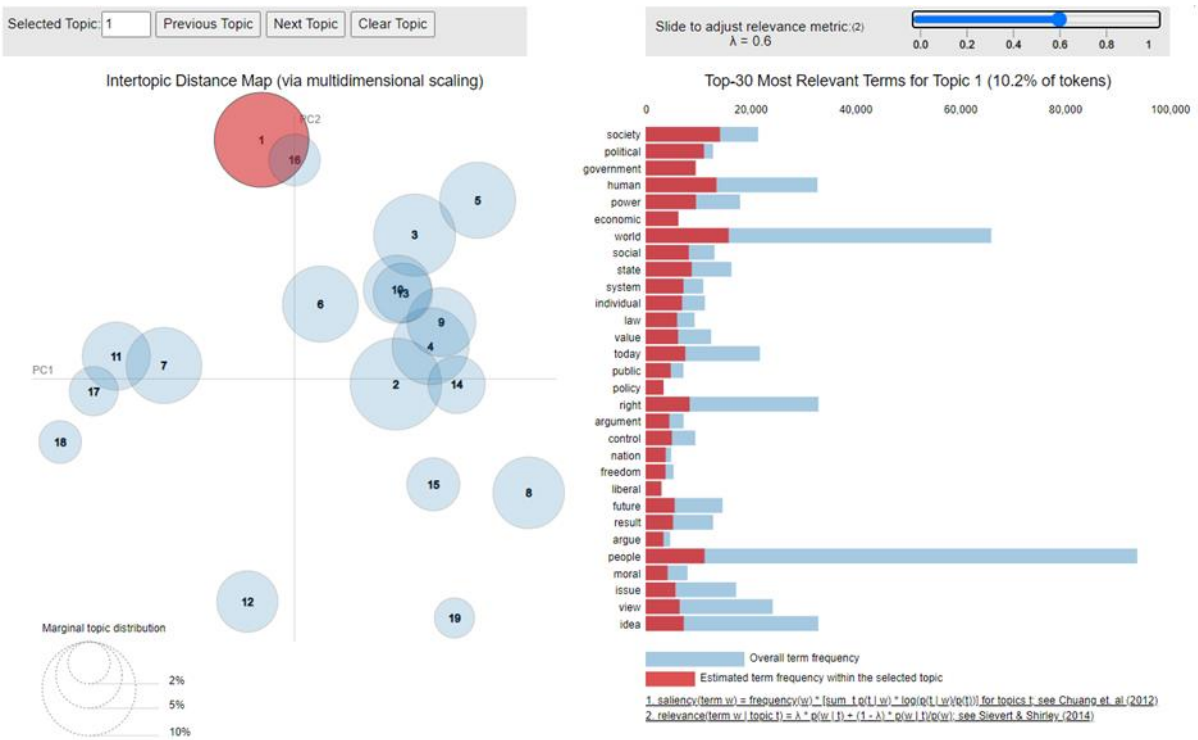


Figure 3. pyLDAvis topic model visualization

Since pyLDAvis was not built for large-scale analysis, as a standard practice with larger datasets, a smaller (200 thousand) sample was created from the corpus to visualize the results. Since the same dictionary and the same model had been applied to the sample of the corpus, the 200 thousand sample was high enough for representativeness. It is important to note, that nevertheless, corpus sampling gives

us the same topics as using the whole corpus, it messes up the order of topics, so only the content will be identical, not the order (both orders can be found in Appendix 5.).

In order to convert the results of the model into interpretable features for a machine learning algorithm, I extracted the probability distribution of the 19 topics for all of the reviews in the dataset. This gives us 19 new features that contain the probability of each of the topics being present in the individual reviews.

5.4. Sentiment analysis

Sentiment analysis was conducted on the sample in order to extract the sentiment polarity of the reviews from the review text, as one of the features identified through the literature review. The analysis was done by utilizing spaCy's spacytextblob extension built for sentiment analysis, and adding it to the pre-trained pipeline I used for tokenization and attribute extraction (part-of-speech tags, etc.) in topic modeling. The result of the analysis, sentiment polarity, had been concatenated to the sample as the *polarity* column.

5.5. Emotion detection

Emotion detection had been conducted to extract the final features identified through the literature review process: distinct emotions. The classifier that had been used for emotion detection (EmoRoBERTa) is a pre-trained transformer-based model, so in order to get it to work and analyze the reviews, first I had to set up a different virtual environment with TensorFlow that serves as a base for the transformers library, and the transformers library. EmoRoBERTa has a limit of 510 tokens per document. After plotting the word count distribution of the sample, it was visible that the proportion of the documents above the limit was marginal (only 3.7%) and could be treated as outliers. Therefore the sample had been filtered down to the reviews that contained a maximum of 510 tokens. As 510 words do not necessarily mean 510 tokens all the time, in order to stay under the limit, all the stop words (eg.: a, to, for, etc.), punctuations, number-like words, and non-alphabetical characters were removed, as certain tokenizers tend to split these into multiple tokens, and they don't add much to the text regarding sentiment.

After all the preparations, all that was left was to pass each document through EmoRoBERTa's pipeline. The results - the probability distribution of the emotions for each review - had been converted into a data frame, where every column represented one type of emotion and every row represented one document, as can be seen in Figure 4. This data frame had been later merged into the main sample.

admiration	neutral	gratitude	pride	approval	optimism	excitement	amusement	joy	relief	...
0.988481	0.006527	0.001576	0.000623	0.000619	0.000617	0.000350	0.000155	0.000120	0.000117	...
0.000513	0.997062	0.000051	0.000008	0.000903	0.000405	0.000019	0.000105	0.000020	0.000005	...
0.012520	0.913324	0.003006	0.000045	0.016417	0.044521	0.000159	0.000159	0.000444	0.000803	...
0.001971	0.956461	0.000097	0.000107	0.000863	0.002978	0.000091	0.000210	0.000040	0.000037	...
0.903452	0.002280	0.001271	0.000134	0.055195	0.004547	0.015594	0.005075	0.006088	0.000578	...
...	...	...	...	...	...	...	...	...	...	...

Figure 4. Dataframe with emotion distribution.

5.6. Target engineering

The perceived helpfulness of the reviews was indicated in the review/helpfulness column of the original dataset. However, it needed to be transformed into a suitable target variable for a machine learning algorithm to predict, which is also known as target engineering. The aim of this process was to transform the content of the review/helpfulness column into a target variable with three classes: helpful, non-helpful, and neutral.

Review helpfulness was indicated in the dataset with a raw helpfulness ratio, that is the number of helpfulness votes out of the number of total votes (eg.: helpfulness of 3/5 means that the review received 3 helpful votes out of the 5 total votes it received). The helpfulness ratio is a better indicator of helpfulness than a simple helpfulness vote count because it balances out the extent of review visibility. Let's say we have two reviews, one with a helpfulness ratio of 2/2, and the other with a helpfulness ratio of 3/6. If we only assess the nr. of helpfulness votes the reviews received, it can seem, that the second review was more helpful, because it received more helpful votes. However, if we also take into account the total number of reviews as well (thus review visibility), the difference becomes clear: one review with a helpfulness ratio of 1, while the other with a helpfulness ratio of 0.5.

I was calling it the “raw helpfulness ratio” because it only included the ratio in a simple string form, like “3/6”. This raw form had to be converted into integers and calculated for each review, to get a number between 0 and 1, indicating how helpful the review was. Based on this number, the reviews had been assigned to one of the three classes. If the helpfulness ratio of a review was equal to 0, it got assigned to the ‘neutral’ class (the review didn’t receive any votes), If the ratio was lower or equal to 0.5 it got assigned to the ‘non-helpful’ class (the review received equal or more non-helpful votes than helpful votes), and if the ratio was higher than 0.5, it got assigned to the ‘helpful’ class. The target class distribution can be seen in Figure 5. below, with a 46-39-15 ratio between helpful-neutral-nonhelpful classes.

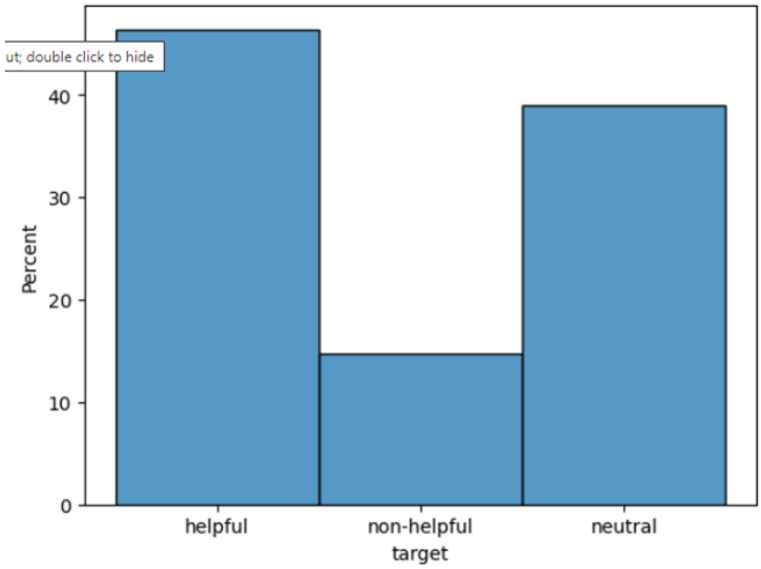


Figure 5. target distribution

5.7. Classification

After all the data preparation had been completed, it was time to use the data to train supervised machine learning models, in order to test whether it is possible to predict the perceived helpfulness of the reviews. During the preparation, the sample (hereafter: dataset) had been reduced to 48 159 observations, and 55 columns, including the target. The final columns are the following: *Year, Month, Day, review/score, ratingsCount, txt_len, Topic 1, Topic 2, Topic 3, Topic 4, Topic 5, Topic 6, Topic 7, Topic 8, Topic 9, Topic 10, Topic 11, Topic 12, Topic 13, Topic 14, Topic 15, Topic 16, Topic 17, Topic 18, Topic 19, polarity, neutral, admiration, gratitude, approval, optimism, caring, joy, relief, excitement, realization, amusement, surprise,*

disappointment, remorse, grief, disapproval, desire, annoyance, love, confusion, pride, curiosity, disgust, fear, sadness, anger, embarrassment, nervousness, target. After the data preparation, the dataset contained only numerical features and no null values, only the target was a 3-class categorical variable.

5.7.1. Baseline model

First, I needed a baseline model that I could use as a basis for comparison during the fine-tuning of the model. I used Logistic Regression for the baseline model (as well as for one of the fine-tuned models). I separated the features from the target and split both into training and testing samples. It is a widespread technique to split the dataset into not 2 but 3 samples, namely training, validation, and testing, the first two being used during the fine-tuning of the model, and keeping the last one to the final test, to evaluate how the model performs on completely unseen data. However, I decided to apply K-fold cross-validation instead, in order to maximize the size of the sample that can be used for training. This way, instead of having 60%-20%-20% training, validation, and testing samples, where only the 60% sample can be used for training, I had an 80% training sample, that has been resampled *k* times into further training and validation sets, each part of the sample being used for validation once. After all the fine-tuning, the final model could be tested on the hold-out test set, which was completely unseen for the model. The holdout test sample contained 20% of all the observations. I applied stratified sampling to keep the distribution of the target classes within the samples, and kept the classes slightly unbalanced, as it was the natural distribution of the random sample and the main dataset as well, and the differences were not that major.

The components of the model had been stored in a pipeline to ensure that the same steps are being executed for both the training and testing samples. The pipeline for the baseline model included 4 steps. 1) Variance threshold, in order to filter out all of the features that can explain less than 1 percent of the variance. This way we can reduce the complexity of the model to some extent, as we are aiming to lower the variance explained previously in the bias-variance trade-off. 2) Yeo-Johnson transformation, in order to normalize our features and achieve a more symmetric distribution, as most of them are skewed by nature (probability distributions, text length, etc.). 3) Feature scaling, in order to standardize the scale of the features, by

all of them having the same scale and prevent certain features from dominating others because of different scales. 4) Logistic Regression, tuned for multiclass classification (the target variable had 3 classes), using one-versus-rest strategy – see Chapter 4.2.1. The results of the baseline model will be briefly assessed in the following paragraph, in order to provide some clarity to the reader about the reason for the decisions made during the fine-tuning of the model.

After fitting the pipeline on the training dataset, and applying K-fold cross-validation with k=5, the mean accuracy of the baseline model was 60.2%. Without any major fine-tuning, the baseline model already had an accuracy that is higher than complete randomness (the average accuracy with three classes by randomness ~33%). In order to identify what could have gone wrong, I plotted the confusion matrix of the model, which can be seen in Figure 6. It is visible that the model does best by predicting the helpful class, and does worse by predicting the non-helpful, especially with the recall of non-helpful, meaning that a lot of truly non-helpful reviews were predicted as other classes. The fact that the model struggles with identifying non-helpful reviews cannot be solely due to the fact that the target classes had been kept unbalanced, as the ratio of accurately identified non-helpful reviews is way smaller than the difference would suggests - later this scenario had been tested with a balanced target, and the accuracy stayed the same.

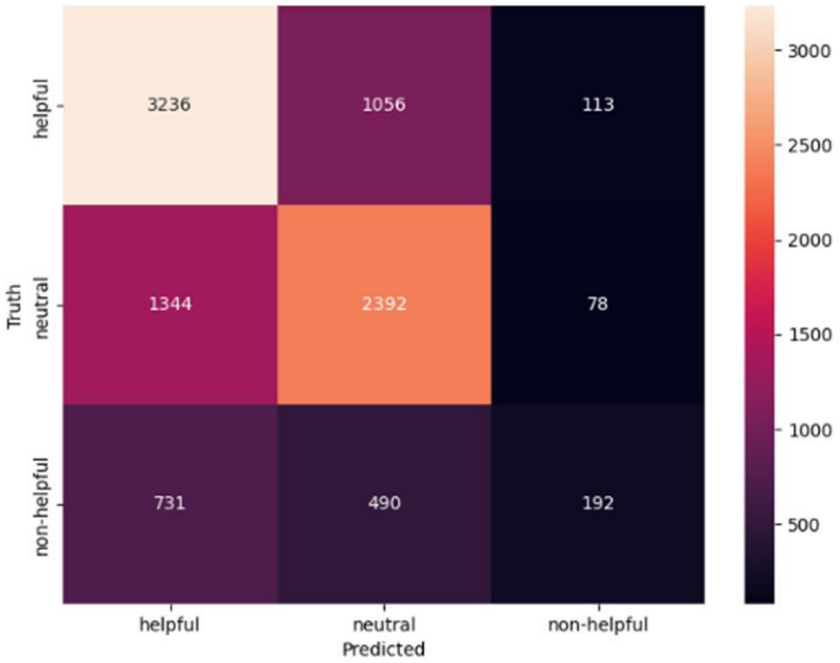


Figure 6. Confusion matrix of baseline logreg model

However, the struggle of the model makes sense, since the main question of the dissertation was how to predict helpful reviews and feature selection had been done accordingly. It can be assumed that the simple lack of helpful components does not straight-forwardly indicate non-helpfulness and other yet non-identified variables might be the right indicators of non-helpfulness. Hence, different variables should be investigated in order to predict what reviews will be perceived as non-helpful.

5.7.2. Fine-tuning

Since the model clearly struggles with identifying non-helpfulness, the non-helpful class had been eliminated from the sample, keeping only the helpful and neutral classes. The neutral class had been upsampled to match the distribution of the helpful class, so now the dataset contained 50-50% helpful and neutral classes. After, stratified sampling had been applied to separate the training and testing set, based on the target column. For the final models, I used Logistic Regression and Neural Networks, and their hyperparameters had been chosen as a result of continuous experimentation, with the aim of maximizing the prediction accuracy.

As already described in Chapter 4.2.1, there are certain assumptions of Logistic Regression such as linearity, constant variance among error terms, uncorrelated errors, more observations than predictors, and no or little multicollinearity. In order to meet the assumption of constant variance, I utilized the Yeo-Johnson transformation, therefore the errors have the same variance across all the features. Since the data does not contain time-series-related observations, previous observations couldn't influence the errors, therefore the assumption that they are not correlated to each other is also met. Moreover, the dataset is large enough to have more observations than features.

However, it cannot be completely ruled out that there is a non-linear relationship between all the features and the target variable, and that there is no multicollinearity, as certain topics and emotions might be very similar to each other (eg.: very negative emotions such as fear and nervousness). That is why Recursive Feature Elimination will be applied, to remove the offending and non-predictive features one by one, until the number of optimal features is reached. Moreover, another classification method (Neural Network) will be applied besides Logistic Regression, which is capable of predicting non-linear relationships as well.

Separate pipelines had been created for both models containing similar, but slightly different steps. Both pipelines contained Yeo-Johnson transformation and Scaling, but they contained different feature selection methods. For Logistic Regression, I used Recursive Feature Elimination, to identify the top n features that can explain most of the variance in the model. The final number of features had been defined through experimentation, and it turned out that using 40 features out of the 54 produced the highest accuracy for our Logistic Regression model. Unfortunately, it is not possible to use RFE with neural networks, therefore I used Variance Threshold to filter out near-zero variance, setting the threshold to 1%.

Even though I found that neural networks with deeper structures tend to perform better, the deeper structure resulted in more overfitting than the wider structure. Through several experimentations, the number of optimal nodes had been identified at 56 in one hidden layer, which produced the best predictions with the least amount of overfitting.

First, the pipelines had been fitted on the whole training set. Then I used k-fold cross-validation to check the generalizability of the findings using 5 folds and an 80-20% training-validation ratio. This way I could assess how the models perform on different samples, and detect potential under or overfitting. Last but not least, the evaluation of the models had been done on the test set. The results and the evaluation of the models can be found in the next chapter.

6. Results

This chapter contains the results of certain steps introduced in the previous chapter that are necessary for the reader to evaluate the findings of the dissertation. These steps are the following: data preparation, topic modeling, sentiment analysis, emotion detection, and classification. The results of the data sampling and target engineering had already been included in Chapter 5.1. and Chapter 5.6. as they were necessary for the detailed explanation of the following steps.

6.1. Result of data preparation

Data preparation had been done to transform the raw data into usable features for the machine learning model and to simultaneously explore and better understand the prepared data including rating, text length, ratings count, and date.

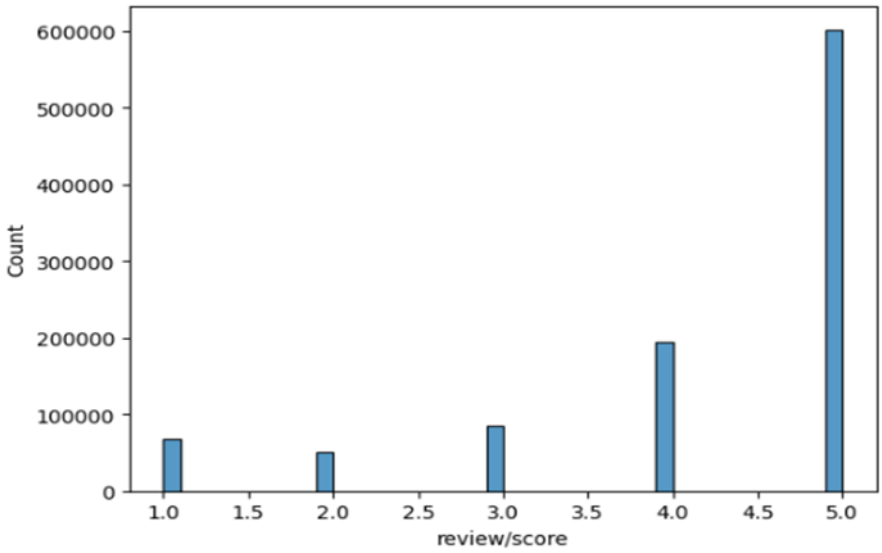


Figure 7. Distribution of ratings

The rating distribution of the sample was expected to be heavily influenced by the underreporting bias (positively skewed, bimodal). The rating distribution (Figure 7) is in fact positively skewed, however, it is not as bimodal as I expected based on the findings of the literature review. There are certainly a bit more negative reviews than neutrals if we group the one- and two-star ratings together, but the difference was expected to be more significant. The distribution still confirms the literature review findings about the underreporting bias, when people with extreme experiences are more likely to write reviews, but in the case of books, it is mainly positive experiences. The cause of this positive skewness and less bimodality might be due to the effect that

if a reader immensely dislikes a certain book, they are more likely to abandon it completely, instead of taking the time to finish it and write a bad review.

	text length	ratings count	year	month	day
min	1	1	1969	1	1
max	32576	4895	2013	12	31
average	824	270			
median	517	10			
most frequent			2012	1	6

Figure 8. Descriptive statistics of text length, rating count, year, month, and day

The descriptive statistics of the text length, ratings count, year, month, and day features are demonstrated in Figure 8.

6.2. Results of topic modeling

Topic modeling had been done to extract the underlying themes from the review texts, thus transforming the text into numerical, interpretable features for the machine learning algorithm.

The final composition of part-of-speech tags included in the training of the model and the final number of topics were determined through experimentation. I created different samples containing different mixtures of part-of-speech tags and conducted coherence checks with them in the range of n topics. Only including verbs and nouns produced a higher coherence score (0.41) with a 10 thousand data sample than including all the part-of-speech tags (0.39), or verbs, nouns, adjectives, and adverbs (0.34). However, in terms of relevancy, the topics of the latter sample provided more descriptive topics in general, while the other samples turned out to be more thematic, mainly revealing the different book genres in the dataset. The sample containing both verbs, nouns, adjectives, and adverbs also showed more stable results as the number of topics increased, making it possible to aim for more topics in the analysis. Hence I decided to progress including verbs, nouns, adjectives, and adverbs. By increasing the sample size from 10 thousand to 100 thousand, the coherence score of the sample increased, peaking at 0.42 at 19 topics - which also set the topic number target at 19.

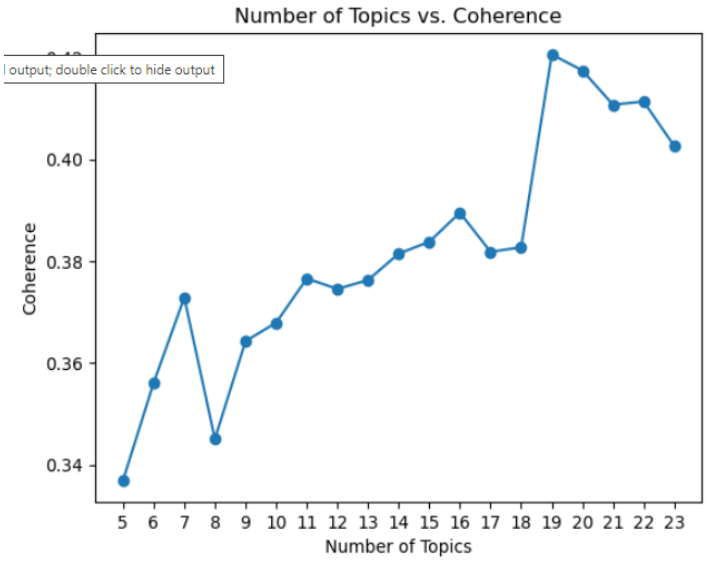


Figure 9. Coherence check for the sample including verbs, nouns, adjectives, and adverbs

The LDA model was trained to provide 19 topics from the collection of reviews as a result. As the exact topic names are not given by the model, the 19 topics had to be interpreted by assessing the top 15 keywords of each topic and identifying similar themes connecting the terms of each topic manually, using the author's judgment. The interactive visualization improved the process of interpreting the top terms and helped better understand the underlying themes of the topics. The 19 topics identified by the LDA model are the following:

- Topic 1: Politics
- Topic 2: Cognitive effort
- Topic 3: Novel characteristics
- Topic 4: Opinion on the plot
- Topic 5: Book attributes (internal content)
- Topic 6: Experience
- Topic 7: Family
- Topic 8: Recommendation
- Topic 9: Improvement/instructions
- Topic 10: Personal development (not a genre)
- Topic 11: Fantasy
- Topic 12: Young age
- Topic 13: History
- Topic 14: Opinion on adaptation
- Topic 15: Book attributes (external)
- Topic 16: Philosophy/Religion
- Topic 17: War

- Topic 18: American History/Western
- Topic 19: Education

The LDA model identified several book genre-related topics (eg.: *Fantasy, War,* etc.) that are present in our book-review dataset, moreover, it managed to recognize several more descriptive topics that characterize reviews, such as *Opinion on plot, Book attributes, Recommendations,* etc. The exact keywords for every 19 topics are included in Appendix 5. , both for the 200 thousand and the one-million sample.

The coherence score of the final model was 0.445, which is a fair result, however, can be further optimized. As described in Chapter 4.1.1., coherence measures the semantic similarity of the topics in our corpus, and the general aim is to maximize this score as much as possible. However, it should not be the only metric to rely on when evaluating the results of our topic modeling model. The topics given by the model are easily interpretable and descriptive in terms of review content. Moreover, besides the major book genres that are naturally included (since these types of books are being reviewed), the topics contain subjects such as recommendations, experience, and opinions. This indicates that the model managed to capture the whole concept of reviews: providing personal opinions on the products and making recommendations. Hence, the model managed to find topics that are a sufficient representation of the content of the review texts.

6.3. Results of sentiment analysis

The simple sentiment analysis that provided the overall emotional tone for the collection of reviews has been plotted and can be seen in Figure 10. The calculated sentiment polarity scores follow a nice, evenly distribution, however, a bit positively skewed, which indicates that the average sentiment of the collection of reviews is moderately positive. Furthermore, the sample has more reviews with extremely positive sentiments than with extremely negative ones. Even though the majority of the reviews are extremely positive based on the ratings, the results of the sentiment analysis suggest that the majority of the reviews were written using only moderately positive language.

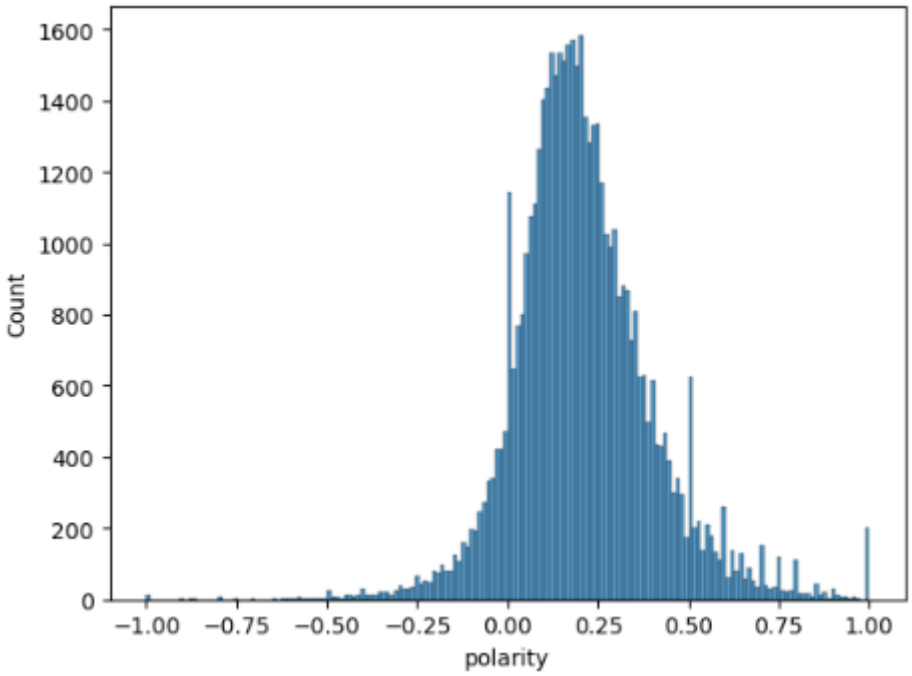


Figure 10. Sentiment polarity distribution of 50k sample

6.4. Results of emotion detection

Using emotion detection, the probability distribution of 28 distinct emotions was extracted from the review text, to use them as features when predicting the perceived helpfulness of reviews. The average probability of each emotion being present in the reviews can be seen in Figure 11. The probability of a review containing admiration or neutral is quite overrepresented in the sample. The number of neutral emotions in the sample is in alignment with the distribution of the sentiment polarity, however, it is not clear whether admiration counts as a moderately positive or extremely positive emotion. In case it is a moderately positive emotion, it would also fit the distribution of sentiment polarity, however, it cannot be ruled out that the overrepresentation of these 2 emotions is caused by certain biases of EmoRoBERTa.

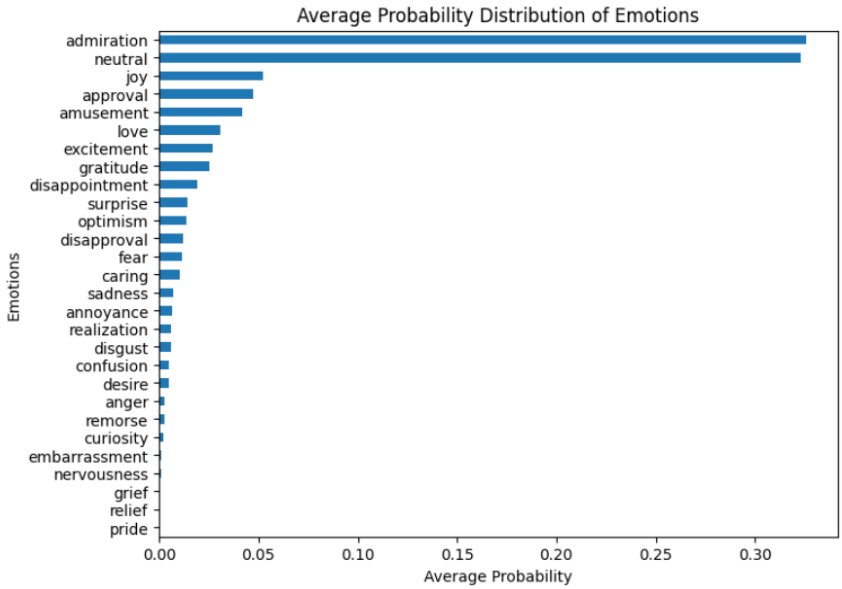


Figure 11. The average probability of emotions in review text

6.5. Results of classification

The classification stage included the creation of the predictive models using different machine learning algorithms. During this stage, I created a baseline model for a basis of comparison (Logistic Regression), and 2 fine-tuned models (Logistic Regression, Neural Networks) which can be compared with the baseline model on how accurately they can predict review helpfulness. This chapter only includes the detailed results of the fine-tuned models, as the results of the baseline model had already been presented in Chapter 5.7.1., to provide some clarity to the reader regarding the decisions that had been made during the fine-tuning.

During the classification stage, pipelines were created for each model to ensure that the same set of steps are being applied to both the training and testing datasets. Fitting the fine-tuned model pipelines on the training dataset resulted in 70% accuracy for Logistic Regression and 78% for Neural Networks. Afterward, I used k-fold cross-validation to check the generalizability of the findings using 5 folds and an 80-20% training-validation ratio. The Logistic Regression model produced the same mean results (70%) in the k-fold cross-validation, however, the Neural Networks resulted in 73.4% mean accuracy, indicating that the results are a little bit overfitting.

The evaluation of the models had been done on the test set by plotting the confirmation matrix as presented in Figure 12. The figure demonstrates that both models did slightly better with predicting the helpful class than the neutral. Moreover,

both models resulted in better recall (LogReg: 71.4%, Neural Net: 75%) than precision (LogReg: 69%, Neural Net: 73%). This means that both models did a better job at predicting the helpful class out of all the truly helpful reviews, than predicting the helpful class out of all the helpfully predicted reviews.

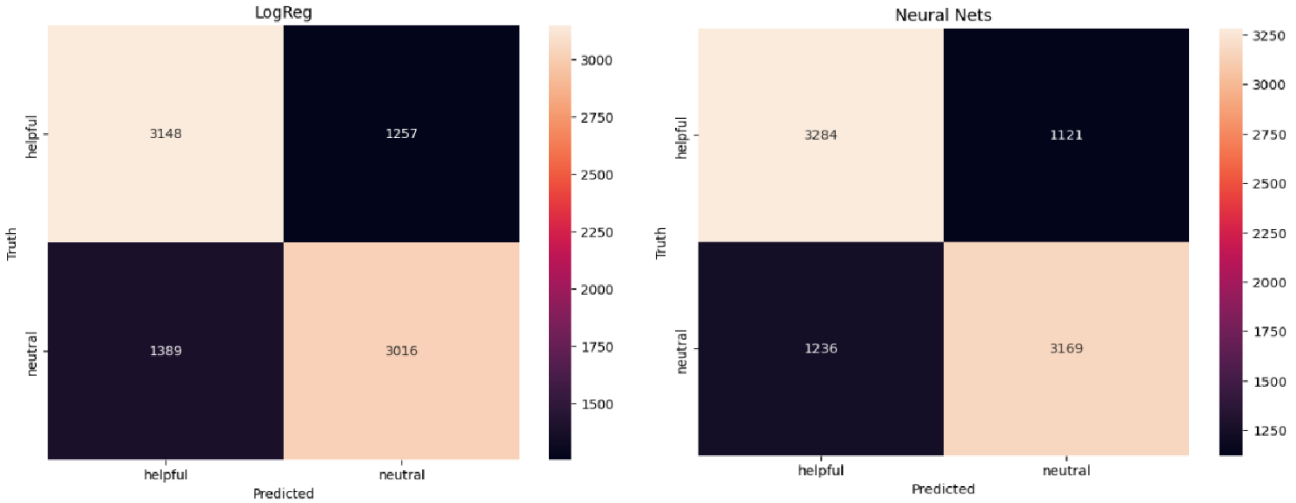


Figure 12. Confirmation matrix of fine-tuned LogReg and Neural Networks

The Receiver Operating Characteristics also have been plotted for both models (Figure 13.) including the Area Under the Curve. As is demonstrated on the ROC curves below, the Neural Network did a better job, since the model has an 81% chance of being able to distinguish between the helpful and neutral classes, while the LogReg has a slightly lower 77% chance.

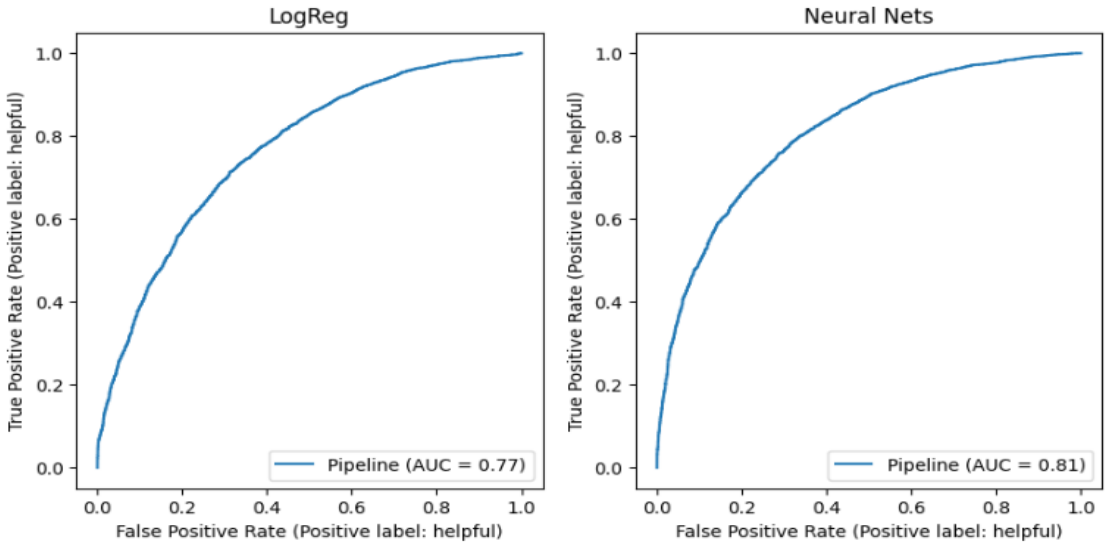


Figure 13. ROC-AUC for fine-tuned LogReg and Neural Networks

As already described in Chapter 4.2.2., not all features have the same predictive ability. It is assumed that certain topics or emotions are perceived to be as more helpful than others, and in order to reduce the complexity of the Logistic Regression model, the number of features had been reduced to 40 with the use of RFE, removing the least predictive features one by one. Removing 14 features actually improved model performance, suggesting that not all the topics and emotions are equally important to our model. While some of them explain a lot of variance, some are negligible. The eliminated features are the following in order, with the last being the least predictive feature: *gratitude, Topic 5 (Opinion on adaptation), desire, neutral, Topic 2 (Experience), Month, sadness, remorse, txt_len, pride, review/score, Year, ratingsCount, Day.*

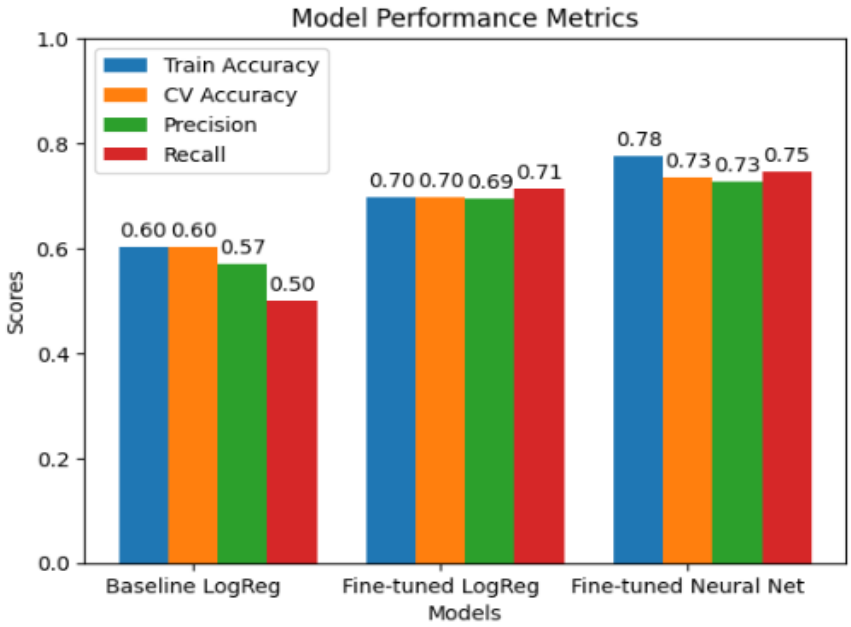


Figure 14. Model performance metrics

The comparison of the results with the baseline model is demonstrated in Figure 14. It is certain that eliminating the non-helpful class increased all the metrics of the model. It is also clear that Neural Networks did an overall better job at predicting helpfulness than the fine-tuned Logistic Regression. As mentioned in Chapter 5.7.2, the non-linear relationship between the features and the target variable cannot be ruled out, and the fact that Neural Networks outperformed Logistic Regression supports this statement. Predicting helpfulness is an overall complex problem, and not necessarily linear. Some neutral reviews might be neutral instead of helpful not because they lack the components of helpful reviews, but simply because no one ever reads them, which makes it more difficult for a machine learning algorithm to predict

helpfulness. Regardless, the results are much better than pure randomness, therefore they are a good indication that it is possible to use machine learning algorithms to predict the perceived helpfulness of reviews.

7. Discussion

This chapter contains the assessment of each of the 4 research questions, outlined in Chapter 1, to show how they were answered in the dissertation. Afterward, potential improvements will be discussed as the direction for future research, with the practical implications of the findings. Last but not least, the chapter will include the collection of the main limitations of the dissertation, that are necessary to objectively assess the findings.

7.1. Research question findings

RQ 1: What is the value of online product reviews and what are the concrete components that generate value?

In Chapter 3.1.2 I introduced the 30 elements of customer value identified by Almquist et al. in 2016, addressing 4 types of customer needs: functional, emotional, life-changing, and social impact. The more elements a product or service has to offer, the higher customer loyalty it can achieve, however, none of them can compensate for the lack of quality, which is the foundational element of customer value. The dissertation identified the 3 elements of value online product reviews contain, all of them being on the functional level, namely: quality, informing, and reducing risk. These 3 elements together capture the true value of online product reviews, which is providing additional quality information about the product for potential customers, to reduce risk by helping them make a better purchasing decision. Hence, the dissertation found the answer to the 1st research question.

RQ 2: What are the challenges of online product review objectivity?

The dissertation answered the 2nd research question by identifying two major challenges to the objectivity of online customer reviews: biases and reviewer strategies. Reviewers are influenced by certain biases when writing reviews which inherently influence review objectivity. The biases found by the dissertation are the underreporting bias, when customers with extreme experiences are more likely to write reviews (Hu et al. 2017), the disconfirmation effect, when a customer with a larger difference between the expected and experienced assessment is more likely to write a review (Ho et al. 2017), and the endowment effect, when someone who had already

purchased a product is biased towards perceiving it as more valuable (Hu et al. 2017). In the case of the underreporting bias, additional supporting evidence had been found in the dataset used in the dissertation, as the distribution of the reviewer ratings was positively skewed towards the 5-star ratings. This majority of extremely positive ratings indicate the presence of the underreporting bias.

Besides certain biases, online reviewers are motivated to gain a better online reputation and more attention with their reviews (Shen et al. 2015). This leads to the application of reviewer strategies such as avoiding crowded review segments, providing differentiated ratings on purpose, and writing reviews that conforms to the underlying opinion of the given segment, thus having a major impact on review objectivity.

RQ 3: What are the variables that influence the perceived helpfulness of online product reviews?

The aim of this research question was to identify variables that can be used as high-quality input features for the machine learning model. During the literature review, I identified 6 distinct variables that influence the perceived helpfulness of online product reviews: rating extremity, length, prior beliefs, sentiment, distinct underlying emotions, and identity disclosure. Out of these 6 variables, the first 5 were used as features alongside review text (in the form of topic modeling) for training the predictive model. Identity disclosure was left out, as the extraction of identity disclosive information would require more extensive analysis and was out of the scope of the dissertation.

However, not all features provide equal predictive abilities, therefore these variables had been filtered during the fine-tuning of the model using Recursive Feature Elimination in order to increase model performance and identify the least predictive features. Out of all the variables identified for this research question, the following had been filtered: ratings, text length, prior beliefs (rating count), and 5 emotions (desire, neutral, sadness, remorse, pride). Furthermore, 2 topics (opinion on adaptation, and experience) and all the date-related variables had been eliminated. The full list of eliminated features in order can be found in Chapter 6.4.

The fact that both rating and text length had been eliminated during RFE provides supporting evidence to the findings of the literature review according to which

more extreme ratings are perceived to be less helpful in the case of experience goods, and text length is, in general, a less influencing factor of perceived helpfulness as well (Mudami and Schuff, 2010). Since average ratings of the product were not available in the original dataset, a rating count had been used to indicate the extent of prior beliefs. The elimination of the ratings count feature indicates that the course taken to substitute average ratings was not sufficient and didn't result in a feature with any major predictive ability.

According to Yin et al. (2014), different distinct emotions might have a different impact on the perceived helpfulness of online product reviews. The findings of this dissertation provide additional supporting evidence for this claim since a handful of emotions (desire, neutral, sadness, remorse, pride) were eliminated during RFE due to not having the same predictive ability of helpfulness as the rest of the emotions. Besides the features above, 2 topics (opinion on adaptation, and experience) and all the date-related features had been eliminated because of having little or no predictive ability of review helpfulness.

The dissertation answered the 3rd research question during the literature review and moreover managed to specify this answer further for book reviews, during the model-building phase. The fine-tuned variables that influence the perceived usefulness of book reviews are the following: sentiment polarity, certain distinct emotions, and certain topics. It is important to note that the ratings feature might have had more predictive ability if the ratings included in the sample were more moderate instead of extremely positive. Furthermore, prior beliefs and identity disclosure still have the potential to provide good predictive ability, if the average ratings of the products and indicators of identity disclosive information are available.

RQ 4: Is it possible to predict the perceived helpfulness of Amazon book reviews using supervised machine learning?

All the previous research questions led to the main question of the entire dissertation, whether it is possible to use supervised machine learning to predict the perceived helpfulness of online product reviews, more specifically Amazon book reviews. The brief answer is: yes, but it is extremely difficult. The model created in the dissertation provides supporting evidence that it is possible to use machine learning algorithms to predict perceived helpfulness with higher accuracy than what would be

the result of pure randomness, both in the case of 3 target classes (helpful/neutral/non-helpful) and 2 target classes (helpful/neutral).

Predicting the non-helpful class of reviews proved to be significantly more difficult for the model than predicting helpful and neutral classes. This struggle was due to the fact that the dissertation was aiming to identify useful predictor variables for predicting helpfulness specifically, not taking into consideration that non-helpful reviews can potentially have other features that should be used for prediction, thus the same features cannot be used for predicting both classes efficiently.

Eliminating the non-helpful class from the sample significantly improved model performance, however, there is still room for improvement. The dissertation took a qualitative approach toward understanding the components behind review helpfulness to make predictions. It turned out that understanding the helpfulness of reviews is a reasonably complex problem, even when differentiating only between helpful and neutral reviews because there is no clear-cut difference between helpful and neutral reviews. The components that drive the perception of helpfulness in online product reviews have been identified in the dissertation. However, some reviews might stay neutral instead of receiving helpful votes because they lack visibility, not because they lack the same components as helpful reviews, which makes the job of the machine learning algorithm more difficult.

7.2. Potential improvements

It is suggested that the predictive model can be further improved in several ways outlined in this section. These ways of potential model improvements set the proposed direction for future research.

First of all, the predictive accuracy can be increased by improving some of the feature engineering steps, namely topic modeling and emotion detection. The performance of topic modeling could be further improved by comparing different topic modeling algorithms and finding the optimal hyperparameter settings through experimentation. This could potentially provide more interpretable and comprehensive topics with higher predictive ability. Based on the results of the emotion detection, it cannot be completely ruled out that the used pre-trained model, EmoRoBERTa was biased towards certain emotions. The quality of the features containing emotions can

potentially be improved by utilizing better pre-trained models for emotion detection, for more accurate predictions.

Second, utilizing additional features might also improve model performance. Features such as average product rating and identity disclosive information were found to be influencing factors of the perceived helpfulness of online product reviews, therefore using these in the right format as predictor features can also lead to better predictions. Moreover, it cannot be ruled out that there are additional variables that influence the perceived usefulness of online product reviews which were not revealed during the literature review process.

Last but not least, the predictive accuracy of the model could potentially be improved by experimenting with other classification methods and different hyperparameter settings. One way the hyperparameters could be improved is by implementing dropout layers (Srivastava et al. 2014) in the hyperparameter settings of a neural network, which can improve the model's accuracy without the threat of overfitting. The dropout layer prevents the neurons of the model to compensate for the mistakes of other neurons, making it too complex and overfitting.

It is important to note that recent Large Language Models provide a novel alternative to the methods used in the dissertation. Large Language Models include GPT-3, ChatGPT, Bard, etc. These models are pre-trained on huge amounts of text and can be further fine-tuned to perform tasks they were not originally trained for. They provide a novel approach for certain steps of the model-building process presented in this dissertation, for example, they could be tuned to identify emotions or underlying themes in the text, even to pre-classify reviews based on their helpfulness.

7.3. Practical implications

Chapter 2. outlined that both customers and e-commerce businesses benefit from review systems that provide high-quality information for potential customers, as they drive both the creation of business and customer value. Throughout the dissertation, I identified and later on, specified the key components of reviews indicating helpfulness, furthermore showed that it is possible to utilize natural language processing to extract valuable features from online product reviews, and supervised machine learning to predict the perceived helpfulness of such reviews.

The findings of the dissertation show the potential for e-commerce businesses to improve their review systems, thus facilitating the creation of more helpful reviews. Based on these findings, the dissertation proposes that it would be relevant for e-commerce businesses to complement their review system with a feature that helps reviewers write better, more helpful reviews. The feature could utilize the above-mentioned techniques to analyze the content of reviews in the text and other review characteristics, identify in advance whether a review would be perceived as helpful by other customers, and make recommendations to the writer accordingly. The dissertation only focused on the extraction of features that potentially indicate helpfulness, and the prediction of helpfulness itself, without attempting to design and create the complete feature per se. The findings of the dissertation can be used later on in the design process of such a feature of review systems.

7.4. Limitations

Naturally, the dissertation has some limitations that need to be taken into consideration when assessing the findings of the dissertation. One of the major limitations of the dissertation is the fact that the majority of the features used for training the model were not naturally present in the original dataset but had been extracted from the review text using various text analysis methods. This opens the door for the creation of biased and incorrect features.

In regards to topic modeling, the dissertation only utilized the LDA approach, without comparing it to other topic modeling methods. This has been done since the aim of this dissertation is not the comparison of multiple topic modeling methods and choosing the best, but to show how the questions of the dissertation can be solved by utilizing machine learning, however, it still sets some limitations for the dissertation. The above is also true for the choice of emotion detection model and machine learning algorithms. The used pre-trained emotion detection model is not 100% accurate and found to be possibly biased towards classifying documents as either neutral or admiration. Moreover, it is not compared to other pre-trained models in the dissertation. In regards to the machine Learning algorithms, since the main focus was not on the comparison of ML algorithms, the dissertation only utilized Logistic Regression and Neural Networks, without further exploring other alternatives. Even

though the above-mentioned choices set some limitations for the dissertation, it must be noted that they were influenced by boundaries set by the natural scope and extent of a dissertation.

The dataset used for the model creation only contained book reviews that are exclusively experience goods. It is possible that the findings would differ by using a dataset that contains other product categories or search goods.

8. Conclusion

The overall objective of the dissertation was to gain meaningful insights about the helpfulness of online product reviews, what are the specific components that are driving helpfulness, and to answer whether it is possible to predict the perceived helpfulness of online product reviews, through a hand-on example. The dissertation identified what are the components of value provided by reviews, what are the obstacles to review objectivity and thus overall quality, and what are the variables that influence the perception of helpfulness. Furthermore, it successfully created a machine learning classifier being able to predict review helpfulness, thus provided supporting evidence that it is possible to utilize machine learning to predict the perceived helpfulness of online product reviews.

References

Acheampong, F.A., Nunoo-Mensah, H. & Chen, W. (2021) Transformer models for text-based emotion detection: a review of BERT-based approaches. Artif Intell Rev 54, 5789–5829

Almquist, Senior, J., & Bloch, N. (2016). The elements of value: measuring - and delivering - what consumers really want. Harvard Business Review, 94(9), 46–.

Barrett, Davidson, E., Prabhu, J., & Vargo, S. L. (2015). Service Innovation in the Digital Age: Key Contributions and Future Directions. MIS Quarterly, 39(1), 135–154.

Bird, S., Klein, E., & Loper, E. (2009). Natural language processing with python. O'Reilly Media.

Boehmke, B. and Greenwell, B. (2020). Hands-On Machine Learning with R, CRC Press

Bonetto, R., & Latzko, V. (2020). Chapter 8 - Machine learning. In F.H.P. Fitzek, F. Granelli, & P. Seeling (Eds.), Computing in Communication Networks (pp. 135-167). Academic Press.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. J. Mach. Learn. Res. 3, null (3/1/2003), 993–1022.

DeLone, & McLean, E. R. (1992). Information Systems Success: The Quest for the Dependent Variable. Information Systems Research, 3(1), 60–95.

DeLone, & McLean, E. R. (2003). The DeLone and McLean Model of Information Systems Success: A Ten-Year Update. Journal of Management Information Systems, 19(4), 9–30.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019) "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 pp. 4171–4186

Forman, Ghose, A., & Wiesenfeld, B. (2008). Examining the Relationship Between Reviews and Sales: The Role of Reviewer Identity Disclosure in Electronic Markets.

Fügener, A., Grahl, J., Gupta, A., & Ketter, W. (2022). Cognitive challenges in Human–Artificial intelligence collaboration: Investigating the path toward productive delegation. Information Systems Research, 33(2), 678-696.

Ho, Wu, J., & Tan, Y. (2017). Disconfirmation Effect on Online Rating Behavior: A Structural Model. Information Systems Research, 28(3), 626–642.

Honnibal, M., & Montani, I. (2017). spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.

Hu, Pavlou, P. A., & Zhang, J. (2017). On Self-Selection Biases in Online Product Reviews. MIS Quarterly, 41(2), 449–475

Jiang, & Guo, H. (2015). Design of Consumer Review Systems and Product Pricing. Information Systems Research, 26(4), 714–730.

Kamath, R., Ghoshal, A., Eswaran, S., and Honnavalli, P. (2022) "An Enhanced Context-based Emotion Detection Model using RoBERTa," IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), Bangalore, India, pp. 1-6, doi: 10.1109/CONECCT55679.2022.9865796.

Kapadia, S. (2019, August 19). Evaluate Topic Models: Latent Dirichlet Allocation (LDA) - A step-by-step guide to building interpretable topic models. Towards Data Science. Retrieved May 12, 2023, from https://towardsdatascience.com/evaluate-topic-model-in-python-latent-dirichlet-allocation-lda-7d57484bb5d0

Kherwa, P. & Bansal, P. (2018). Topic Modeling: A Comprehensive Review. ICST Transactions on Scalable Information Systems. 7. 159623. 10.4108/eai.13-7-2018.159623.

Kumar, N., & Benbasat, I. (2006). The influence of recommendations and consumer reviews on evaluations of websites. Information Systems Research, 17(4), 425-439.

Kwark, Chen, J., & Raghunathan, S. (2014). Online Product Reviews: Implications for Retailers and Competing Manufacturers. Information Systems Research : ISR., 25(1), 93–110.

Li, Tan, C.-H., Wei, K.-K., & Wang, K. (2017). Sequentiality of Product Review Information Provision: An Information Foraging Perspective. MIS Quarterly, 41(3), 867–A7.

Liu, Li, Y., & Xu, S. (2021). Assessing the Unacquainted: Inferred Reviewer Personality and Review Helpfulness. MIS Quarterly, 45(3), 1113–1148.

Mudambi, & Schuff, D. (2010). WHAT MAKES A HELPFUL ONLINE REVIEW? A STUDY OF CUSTOMER REVIEWS ON AMAZON.COM.

Narkhede, S. (2018, June 26). Understanding AUC - ROC Curve. Towards Data Science. Retrieved May 14, 2023, from https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

Pedregosa, F., Varoquaux, Ga"el, Gramfort, A., Michel, V., Thirion, B., Grisel, O., … others. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825–2830.

Petter, DeLone, W., & McLean, E. R. (2013). Information Systems Success: The Quest for the Independent Variables. Journal of Management Information Systems, 29(4), 7–62.

Rehurek, R., & Sojka, P. (2011). Gensim–python framework for vector space modelling. NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic, 3(2).

Ross, Jeanne W ; Beath, Cynthia M ; Mocker, Martin (2019) Designed for Digital: How to Architecht Your Business for Sustained Success

Scientist, N. (2017). Machines that Think: Everything you need to know about the coming age of artificial intelligence. Hachette UK.

Shen, Hu, Y. J., & Ulmer, J. R. (2015). Competing for Attention: An Empirical Study of Online Reviewers' Strategic Behavior. MIS Quarterly, 39(3), 683–696.

Silge, & Robinson, D. (2017). Text mining with R : a tidy approach. (1. ed.).

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15, 1, 1929–1958.

Subasi. (2020). Practical machine learning for data analysis using python. Academic Press.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., L., Kaiser, L., and Polosukhin, I. (2017) "Attention is all you need," in Advances in neural information processing systems, pp. 5998–6008

Yi, J., Nasukawa, T., Bunescu, R., Niblack, W. (2003). Sentiment Analyzer: Extracting sentiments about a given topic using natural language processing techniques. Proceedings - IEEE International Conference on Data Mining, ICDM. 427- 434. 10.1109/ICDM.2003.1250949.

Yin, Bond, S. D., & Zhang, H. (2014). Anxious or Angry? Effects of Discrete Emotions on the Perceived Helpfulness of Online Reviews. MIS Quarterly, 38(2), 539–560.

Yin, Bond, S., & Zhang, H. (2021). Anger in Consumer Reviews: Unhelpful but Persuasive? MIS Quarterly, 45(3), 1059–1086.

Yin, Mitra, S., & Zhang, H. (2016). When Do Consumers Value Positive vs. Negative Reviews? An Empirical Investigation of Confirmation Bias in Online Word of Mouth. Information Systems Research, 27(1), 131–144.

Zhou. (2021). Machine learning. Springer.

Appendix

Appendix 1.1 - Scopus search

No.	Keyword/Searchterm	Problem	Reasoning	Search hits	Assessed	Platform	Date	Link for search
1	Predictive analytics	Information gathering on predictive analytics appearing in IS research	The main topic of the thesis is how to use predictive analytics to improve customer experience.	15	3	Scopus	23/01/23	https://www.scopus.com/resul
2	Predictive analysis	Information gathering on predictive analytics appearing in IS research	The main topic of the thesis is how to use predictive analytics to improve customer experience.	31	2	Scopus	23/01/23	https://www.scopus.com/resul
3	Machine learning	How machine learning as a topic appears in IS research	The core part of the secondary research will be a ML model building, so nit's relevant to see how the literature approaches the topic itself.	46	7	Scopus	23/01/23	https://www.scopus.com/resul
4	Artificial intelligence	AI and ML are often used interchangeably.	What are the exact differences, when can we talk about which?	66	10	Scopus	23/01/23	https://www.scopus.com/resul
5	AI	AI and ML are often used interchangeably.	What are the exact differences, when can we talk about which?	15	0	Scopus	23/01/23	https://www.scopus.com/resul
6	Reviews	What is the purpose of reviews? How do they nfluence customer decision making? does it give any additional satisfaction to those who submit the reviews? etc.	In order to analyze reviews, we have to understand reviews a bit more.	223	28	Scopus	23/01/23	https://www.scopus.com/resul
7	Review	What is the purpose of reviews? How do they nfluence customer decision making? does it give any additional satisfaction to those who submit the reviews? etc.	In order to analyze reviews, we have to understand reviews a bit more.	Had the same amount of hits as "Reviews". It is assumed that they provided the same hits.		Scopus	23/01/23	https://www.scopus.com/resul
8	Product reviews	What is the purpose of reviews? How do they nfluence customer decision making? does it give any additional satisfaction to those who submit the reviews? etc.	In order to analyze reviews, we have to understand reviews a bit more.	60	11	Scopus	29/01/23	https://www.scopus.com/resul
9	Useful Reviews	What constitutes a usefull review?	WHat is usefullness?	11	0	Scopus	29/01/23	https://www.scopus.com/resul
10	Online reviews	What constitutes a usefull review?	WHat is usefullness?	81	0	Scopus	29/01/23	https://www.scopus.com/resul
11	Usefullness	What is the definition of useful in the eyes of the customers		91	1	Scopus	29/01/23	https://www.scopus.com/resul
12	Text Mining reviews	How does text mining reviews work	LOoking older studies to check how they applied text mining methods on reviews specifically	7	1	Scopus	29/01/23	https://www.scopus.com/resul
13	Machine learning review assessment			1	0	Scopus	29/01/23	https://www.scopus.com/resul
14	Amazon	Understanding the research context	IS research conducted on Amazon	48	6	Scopus	29/01/23	https://www.scopus.com/resul
15	Value creation	No IS or feature can be considered as a sucsess without the intended users actually using it.	As we explore how to increase customer experience in reviews through predictive analytics, it is necessary to check up on value creation as well, as it is the outer context	57	4	Scopus	20/2/23	https://www.scopus.com/resul

16	IT value and system design	What system design aspects should be taken into consideration while applying ML on review assessment?	System design is the outer context of the whole topic, which we will investigate through vale creation.	24	2	Scopus	20/2/23	https://www.scopus.com/resul
17	Digital offerings	WHat is the intersect between what a company is capable of technologywise and what the customers are willing to use?		12	3	Scopus	20/2/23	https://www.scopus.com/resul
18	Value creation and Online communities	What constitutes as value in the online communities?	What are customers willing to pay for?	3	1	Scopus	20/2/23	https://www.scopus.com/resul

Excluded:	**Reason**
Business intelligence	too broad of a topic, it would make sense if Amazon itself would like to analyse reviews etc.
Big data	too broad, even though big data enables ML methods, this time I am focusing on a specific part of big data – reviews, thus in order to make the scope of the lit review more narrow it is excluded.
e-commerce	the topic of e-commerce is included in reviews, there is no need to increase the scope of the research to other e-commerce related topics
Information management	information management related studies are included through papers from the curriculum. Otherwise we are not interested in the field of research IM in itself.

Appendix 1.2 - First iteration

Collection of promising results. Selection solely based on title.

Individual literature search

Relevant
Neutral
Irrelevant

Keyword	Title	Link	Reasoning	Date	Related to searchterm no.	Relevancy
Predictive analytics	Predictive Analytics in Information Systems Research	https://doi.org/10.2307/23042796		23/01/23	1.	
Predictive analytics	Big Data, Data Science, and Analytics: The Opportunity and Challenge for IS Research	https://doi.org/10.1287/isre.2014.054		23/01/23	1.	
Predictive analytics	ASSESSING THE UNACQUAINTED: INFERRED REVIEWER PERSONALITY AND REVIEW HELPFULNESS	https://web-p-ebscohost-com.ez.statsbib		23/01/23	1.	
Predictive analysis	On the Assessment of the Strategic Value of Information Technologies: Conceptual and Analytical Approaches	https://doi.org/10.2307/25148790		23/01/23	2.	
Predictive analysis	**Mind the gap: Accounting for measurement error and misclassification in variables generated via data mining**	https://www.scopus.com/record/display.u		23/01/23	2.	
Machine learning	Expecting the unexpected: Effects of data collection design choices on the quality of crowdsourced user-generated content1	https://www.scopus.com/record/display.u		23/01/23	3.	
Machine learning	Inductive expert system design: Maximizing system value	https://www.scopus.com/record/display.u		23/01/23	3.	
Machine learning	WIll humans-in-the-loop become borgs? merits and pitfalls of working with AI	https://www.scopus.com/record/display.u		23/01/23	3.	
Machine learning	Developing a Quality of Experience (QoE) model for Web Applications	https://www.scopus.com/record/display.u		23/01/23	3.	
Machine learning	Financial incentives dampen altruism in online prosocial contributions: A study of online reviews	https://www.scopus.com/record/display.u		23/01/23	3.	
Machine learning	Crowds, lending, machine, and bias	https://www.scopus.com/record/display.u		23/01/23	3.	
Machine learning	Cognitive Challenges in Human–Artificial Intelligence Collaboration: Investigating the Path Toward Productive Delegation	https://www.scopus.com/record/display.u		23/01/23	3.	
Artificial Intelligence	The impact of fake reviews on online visibility: A vulnerability assessment of the hotel industry	https://www.scopus.com/record/display.u		23/01/23	4.	
Artificial Intelligence	A machine learning approach to improving dynamic decision making	https://www.scopus.com/record/display.u		23/01/23	4.	
Artificial Intelligence	Examining the impact of keyword ambiguity on search advertising performance: A topic model approach	https://www.scopus.com/record/display.u		23/01/23	4.	
Artificial Intelligence	When algorithmic predictions use human-generated data: A bias-aware classification algorithm for breast cancer diagnosis	https://www.scopus.com/record/display.u		23/01/23	4.	
Artificial Intelligence	Estimating the impact of “humanizing” customer service chatbots	https://www.scopus.com/record/display.u		23/01/23	4.	
Artificial Intelligence	Is AI ground truth really true? The dangers of training and evaluating AI tools based on experts’ know-what	https://www.scopus.com/record/display.u		23/01/23	4.	
Artificial Intelligence	Avoiding an oppressive future of machine learning: A design theory for emancipatory assistants	https://www.scopus.com/record/display.u		23/01/23	4.	
Artificial Intelligence	Editorial for the special section on humans, algorithms, and augmented intelligence: The future of work, organizations, and society	https://www.scopus.com/record/display.u		23/01/23	4.	
Artificial Intelligence	Cognitive Challenges in Human–Artificial Intelligence Collaboration: Investigating the Path Toward Productive Delegation	https://www.scopus.com/record/display.u		23/01/23	4.	
Artificial Intelligence	Managing artificial intelligence projects: Key insights from an AI consulting firm	https://www.scopus.com/record/display.u		23/01/23	4.	
Reviews	What makes a helpful online review? A study of customer reviews on amazon.com	https://www.scopus.com/record/display.u		23/01/23	6.	
Reviews	Examining the relationship between reviews and sales: The role of reviewer identity disclosure in electronic markets	https://www.scopus.com/record/display.u		23/01/23	6.	
Reviews	Self-selection and information role of online product reviews	https://www.scopus.com/record/display.u		23/01/23	6.	
Reviews	e-commerce product recommendation agents: Use, characteristics, and impact	https://www.scopus.com/record/display.u		23/01/23	6.	
Reviews	Service innovation in the digital age: Key contributions and future directions	https://www.scopus.com/record/display.u		23/01/23	6.	
Reviews	Anxious or angry? Effects of discrete emotions on the perceived helpfulness of online reviews	https://www.scopus.com/record/display.u		23/01/23	6.	
Reviews	The influence of recommendations and consumer reviews on evaluations of websites	https://www.scopus.com/record/display.u		23/01/23	6.	

Reviews	"Popularity effect" in user-generated content: Evidence from online product reviews	https://www.scopus.com/record/display.u		23/01/23	6.	
Reviews	Business intelligence in Blogs: Understanding consumer interactions and communities	https://www.scopus.com/record/display.u		23/01/23	6.	
Reviews	Online product reviews: Implications for retailers and competing manufacturers	https://www.scopus.com/record/display.u		23/01/23	6.	
Reviews	When do consumers value positive vs. negative reviews? An empirical investigation of confirmation bias in online word of mouth	https://www.scopus.com/record/display.u		23/01/23	6.	
Reviews	Big data, big risks	https://www.scopus.com/record/display.u		23/01/23	6.	
Reviews	On self-selection biases in online product reviews	https://www.scopus.com/record/display.u		23/01/23	6.	
Reviews	Competing for attention: An empirical study of online reviewers' strategic behavior	https://www.scopus.com/record/display.u		23/01/23	6.	
Reviews	Timing of adaptive web personalization and its effects on online consumer behavior	https://www.scopus.com/record/display.u		23/01/23	6.	
Reviews	Disconfirmation effect on online rating behavior: A structural model	https://www.scopus.com/record/display.u		23/01/23	6.	
Reviews	The impact of online product reviews on product returns	https://www.scopus.com/record/display.u		23/01/23	6.	
Reviews	Extrinsic versus intrinsic rewards for contributing reviews in an online platform	https://www.scopus.com/record/display.u		23/01/23	6.	
Reviews	Sequentiality of product review information provision: An information foraging perspective	https://www.scopus.com/record/display.u		23/01/23	6.	
Reviews	Leveraging user-generated content for product promotion: The effects of firm-highlighted reviews	https://www.scopus.com/record/display.u		23/01/23	6.	
Reviews	Mind the gap: Accounting for measurement error and misclassification in variables generated via data mining	https://www.scopus.com/record/display.u		23/01/23	6.	
Reviews	When seeing helps believing: The interactive effects of previews and reviews on e-book purchases	https://www.scopus.com/record/display.u		23/01/23	6.	
Reviews	On the spillover effects of online product reviews on purchases: Evidence from clickstream data	https://www.scopus.com/record/display.u		23/01/23	6.	
Reviews	Assessing the unacquainted: Inferred reviewer personality and review helpfulness	https://www.scopus.com/record/display.u		23/01/23	6.	
Reviews	Focus within or on others: The impact of reviewers' attentional focus on review helpfulness	https://www.scopus.com/record/display.u		23/01/23	6.	
Reviews	Anger in consumer reviews: Unhelpful but persuasive?	https://www.scopus.com/record/display.u		23/01/23	6.	
Reviews	Measuring Product Type and Purchase Uncertainty with Online Product Ratings: A Theoretical Model and Empirical Application	https://www.scopus.com/record/display.u		23/01/23	6.	
Reviews	Know Thy Context: Parsing Contextual Information from User Reviews for Recommendation Purposes	https://www.scopus.com/record/display.u		23/01/23	6.	
Product reviews	What makes a helpful online review? A study of customer reviews on amazon.com	https://soeg.kb.dk/discovery/openurl?ins		29/01/23	8.	
Product reviews	Self-selection and information role of online product reviews	https://soeg.kb.dk/discovery/openurl?ins		29/01/23	8.	
Product reviews	"Popularity effect" in user-generated content: Evidence from online product reviews	https://soeg.kb.dk/discovery/openurl?ins		29/01/23	8.	
Product reviews	Business intelligence in Blogs: Understanding consumer interactions and communities	https://soeg.kb.dk/discovery/openurl?ins		29/01/23	8.	
Product reviews	When do consumers value positive vs. negative reviews? An empirical investigation of confirmation bias in online word of mouth	https://soeg.kb.dk/discovery/openurl?ins		29/01/23	8.	
Product reviews	VOCAL minority and silent majority: How do online ratings reflect population perceptions of quality	https://soeg.kb.dk/discovery/openurl?ins		29/01/23	8.	
Product reviews	On self-selection biases in online product reviews	https://soeg.kb.dk/discovery/openurl?ins		29/01/23	8.	
Product reviews	Extrinsic versus intrinsic rewards for contributing reviews in an online platform	https://soeg.kb.dk/discovery/openurl?ins		29/01/23	8.	
Product reviews	Design of consumer review systems and product pricing	https://soeg.kb.dk/discovery/openurl?ins		29/01/23	8.	
Product reviews	Assessing the unacquainted: Inferred reviewer personality and review helpfulness	https://soeg.kb.dk/discovery/openurl?ins		29/01/23	8.	
Product reviews	Mining bilateral reviews for online transaction prediction: A relational topic modeling approach	https://soeg.kb.dk/discovery/openurl?ins		29/01/23	8.	
Usefulness	Perceived usefulness, perceived ease of use, and user acceptance of information technology	https://soeg.kb.dk/discovery/openurl?ins		29/01/23	11.	
Text mining reviews	Mining bilateral reviews for online transaction prediction: A relational topic modeling approach	https://soeg.kb.dk/discovery/openurl?ins		29/01/23	12.	
Amazon	What makes a helpful online review? A study of customer reviews on amazon.com	https://soeg.kb.dk/discovery/openurl?ins		29/01/23	14.	
Amazon	The impact of external word-of-mouth sources on retailer sales of high-involvement products	https://soeg.kb.dk/discovery/openurl?ins		29/01/23	14.	
Amazon	Recommendation networks and the long tail of electronic commerce	https://soeg.kb.dk/discovery/openurl?ins		29/01/23	14.	

Amazon	Competing for attention: An empirical study of online reviewers' strategic behavior	https://soeg.kb.dk/discovery/openurl?ins		29/01/23	14.	
Amazon	How is your user feeling? Inferring emotion through human-computer interaction devices	https://soeg.kb.dk/discovery/openurl?ins		29/01/23	14.	
Amazon	On the spillover effects of online product reviews on purchases: Evidence from clickstream data	https://soeg.kb.dk/discovery/openurl?ins		29/01/23	14.	
Value creation	The ecosystem of software platform: A study of asymmetric cross-side network effects and platform governance	https://www.scopus.com/record/display.u		20/2/23	15.	
Value creation	Online community as space for knowledge flows	https://www.scopus.com/record/display.u		20/2/23	15.	
Value creation	Bridging the service divide through digitally enabled service innovations: Evidence from Indian healthcare service providers	https://www.scopus.com/record/display.u		20/2/23	15.	
Value creation	Service innovation: A service-dominant logic perspective	https://www.scopus.com/record/display.u		20/2/23	15.	
ITvalue and system (	Expecting the unexpected: Effects of data collection design choices on the quality of crowdsourced user-generated content1	https://www.scopus.com/record/display.u		20/2/23	16.	
ITvalue and system (	Technologies for value creation: An exploration of remote diagnostics systems in the manufacturing industry	https://www.scopus.com/record/display.u		20/2/23	16.	
Digital offerings	Content or community? A digital business strategy for content providers in the social age	https://www.scopus.com/record/display.u		20/2/23	17.	
Digital offerings	Balancing IT with the human touch: Optimal investment in IT-based customer service	https://www.scopus.com/record/display.u		20/2/23	17.	
Digital offerings	Impact of communication medium and computer support on group perceptions and performance: A comparison of face-to-face and dispersed meetings	https://www.scopus.com/record/display.u		20/2/23	17.	
Value creation and o	Online community as space for knowledge flows	https://www.scopus.com/record/display.u		20/2/23	18.	

IM program literature

Keyword	Title	Link	Reasoning	Date	Related to searchterm no.	Relevancy	Course
Value creation	Value creation using the mission breakdown structure	https://biopen.bi.no/bi-xmlui/bitstream/ha		20/2/23			Project management
Value creation	Rethinking IT project management: Evidence of a new mindset and its implications.	https://www-sciencedirect-com.ez.statsb		20/2/23			Project management
Value creation	Revisiting IS business value research: what we already know, what we still need to know, and how we can get there.	https://epub.uni-regensburg.de/23082/1/		20/2/23			Information Systems Development
Value creation	Review: IT-Dependent Strategic Initiatives and Sustained Competitive Advantage: A Review and Synthesis of the Literature	https://www.proquest.com/docview/2181		20/2/23			Information Systems Development
Value creation	Managing and using information systems: A strategic approach	Not available online		20/2/23			Information Systems Strategy
Value creation	The elements of value.	https://brightspace.au.dk/d2l/le/lessons/5		20/2/23			Information Systems Strategy
Value creation	Capture more value	https://brightspace.au.dk/d2l/le/lessons/5		20/2/23			Information Systems Strategy
Value creation	Useful business cases: value creation in IS projects.	https://vbn.aau.dk/ws/files/240770222/P		20/2/23			Information Systems Strategy
System design	Information systems success: The quest for the independent variables	https://web-s-ebscohost-com.ez.statsbib		20/2/23			Information Systems Development
System design	If we build it, they will come: Designing information systems that people want to use.	https://www.proquest.com/docview/1302		20/2/23			Information Systems Development
System design	Software engineering: a practitioner's approach. Palgrave Macmillan. Chapter 9 Requiments modelling : Scenario-based models	https://brightspace.au.dk/d2l/le/lessons/2		20/2/23			Information Systems Development
System design	Apprenticing with the customer.	https://dl-acm-org.ez.statsbiblioteket.dk/		20/2/23			Information Systems Development

Artificial Intelligence	Reshaping Business With Artificial Intelligence: Closing the Gap Between Ambition and Action	https://www.proquest.com/docview/1950		20/2/23			Digital Innovation
Artificial Intelligence	**AI, the IOT, and Content: Ethics and Opportunities**	https://www.proquest.com/docview/2261		20/2/23			Digital Innovation
Artificial Intelligence	Preparing for the Cognitive Generation of Decision Support	https://web-s-ebscohost-com.ez.statsbib		20/2/23			Digital Innovation
Online communities	Special Section Introduction—Online Community as Space for Knowledge Flows	https://web-s-ebscohost-com.ez.statsbib		20/2/23			Digital Innovation
Digital offerings	Building shared customer insigths	https://kbdk-kgl.alma.exlibrisgroup.com/l		20/2/23			Digital Innovation

Appendix 1.3 - Second iteration

Relevant
Neutral
Irrelevant

Collection of promising results. Duplicates removed. Selection based on abstract

Individual literature search

							If Included			
Keyword	**TItle**	**Link**	**Date**	**Related to searchterm no.**	**Relevancy**	**Reasoning Exclusion/Inclusion**	**Focus 1.**	**Focus 2**	**Focus 3**	**Focus 4**
Predictive analytics	Predictive Analytics in Information Systems Research	https://doi.org/1	23/01/23	1		ML/predictive analytics to improve IS research. Details: Explains that IS research exclusively use explanatory analytics instead of predictive analytics while assuming predictive power automatically. The paper tries to outline how predictive analytics can help with theory generation, building, improvement, etc.		.		
Predictive analytics	Big Data, Data Science, and Analytics: The Opportunity and Challenge for IS Research	https://doi.org/1	23/01/23	1		Main topic is IS research. Details: Discusses whether the underlying questions for big data, analytics and data science are fundamentally different than those of prior IS research.		.		
Predictive analytics	ASSESSING THE UNACQUAINTED: INFERRED REVIEWER PERSONALITY AND REVIEW HELPFULNESS	https://web-p-eb	23/01/23	1		Factors that influence quality content creation in online communities. Trains a deep learning algorithm to identify personality traits of reviewers, to predict who is more likely to produce helpful product reviews.	.	.	.	
Predictive analysis	On the Assessment of the Strategic Value of Information Technologies: Conceptual and Analytical Approaches	https://doi.org/1	23/01/23	2		No connection to reviews or NLP or ML. ROI of IT in terms of strategic value. Interviews CEOs and CIOs about the value of IT		.		
Predictive analysis	**Mind the gap: Accounting for measurement error and misclassification in variables generated via data mining**	https://www.sco	23/01/23	2		Not content specific ML, but error correction methods might be relevant. Letss see later. Details: The threat of ignored errors in predictive data mining models in IS research. Focuses on error correction methods.		.		
Machine learning	Expecting the unexpected: Effects of data collection design choices on the quality of crowdsourced user-generated content1	https://www.sco	23/01/23	3		Quality content creation in online communities (at least ppl outside of the org.) Details: How to ensure that data provided by ppl outside the org. has a suitable quality. Data collection method's impact on accuracy. Maybe if I need some inspiration regarding data collection. How post processing can lead to precision gains (follow up study) - that can be very interesting.	.	.	.	
Machine learning	Inductive expert system design: Maximizing system value	https://www.sco	23/01/23	3		ML classification, but not based on online content. Inductive expert system building.		.		
Machine learning	WIll humans-in-the-loop become borgs? merits and pitfalls of working with AI	https://www.sco	23/01/23	3		Non content specific AI. Looks interesting tho, only include if in the end there is not enough research I can use. Human- AI complementarity. Argues that groups with AI ar less effective than groups without AI. MIght be interesting in a more general sense, "leaning towards AI".		.		
Machine learning	Developing a Quality of Experience (QoE) model for Web Applications	https://www.sco	23/01/23	3		User experience related to system design. Details : How to develop high quality experience web apps. Quality requirement factors from web architecture. Relevant if we want to build a web service - it should have high quality.		.		
Machine learning	Financial incentives dampen altruism in online prosocial contributions: A study of online reviews	https://www.sco	23/01/23	3		Machine learning on amazon review dataset. The aim here is to identify those who got financial incentives to make reviews.	Reviews	Machine learning	Financially incentivised reviews	
Machine learning	Crowds, lending, machine, and bias	https://www.sco	23/01/23	3		Using predictive analytics in online platforms.	Predictive analytics	Algorithm biases	Crowd lending	

Machine learning	Cognitive Challenges in Human–Artificial Intelligence Collaboration: Investigating the Path Toward Productive Delegation	https://www.sco	23/01/23	3		Benefits of using AI in collaboration with humans. Study suggests that humans working with AI can outperform both humans and AI in CLASSIFICATION TASKS, in case the delegation is from the side of AI.	Human AI collaboration	Delegation	Classification	Performance improvement
Artificial Intelligence	The impact of fake reviews on online visibility: A vulnerability assessment of the hotel industry	https://www.sco	23/01/23	4		(Fake) reviews in online platforms, but with a twist: the effect of fake review attacks towards small businesses. Review assessment (2.3 m) - assumption is that with the use of ML as they try to identify factors in big data.	Fake reviews	Review assessment	IMpact of fake review attacks	
Artificial Intelligence	A machine learning approach to improving dynamic decision making	https://www.sco	23/01/23	4		Medical study.				
Artificial Intelligence	Examining the impact of keyword ambiguity on search advertising performance: A topic model approach	https://www.sco	23/01/23	4		Effects of keyword ambiguity in online generated data. But it focuses on search engine keywords. The ambiguous keywords part can be interesting from a topic modeling point of view, but the context is quite different (keyword optimization for CTR) from whet Im searching for.	Keyword ambiguity	Topic modeling	Search engine keywords	CTR optimization
Artificial Intelligence	When algorithmic predictions use human-generated data: A bias-aware classification algorithm for breast cancer diagnosis	https://www.sco	23/01/23	4		Medical study. Too bad, algorithmic predictions on human generated data.				
Artificial Intelligence	Estimating the impact of “humanizing” customer service chatbots	https://www.sco	23/01/23	4		Non online content specific ML. HUmanizing chatbots with huour, delayed answers etc.		.		
Artificial Intelligence	Is AI ground truth really true? The dangers of training and evaluating AI tools based on experts’ know-what	https://www.sco	23/01/23	4		Risks and benefits of AI tools. Difference between AI know what and experts know how - high accuracy performance didnt meet the expectations in practice. Medical setting.		.		
Artificial Intelligence	Avoiding an oppressive future of machine learning: A design theory for emancipatory assistants	https://www.sco	23/01/23	4		Non online content specific ML. How to avoid an oppressive dystopia created by AI by creating an emancipatory assistant.		.		
Artificial Intelligence	Editorial for the special section on humans, algorithms, and augmented intelligence: The future of work, organizations, and society	https://www.sco	23/01/23	4		Non online content specific ML. Benefits of intelligence augmentation (IA) - AI and humans working together.				
Artificial Intelligence	Managing artificial intelligence projects: Key insights from an AI consulting firm	https://www.sco	23/01/23	4		NOn online content specific AI, HOw to succeed with AI projects based on the insights of an AI consulting firm. Might be relevant while considering what to pay attention to with a project using AI.				
Reviews	What makes a helpful online review? A study of customer reviews on amazon.com	https://www.sco	23/01/23	6		Helpfullness of Amazon reviews	Factors influencing the helpfullness of reviews	Search vs experience goods	Model of customer review helpfullness	
Reviews	Examining the relationship between reviews and sales: The role of reviewer identity disclosure in electronic markets	https://www.sco	23/01/23	6		How identity disclosure in reviews can affect the perception of reviews in terms of helpfullness.	Identity disclosure of reviewr	Helpfullness perception	Review and sales relationship	Missing information supplemented/replaced by identity info
Reviews	Self-selection and information role of online product reviews	https://www.sco	23/01/23	6		How ealry reviews affect long term cons. purchase behaviour.	Early reviews	Self selection bias	structure of product ratings over time	Preference differences betweel early and later buyers
Reviews	e-commerce product recommendation agents: Use, characteristics, and impact	https://www.sco	23/01/23	6		Product recommendation agents				
Reviews	Service innovation in the digital age: Key contributions and future directions	https://www.sco	23/01/23	6		Review of other studies on how to innovate services in the digital age. MIght be relevant.	Service Innovation	Service ecosystem	service platform	Value cocreation

Reviews	Anxious or angry? Effects of discrete emotions on the perceived helpfulness of online reviews	https://www.sco	23/01/23	6		Effects o discrete emotions on perception of helpfullness of online reviews	Review helpfullness	Effects of emotions in reviews	Angry vvs enxious	
Reviews	The influence of recommendations and consumer reviews on evaluations of websites	https://www.sco	23/01/23	6		Effect of reviews on the evaluation of websites	Why reviews are an important feature of B2C websites	Perceived usefullness		
Reviews	"Popularity effect" in user-generated content: Evidence from online product reviews	https://www.sco	23/01/23	6		Popularity effect: how those who become more popular reviewers produce more but more negatively evaluated reviews?	Popularity effect	Trade off between popularity and ratings of the review		
Reviews	Business intelligence in Blogs: Understanding consumer interactions and communities	https://www.sco	23/01/23	6		How to effectively collect, extract and analyze user geerated content from blogs	Framework on how to analyse user content	Buiness intelligence	Blogs	
Reviews	Online product reviews: Implications for retailers and competing manufacturers	https://www.sco	23/01/23	6		Importance and effect of online product reviews from the retailers point of view.	Importance of product reviews	Retailers	Competition	
Reviews	When do consumers value positive vs. negative reviews? An empirical investigation of confirmation bias in online word of mouth	https://www.sco	23/01/23	6		What do customers perceive more helpful - positive or negative reviews?	Positive vs negative reviews	Confirmation bias	Consumers' initial beliefs	
Reviews	Big data, big risks	https://www.sco	23/01/23	6		Threats imposed by big data. good for limitations part	Risks of big data	Moral and legal responsibility		
Reviews	On self-selection biases in online product reviews	https://www.sco	23/01/23	6		Biases in online product reviews. Acquisition bias, underreporting bias (those with extreme opinions are more likely to wrtie reviews).	Biases in reviews	Distribution of online reviews		
Reviews	Competing for attention: An empirical study of online reviewers' strategic behavior	https://www.sco	23/01/23	6		Strategic behaviour of reviewrs to gain more attention	COmpetition between reviewers	Reviewer reputation	COmpetition effect	Amazon
Reviews	Timing of adaptive web personalization and its effects on online consumer behavior	https://www.sco	23/01/23	6		Personalizing website content and recommendations to customers. Rather about recommendations and when to present personalized content based on customer preferences.	Web personalizati on	Personalized recomnedati on	Personalized content timing	
Reviews	Disconfirmation effect on online rating behavior: A structural model	https://www.sco	23/01/23	6		Likeliness of leaving a review and what review to leave influenced by disconfirmation. Soemone is more likely to leave a review/rating if the disconfirmation she experiences is larger, thus the difference between the aggregate rating and the own user experience - expected and experienced assesment of the product	Disconfirmati on effect	Likelihood of leaving a review	Difference between expectation and experience.	
Reviews	The impact of online product reviews on product returns	https://www.sco	23/01/23	6		Effect of reviews on product returns.Less relevant as it impacts the retailer's side more and we want to study the customer experience side.				
Reviews	Extrinsic versus intrinsic rewards for contributing reviews in an online platform	https://www.sco	23/01/23	6		Introducing rewards for online reviews				
Reviews	Sequentiality of product review information provision: An information foraging perspective	https://www.sco	23/01/23	6		Study on how to provide product reviews on a website - offering varying genres of product reviews to enhance customer decision making.	Provision of product reviews	How to provide different genres of reviews	useage vs attribute oriented reviews	

Reviews	Leveraging user-generated content for product promotion: The effects of firm-highlighted reviews	https://www.sco	23/01/23	6		How to use user generated content for product promotion. SOunds interesting, but we are not looking at how companies can leverage positive reviews thrugh highlighting.	Leveraging reviews	Firm highlighted reviews	Consumer skepticism towards highlighted positive reviews	
Reviews	When seeing helps believing: The interactive effects of previews and reviews on e-book purchases	https://www.sco	23/01/23	6		Interactions between previews and reviews, how previews complement reviews.				
Reviews	On the spillover effects of online product reviews on purchases: Evidence from clickstream data	https://www.sco	23/01/23	6		Reviews effect on purchases, accompanied by some ML	Reviews on purchases	Spillover effect	Machine Learning model on reviews	
Reviews	Focus within or on others: The impact of reviewers' attentional focus on review helpfulness	https://www.sco	23/01/23	6		Focus of attention: own experience / others-prospective. How, why and when the attentional focus may influence review helpfullness. HIGHLY RELEVANT!!!	Reviewers attentional focus (self/other)	Review helpfullness		
Reviews	Anger in consumer reviews: Unhelpful but persuasive?	https://www.sco	23/01/23	6		How aggressive reviews are perceived as helpful	Agressive reviews	review helpfullness	influence on reader attitude	emotions
Reviews	Measuring Product Type and Purchase Uncertainty with Online Product Ratings: A Theoretical Model and Empirical Application	https://www.sco	23/01/23	6		Reviews, but solely ratings and how they reduce uncertainty.	Only ratings	Uncretainty reduction		
Reviews	Know Thy Context: Parsing Contextual Information from User Reviews for Recommendation Purposes	https://www.sco	23/01/23	6		According to the inclusion criteria, it should be included, but for some reason it doesnt fit. Revisit later if lack of studies	Parsing contextual info	how to recommend reviews		
Product reviews	VOCAL minority and silent majority: How do online ratings reflect population perceptions of quality	https://soeg.kb.d	29/01/23	8		How do ratings reflect perception of quality. How the reviews reflect the opinion of the population at large. intereting topic but! compares the online opinion with real life opinion of other subjects. Thus not sure. Also made on physician evaluation, so quite far from ecommerce. Ill say no.				
Product reviews	Design of consumer review systems and product pricing	https://soeg.kb.d	29/01/23	8		How the design of customer review systems affect the review outcome. HIGHLY RELEVANT!!!	Design of review systems	When to use which design	Product pricing	
Product reviews	Mining bilateral reviews for online transaction prediction: A relational topic modeling approach	https://soeg.kb.d	29/01/23	8		Prediction of transaction based on topic modeling of bilateral reviews - an approach when the seller can also write a review.	Topic modeling on reviews	Bilateral reviews (seller can write review as well)		
Usefulness	Perceived usefulness, perceived ease of use, and user acceptance of information technology	https://soeg.kb.d	29/01/23	11		Usefullness and user acceptance of IT. Can be relevant from a system design perspective	User acceptance of IT systems	Perceiverd usefullness of IT systems	Perceived ease of use of IT systems	
Amazon	The impact of external word-of-mouth sources on retailer sales of high-involvement products	https://soeg.kb.d	29/01/23	14		IMpact on reviews on high invilvment products specifically. Focus on external reviews	Effects of reviews	High involvment products (eg.: cameras)		Focus on external reviews.
Amazon	Recommendation networks and the long tail of electronic commerce	https://soeg.kb.d	29/01/23	14		The effect of recommendations (reviews) on the distribution of demand among categories. The more a category is influenced by recommendations, the higher the demand. Can it be th other way around?	Reviews	Distribution of demand	Product categories	
Amazon	How is your user feeling? Inferring emotion through human-computer interaction devices	https://soeg.kb.d	29/01/23	14		Infering emotion based on human-computer interaction. Not relevant.				
Value creation	The ecosystem of software platform: A study of asymmetric cross-side network effects and platform governance	https://www.scop	20/2/23	15.		Too technical				
Value creation	Online community as space for knowledge flows	https://www.scop	20/2/23	15.		Collective knowledge floew among online commnity participants which represents value. Tacit /Explicit knowledge	Value of knowledge flows	Online comunities	Tacit/Explicit knowledge	

Value creation	Bridging the service divide through digitally enabled service innovations: Evidence from Indian healthcare service providers	https://www.scop	20/2/23	15.		Healthcare service improvement. Not relevant.				
Value creation	Service innovation: A service-dominant logic perspective	https://www.scop	20/2/23	15.		Service innovation. Is it digital tho? not clear from the abstract.	Service innovation	Service platforms	Value cocreation	
ITvalue and system design	Technologies for value creation: An exploration of remote diagnostics systems in the manufacturing industry	https://www.scop	20/2/23	16.		Remote diagnostics system (not relevant) + manufacturing industry (not relevant)				
Digital offerings	Content or community? A digital business strategy for content providers in the social age	https://www.scop	20/2/23	17.		Social media and content creation + premium services.				
Digital offerings	Balancing IT with the human touch: Optimal investment in IT-based customer service	https://www.scop	20/2/23	17.		IT based customer service. How to balance the mix of human and computer customer service "agents".				
Digital offerings	Impact of communication medium and computer support on group perceptions and performance: A comparison of face-to-face and dispersed meetings	https://www.scop	20/2/23	17.		Online meetings				

IM program literature

							If Included			
Keyword	**Title**	**Link**	**Date**	**Course**	**Relevancy**	**Reasoning Exclusion/Inclusion**	**Focus 1.**	**Focus 2**	**Focus 3**	**Focus 4**
Value creation	Value creation using the mission breakdown structure	https://biopen.bi.	20/2/23	Project manag		Value creation from the organization's perspective. Not customer value. Also focusing on the MIsion breakdown structure				
Value creation	Rethinking IT project management: Evidence of a new mindset and its implications.	https://www-scie	20/2/23	Project manag		IT project management				
Value creation	Revisiting IS business value research: what we already know, what we still need to know, and how we can get there.	https://epub.uni-r	20/2/23	Information Sy		IT business value - organisational perspective.				
Value creation	Review: IT-Dependent Strategic Initiatives and Sustained Competitive Advantage: A Review and Synthesis of the Literature	https://www.prog	20/2/23	Information Sy		IT dependent strategic initiatives - sounds like organizational perspective.				
Value creation	Managing and using information systems: A strategic approach	Not available onl	20/2/23	Information Sy						
Value creation	The elements of value.	https://brightspac	20/2/23	Information Sy		Measuring and delivering what customers really want.	What is valued by customers	How to deliver it	Elements (attributes) of value	
Value creation	Capture more value	https://brightspac	20/2/23	Information Sy		Methods for next level value creation - how to capture MORE value.	Value creation	Value capture	Methods through examples.	
Value creation	Useful business cases: value creation in IS projects.	https://vbn.aau.d	20/2/23	Information Sy		Usefullness of business cases in IS research				
System design	Information systems success: The quest for the independent variables	https://web-s-ebs	20/2/23	Information Sy		Variables that influence IS success. Social/Individual might be interesting	Variables influencing IS success	Social, Individual		
System design	If we build it, they will come: Designing information systems that people want to use.	https://www.prog	20/2/23	Information Sy		Organization's perspective				
System design	Software engineering: a practitioner's approach. Palgrave Macmillan. Chapter 9 Requiments modelling : Scenario-based models	https://brightspac	20/2/23	Information Sy		System design, too technical				
System design	Apprenticing with the customer.	https://dl-acm-or	20/2/23	Information Sy		How to observe users of a future system. Rather making work processes more effective. we are not focusing on work processes.				
Artificial Intelligence	Reshaping Business With Artificial Intelligence: Closing the Gap Between Ambition and Action	https://www.prog	20/2/23	Digital Innovat		Compare AI ambitions and efforts of companies.	AI ambition	AI effort	CUrrent picture about companies	
Artificial Intelligence	**AI, the IOT, and Content: Ethics and Opportunities**	https://www.prog	20/2/23	Digital Innovat		IoT				

Artificial Intelligence	Preparing for the Cognitive Generation of Decision Support	https://web-s-ebs	20/2/23	Digital Innovat		Not relevant, primarily about decision support.				
Online communities and innovation	Special Section Introduction—Online Community as Space for Knowledge Flows	https://web-s-ebs	20/2/23	Digital Innovat		Online communities.	Value creation	online communities	Tacit to tacit / tacit to explicit etc. knowledge	
Digital offerings	Building shared customer insigths	https://kbdk-kgl.a	20/2/23	Digital Innovat		Relevant. Finding the intersection between what a company can do (technologywise) and what represents value to the customer.	DIgital offerings	Technological opportunities	WHat customers are willing to pay for.	

Appendix 1.4 - Third iteration

Relevant
Neutral
Irrelevant

Individual literature search

Keyword	Title	Link	Date	Related to searchterm no.	Reasoning for 2nd it. Inclusion	Focus 1.	Focus 2	Focus 3	Focus 4	Relevancy after 3rd it.	Reasoning
Predictive analytics	ASSESSING THE UNACQUAINTED: INFERRED REVIEWER PERSONALITY AND REVIEW HELPFULNESS	https://web-p-eb	23/01/23	1	Factors that influence quality content creation in online communities. Trains a deep learning algorithm to identify personality traits of reviewers, to predict who is more likely to produce helpful product reviews.	Factors of quality	Predicting helpfullness	Quality content	Deep learning		
Machine learning	Expecting the unexpected: Effects of data collection design choices on the quality of crowdsourced user-generated content1	https://www.sco	23/01/23	3	Quality content creation in online communities (at least ppl outside of the org.) Details: How to ensure that data provided by ppl outside the org. has a suitable quality. Data collection method's impact on accuracy. Maybe if I need some inspiration regarding data collection. How post processing can lead to precision gains (follow up study) - that can be very interesting.	user generated content	data quality	information quality management	supervised ML		Related to my topic, I just dont see how it can be included into my current structure.
Machine learning	Financial incentives dampen altruism in online prosocial contributions: A study of online reviews	https://www.sco	23/01/23	3	Machine learning on amazon review dataset. The aim here is to identify those who got financial incentives to make reviews.	Reviews	Machine learning	Financially incentivised reviews	spillover effecet		Important concepts about reviews, also relevant example of ML and stats on reviews. HOWEVER! we are not studying financial incentives.
Machine learning	Crowds, lending, machine, and bias	https://www.sco	23/01/23	3	Using predictive analytics in online platforms.	Predictive analytics	Algorithm biases	Crowd lending			Couldnt access
Machine learning	Cognitive Challenges in Human–Artificial Intelligence Collaboration: Investigating the Path Toward Productive Delegation	https://www.sco	23/01/23	3	Benefits of using AI in collaboration with humans. Study suggests that humans working with AI can outperform both humans and AI in CLASSIFICATION TASKS, in case the delegation is from the side of AI.	Human AI collaboration	Delegation	Classification	Performance improvement		Relevant concepts regarding AI
Artificial Intelligence	The impact of fake reviews on online visibility: A vulnerability assessment of the hotel industry	https://www.sco	23/01/23	4	(Fake) reviews in online platforms, but with a twist: the effect of fake review attacks towards small businesses. Review assessment (2.3 m) - assumption is that with the use of ML as they try to identify factors in big data.	Fake reviews	Review assessment	Visibility (placement of product in results) based on reviews			Includes some great ideas, but in general we are not focusing on fake/valid revies, but helpul content.
Reviews	What makes a helpful online review? A study of customer reviews on amazon.com	https://www.sco	23/01/23	6	Helpfullness of Amazon reviews	Factors influencing the helpfullness of reviews	Search vs experience goods	Model of customer review helpfullness			

Reviews	Examining the relationship between reviews and sales: The role of reviewer identity disclosure in electronic markets	https://www.sco	23/01/23	6	How identity disclosure in reviews can affect the perception of reviews in terms of helpfullness.	Identity disclosure of reviewr	Helpfullness perception	Review and sales relationship	Missing information supplemented/replaced by identity info		Relevant info on reviews
Reviews	Service innovation in the digital age: Key contributions and future directions	https://www.sco	23/01/23	6	Review of other studies on how to innovate services in the digital age. MIght be relevant.	Service Innovation	Service ecosystem	service platform	Value cocreation		There are a few relevant ideas
Reviews	Anxious or angry? Effects of discrete emotions on the perceived helpfulness of online reviews	https://www.sco	23/01/23	6	Effects o discrete emotions on perception of helpfullness of online reviews	Review helpfullness	Effects of emotions in reviews	Angry vvs enxious			
Reviews	The influence of recommendations and consumer reviews on evaluations of websites	https://www.sco	23/01/23	6	Effect of reviews on the evaluation of websites	Importance of reviews 2 B2C sites	Perceived usefullness				Relevant concepts on the value carried by the review feature.
Reviews	Business intelligence in Blogs: Understanding consumer interactions and communities	https://www.sco	23/01/23	6	How to effectively collect, extract and analyze user geerated content from blogs	Framework on how to analyse user content	Buiness intelligence	Blogs			Looked relevant at first, but too focused on blogs and interaction networks in blogs.
Reviews	Online product reviews: Implications for retailers and competing manufacturers	https://www.sco	23/01/23	6	Importance and effect of online product reviews from the retailers point of view.	Importance of product reviews	Retailers	Competition			Business value of reviews from the retailers perspective.
Reviews	When do consumers value positive vs. negative reviews? An empirical investigation of confirmation bias in online word of mouth	https://web-p-eb	23/01/23	6	What do customers perceive more helpful - positive or negative reviews?	Positive vs negative reviews	Confirmation bias	Consumers' initial beliefs			Rlevant
Reviews	Big data, big risks	https://www.sco	23/01/23	6	Threats imposed by big data. good for limitations part	Risks of big data	Moral and legal responsibility				Risks of big data. Revisit if needed for limitations.
Reviews	On self-selection biases in online product reviews	https://www.sco	23/01/23	6	Biases in online product reviews. Acquisition bias, underreporting bias (those with extreme opinions are more likely to wrtie reviews).	Biases in reviews	Distribution of online reviews				Relevant on review characteristics
Reviews	Competing for attention: An empirical study of online reviewers' strategic behavior	https://www.sco	23/01/23	6	Strategic behaviour of reviewrs to gain more attention	COmpetition between reviewers	Reviewer reputation	COmpetition effect	Amazon		
Reviews	Disconfirmation effect on online rating behavior: A structural model	https://www.sco	23/01/23	6	Likeliness of leaving a review and what review to leave influenced by disconfirmation. Soemone is more likely to leave a review/rating if the disconfirmation she experiences is larger, thus the difference between the aggregate rating and the own user experience - expected and experienced assesment of the product	Disconfirmation effect	Likelihood of leaving a review	Expectation vs experience			

Reviews	Sequentiality of product review information provision: An information foraging perspective	https://www.sco	23/01/23	6	Study on how to provide product reviews on a website - offering varying genres of product reviews to enhance customer decision making.	Provision of product reviews	different genres of reviews	useage vs attribute oriented reviews			
Reviews	On the spillover effects of online product reviews on purchases: Evidence from clickstream data	https://www.sco	23/01/23	6	Reviews effect on purchases, accompanied by some ML	Reviews on purchases	Spillover effect	Machine Learning model on reviews			couldnt access
Reviews	Focus within or on others: The impact of reviewers' attentional focus on review helpfulness	https://www.sco	23/01/23	6	Focus of attention: own experience / others-prospective. How, why and when the attentional focus may influence review helpfullness. HIGHLY RELEVANT!!!	Reviewers attentional focus (self/other)	Review helpfullness				couldnt access
Reviews	Anger in consumer reviews: Unhelpful but persuasive?	https://www.sco	23/01/23	6	How aggressive reviews are perceived as helpful	Agressive reviews	review helpfullness	influence on reader attitude	emotions		
Product reviews	Design of consumer review systems and product pricing	https://soeg.kb.c	29/01/23	8	How the design of customer review systems affect the review outcome. HIGHLY RELEVANT!!!	Design of review systems	When to use which design	Product pricing			
Product reviews	Mining bilateral reviews for online transaction prediction: A relational topic modeling approach	https://soeg.kb.c	29/01/23	8	Prediction of transaction based on topic modeling of bilateral reviews - an approach when the seller can also write a review.	Topic modeling on reviews	Bilateral reviews				couldnt access
Usefulness	Perceived usefulness, perceived ease of use, and user acceptance of information technology	https://soeg.kb.c	29/01/23	11	Usefullness and user acceptance of IT. Can be relevant from a system design perspective	User acceptance of IT systems	Perceiverd usefullness of IT systems	Perceived ease of use of IT systems			perceived usefullness is an importan concept of IS acceptance. Even though a connection can be drawn between PU and the usefullness of review systems, the content of the study is not directly related.
Amazon	The impact of external word-of-mouth sources on retailer sales of high-involvement products	https://soeg.kb.c	29/01/23	14	IMpact on reviews on high invilvment products specifically. Focus on external reviews	Effects of reviews	High involvment products		Focus on external reviews.		External WOM (not retailer hosted)
Amazon	Recommendation networks and the long tail of electronic commerce	https://soeg.kb.c	29/01/23	14	The effect of recommendations on the distribution of demand among categories. The more a category is influenced by recommendations, the higher the demand. Can it be th other way around?		Distribution of demand	Product categories			Not review specific, only product recommendations on the site
Value creation	Online community as space for knowledge flows	https://www.scop	20/2/23	15.	Collective knowledge floew among online commnity participants which represents value. Tacit /Explicit knowledge	Value of knowledge flows	Online comunities	Tacit/Explicit knowledge			focus on online communities which I decided to exlude. Maybe tacit-to tactit knowledge.

Value creation	Service innovation: A service-dominant logic perspective	https://www.scop	20/2/23	15.	Service innovation. Is it digital tho? not clear from the abstract.	Service innovation	Service platforms	Value cocreation			Focus on service dominant logic, doesnt seem applicable to my topic.

IM program literature

						If Included					
Keyword	**Title**	**Link**	**Date**	**Course**	**Reasoning for 2nd it. Inclusion**	**Focus 1.**	**Focus 2**	**Focus 3**	**Focus 4**	**Relevancy after 3**	**Reasoning**
Value creation	The elements of value.	https://brightspac	20/2/23	Information S	Measuring and delivering what customers really want.	What is valued by customers	How to deliver it	Elements (attributes) of value			Elements of value
Value creation	Capture more value	https://brightspac	20/2/23	Information S	Methods for next level value creation - how to capture MORE value.	Value creation	Value capture	Methods through examples.			Capturing more value here is directly referring to pricing. Since the review feature of amazon doesnt require any monetary contribution (except that someone in fact buys the product), it doesnt seem relevant.
System design	Information systems success: The quest for the independent variables	https://web-s-ebs	20/2/23	Information S	Variables that influence IS success. Social/Individual might be interesting	Variables influencing IS success	Social, Individual				Relevant
Artificial Intelligence	Reshaping Business With Artificial Intelligence: Closing the Gap Between Ambition and Action	https://www.proq	20/2/23	Digital Innov	Compare AI ambitions and efforts of companies.	AI ambition	AI effort	CUrrent picture about companies			
Digital offerings	Building shared customer insigths	https://kbdk-kgl.a	20/2/23	Digital Innov	Relevant. Finding the intersection between what a company can do (technologywise) and what represents value to the customer.	DIgital offerings	Technological opportunities	WHat customers are willing to pay for.			

Through Citation											
IS success original	DeLone, & McLean, E. R. (2003). The DeLone and McLean Model of Information Systems Success: A Ten-Year Update. Journal of Management Information Systems, 19(4), 9–30.										Individual and organizational impact as IS success dimensions

IS success update	Information Systems Success: The Quest for the Dependent Variable										Net benefits as a dimension of IS success. Also assesses the model using e-commerce sites

Appendix 1.5

Literature review template

Check the Lit review - how to for more information.

Task description received:

Review completed by: Rezso Roland Gimesi

Lit review scoping

Question	**Utilization of Machine Learning to predict helpfulness of online product reviews**
Topic/area of research	Online product reviews, Machine Learning
Purpose of literature review - why is this being done and for what purpose?	- *What is the value of online product reviews and what are the concrete components that generate value?* - *What are the challenges of online product review objectivity?* - *What are the variables that influence the perceived helpfulness of online product reviews?* - *Is it possible to predict the perceived helpfulness of Amazon book reviews using supervised machine learning?*

Scope of literature review (what is included and excluded) - this is extremely important as a lit review might otherwise get too broad (and take forever), or too narrow and miss important papers	Inclusion: - All research on reviews in ecommerce, customer perception of usefulness. - Quality content creation in online communities. - Personality types and other factors that influence quality content creation in online communities. - NLP related studies involving reviews or content generated in online communities - Increased user experience related to AI - Benefits of using AI in web platforms - User experience related to system design - Value creation related to online activities - AI potential for companies (innovation, etc) Exclusion: - Medical studies - ML/predictive analytics to improve IS research - Main topic is IS research itself - No connection to reviews or NLP or ML - Non online content specific NLP - Non online content specific ML - FInancial incentivization of reviews -External Word of Mouth - Studies on reviews which largely deviate from the customer related aspects - Studies focusing mainly on the retailer/organization's perspective - Value creation in organizational context - Business model literature - Previews - system design which is more technical focused - Other, highly deviated industries (eg.: manufacturing) - Social media and online communities - Online meetings - Project management - IoT

Key search terms - what are the terms you used for your search? Did you exclude any obvious terms, and why? Reflect on what terms other researchers may use for related topics that could be important. Find a couple of relevant articles, which keywords are they using?	Included: Predictive analytics Predictive analysis Machine learning Artificial intelligence AI Reviews Product reviews Useful Reviews Online reviews Usefullness Text Mining reviews Machine learning review assessment Amazon Excluded: Review – gave the same results as reviews Business intelligence – too broad of a topic, it would make sense if Amazon itself would like to analyse reviews etc. Big data – too broad, even though big data enables ML methods, this time I am focusing on a specific part of big data – reviews, thus in order to make the scope of the lit review more narrow it is excluded. e-commerce – the topic of e-commerce is included in reviews, there is no need to increase the scope of the research to other e-commerce related topics information management – information management related studies are included through papers from the curriculum. Otherwise we are not interested in the field of research IM in itself.

Appendix 2. Basket of eight

MIS Quarterly,

European Journal of Information Systems,

Information Systems Journal,

Information Systems Research,

Journal of Information Technology,

Journal of Management Information Systems,

Journal of Strategic Information Systems,

Journal of the Association for Information Systems

Appendix 2.1. Scopus search string

TITLE-ABS-KEY(*SEARCHTERM*) AND (LIMIT-TO (EXACTSRCTITLE,"MIS Quarterly Management Information Systems") OR LIMIT-TO (EXACTSRCTITLE,"Information Systems Research") OR LIMIT-TO (EXACTSRCTITLE,"European Journal of Information Systems") OR LIMIT-TO (EXACTSRCTITLE,"Information Systems Journal") OR LIMIT-TO (EXACTSRCTITLE,"Journal of the Association of Information Systems") OR LIMIT-TO (EXACTSRCTITLE,"Journal of Information Technology") OR LIMIT-TO (EXACTSRCTITLE,"Journal of Management Information Systems") OR LIMIT-TO (EXACTSRCTITLE,"Journal of Strategic Information Systems"))

SEARCHTERM was replaced with each of the search terms.

Appendix 3 - Coherence checks for topic modeling

Coherence checks

In [ ]:
```
import spacy
from spacy import displacy
import pandas as pd
import numpy as np
import gensim
from gensim.corpora import Dictionary
from gensim.models import LdaModel, CoherenceModel, LsiModel, HdpModel
```

In [186…
```
# Load 10k dataset
review = pd.read_csv('C:/Users/grezs/Desktop/Rencool/MAster/Aarhus/BSS stuff/4th semester - thesis/data/data_prep_10k.c

# connect the review text, where each review is a different line
text = '\n'.join(review['review/text'].tolist())
```

In [187…
```
nlp = spacy.load("en_core_web_md")
nlp.max_length = 10000000
#tokenize text - Spacy
docs = nlp(text, disable = ['ner', 'parser'])
```

In [154…
```
# count length now
len(docs)
```

Out[154]:
```
1648703
```

Include only Verbs and Nouns

In [189…
```
#Remove punctuations, stopwords, numbers (10, ten, etc), symbols, filter nouns and verbs and lemmatize - Spacy
texts, article = [], []
for word in docs:

    if word.text != '\n' and not word.is_stop and not word.is_punct\
                         and not word.like_num and word.is_alpha and word.text != 'I' and word.pos_ in['NOUN', 'VERB']:
        article.append(word.lemma_)
```

```
    if word.text == '\n':      #every text is separated by new lines (should use the same when separate rows of the df)
        texts.append(article)
        article = []
```

In [190…

```
# Compute bigrams.
from gensim.models import Phrases

# Add bigrams and trigrams to docs (only ones that appear 20 times or more).
bigram = Phrases(texts, min_count=20)
for idx in range(len(texts)):
    for token in bigram[texts[idx]]:
        if '_' in token:
            # Token is a bigram, add to document.
            texts[idx].append(token)
```

```
2023-04-12 10:50:16,176 : INFO : collecting all words and their counts
2023-04-12 10:50:16,212 : INFO : PROGRESS: at sentence #0, processed 0 words and 0 word types
2023-04-12 10:50:16,752 : INFO : collected 278068 token types (unigram + bigrams) from a corpus of 410196 words and 999
9 sentences
2023-04-12 10:50:16,753 : INFO : merged Phrases<278068 vocab, min_count=20, threshold=10.0, max_vocab_size=40000000>
2023-04-12 10:50:16,799 : INFO : Phrases lifecycle event {'msg': 'built Phrases<278068 vocab, min_count=20, threshold=1
0.0, max_vocab_size=40000000> in 0.61s', 'datetime': '2023-04-12T10:50:16.792725', 'gensim': '4.3.1', 'python': '3.10.9
| packaged by Anaconda, Inc. | (main, Mar  1 2023, 18:18:15) [MSC v.1916 64 bit (AMD64)]', 'platform': 'Windows-10-10.
0.19044-SP0', 'event': 'created'}
```

In [157…

```
len(texts)
```

Out[157]:

```
9999
```

In [192…

```
# Remove rare and common tokens.
from gensim.corpora import Dictionary

# Create a dictionary representation of the documents.
dictionary = Dictionary(texts)

# Filter out words that occur less than 20 documents, or more than 50% of the documents.
dictionary.filter_extremes(no_below=20, no_above=0.5)
```

```
2023-04-12 10:50:46,452 : INFO : adding document #0 to Dictionary<0 unique tokens: []>
2023-04-12 10:50:47,047 : INFO : built Dictionary<17988 unique tokens: ['author', 'book', 'file', 'indicate', 'militanc
y']...> from 9999 documents (total 412502 corpus positions)
2023-04-12 10:50:47,048 : INFO : Dictionary lifecycle event {'msg': "built Dictionary<17988 unique tokens: ['author',
'book', 'file', 'indicate', 'militancy']...> from 9999 documents (total 412502 corpus positions)", 'datetime': '2023-04
-12T10:50:47.048623', 'gensim': '4.3.1', 'python': '3.10.9 | packaged by Anaconda, Inc. | (main, Mar  1 2023, 18:18:15)
[MSC v.1916 64 bit (AMD64)]', 'platform': 'Windows-10-10.0.19044-SP0', 'event': 'created'}
2023-04-12 10:50:47,092 : INFO : discarding 15777 tokens: [('book', 7849), ('militancy', 1), ('paranoia', 13), ('adapti
on', 1), ('animate', 11), ('discrepancy', 6), ('ensue', 6), ('read', 5367), ('congregation', 8), ('cream', 13)]...
2023-04-12 10:50:47,093 : INFO : keeping 2211 tokens which were in no less than 20 and no more than 4999 (=50.0%) docum
ents
2023-04-12 10:50:47,111 : INFO : resulting dictionary: Dictionary<2211 unique tokens: ['author', 'file', 'indicate', 'r
elationship', 'shame']...>
```

In [159…
```
#count length with extra words
len(dictionary)
```

Out[159]:
```
3628
```

In [193…
```
#count length only nouns and verbs
len(dictionary)
```

Out[193]:
```
2211
```

Create corpus

In [194…
```
# Bag-of-words representation of the documents.
corpus = [dictionary.doc2bow(text) for text in texts]
```

In [195…
```
#Let’s see how many tokens and documents we have to train on.
print('Number of unique tokens: %d' % len(dictionary))
print('Number of documents: %d' % len(corpus))
```

```
Number of unique tokens: 2211
Number of documents: 9999
```

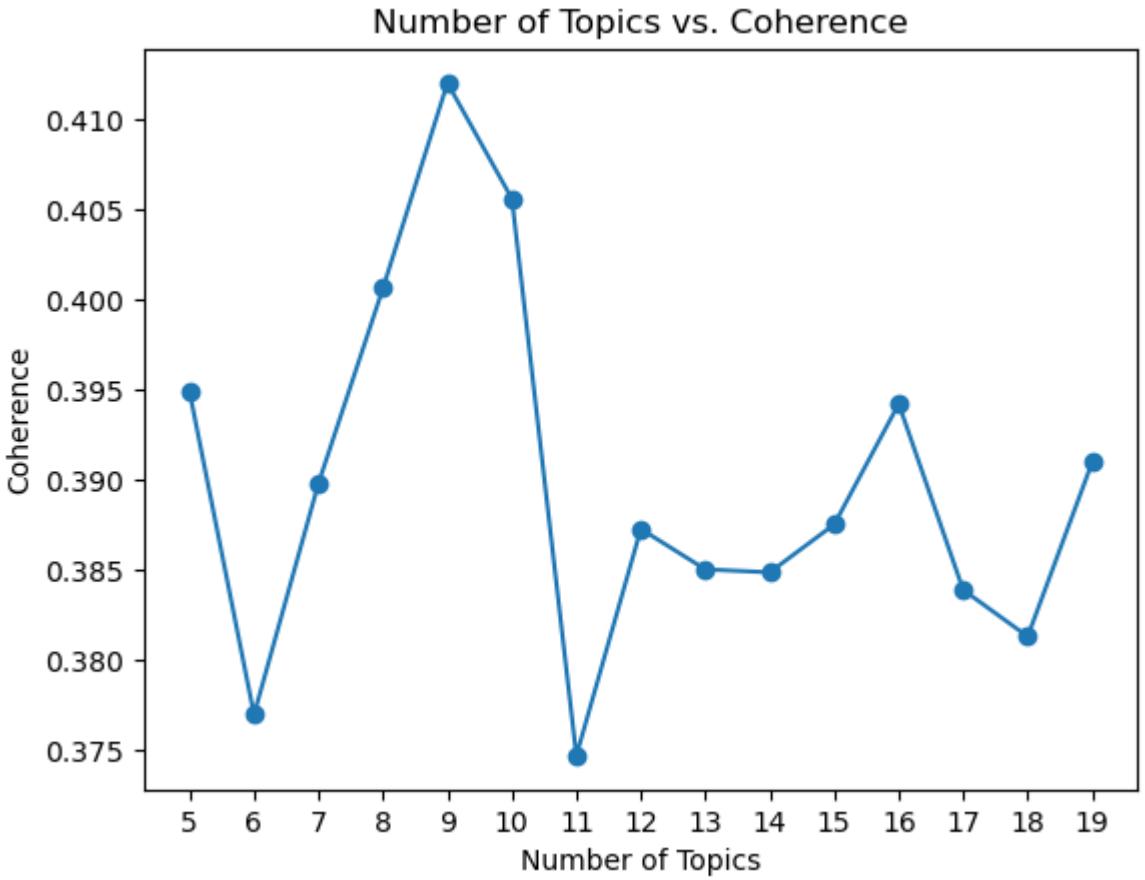
In [170…
```
import logging
logging.basicConfig(format='%(asctime)s : %(levelname)s : %(message)s', level=logging.INFO)
```

Check topic coherence

In [ ]:
```
# Plot topic coherence to get the nr of topics
coherence = []
for k in range(5,20):
    print('Round: '+str(k))
    Lda = gensim.models.ldamodel.LdaModel
    ldamodel = Lda(corpus, num_topics=k, \
                id2word = dictionary, passes=20,\
                iterations=400, chunksize = 2000, eval_every = None)

    cm = gensim.models.coherencemodel.CoherenceModel(\
         model=ldamodel, texts=texts,\
         dictionary=dictionary, coherence='c_v')

    coherence.append((k,cm.get_coherence()))
```

In [197…
```
import matplotlib.pyplot as plt
x_val = [x[0] for x in coherence]
y_val = [x[1] for x in coherence]
```

In [198…
```
plt.plot(x_val,y_val)
plt.scatter(x_val,y_val)
plt.title('Number of Topics vs. Coherence')
plt.xlabel('Number of Topics')
plt.ylabel('Coherence')
plt.xticks(x_val)
plt.show()

# Topic coherence higher at topic nr 9 than when we included other pos tags, but less gradual, higher fluctuations
# Since we want to maximize this score, lets continue with this
```



Train model

In [ ]:
```
from gensim.models import LdaModel

# Set training parameters.
num_topics = 9 #have to experiment, can be 10-50-100-etc.
chunksize = 2000 #how many docs are processed at once
passes = 20 # how often we train the model
iterations = 400 #how often we repeat a loop over every document
eval_every = None  # Don't evaluate model perplexity, takes too much time.

#####Passes/iterations fine tuning:
###### First set eval_every = 1
###### check log to make sure that by the final passes most of the documents converged, have to set
###### both passes and iterations high enough for that to happen.
```

```
###### Then set it back to None


# Make an index to word dictionary.
temp = dictionary[0]  # This is only to "load" the dictionary.
id2word = dictionary.id2token

model = LdaModel(
    corpus=corpus,
    id2word=id2word,
    chunksize=chunksize,
    alpha='auto',
    eta='auto',
    iterations=iterations,
    num_topics=num_topics,
    passes=passes,
    eval_every=eval_every
)
```

In [ ]:
```
# Check topic coherence
top_topics = model.top_topics(corpus)

# Average topic coherence is the sum of topic coherences of all topics, divided by the number of topics.
avg_topic_coherence = sum([t[1] for t in top_topics]) / num_topics
print('Average topic coherence: %.4f.' % avg_topic_coherence)

from pprint import pprint
pprint(top_topics)
```

Extract topics

In [204…
```
all_topics = {}
num_terms = 10 # Adjust number of words to represent each topic
lambd = 0.4 # Adjust this accordingly based on tuning above
for i in range(1,10): #Adjust this to reflect number of topics chosen for final LDA model
    topic = topic_data.topic_info[topic_data.topic_info.Category == 'Topic'+str(i)].copy()
    topic['relevance'] = topic['loglift']*(1-lambd)+topic['logprob']*lambd
    all_topics['Topic '+str(i)] = topic.sort_values(by='relevance', ascending=False).Term[:num_terms].values
```

In [205…
```
pd.DataFrame(all_topics).T
```

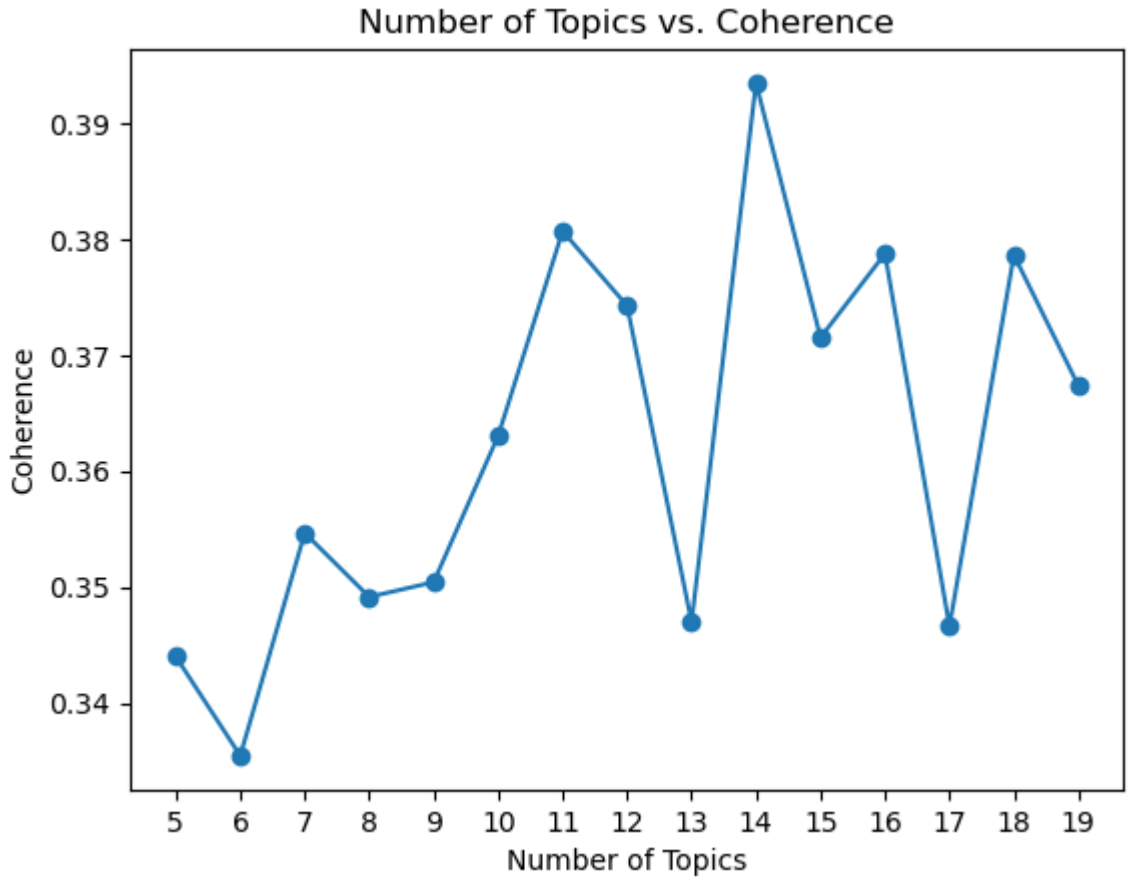
Out[205]:

	0	1	2	3	4	5	6	7	8	9
Topic 1	novel	reader	story	character	event	world	tale	fiction	narrative	create
Topic 2	like	think	story	love	series	character	get	enjoy	want	time
Topic 3	people	idea	step	theory	church	religion	philosophy	business	belief	value
Topic 4	information	text	edition	cover	reference	material	chapter	section	include	print
Topic 5	woman	life	family	father	mother	man	husband	live	relationship	marry
Topic 6	war	government	case	soldier	police	health	victim	disease	mission	murder
Topic 7	child	kid	recipe	dog	boy	adult	house	trip	cookbook	animal
Topic 8	school	student	teacher	lesson	class	college	teach	language	textbook	cat
Topic 9	movie	film	art	poetry	music	poem	artist	version	poet	painting

All part-of-speech tags included

In [185…

```
# We can see that topic coherence increased until 11 topic, peaked once more at 14, but then started declining.
```



Nouns, Verbs, Adjectives and Adverbs

In [1]:
```
import os
os.environ['BLAS_NUM_THREADS'] = '2'
import spacy
from spacy import displacy
import pandas as pd
import numpy

import gensim
from gensim.corpora import Dictionary
from gensim.models import LdaModel, CoherenceModel, LsiModel, HdpModel
from gensim.models.ldamulticore import LdaMulticore
```

Load and Set up dataset

In [2]:
```
# Load 10k dataset
review = pd.read_csv('C:/Users/grezs/Desktop/Rencool/MAster/Aarhus/BSS stuff/4th semester - thesis/data/data_prep_10k.c

#Check if column contains null
review['review/text'].isnull().sum()
# 8 precisely, must be deleted

#delete rows with null
review.dropna(subset=['review/text'], inplace=True)

# connect the review text
texts = review['review/text'].tolist()
```

Tokenize with Spacy

In [5]:
```
from tqdm import tqdm
from spacy.tokens import DocBin

nlp = spacy.load("en_core_web_md")

doc_bin = DocBin(attrs=["LEMMA", "POS"])
doc_bin_all = DocBin(attrs=["LEMMA", "POS"])

batch_size = 500
counter = 0
for doc in tqdm(nlp.pipe(texts, n_process=2, disable=["parser", "ner"]), total=len(texts)):
    doc_bin.add(doc)
    counter += 1

    if counter == batch_size:
        doc_bin_all.merge(doc_bin)
        doc_bin = DocBin(attrs=["LEMMA", "POS"])
        counter = 0
#doc_bins.append(doc_bin) ##ONLY USE IF TOTAL NUMBER IS REDUCED BECAUSE OF NULLS!!!!!!!!!!!!
```

```
100%|██████████| 10000/10000 [02:13<00:00, 74.74it/s]
```

In [24]:
```
# Serialize
bytes_data = doc_bin_all.to_bytes()
```

In [26]:
```
#Desirialize
nlp = spacy.blank("en")
doc_bin = DocBin().from_bytes(bytes_data)
docs = list(doc_bin.get_docs(nlp.vocab))

#Check the length just in case
len(docs)
```

In [29]:
```
#Save doc_bin_all for later use
doc_bin_all.to_disk('C:/Users/grezs/Desktop/Rencool/MAster/Aarhus/BSS stuff/4th semester - thesis/data/doc_bin_all_10k'
```

In [37]:
```
#Load docbin and convert it to an iterable docs collection
doc_bin_all = DocBin().from_disk("C:/Users/grezs/Desktop/Rencool/MAster/Aarhus/BSS stuff/4th semester - thesis/data/doc
docs = list(doc_bin_all.get_docs(nlp.vocab))
```

Convert to a nested list of lemmatized filtered words, that gensim can use later

In [38]:
```
#### THIS LOOKS LIKE THE ONE!!!
#Remove punctuations, stopwords, numbers (10, ten, etc), symbols, filter nouns and verbs and lemmatize - Spacy
texts, article = [], []
for doc in docs:
    for word in doc:
        if word.text is not word.is_stop and not word.is_punct\
                    and not word.like_num and word.is_alpha and word.text != 'I' and word.pos_ in['NOUN', 'VERB', 'ADJ'
            article.append(word.lemma_)

    texts.append(article)
    article = []
```

In [42]:
```
# Compute bigrams.
from gensim.models import Phrases

# Add bigrams and trigrams to docs (only ones that appear 20 times or more).
bigram = Phrases(texts, min_count=20)
for idx in range(len(texts)):
    for token in bigram[texts[idx]]:
        if '_' in token:
            # Token is a bigram, add to document.
            texts[idx].append(token)
```

Set up dictionary and corpus

In [43]:
```
# Remove rare and common tokens.
# Create a dictionary representation of the documents.
dictionary = Dictionary(texts)

# Filter out words that occur less than 20 documents, or more than 50% of the documents.
dictionary.filter_extremes(no_below=20, no_above=0.5)
```

In [44]:
```
# Bag-of-words representation of the documents.
corpus = [dictionary.doc2bow(text) for text in texts]
```

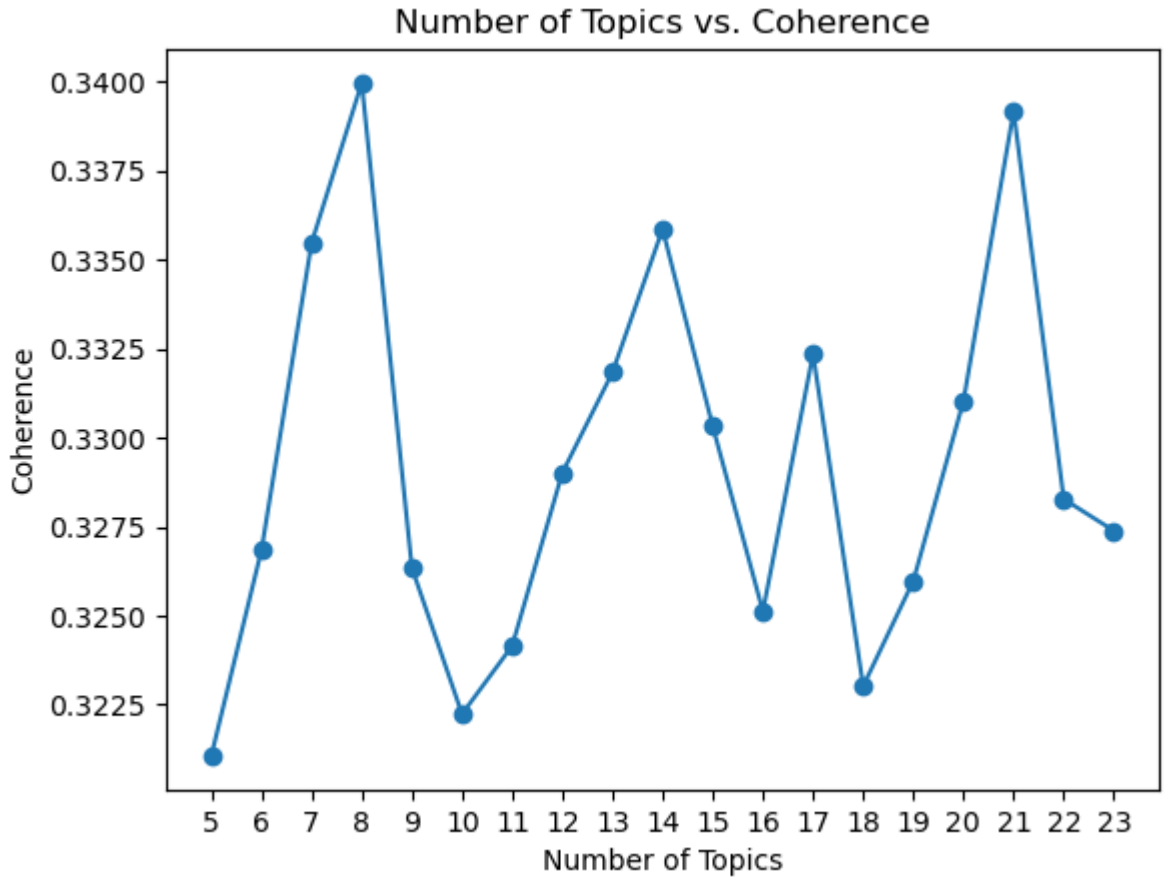
In [45]:
```
import logging
logging.basicConfig(format='%(asctime)s : %(levelname)s : %(message)s', level=logging.INFO)
```

Coherence check

In [ ]:
```
# Plot topic coherence to get the nr of topics
coherence = []
for k in range(5,24):
    print('Round: '+str(k))

    ldamodel = LdaMulticore(corpus, num_topics=k, \
                id2word = dictionary, passes=20,\
                iterations=400, chunksize = 2000, eval_every = None)

    cm = gensim.models.coherencemodel.CoherenceModel(\
         model=ldamodel, texts=texts,\
         dictionary=dictionary, coherence='c_v')

    coherence.append((k,cm.get_coherence()))
```

In [47]:
```
import matplotlib.pyplot as plt
x_val = [x[0] for x in coherence]
y_val = [x[1] for x in coherence]

plt.plot(x_val,y_val)
plt.scatter(x_val,y_val)
plt.title('Number of Topics vs. Coherence')
plt.xlabel('Number of Topics')
plt.ylabel('Coherence')
plt.xticks(x_val)
plt.show()
```

```
# Topic coherence seems the most fluctuating in this sample, which contains the adj and advs as well
# but in fact it is the most stable, range of fluctuation is the smallest here-
# Overall topic coh. is lowest here, but the topics are more descriptive, not just thematic as compared to the previou
# SO far 3 samples, decreasing by coherence:
#       1. Only nouns and verbs peak at 0.41 - Can it be because it rather revealed the "genres"
#       2. Everything, peak at 0.39 - most gradual
#       3. Nouns, verbs, adj, adv 0.34 - seems most fluctuating, but least difference between highest and lowest. Can a
```



In [ ]:
```
# ONLY NOUNS AND VERBS
import pyLDAvis
import pyLDAvis.gensim_models

pyLDAvis.enable_notebook()
topic_data_parLDA = pyLDAvis.gensim_models.prepare(model, corpus, dictionary)
pyLDAvis.display(topic_data_parLDA)
```

```
#Topic 1: Humanities
#Topic 2: Novels
#Topic 3: Kids-Teenagers
#Topic 4: Book attributes (both content and "appearance")
#Topic 5: Instructions
#Topic 6: Family
#Topic 7: Crime
#Topic 8: Horror-Crime
#Topic 9: Adventure-Romance
```

In [ ]:
```
# Nouns, verbs, adjectives and adverbs
import pyLDAvis
import pyLDAvis.gensim_models

pyLDAvis.enable_notebook()
topic_data_parLDA = pyLDAvis.gensim_models.prepare(model, corpus, dictionary)
pyLDAvis.display(topic_data_parLDA)


#Topic 1: Histroic novel
#Topic 2: Romance? not sure
#Topic 3: Studies/Book attributes
#Topic 4: Story aspects/content of novels
#Topic 5: Arts/Artistic
#Topic 6: School
#Topic 7: Instructions
#Topic 8: Fantasy/child stories
#Topic 9: Versions (movie/film/audio)
#The topics are more distant than when using only nouns and verbs
#Only noun and verb topics are more categorical, while the topics including adjectives and adverbs are more descriptive
#Since we are focusing on experience goods, might be better to include adj and adv, since they help to be more descript
```

100k dataset

In [48]:
```
# Load whole dataset
review = pd.read_csv('C:/Users/grezs/Desktop/Rencool/MAster/Aarhus/BSS stuff/4th semester - thesis/data/Books_rating.cs
# sample 100k
review = review.sample(n=100000)

#Check if column contains null
review['review/text'].isnull().sum()
# 8 precisely, must be deleted
```

```
#delete rows with null
review.dropna(subset=['review/text'], inplace=True)

# connect the review text
texts = review['review/text'].tolist()
```

In [49]:
```
len(texts)
```

Out[49]:
```
100000
```

In [50]:
```
from tqdm import tqdm
from spacy.tokens import DocBin

nlp = spacy.load("en_core_web_md")

doc_bin = DocBin(attrs=["LEMMA", "POS"])
doc_bin_all = DocBin(attrs=["LEMMA", "POS"])

batch_size = 500
counter = 0
for doc in tqdm(nlp.pipe(texts, n_process=2, disable=["parser", "ner"]), total=len(texts)):
    doc_bin.add(doc)
    counter += 1

    if counter == batch_size:
        doc_bin_all.merge(doc_bin)
        doc_bin = DocBin(attrs=["LEMMA", "POS"])
        counter = 0
#doc_bins.append(doc_bin) ##ONLY USE IF TOTAL NUMBER IS REDUCED BECAUSE OF NULLS!!!!!!!!!!!
```

```
100%|██████████| 100000/100000 [21:59<00:00, 75.77it/s]
```

In [55]:
```
#Save doc_bin_all for later use
doc_bin_all.to_disk('C:/Users/grezs/Desktop/Rencool/MAster/Aarhus/BSS stuff/4th semester - thesis/data/doc_bin_all_100k
docs = list(doc_bin_all.get_docs(nlp.vocab))
```

Lemma, stopw. etc.

In [56]:
```
#### THIS LOOKS LIKE THE ONE!!!
#Remove punctuations, stopwords, numbers (10, ten, etc), symbols, filter nouns and verbs and lemmatize - Spacy
texts, article = [], []
for doc in docs:
```

```
    for word in doc:
        if word.text is not word.is_stop and not word.is_punct\
                    and not word.like_num and word.is_alpha and word.text != 'I' and word.pos_ in['NOUN', 'VERB', 'ADJ'
            article.append(word.lemma_)

    texts.append(article)
    article = []
```

In [58]:
```
# Compute bigrams.
from gensim.models import Phrases

# Add bigrams and trigrams to docs (only ones that appear 20 times or more).
bigram = Phrases(texts, min_count=20)
for idx in range(len(texts)):
    for token in bigram[texts[idx]]:
        if '_' in token:
            # Token is a bigram, add to document.
            texts[idx].append(token)
```

```
2023-04-17 21:14:38,414 : INFO : collecting all words and their counts
2023-04-17 21:14:38,417 : INFO : PROGRESS: at sentence #0, processed 0 words and 0 word types
2023-04-17 21:14:39,265 : INFO : PROGRESS: at sentence #10000, processed 630716 words and 419974 word types
2023-04-17 21:14:40,292 : INFO : PROGRESS: at sentence #20000, processed 1271806 words and 736788 word types
2023-04-17 21:14:41,233 : INFO : PROGRESS: at sentence #30000, processed 1911876 words and 1016041 word types
2023-04-17 21:14:42,156 : INFO : PROGRESS: at sentence #40000, processed 2548274 words and 1268904 word types
2023-04-17 21:14:43,165 : INFO : PROGRESS: at sentence #50000, processed 3188128 words and 1505400 word types
2023-04-17 21:14:44,203 : INFO : PROGRESS: at sentence #60000, processed 3836326 words and 1730937 word types
2023-04-17 21:14:45,355 : INFO : PROGRESS: at sentence #70000, processed 4480085 words and 1941974 word types
2023-04-17 21:14:46,662 : INFO : PROGRESS: at sentence #80000, processed 5122654 words and 2143566 word types
2023-04-17 21:14:47,664 : INFO : PROGRESS: at sentence #90000, processed 5767719 words and 2336364 word types
2023-04-17 21:14:48,639 : INFO : collected 2519726 token types (unigram + bigrams) from a corpus of 6408507 words and 1
00000 sentences
2023-04-17 21:14:48,640 : INFO : merged Phrases<2519726 vocab, min_count=20, threshold=10.0, max_vocab_size=40000000>
2023-04-17 21:14:48,643 : INFO : Phrases lifecycle event {'msg': 'built Phrases<2519726 vocab, min_count=20, threshold=
10.0, max_vocab_size=40000000> in 10.23s', 'datetime': '2023-04-17T21:14:48.643785', 'gensim': '4.3.1', 'python': '3.1
0.9 | packaged by Anaconda, Inc. | (main, Mar  1 2023, 18:18:15) [MSC v.1916 64 bit (AMD64)]', 'platform': 'Windows-10-
10.0.19044-SP0', 'event': 'created'}
```

Set up dictionary and corpus

In [59]:
```
# Remove rare and common tokens.
# Create a dictionary representation of the documents.
dictionary = Dictionary(texts)
```

```
# Filter out words that occur less than 5% documents, or more than 50% of the documents.
dictionary.filter_extremes(no_below=0.05, no_above=0.5)
```

```
2023-04-17 21:16:03,194 : INFO : adding document #0 to Dictionary<0 unique tokens: []>
2023-04-17 21:16:04,227 : INFO : adding document #10000 to Dictionary<26941 unique tokens: ['ability', 'able', 'accoun
t', 'affect', 'alive']...>
2023-04-17 21:16:05,465 : INFO : adding document #20000 to Dictionary<36405 unique tokens: ['ability', 'able', 'accoun
t', 'affect', 'alive']...>
2023-04-17 21:16:06,996 : INFO : adding document #30000 to Dictionary<43675 unique tokens: ['ability', 'able', 'accoun
t', 'affect', 'alive']...>
2023-04-17 21:16:08,059 : INFO : adding document #40000 to Dictionary<49729 unique tokens: ['ability', 'able', 'accoun
t', 'affect', 'alive']...>
2023-04-17 21:16:09,043 : INFO : adding document #50000 to Dictionary<55232 unique tokens: ['ability', 'able', 'accoun
t', 'affect', 'alive']...>
2023-04-17 21:16:10,097 : INFO : adding document #60000 to Dictionary<60234 unique tokens: ['ability', 'able', 'accoun
t', 'affect', 'alive']...>
2023-04-17 21:16:10,962 : INFO : adding document #70000 to Dictionary<64894 unique tokens: ['ability', 'able', 'accoun
t', 'affect', 'alive']...>
2023-04-17 21:16:11,808 : INFO : adding document #80000 to Dictionary<69106 unique tokens: ['ability', 'able', 'accoun
t', 'affect', 'alive']...>
2023-04-17 21:16:12,650 : INFO : adding document #90000 to Dictionary<72993 unique tokens: ['ability', 'able', 'accoun
t', 'affect', 'alive']...>
2023-04-17 21:16:13,555 : INFO : built Dictionary<76714 unique tokens: ['ability', 'able', 'account', 'affect', 'aliv
e']...> from 100000 documents (total 6641361 corpus positions)
2023-04-17 21:16:13,556 : INFO : Dictionary lifecycle event {'msg': "built Dictionary<76714 unique tokens: ['ability',
'able', 'account', 'affect', 'alive']...> from 100000 documents (total 6641361 corpus positions)", 'datetime': '2023-04
-17T21:16:13.556654', 'gensim': '4.3.1', 'python': '3.10.9 | packaged by Anaconda, Inc. | (main, Mar  1 2023, 18:18:15)
[MSC v.1916 64 bit (AMD64)]', 'platform': 'Windows-10-10.0.19044-SP0', 'event': 'created'}
2023-04-17 21:16:13,622 : INFO : discarding 2 tokens: [('book', 79082), ('read', 53643)]...
2023-04-17 21:16:13,623 : INFO : keeping 76712 tokens which were in no less than 0 and no more than 50000 (=50.0%) docu
ments
2023-04-17 21:16:13,731 : INFO : resulting dictionary: Dictionary<76712 unique tokens: ['ability', 'able', 'account',
'affect', 'alive']...>
```

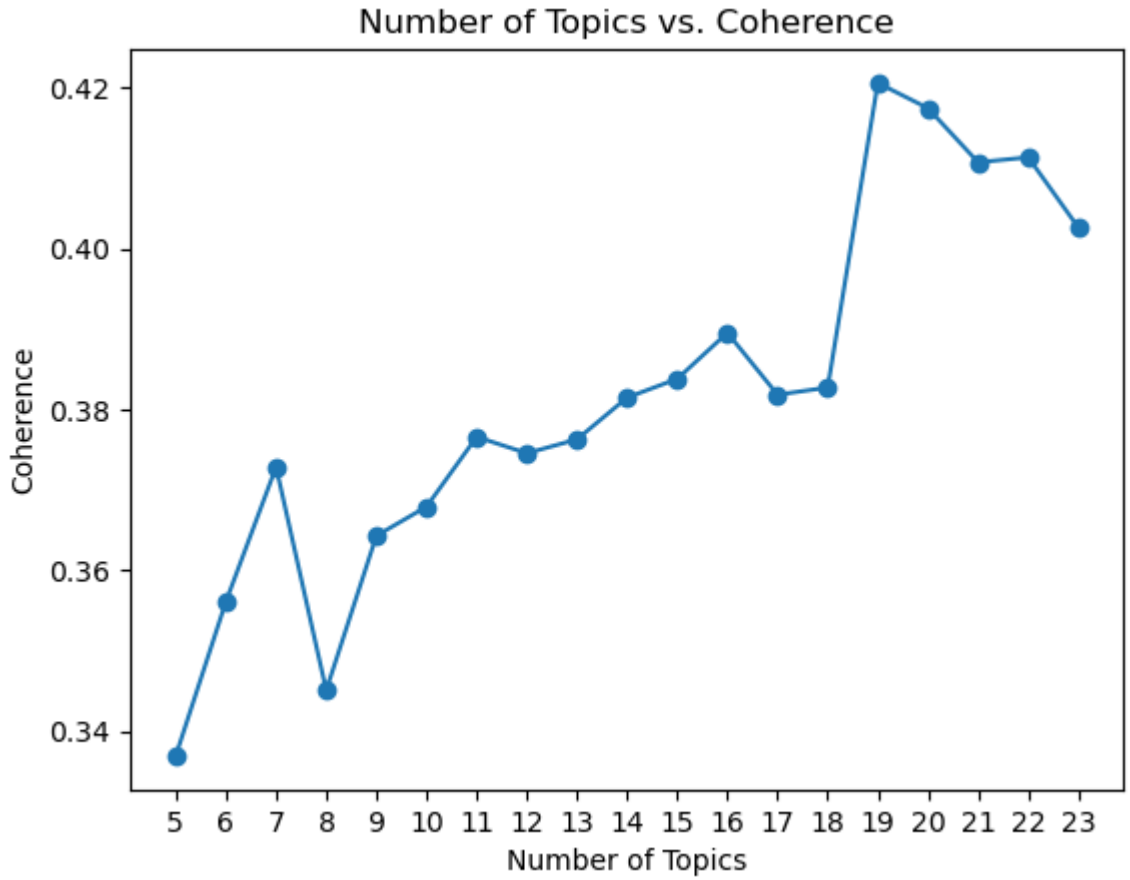
In [60]:
```
# Bag-of-words representation of the documents.
corpus = [dictionary.doc2bow(text) for text in texts]
```

In [62]:
```
import logging
logging.basicConfig(format='%(asctime)s : %(levelname)s : %(message)s', level=logging.INFO)
```

Coherence check

In [ ]:
```
# Plot topic coherence to get the nr of topics
coherence = []
for k in range(5,24):
    print('Round: '+str(k))

    ldamodel = LdaMulticore(corpus, num_topics=k, \
                id2word = dictionary, passes=20,\
                iterations=400, chunksize = 2000, eval_every = None)

    cm = gensim.models.coherencemodel.CoherenceModel(\
         model=ldamodel, texts=texts,\
         dictionary=dictionary, coherence='c_v')

    coherence.append((k,cm.get_coherence()))
```

In [64]:
```
import matplotlib.pyplot as plt
x_val = [x[0] for x in coherence]
y_val = [x[1] for x in coherence]

plt.plot(x_val,y_val)
plt.scatter(x_val,y_val)
plt.title('Number of Topics vs. Coherence')
plt.xlabel('Number of Topics')
plt.ylabel('Coherence')
plt.xticks(x_val)
plt.show()

#Beautiful, gradual
#Peaks at 19 topics, lets do this
```


Number of Topics vs. Coherence
Coherence
0.42
0.40
0.38
0.36
0.34
5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
Number of Topics

In [ ]:

Appendix 4. -Topic modeling

Load dataset

In [1]:
```
# Allow BLAS to use 2 threads
import os
os.environ['BLAS_NUM_THREADS'] = '2'
import spacy
#from spacy import displacy
import pandas as pd
import numpy as np

import gensim
from gensim.corpora import Dictionary
from gensim.models import LdaModel, CoherenceModel, LsiModel, HdpModel
from gensim.models.ldamulticore import LdaMulticore
from tqdm import tqdm
from spacy.tokens import DocBin
```

In [3]:
```
# Load whole dataset
review = pd.read_csv('C:/Users/grezs/Desktop/Rencool/MAster/Aarhus/BSS stuff/4th semester - thesis/data/Books_rating')
```

In [ ]:
```
review = review.sample(n=1000000)
```

In [4]:
```
#Check if column contains null
review['review/text'].isnull().sum()
# 2 precisely, must be deleted

#delete rows with null
review.dropna(subset=['review/text'], inplace=True)
```

In [ ]:
```
review.to_csv("C:/Users/rezso.gimesi/Desktop/Rezso/Personal stuff/thesis/archive/review_samp_1m.csv")
```

In [ ]:
```
review = pd.read_csv('C:/Users/grezs/Desktop/Rencool/MAster/Aarhus/BSS stuff/4th semester - thesis/data/review_samp_1m.
```

In [5]:
```
# connect the review text
texts = review['review/text'].tolist()
```

In [ ]:
```
nlp = spacy.load("en_core_web_md")

doc_bin = DocBin(attrs=["LEMMA", "POS"])
doc_bin_all = DocBin(attrs=["LEMMA", "POS"])

batch_size = 50000
counter = 0
for doc in tqdm(nlp.pipe(texts, n_process=2, disable=["parser", "ner"]), total=len(texts)):
    doc_bin.add(doc)
    counter += 1

    if counter == batch_size:
        doc_bin_all.merge(doc_bin)
        doc_bin = DocBin(attrs=["LEMMA", "POS"])
        counter = 0
doc_bin_all.merge(doc_bin)
```

In [ ]:
```
# Serialize
bytes_data = doc_bin.to_bytes()

#Desirialize
doc_bin = DocBin().from_bytes(bytes_data)
docs = list(doc_bin.get_docs(nlp.vocab))
```

In [ ]:
```
doc_bin_all.to_disk("C:/Users/rezso.gimesi/Desktop/Rezso/Personal stuff/thesis/docbin_1m")
```

In [2]:
```
doc_bin_all = DocBin().from_disk("C:/Users/grezs/Master thesis/docbin_1m")
```

In [3]:
```
nlp = spacy.load("en_core_web_md")
docs = list(doc_bin_all.get_docs(nlp.vocab))
```

In [4]:
```
del(doc_bin_all)
del(nlp)
```

In [5]:
```
#Remove punctuations, stopwords, numbers (10, ten, etc), symbols, filter nouns and verbs and lemmatize - Spacy
texts, article = [], []
for doc in tqdm(docs, total=len(docs)):
    for token in doc:
        if not token.is_stop and not token.is_punct\
```

```
                    and not token.like_num and token.is_alpha and token.text != 'I' and token.pos_ in['NOUN', 'VERB', '
            article.append(token.lemma_)

    texts.append(article)
    article = []
```

```
100%|██████████| 999998/999998 [05:56<00:00, 2803.10it/s]
```

In [6]:
```
import logging
logging.basicConfig(format='%(asctime)s : %(levelname)s : %(message)s', level=logging.INFO)
```

In [ ]:
```
# Compute bigrams.
from gensim.models import Phrases

# Add bigrams to docs (only ones that appear 1000 times or more).
bigram = Phrases(texts, min_count=1000)
for idx in tqdm(range(len(texts))):
    for token in bigram[texts[idx]]:
        if '_' in token:
            # Token is a bigram, add to document.
            texts[idx].append(token)
```

In [ ]:
```
# Create a dictionary representation of the documents.
dictionary = Dictionary(texts)
```

In [9]:
```
# Filter out words that occur less than 10% of the documents, or more than 50% of the documents.
dictionary.filter_extremes(no_below=0.1, no_above=0.5)
```

```
2023-05-06 22:14:05,334 : INFO : discarding 144749 tokens: [('book', 789032), ('read', 539568), ('pashent', 1), ('scoio
sis', 1), ('slayground', 1), ('reperative', 1), ('simperer', 1), ('aplusphysic', 1), ('stareview', 1), ('counterroris
m', 1)]...
2023-05-06 22:14:05,334 : INFO : keeping 100000 tokens which were in no less than 0 and no more than 499999 (=50.0%) do
cuments
2023-05-06 22:14:05,561 : INFO : resulting dictionary: Dictionary<100000 unique tokens: ['corporate', 'day', 'definit
e', 'easy', 'email']...>
```

In [10]:
```
# Bag-of-words representation of the documents.
corpus = [dictionary.doc2bow(text) for text in tqdm(texts, total=len(texts))]
```

```
100%|██████████| 999998/999998 [00:51<00:00, 19454.13it/s]
```

In [11]:
```
del(texts)
```

In [ ]:
```
# Set training parameters.
num_topics = 19
chunksize = 2000 #how many docs are processed at once
passes = 20 # how often we train the model
iterations = 400 #how often we repeat a loop over every document
eval_every = None

# Make an index to word dictionary.
temp = dictionary[0]
id2word = dictionary.id2token

model = LdaMulticore(
    corpus=corpus,
    id2word=id2word,
    chunksize=chunksize,
    eta='auto',
    iterations=iterations,
    num_topics=num_topics,
    passes=passes,
    eval_every=eval_every
)
```

In [ ]:
```
#Save the model to disk
from gensim.test.utils import datapath
from gensim.corpora import MmCorpus, Dictionary

#Save the model
temp_file_model = datapath("C:/Users/grezs/Master thesis/lda_model_1m_v2")
model.save(temp_file_model)

# save the corpus
MmCorpus.serialize('corpus_1m_v2.mm', corpus)

# save the dictionary
dictionary.save('dictionary_1m_v2.gensim')
```

In [1]:
```
# import gensim
from gensim import corpora
from gensim.corpora import Dictionary
from gensim.test.utils import datapath
from gensim.models import LdaModel, CoherenceModel, LsiModel, HdpModel
from gensim.models.ldamulticore import LdaMulticore
# load model from disk
# Load the model from disk
```

```
temp_file_model = datapath("C:/Users/grezs/Master thesis/lda_model_1m_v2")
model = LdaModel.load(temp_file_model)

#Load the dictionary from disk
dictionary = corpora.Dictionary.load('dictionary_1m_v2.gensim')

# Load the corpus from disk
corpus = corpora.MmCorpus('corpus_1m_v2.mm')
```

Visualization

In [10]:
```
#200k sample of corpus for visualization
import random
# set the sampling size
sample_size = 200000

# randomly sample the texts
sampled_texts = random.sample(texts, sample_size)

#Create sampled_corpus
sampled_corpus = [dictionary.doc2bow(text) for text in tqdm(sampled_texts, total=len(sampled_texts))]
```

```
100%|██████████| 200000/200000 [00:09<00:00, 21313.86it/s]
```

In [ ]:
```
import pyLDAvis
import pyLDAvis.gensim_models

pyLDAvis.enable_notebook()
topic_data_parLDA = pyLDAvis.gensim_models.prepare(model, sampled_corpus, dictionary)
pyLDAvis.display(topic_data_parLDA)

#Topic 1: Politics
#Topic 2: Cognitive effort
#Topic 3: Novel characteristics
#Topic 4: Opinion on plot
#Topic 5: Book attributes (internal content)
#Topic 6: Experience
#Topic 7: Family
#Topic 8: Recommendation
#Topic 9: Improvement/instructions
#Topic 10: Personal development (not a genre)
#Topic 11: Fantasy
#Topic 12: Young age
```

```
#Topic 13: History
#Topic 14: Opinion on adaptation
#Topic 15: Book attributes (external)
#Topic 16: Philosophy/Religion
#Topic 17: War
#Topic 18: American history/Western
#Topic 19: Education
```

In [13]:
```
#Save topics from 200k sample to csv
all_topics = {}
num_terms = 15
lambd = 0.6 # Adjust this accordingly based on tuning above
for i in range(1,20): # Because we have 19 topics
    topic = topic_data_parLDA.topic_info[topic_data_parLDA.topic_info.Category == 'Topic'+str(i)].copy()
    topic['relevance'] = topic['loglift']*(1-lambd)+topic['logprob']*lambd
    all_topics['Topic '+str(i)] = topic.sort_values(by='relevance', ascending=False).Term[:num_terms].values

topics_df = pd.DataFrame(all_topics).T
topics_df.to_csv('C:/Users/grezs/Desktop/Rencool/MAster/Aarhus/BSS stuff/4th semester - thesis/topics.csv')
```

In [6]:
```
# save topics from 1m sample to csv
topics = model.show_topics(num_topics=-1, num_words=15, formatted=False)

# Create a dictionary for each topic and its keywords
topic_keywords = {}
for topic_num, topic_words in topics:
    keywords = [word for word, _ in topic_words]
    topic_keywords[topic_num] = keywords


df_topics = pd.DataFrame.from_dict(topic_keywords, orient='index')
df_topics.columns = [f"Keyword {i+1}" for i in range(df_topics.shape[1])]


df_topics.insert(0, 'Topic', df_topics.index)

df_topics = df_topics.reset_index(drop=True)
df_topics
```

Out[6]:

	Topic	Keyword 1	Keyword 2	Keyword 3	Keyword 4	Keyword 5	Keyword 6	Keyword 7	Keyword 8	Keyword 9	Keyword 10	Keyword 11
0	0	year	child	old	young	ago	kid	year_old	story	boy	girl	year_ago
1	1	life	story	time	love	feel	experience	world	heart	way	live	find
2	2	edition	recipe	look	illustration	cover	great	page	buy	picture	version	copy
3	3	think	know	thing	go	get	time	good	want	way	people	bad
4	4	movie	review	buy	version	write	star	see	film	money	time	waste
5	5	world	society	human	people	political	work	power	government	state	right	social
6	6	novel	story	character	work	write	reader	writing	style	time	writer	classic
7	7	world	story	adventure	tale	man	find	evil	fantasy	come	take	ship
8	8	information	good	learn	need	help	use	great	work	find	easy	business
9	9	question	religion	answer	christian	faith	church	religious	philosophy	believe	truth	spiritual
10	10	woman	man	love	family	life	young	father	wife	husband	mother	girl
11	11	war	kill	fight	battle	man	murder	military	crime	game	death	soldier
12	12	highly	recommend	school	highly_recommend	class	high	student	great	high_school	teacher	reading
13	13	history	write	historical	time	author	life	great	year	interesting	detail	event
14	14	love	good	story	great	think	like	time	enjoy	write	want	find
15	15	black	white	travel	town	place	american	country	small	city	people	live
16	16	people	life	think	way	help	thing	good	understand	change	person	feel
17	17	chapter	text	author	find	reader	work	subject	word	example	study	good
18	18	character	story	novel	plot	series	good	reader	end	mystery	main	find

In [8]:
```
df_topics.to_csv('C:/Users/grezs/Desktop/Rencool/MAster/Aarhus/BSS stuff/4th semester - thesis/all_topics_1m.csv')
```

Calculate baseline coherence score

In [9]:
```
# model v2
from gensim.models import CoherenceModel

# Compute Coherence Score
cm = gensim.models.coherencemodel.CoherenceModel(\
         model=model, texts=texts,\
         dictionary=dictionary, coherence='c_v')
coherence_lda = cm.get_coherence()
print('Coherence Score: ', coherence_lda)
#Can be further fine tuned, but its not in the scope of this paper
#Coherence Score:  0.4451422703738074
```

```
2023-05-07 18:56:48,392 : INFO : using ParallelWordOccurrenceAccumulator<processes=3, batch_size=64> to estimate probab
ilities from sliding windows
2023-05-07 19:18:03,133 : INFO : 3 accumulators retrieved from output queue
2023-05-07 19:18:03,228 : INFO : accumulated word occurrence stats for 10301860 virtual documents
Coherence Score:  0.4451422703738074
```

Get topic distribution

In [14]:
```
from tqdm import tqdm
train_vecs = []
for i in tqdm(range(999998)):
    top_topics = (
        model.get_document_topics(corpus[i],
                                  minimum_probability=0.0)
    )
    topic_vec = [top_topics[i][1] for i in range(19)]
    train_vecs.append(topic_vec)
```

```
100%|██████████| 999998/999998 [24:16<00:00, 686.65it/s]
```

In [18]:
```
train_vecs[0]
```

Out[18]:
```
[0.0019498863,
 0.16724145,
 0.0019498863,
 0.0019498866,
 0.0019498866,
 0.0019498863,
 0.0019498863,
 0.0019498863,
 0.5997592,
 0.0019498863,
 0.0019498863,
 0.09229446,
 0.0019498863,
 0.0019498866,
 0.11145656,
 0.0019498863,
 0.0019498866,
 0.0019498863,
 0.0019498863]
```

In [16]:
```
import pandas as pd
column_names = ['Topic 1', 'Topic 2', 'Topic 3', 'Topic 4', 'Topic 5', 'Topic 6', 'Topic 7', 'Topic 8', 'Topic 9', 'Top
topics_df = pd.DataFrame(train_vecs, columns = column_names)
topics_df.head()
```

Out[16]:

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10	Topic 11	Topic 12	Topic 13	Topic 14
0	0.001950	0.167241	0.001950	0.001950	0.001950	0.001950	0.001950	0.001950	0.599759	0.001950	0.001950	0.092294	0.001950	0.001950
1	0.000502	0.000502	0.000502	0.000502	0.145020	0.000502	0.159989	0.000502	0.000502	0.048453	0.354032	0.000502	0.026404	0.000502
2	0.002409	0.002409	0.259074	0.105644	0.002409	0.002409	0.002409	0.002409	0.002409	0.190204	0.002409	0.002409	0.002409	0.002409
3	0.004078	0.004078	0.178871	0.004078	0.004078	0.004078	0.004078	0.004078	0.004078	0.004078	0.004078	0.004078	0.123712	0.004078
4	0.320628	0.393691	0.001505	0.001505	0.001505	0.001505	0.001505	0.001505	0.001505	0.001505	0.086873	0.001505	0.001505	0.001505

Add topic distribution to review dataset based on index

In [19]:
```
review = pd.read_csv('C:/Users/grezs/Desktop/Rencool/MAster/Aarhus/BSS stuff/4th semester - thesis/data/review_samp_1m.
```

In [20]:
```
review_topics =pd.concat([review, topics_df], axis=1)
```

In [21]:
```
review_topics.head()
```

Out[21]:

	Unnamed: 0	Id	Title	Price	User_id	profileName	review/helpfulness	review/score	review/time	review/summar
0	2917730	0974930415	Total Workday Control Using Microsoft Outlook:...	17.12	A4QLH3O9XXQB0	William M. Stabler	1/1	5.0	1186617600	Great Book- Work
1	948777	0451518845	Jane Eyre (Signet classics)	NaN	A2LJNVTOQTKEP0	T. Misbach	1/1	5.0	1321747200	Jane Eyr
2	641034	0740733265	Everybody Loves Ramen: Recipes, Stories, Games...	NaN	A1XCJRNPC4U1I6	Mallydoodle	0/0	5.0	1294012800	who doesn't lov rame
3	2107536	B0006AHG4C	The prince and the pauper;: A tale for young p...	NaN	ABANOHPDMIEYC	Daniel Krawisz	0/0	5.0	921888000	Extraordinar book! My favorit Mark Twai
4	1441547	1562790749	Synonym for Love	NaN	A1RH4BLM6BOHNH	Emily Pratt "raindol"	1/1	5.0	1098748800	great book at th right tim

5 rows × 30 columns

In [22]:
```
#Save topics appended review dataset
review_topics.to_csv("C:/Users/grezs/Desktop/Rencool/MAster/Aarhus/BSS stuff/4th semester - thesis/data/review_topics_1
```

Appendix 5. Topics 200k sample (first), Topics 1m sample (second)

	Topic	Keyword 1	Keyword 2	Keyword 3	Keyword 4	Keyword 5	Keyword 6	Keyword 7	Keyword 8	Keyword 9	Keyword 10	Keyword 11	Keyword 12	Keyword 13	Keyword 14	Keyword 15
1	**Politics**	society	political	government	human	power	economic	world	social	system	individual	law	value	today	public	
2	**Cognitive effort**	thing	think	know	get	go	bad	say	guy	try	time	maybe	good	people	way	
3	**Novel characteristics**	novel	writing	style	literature	reader	writer	literary	character	classic	prose	story	write	narrative	language	
4	**Opinion on plot**	character	plot	story	novel	series	mystery	main	main_character	ending	twist	end	action	reader	suspense	interesting
5	**Book attributes (internal content)**	chapter	text	topic	subject	theory	section	introduction	translation	study	example	language	reference	author	material	essay
6	**Experience**	life	heart	experience	feel	love	time	story	human	emotion	live	truly	world	soul	true	dream
7	**Family**	woman	man	love	family	wife	father	husband	marry	marriage	young	sister	mother	life	girl	relationship
8	**Recommendation**	love	good	like	story	great	think	enjoy	favorite	time	want	finish	recommend	series	write	get
9	**Improvement/ instructions**	information	learn	business	guide	need	technique	use	help	step	easy	useful	basic	helpful	program	tool
10	**Personal development (not genre)**	people	life	think	help	understand	change	person	way	thing	view	different	problem	point	deal	point_view
11	**Fantasy**	adventure	tale	world	evil	magic	ship	fantasy	sea	creature	island	land	planet	alien	king	story
12	**Young age**	child	year	old	ago	year_old	kid	boy	year_ago	young	adult	parent	girl	dog	son	age
13	**History**	history	historical	account	event	biography	research	period	write	detail	early	fascinating	author	century	interesting	historian
14	**Opinion on adaptation**	movie	review	film	version	waste	money	star	buy	see	rating	reviewer	publisher	copy	see_movie	waste_time
15	**Book attributes (external)**	edition	recipe	illustration	poem	food	photo	picture	color	cover	print	audio	music	version	cook	copy
16	**Philosophy/ Religion**	question	religion	christian	church	faith	answer	religious	philosophy	spiritual	belief	biblical	truth	prayer	catholic	teaching
17	**War**	war	military	crime	battle	fight	kill	murder	soldier	game	police	officer	army	team	camp	player
18	**American history/Western**	black	white	town highly_	travel	city	native	horse	trip	country	visit	plant science_	painting	american	slave	southern
19	**Education**	highly	school	recommend	recommend	class	student	high_school	high	teacher	college	fiction	condition	science	fi	reading

	Topic	Keyword 1	Keyword 2	Keyword 3	Keyword 4	Keyword 5	Keyword 6	Keyword 7	Keyword 8	Keyword 9	Keyword 10	Keyword 11	Keyword 12	Keyword 13	Keyword 14	Keyword 15
1	Young Age	year	child	old	young	ago	kid	year_old	story	boy	girl	year_ago	parent	love	adult	family
2	Experience	life	story	time	love	feel	experience	world	heart	way	live	find	write	reader	human	true
3	Book attributes (extrernal)	edition	recipe	look	illustration	cover	great	page	buy	picture	version	copy	good	food	poem	print
4	Cognitive effort	think	know	thing	go	get	time	good	want	way	people	bad	say	try	find	come
5	Opinion on adaptation	movie	review	buy	version	write	star	see	film	money	time	waste	well	author	copy	page
6	Politics	world	society	human	people	political	work	power	government	state	right	social	today	fact	idea	system
7	Novel characteristics	novel	story	character	work	write	reader	writing	style	time	writer	classic	literature	great	literary	good
8	Fantasy	world	story	adventure	tale	man	find	evil	fantasy	come	take	ship	magic	human	journey	land
9	Improvement/ instructions	information	good	learn	need	help	use	great	work	find	easy	business	guide	new	want	provide
10	Philosophy/ Religion	question	religion	answer	christian	faith	church	religious	philosophy	believe	truth	spiritual	belief	say	ask	man
11	Family	woman	man	love	family	life	young	father	wife	husband	mother	girl	marry	find	marriage	sister
12	War	war	kill	fight	battle	man	murder	military	crime	game	death	soldier	play	police	case	win
13	Education	highly	recommend	school	highly_recommend	class	high	student	great	high_school	teacher	reading	college	science	good	condition
14	History	history	write	historical	time	author	life	great	year	interesting	detail	event	account	early	work	know
15	Recommendation	love	good	story	great	think	like	time	enjoy	write	want	find	recommend	get	favorite	know
16	American history/Western	black	white	travel	town	place	american	country	small	city	people	live	trip	visit	horse	culture
17	Personal development (not genre)	people	life	think	way	help	thing	good	understand	change	person	feel	want	know	different	point
18	Book attributes (internal content)	chapter	text	author	find	reader	work	subject	word	example	study	good	include	language	write	section
19	Opinion on plot	character	story	novel	plot	series	good	reader	end	mystery	main	find	enjoy	interesting	main_character	action

Appendix 6 - Rest of the model building

In [17]:
```
# Import necessary libraries
import os
os.environ['BLAS_NUM_THREADS'] = '2'
import pandas as pd
import numpy as np
from tqdm import tqdm
import matplotlib as plt
import seaborn as sns
```

Data preparation

Import dataset

In [18]:
```
# Load 1m dataset already appended with topic distribution
review_df = pd.read_csv('C:/Users/grezs/Desktop/Rencool/MAster/Aarhus/BSS stuff/4th semester - thesis/data/review_topic

review_df.head()
```

Out[18]:

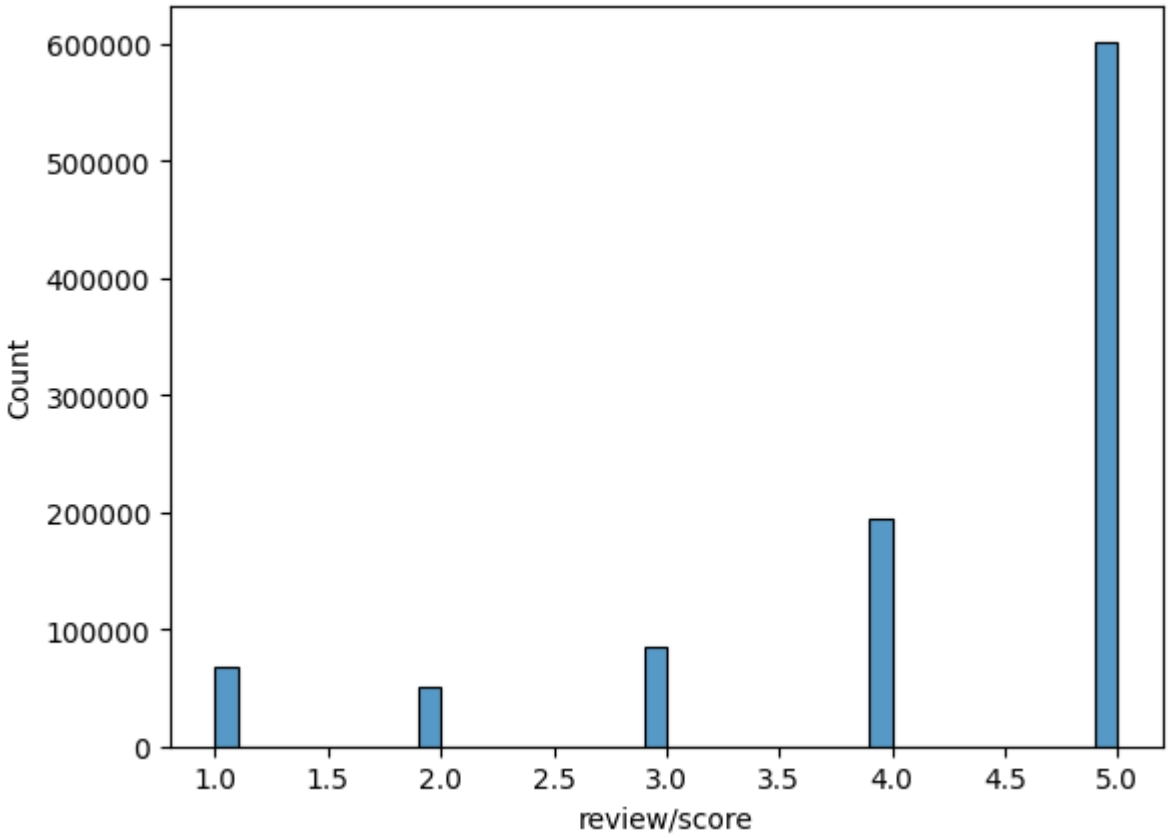
	Unnamed: 0.1	Unnamed: 0	Id	Title	Price	User_id	profileName	review/helpfulness	review/score	review/time	...
0	0	2917730	0974930415	Total Workday Control Using Microsoft Outlook:...	17.12	A4QLH3O9XXQB0	William M. Stabler	1/1	5.0	1186617600	...
1	1	948777	0451518845	Jane Eyre (Signet classics)	NaN	A2LJNVTOQTKEP0	T. Misbach	1/1	5.0	1321747200	...
2	2	641034	0740733265	Everybody Loves Ramen: Recipes, Stories, Games...	NaN	A1XCJRNPC4U1I6	Mallydoodle	0/0	5.0	1294012800	...
3	3	2107536	B0006AHG4C	The prince and the pauper;: A tale for young p...	NaN	ABANOHPDMIEYC	Daniel Krawisz	0/0	5.0	921888000	...
4	4	1441547	1562790749	Synonym for Love	NaN	A1RH4BLM6BOHNH	Emily Pratt "raindol"	1/1	5.0	1098748800	...

5 rows × 31 columns

Ratings

In [4]:
```
# visualize rating distribution
import seaborn
import matplotlib.pyplot as plt
seaborn.histplot(data=review_df, x = 'review/score', binwidth = 0.1)
```

Out[4]:
```
<Axes: xlabel='review/score', ylabel='Count'>
```



In [6]:
```
review_df['review/score'].value_counts()
```

Out[6]:
```
review/score
5.0    602049
4.0    194994
3.0     85239
1.0     67218
2.0     50498
Name: count, dtype: int64
```

Text length

In [19]:
```
#Add text length count
review_df['txt_len'] = review_df['review/text'].apply(lambda x: len(x))
review_df.head()
```

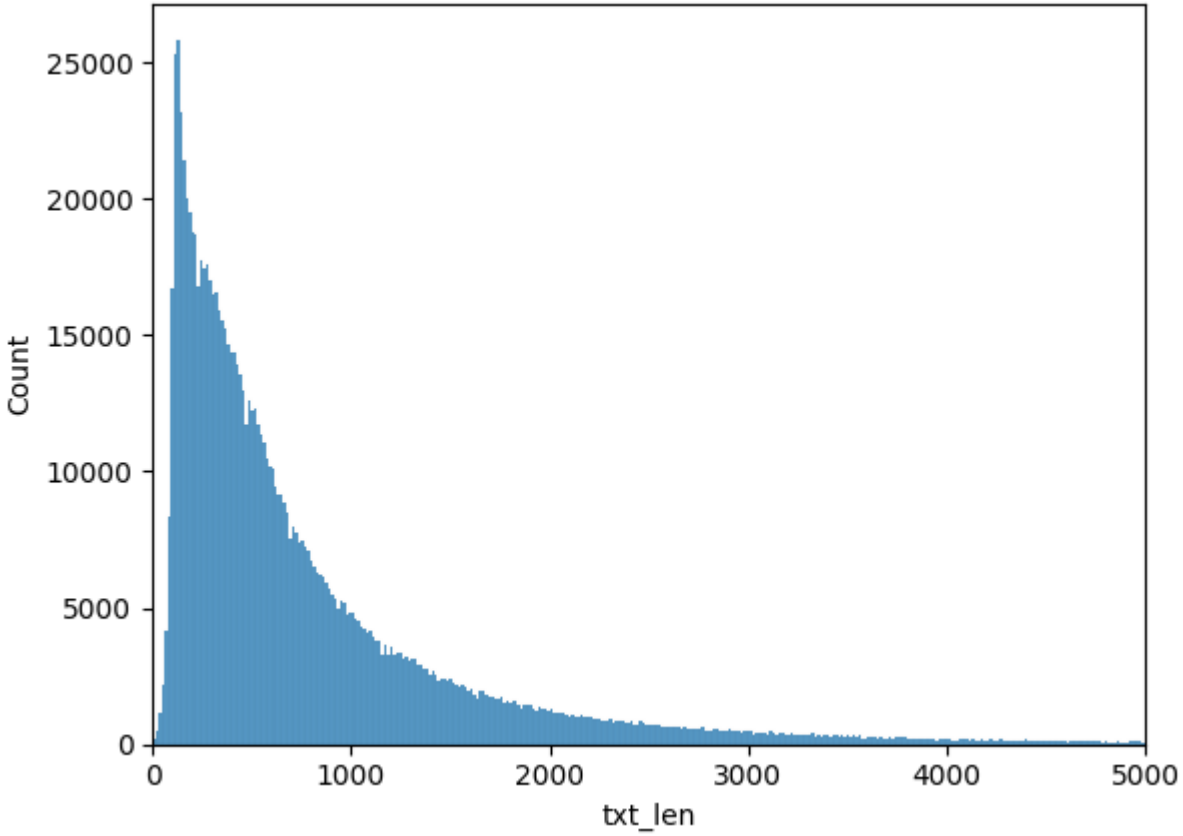
Out[19]:

	Unnamed: 0.1	Unnamed: 0	Id	Title	Price	User_id	profileName	review/helpfulness	review/score	review/time	...
0	0	2917730	0974930415	Total Workday Control Using Microsoft Outlook:...	17.12	A4QLH3O9XXQB0	William M. Stabler	1/1	5.0	1186617600	...
1	1	948777	0451518845	Jane Eyre (Signet classics)	NaN	A2LJNVTOQTKEP0	T. Misbach	1/1	5.0	1321747200	...
2	2	641034	0740733265	Everybody Loves Ramen: Recipes, Stories, Games...	NaN	A1XCJRNPC4U1I6	Mallydoodle	0/0	5.0	1294012800	...
3	3	2107536	B0006AHG4C	The prince and the pauper;: A tale for young p...	NaN	ABANOHPDMIEYC	Daniel Krawisz	0/0	5.0	921888000	...
4	4	1441547	1562790749	Synonym for Love	NaN	A1RH4BLM6BOHNH	Emily Pratt "raindol"	1/1	5.0	1098748800	...

5 rows × 32 columns

In [8]:

```
#Text length distribution
import matplotlib.pyplot as plt
import seaborn
seaborn.histplot(data=review_df, x = 'txt_len')

plt.xlim(0, 5000)

plt.show()

#negatively skewed, outliers after 2000 characters
```



In [21]:
```
review_df['txt_len'].median()
```

Out[21]:
```
517.0
```

In [27]:
```
review_df['txt_len'].describe(percentiles=[0.9])
```

Out[27]:
```
count    999998.000000
mean        824.773296
std         969.620706
min           1.000000
50%         517.000000
90%        1839.000000
max       32576.000000
Name: txt_len, dtype: float64
```

Ratings count

In [9]:
```
# Import book details
books = pd.read_csv('C:/Users/grezs/Desktop/Rencool/MAster/Aarhus/BSS stuff/4th semester - thesis/data/books_data.csv')
books.head()
```

Out[9]:

	Title	description	authors	image	previewLink	publisher	publishedDate	
0	Its Only Art If Its Well Hung!	NaN	['Julie Strain']	http://books.google.com/books/content?id=DykPA...	http://books.google.nl/books?id=DykPAAAACAAJ&d...	NaN	1996	http://books id=Dy
1	Dr. Seuss: American Icon	Philip Nel takes a fascinating look into the k...	['Philip Nel']	http://books.google.com/books/content?id=IjvHQ...	http://books.google.nl/books?id=IjvHQsCn_pgC&p...	A&C Black	2005-01-01	http://books id=I
2	Wonderful Worship in Smaller Churches	This resource includes twelve principles in un...	['David R. Ray']	http://books.google.com/books/content?id=2tsDA...	http://books.google.nl/books?id=2tsDAAAACAAJ&d...	NaN	2000	http://books id=2t
3	Whispers of the Wicked Saints	Julia Thomas finds her life spinning out of co...	['Veronica Haddon']	http://books.google.com/books/content?id=aRSIg...	http://books.google.nl/books?id=aRSIgJlq6JwC&d...	iUniverse	2005-02	http://books id=
4	Nation Dance: Religion, Identity and Cultural ...	NaN	['Edward Long']	NaN	http://books.google.nl/books?id=399SPgAACAAJ&d...	NaN	2003-03-01	http://books id=39

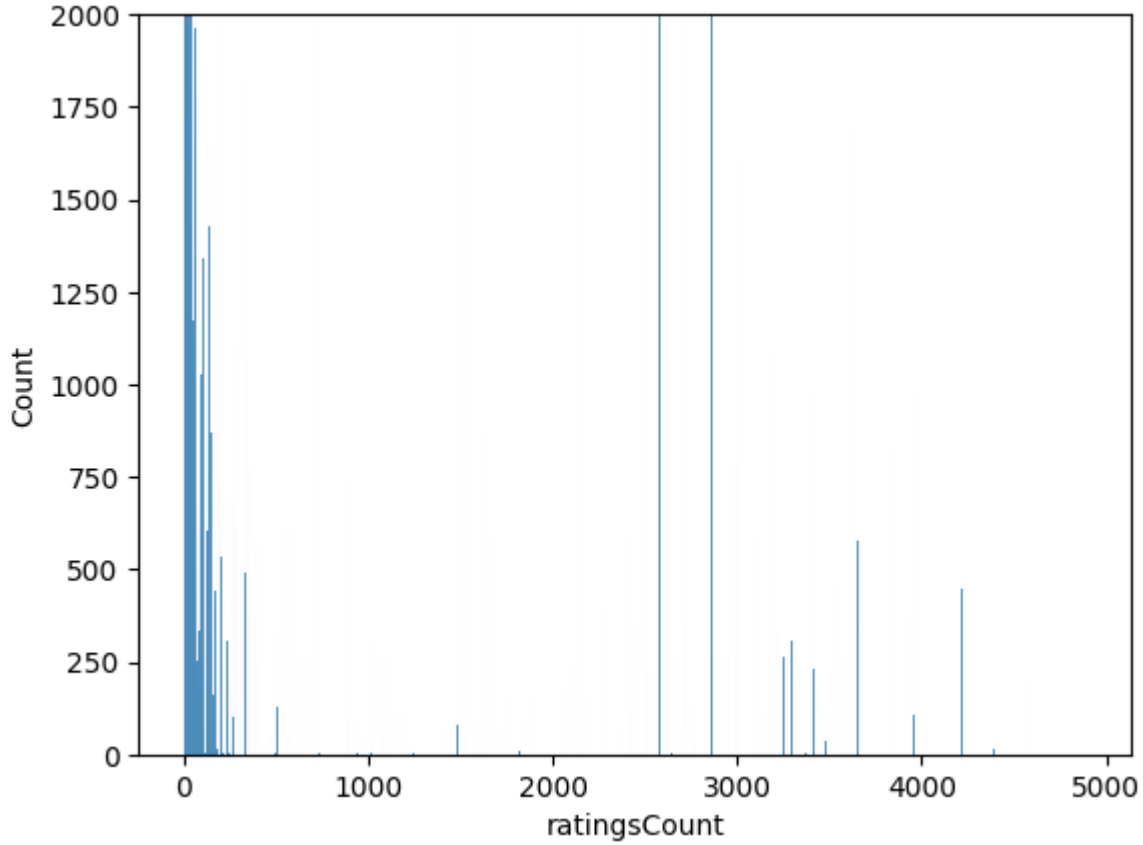
In [10]:
```
#Create subset - We only need Title and ratingsCount
book_ratings = books[['Title', 'ratingsCount']]
book_ratings.head()
```

Out[10]:

	Title	ratingsCount
0	Its Only Art If Its Well Hung!	NaN
1	Dr. Seuss: American Icon	NaN
2	Wonderful Worship in Smaller Churches	NaN
3	Whispers of the Wicked Saints	NaN
4	Nation Dance: Religion, Identity and Cultural ...	NaN

In [ ]:
```
# join books to reviews by title
# Works as a left join
review_df = pd.merge(review_df, book_ratings, on = 'Title')

review_df.info()
```

In [12]:
```
import seaborn
import matplotlib.pyplot as plt
seaborn.histplot(data=review_df, x = 'ratingsCount')
plt.ylim(0, 2000)

plt.show()
```



In [13]:
```
# We have some null values on the title and ratingsCount columns, lets delete them
title_null = review_df['Title'].isna().sum()
ratings_null = review_df['ratingsCount'].isna().sum()

print(title_null)
print(ratings_null)
```

```
63
453355
```

In [14]:
```
#Delete rows containing null values in Title column
review_df.dropna(subset=['Title'], inplace=True)
review_df.dropna(subset=['ratingsCount'], inplace=True)

title_null = review_df['Title'].isna().sum()
ratings_null = review_df['ratingsCount'].isna().sum()
```

```
print(title_null)
print(ratings_null)
```

```
0
0
```

In [ ]:
```
review_df.info()
```

In [16]:
```
# Save ratingscount appended dataframe to disk
# This one already has the topic scores and haveRatings appended
review_df.to_csv('C:/Users/grezs/Desktop/Rencool/MAster/Aarhus/BSS stuff/4th semester - thesis/data/reviews_ratingscoun
```

In [29]:
```
review_df = pd.read_csv('C:/Users/grezs/Desktop/Rencool/MAster/Aarhus/BSS stuff/4th semester - thesis/data/reviews_rati
```

In [31]:
```
review_df['ratingsCount'].describe()
```

Out[31]:
```
count    546580.000000
mean        270.342585
std         785.813633
min           1.000000
25%           3.000000
50%          10.000000
75%          54.000000
max        4895.000000
Name: ratingsCount, dtype: float64
```

In [32]:
```
review_df['ratingsCount'].median()
```

Out[32]:
```
10.0
```

Set time

In [34]:
```
# Convert timestamp to datetime
from datetime import datetime
review_df['review/time'] = pd.to_datetime(review_df['review/time'], unit='s')
review_df.head()
```

Out[34]:

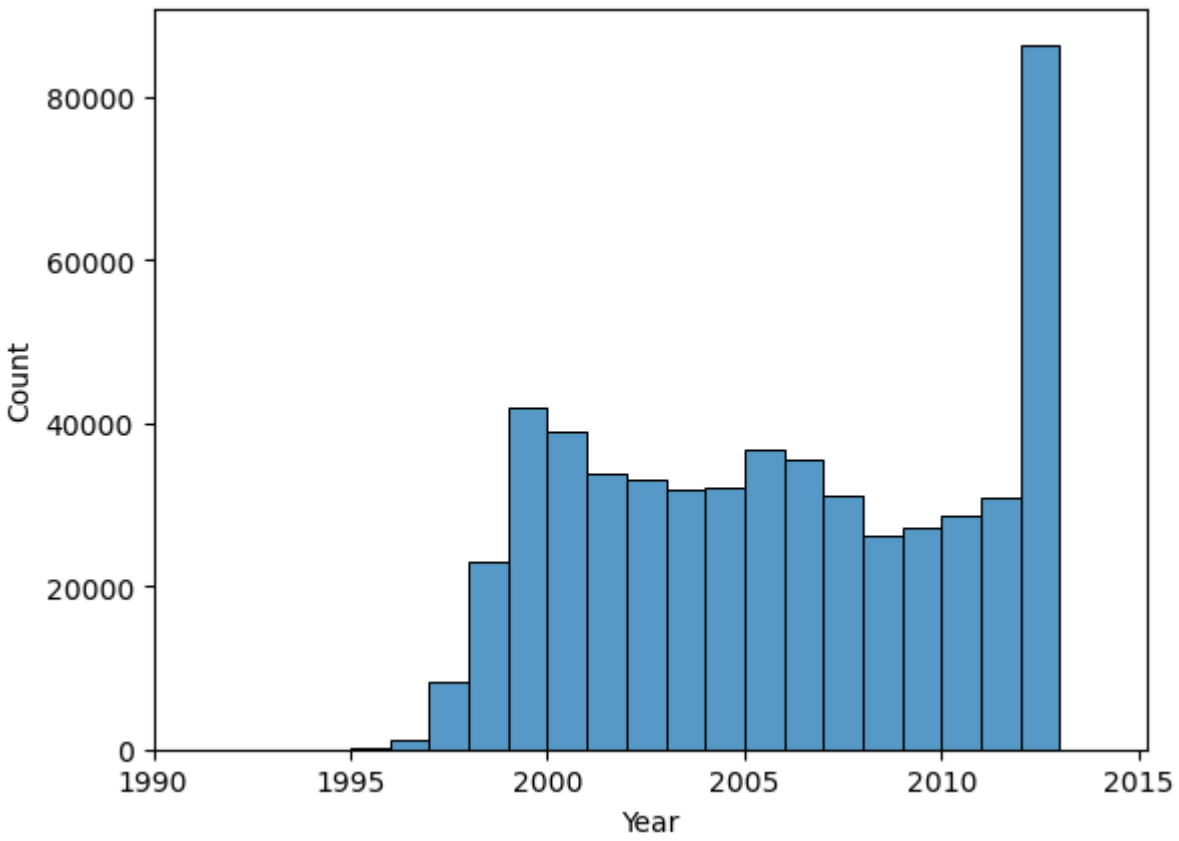
	Unnamed: 0.2	Unnamed: 0.1	Unnamed: 0	Id	Title	Price	User_id	profileName	review/helpfulness	review/score
0	43	1	948777	0451518845	Jane Eyre (Signet classics)	NaN	A2LJNVTOQTKEP0	T. Misbach	1/1	5.0
1	44	472	948767	0451518845	Jane Eyre (Signet classics)	NaN	A7WIT4FWF7LE7	Sara	2/2	4.0
2	45	3593	949695	0451518845	Jane Eyre (Signet classics)	NaN	NaN	NaN	1/2	5.0
3	46	4022	950000	0451518845	Jane Eyre (Signet classics)	NaN	A26CPHYQ17K806	Beatrice Asira	0/0	5.0
4	47	5463	949887	0451518845	Jane Eyre (Signet classics)	NaN	AEGONKCC7WE5O	Jen (jeneveive@hotmail.com)	2/2	4.0

5 rows × 34 columns

In [ ]:
```
#Convert time to datetime
from datetime import datetime
review_df['review/time'] = pd.to_datetime(review_df['review/time'], format='ISO8601')

#Now split it into years-moths-days
review_df['Year'] = review_df['review/time'].dt.year
review_df['Month'] = review_df['review/time'].dt.month
review_df['Day'] = review_df['review/time'].dt.day
review_df.info()
```

In [15]:
```
#Visualize years distribution
import matplotlib.pyplot as plt
seaborn.histplot(data=review_df, x = 'Year', binwidth=1)
```

```
plt.xlim(1990, None)
plt.show()
```



In [ ]:
```
review_df['Year'].value_counts()
```

In [43]:
```
review_df['Month'].value_counts()
```

Out[43]:
```
Month
1     64518
2     51907
12    51581
3     44262
11    43628
10    43358
7     42179
5     41733
8     41497
6     41071
4     40637
9     40209
Name: count, dtype: int64
```

In [44]:
```
review_df['Day'].value_counts()
```

Out[44]:
```
Day
6     19269
9     19106
28    18428
3     18350
8     18319
5     18298
21    18294
2     18221
10    18210
11    18101
19    18086
7     18083
27    18045
18    18030
26    18021
12    18002
13    17934
4     17872
20    17655
17    17621
16    17613
22    17500
23    17453
14    17449
24    17408
1     17354
15    17191
25    17048
30    16707
29    16626
31    10286
Name: count, dtype: int64
```

Delete unnecessary columns

In [16]:
```
review_df['Price'].isna().sum()
# To many values missing from price, doesnt worth risking imputation - should delete whole feature.
```

Out[16]:
```
470756
```

In [23]:
```
# Drop all columns we will likely not need
#'Unnamed: 0.1', 'Unnamed: 0_x', 'Id', 'Title', 'Price', 'User_id', 'profileName', 'review/sumary', 'Unnamed: 0_y'
```

```
review_clean = review_df.drop(columns=[ 'Unnamed: 0.1', 'Unnamed: 0', 'review/time', 'Id', 'Title', 'Price', 'User_id',
review_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 546580 entries, 43 to 999997
Data columns (total 27 columns):
 #   Column              Non-Null Count   Dtype  
---  ------              --------------   -----  
 0   review/helpfulness  546580 non-null  object 
 1   review/score        546580 non-null  float64
 2   review/text         546580 non-null  object 
 3   Topic 1             546580 non-null  float64
 4   Topic 2             546580 non-null  float64
 5   Topic 3             546580 non-null  float64
 6   Topic 4             546580 non-null  float64
 7   Topic 5             546580 non-null  float64
 8   Topic 6             546580 non-null  float64
 9   Topic 7             546580 non-null  float64
 10  Topic 8             546580 non-null  float64
 11  Topic 9             546580 non-null  float64
 12  Topic 10            546580 non-null  float64
 13  Topic 11            546580 non-null  float64
 14  Topic 12            546580 non-null  float64
 15  Topic 13            546580 non-null  float64
 16  Topic 14            546580 non-null  float64
 17  Topic 15            546580 non-null  float64
 18  Topic 16            546580 non-null  float64
 19  Topic 17            546580 non-null  float64
 20  Topic 18            546580 non-null  float64
 21  Topic 19            546580 non-null  float64
 22  ratingsCount        546580 non-null  float64
 23  txt_len             546580 non-null  int64  
 24  Year                546580 non-null  int64  
 25  Month               546580 non-null  int64  
 26  Day                 546580 non-null  int64  
dtypes: float64(21), int64(4), object(2)
memory usage: 116.8+ MB
```

In [24]:
```
review_clean.to_csv('C:/Users/grezs/Desktop/Rencool/MAster/Aarhus/BSS stuff/4th semester - thesis/data/reviews_clean_50
```

Target engineering

In [26]:
```
# Transform review/helpfulness column into helpf. votes and total votes
review_clean[['helpf.votes', 'total.votes']] = review_clean['review/helpfulness'].str.split("/", expand = True)

#convert new columns into integer
review_clean[['helpf.votes', 'total.votes']] = review_clean[['helpf.votes', 'total.votes']].astype('int')

# Make calculation for helpfulness ratio
review_clean['helpfulnessRatio'] = review_clean['helpf.votes'] / review_clean['total.votes']

review_clean.head(5)
```

Out[26]:

	Unnamed: 0	review/helpfulness	review/score	review/text	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	...	Topic 18	Topic
0	43	1/1	5.0	I decided to read the book after seeing the 19...	0.000502	0.000502	0.000502	0.000502	0.145020	0.000502	...	0.000502	0.0214
1	44	2/2	4.0	This book, like my title says, was hard for me...	0.001646	0.224390	0.001646	0.001646	0.001646	0.001646	...	0.001646	0.1494
2	45	1/2	5.0	Just imagine. You're reading along, you come t...	0.000337	0.320400	0.031533	0.233947	0.000337	0.100846	...	0.000337	0.0765
3	46	0/0	5.0	This is a book to read again and again I read ...	0.449445	0.004388	0.004388	0.004388	0.004388	0.004388	...	0.004388	0.0043
4	47	2/2	4.0	This was a good novel with an unpredictable pl...	0.001033	0.001033	0.001033	0.138355	0.090360	0.001033	...	0.001033	0.2010

5 rows × 31 columns

In [27]:
```
# Replace NaN in Helpfulness Ratio with 0 (0/0 gives NaN)
#Replace with 0
review_clean['helpfulnessRatio'] = review_clean['helpfulnessRatio'].fillna(0)

review_clean.head()
```

Out[27]:

	Unnamed: 0	review/helpfulness	review/score	review/text	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	...	Topic 18	Topic
0	43	1/1	5.0	I decided to read the book after seeing the 19...	0.000502	0.000502	0.000502	0.000502	0.145020	0.000502	...	0.000502	0.0214
1	44	2/2	4.0	This book, like my title says, was hard for me...	0.001646	0.224390	0.001646	0.001646	0.001646	0.001646	...	0.001646	0.1494
2	45	1/2	5.0	Just imagine. You're reading along, you come t...	0.000337	0.320400	0.031533	0.233947	0.000337	0.100846	...	0.000337	0.0765
3	46	0/0	5.0	This is a book to read again and again I read ...	0.449445	0.004388	0.004388	0.004388	0.004388	0.004388	...	0.004388	0.0043
4	47	2/2	4.0	This was a good novel with an unpredictable pl...	0.001033	0.001033	0.001033	0.138355	0.090360	0.001033	...	0.001033	0.2010

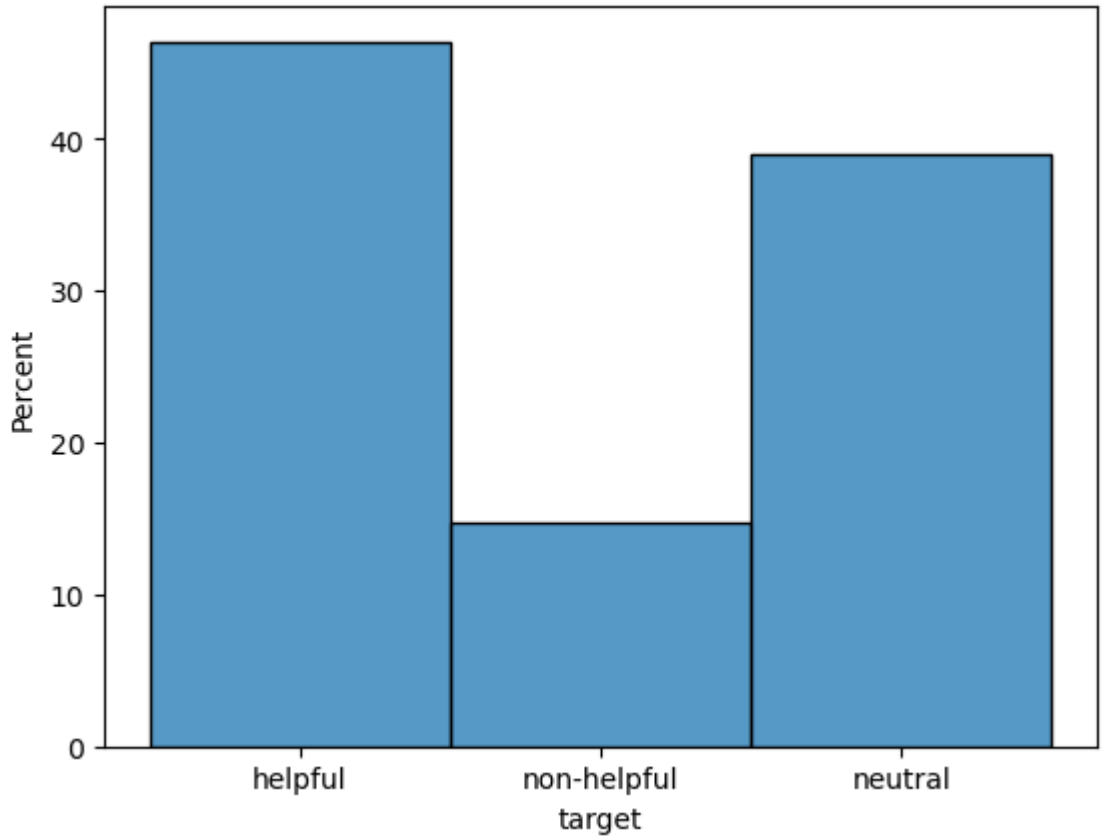
5 rows × 31 columns

In [ ]:
```
# Setting up target classes
# Classes: Non-helpful(helpfulnessRatio <= 0.5), Helpful (helpfulnessRatio > 0.5), Neutral (helpfulnessRatio = 0)
from tqdm import tqdm
review_clean['target'] = ''
```

```
for i, row in tqdm(review_clean.iterrows(), total=len(review_clean)):
    val = row['helpfulnessRatio']
    if val == 0:
        review_clean.at[i, 'target'] = 'neutral'
    elif val <= 0.5:
        review_clean.at[i, 'target'] = 'non-helpful'
    elif val > 0.5:
        review_clean.at[i, 'target'] = 'helpful'
    else:
        print('Error')

review_clean.head(10)
```

In [29]:

```
import matplotlib.pyplot as plt
import seaborn
seaborn.histplot(data=review_clean, x = 'target', stat='percent')
plt.show()
```



In [30]:
```
#check target class ratio
review_clean['target'].value_counts(normalize=True)
```

Out[30]:
```
helpful        0.463842
neutral        0.389153
non-helpful    0.147005
Name: target, dtype: float64
```

In [31]:
```
# delete helpfulnessRatio
review_clean = review_clean.drop(columns=['Unnamed: 0', 'helpfulnessRatio', 'helpf.votes', 'total.votes'])
```

In [33]:
```
review_clean.to_csv('C:/Users/grezs/Desktop/Rencool/MAster/Aarhus/BSS stuff/4th semester - thesis/data/reviews_withtarg
```

Sampling

In [77]:
```
review_clean = pd.read_csv('C:/Users/grezs/Desktop/Rencool/MAster/Aarhus/BSS stuff/4th semester - thesis/data/reviews_w
```

In [78]:
```
# Check the proportions for target of the main dataset
# We can use this to check whether random sampling will be as good as stratified sampling
class_counts = review_clean['target'].value_counts()
class_proportions = class_counts / len(review_clean)

# print the proportions
print(class_proportions)
```

```
helpful        0.463842
neutral        0.389153
non-helpful    0.147005
Name: target, dtype: float64
```

In [79]:
```
#Lets make a sample of 10k
clean_sample = review_clean.sample(n=50000)

#check proportions
class_counts = clean_sample['target'].value_counts()
class_proportions = class_counts / len(clean_sample)

# print the proportions
print(class_proportions)

#Less than 0.01% difference in sample size 50k in proportions, random is as good as stratified.
```

```
helpful        0.46722
neutral        0.38656
non-helpful    0.14622
Name: target, dtype: float64
```

In [80]:
```
#Save sample
clean_sample.to_csv('C:/Users/grezs/Desktop/Rencool/MAster/Aarhus/BSS stuff/4th semester - thesis/data/clean_sample_50k
```

Feature engineering

Sentiment analysis

In [ ]:
```
! pip install spacytextblob
```

In [ ]:
```
! python -m textblob.download_corpora
```

In [83]:
```
import pandas as pd
clean_sample = pd.read_csv('C:/Users/grezs/Desktop/Rencool/MAster/Aarhus/BSS stuff/4th semester - thesis/data/clean_sam
```

In [84]:
```
# connect the review text, where each review is a different line
samp_txt = clean_sample['review/text'].tolist()
```

In [85]:
```
import spacy
from spacytextblob.spacytextblob import SpacyTextBlob
from spacy.tokens import DocBin

nlp = spacy.load("en_core_web_md")
nlp.add_pipe('spacytextblob')
```

Out[85]:
```
<spacytextblob.spacytextblob.SpacyTextBlob at 0x1fdf1ac65f0>
```

In [86]:
```
from tqdm import tqdm

sentiment = pd.DataFrame(columns=['id', 'polarity'])
for i in tqdm(range(len(samp_txt))):

    text = samp_txt[i]
    # put every text to nlp
    doc = nlp(text)
    # call blob
    polarity = doc._.blob.polarity

    new_df = pd.DataFrame({'id' : i, 'polarity' : polarity}, index=[i])
    sentiment = pd.concat([sentiment, new_df], ignore_index = True, sort = False)
```

```
100%|██████████| 50000/50000 [32:14<00:00, 25.85it/s]
```

In [87]:
```
sentiment
```

Out[87]:

	id	polarity
0	0	0.390833
1	1	0.116667
2	2	0.303106
3	3	0.013381
4	4	0.650000
...	...	...
49995	49995	0.500000
49996	49996	0.010802
49997	49997	0.437500
49998	49998	0.071875
49999	49999	0.154545

50000 rows × 2 columns

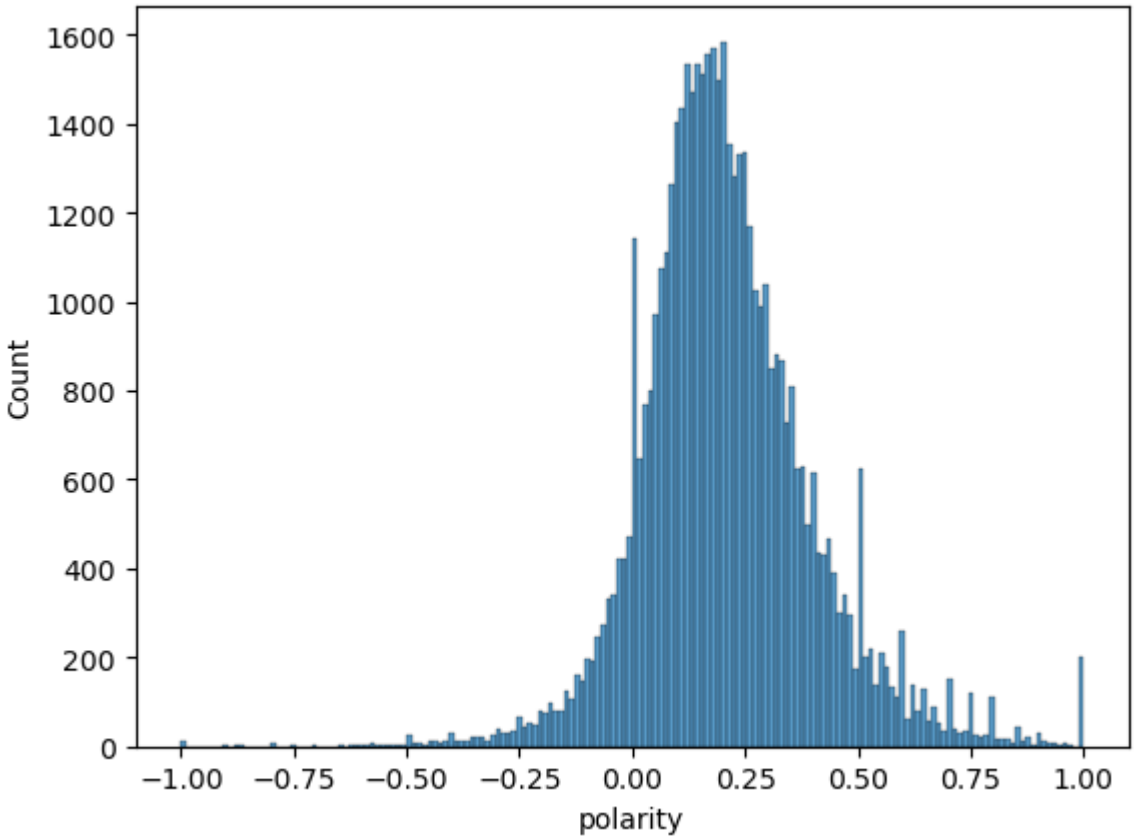
In [88]:
```
# Safety check if the sentiment df is is in the right order - should be
text1 = nlp(samp_txt[0])
textlast = nlp(samp_txt[49999])

polarity1 = text1._.blob.polarity
polarityl = textlast._.blob.polarity

print(polarity1, polarityl)
#yes it is
```

```
0.39083333333333337 0.15454545454545454
```

In [89]:
```
#visualize sentiment distribution
import seaborn
import matplotlib.pyplot as plt
seaborn.histplot(data=sentiment, x = 'polarity')

# The sentiment scores have a very nice, even distribution, however a bit positively skewed.
```

Out[89]:

```
<Axes: xlabel='polarity', ylabel='Count'>
```



In [ ]:
```
sample_sent = pd.concat([clean_sample, sentiment['polarity']], axis=1)
sample_sent.head()
```

In [91]:
```
#Save sentiment appended 50k sample
sample_sent.to_csv('C:/Users/grezs/Desktop/Rencool/MAster/Aarhus/BSS stuff/4th semester - thesis/data/sample_50k_clean_
```

Emotion Detection

In [1]:
```
import tensorflow as tf
from transformers import RobertaTokenizerFast, TFRobertaForSequenceClassification, pipeline

tokenizer = RobertaTokenizerFast.from_pretrained("arpanghoshal/EmoRoBERTa")
model = TFRobertaForSequenceClassification.from_pretrained("arpanghoshal/EmoRoBERTa")

emotion = pipeline('sentiment-analysis', model='arpanghoshal/EmoRoBERTa', top_k= None)
```

```
C:\Users\grezs\anaconda_newlocation\anaconda3\envs\tensorflow\lib\site-packages\tqdm\auto.py:21: TqdmWarning: IProgress
not found. Please update jupyter and ipywidgets. See https://ipywidgets.readthedocs.io/en/stable/user_install.html
  from .autonotebook import tqdm as notebook_tqdm
All model checkpoint layers were used when initializing TFRobertaForSequenceClassification.

All the layers of TFRobertaForSequenceClassification were initialized from the model checkpoint at arpanghoshal/EmoRoBE
RTa.
If your task is similar to the task the model of the checkpoint was trained on, you can already use TFRobertaForSequenc
eClassification for predictions without further training.
All model checkpoint layers were used when initializing TFRobertaForSequenceClassification.

All the layers of TFRobertaForSequenceClassification were initialized from the model checkpoint at arpanghoshal/EmoRoBE
RTa.
If your task is similar to the task the model of the checkpoint was trained on, you can already use TFRobertaForSequenc
eClassification for predictions without further training.
```

In [4]:
```
sample_sent = pd.read_csv('C:/Users/grezs/Desktop/Rencool/MAster/Aarhus/BSS stuff/4th semester - thesis/data/sample_50k
```

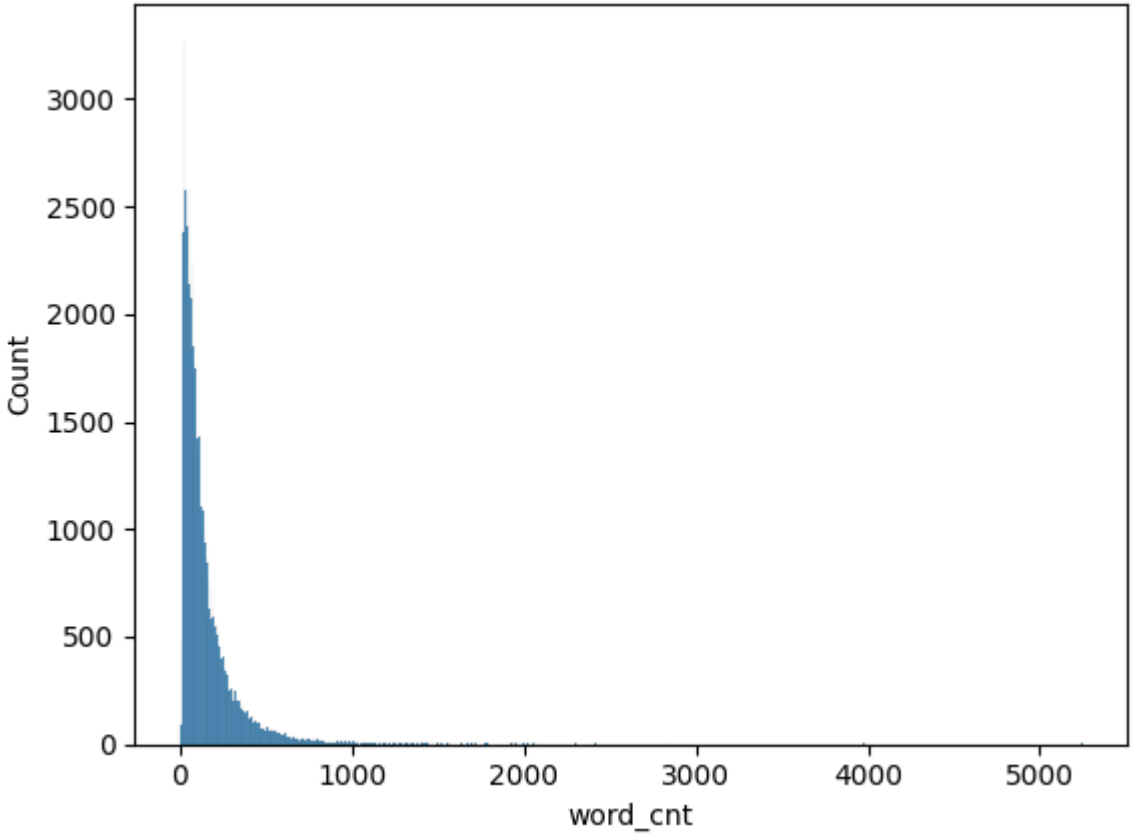
In [5]:
```
# EmoRoBERTa only takes in review with maximum 510 tokens, so we will have to remove everything above that.
# Lets start with removing all reviews above 510 words (apprx 1 page)

#create wordcount
sample_sent['word_cnt'] = sample_sent['review/text'].apply(lambda x: len(x.split()))
sample_sent.head()
```

Out[5]:

	Unnamed: 0.2	Unnamed: 0.1	Unnamed: 0	review/helpfulness	review/score	review/text	Topic 1	Topic 2	Topic 3	Topic 4	...	Topic 18	Topi
0	0	364080	364080	1/1	5.0	If it can be found living in the wilds (or not...	0.001122	0.001122	0.140510	0.001122	...	0.001122	0.00
1	1	312056	312056	0/4	1.0	Simon, Prince Joshua and Simon's old friend Bi...	0.151641	0.003116	0.003116	0.358671	...	0.003116	0.00
2	2	293679	293679	0/1	5.0	this is one book all new and begining knitters...	0.003104	0.003104	0.003104	0.213502	...	0.003104	0.00
3	3	434538	434538	0/0	5.0	'Sinners in the Hands of an Angry God', goes s...	0.000451	0.077336	0.000451	0.191841	...	0.060277	0.03
4	4	521796	521796	0/0	5.0	The book is quite the tongue twister, but fun ...	0.007538	0.007538	0.007538	0.007538	...	0.007538	0.00

5 rows × 33 columns

In [6]:
```
# Plot word count
import matplotlib.pyplot as plt
import seaborn
seaborn.histplot(data=sample_sent, x = 'word_cnt')

plt.show()
# we can see that the observations above 510 words are marginal, and can be treatedc as are outliers (<5%), we can dele
```



In [7]:
```
# drop above 510
#no_below = sample_sent.drop(sample_sent[sample_sent['word_cnt'] > 510].index)
no_above = sample_sent[sample_sent['word_cnt'] <= 510].reset_index(drop=True)
# convert them to a list
samp_list = no_above['review/text'].tolist()
# Check length to see how much we dropped
print(len(samp_list)/len(sample_sent))

#Observations over 510 words represented less than 3.7% of all observations, deleting them should not have a large impa
```

```
0.96326
```

In [16]:
```
# I found that there are 2 more occasions, where our text is above 510 tokens.
# I did nothing wrong, these texts include less than 510 words, however even though we used an english database,
# these reviews include spanish text and most likely Spacy's english vocabulary based tokenizer can split up the non-eng
# and count them as multile tokens. Since I only have 2 instances, I can remove them manually and procede.
no_above = no_above.drop([1220, 26933, 31172, 33039]).reset_index(drop=True)
```

In [17]:
```
no_above.to_csv('C:/Users/grezs/Desktop/Rencool/MAster/Aarhus/BSS stuff/4th semester - thesis/data/messin_around/samp_l
```

In [18]:
```
# convert them to a list
samp_list = no_above['review/text'].tolist()
```

In [19]:
```
import spacy
from tqdm import tqdm
from spacy.tokens import DocBin

nlp = spacy.load("en_core_web_md")

doc_bin = DocBin(attrs=["LEMMA", "POS"])
doc_bin_all = DocBin(attrs=["LEMMA", "POS"])

batch_size = 5
counter = 0
for doc in tqdm(nlp.pipe(samp_list, n_process=2, disable=["parser", "ner"]), total=len(samp_list)):
    doc_bin.add(doc)
    counter += 1

    if counter == batch_size:
        doc_bin_all.merge(doc_bin)
        doc_bin = DocBin(attrs=["LEMMA", "POS"])
        counter = 0
doc_bin_all.merge(doc_bin)
```

```
100%|██████████| 48159/48159 [07:04<00:00, 113.38it/s]
```

In [20]:
```
# save docbin to disk for safety
doc_bin_all.to_disk("C:/Users/grezs/Desktop/Rencool/MAster/Aarhus/BSS stuff/4th semester - thesis/data/docbin_50k_forem
```

In [ ]:
```
#if we need to reload
doc_bin_all = DocBin().from_disk("C:/Users/grezs/Desktop/Rencool/MAster/Aarhus/BSS stuff/4th semester - thesis/data/doc
```

In [21]:
```
# get documents from Spacy docbin
docs = list(doc_bin_all.get_docs(nlp.vocab))
```

In [22]:
```
# Now lets remove all the extra stuff, ensuring that we have less than 510 tokens
texts, article = [], []
for doc in tqdm(docs, total=len(docs)):
    for token in doc:
        if not token.is_stop and not token.is_punct and not token.like_num and token.is_alpha:
```

```
            article.append(token.lemma_)

    texts.append(article)
    article = []

len(texts)#Safety checkup
```

```
100%|██████████| 48159/48159 [00:03<00:00, 14932.05it/s]
```

Out[22]:
```
48159
```

In [23]:
```
#EmoRoBERTa takes input as a string or a list of strings
#Convert the nested list back to a full list made of strings
list_of_strings = [' '.join(article) for article in texts]
#print(list_of_strings)
```

In [24]:
```
for i, seq in enumerate(list_of_strings):
    tokenized_seq = tokenizer.encode(seq, add_special_tokens=True)
    if len(tokenized_seq) > 510:
        print(f"Error: Sequence {i} has length {len(tokenized_seq)} {seq}")
    else:
        continue

#Our token counter didn't drop any errors, so everything is below 510 tokens, we are good to proceed.
```

In [25]:
```
def process_string(string, update_fn):
    result = emotion([string])
    update_fn()
    return result[0]

with tqdm(total=len(list_of_strings)) as pbar:
    emo_collection = [process_string(string, pbar.update) for string in list_of_strings]
```

```
100%|██████████| 48159/48159 [7:01:13<00:00,  1.91it/s]
```

In [ ]:
```
emo_collection
```

In [26]:
```
df = pd.DataFrame()
for emos in tqdm(emo_collection, total=len(emo_collection)):
    new_df = pd.DataFrame(emos)
    new_df = new_df.set_index('label').T
    df = pd.concat([df, new_df])
df.reset_index(drop=True, inplace=True)
```

```
100%|██████████| 48159/48159 [03:05<00:00, 259.10it/s]
```

In [27]:
```
df
```

Out[27]:

	label	admiration	neutral	gratitude	pride	approval	optimism	excitement	amusement	joy	relief	...	disappointment	ca
0		0.988481	0.006527	0.001576	0.000623	0.000619	0.000617	0.000350	0.000155	0.000120	0.000117	...	0.000050	0.000
1		0.000513	0.997062	0.000051	0.000008	0.000903	0.000405	0.000019	0.000105	0.000020	0.000005	...	0.000100	0.000
2		0.012520	0.913324	0.003006	0.000045	0.016417	0.044521	0.000159	0.000159	0.000444	0.000803	...	0.000222	0.004
3		0.001971	0.956461	0.000097	0.000107	0.000863	0.002978	0.000091	0.000210	0.000040	0.000037	...	0.000205	0.001
4		0.903452	0.002280	0.001271	0.000134	0.055195	0.004547	0.015594	0.005075	0.006088	0.000578	...	0.000060	0.001
...		...	...	...	...	...	...	...	...	...	...	...	...	
48154		0.000201	0.006577	0.000192	0.000029	0.004854	0.000079	0.016153	0.002578	0.966674	0.000075	...	0.000045	0.000
48155		0.001275	0.995490	0.000181	0.000008	0.000772	0.000734	0.000021	0.000077	0.000017	0.000008	...	0.000152	0.000
48156		0.354630	0.509981	0.071184	0.000099	0.040449	0.015337	0.000650	0.000286	0.001083	0.001180	...	0.000320	0.002
48157		0.959521	0.004109	0.001254	0.000209	0.010590	0.011058	0.005782	0.001774	0.002350	0.000371	...	0.000034	0.000
48158		0.992044	0.000520	0.000323	0.000090	0.003455	0.000783	0.000395	0.000100	0.000509	0.000117	...	0.000065	0.000

48159 rows × 28 columns

In [28]:
```
# Add emotions to main data sample
samp_sent_emo = pd.concat([no_above, df], axis=1)
samp_sent_emo.head()
```

Out[28]:

	Unnamed: 0.2	Unnamed: 0.1	Unnamed: 0	review/helpfulness	review/score	review/text	Topic 1	Topic 2	Topic 3	Topic 4	...	disappointmen
0	0	364080	364080	1/1	5.0	If it can be found living in the wilds (or not...	0.001122	0.001122	0.140510	0.001122	...	0.00005
1	1	312056	312056	0/4	1.0	Simon, Prince Joshua and Simon's old friend Bi...	0.151641	0.003116	0.003116	0.358671	...	0.00010
2	2	293679	293679	0/1	5.0	this is one book all new and begining knitters...	0.003104	0.003104	0.003104	0.213502	...	0.00022
3	3	434538	434538	0/0	5.0	'Sinners in the Hands of an Angry God', goes s...	0.000451	0.077336	0.000451	0.191841	...	0.00020
4	4	521796	521796	0/0	5.0	The book is quite the tongue twister, but fun ...	0.007538	0.007538	0.007538	0.007538	...	0.00006

5 rows × 61 columns

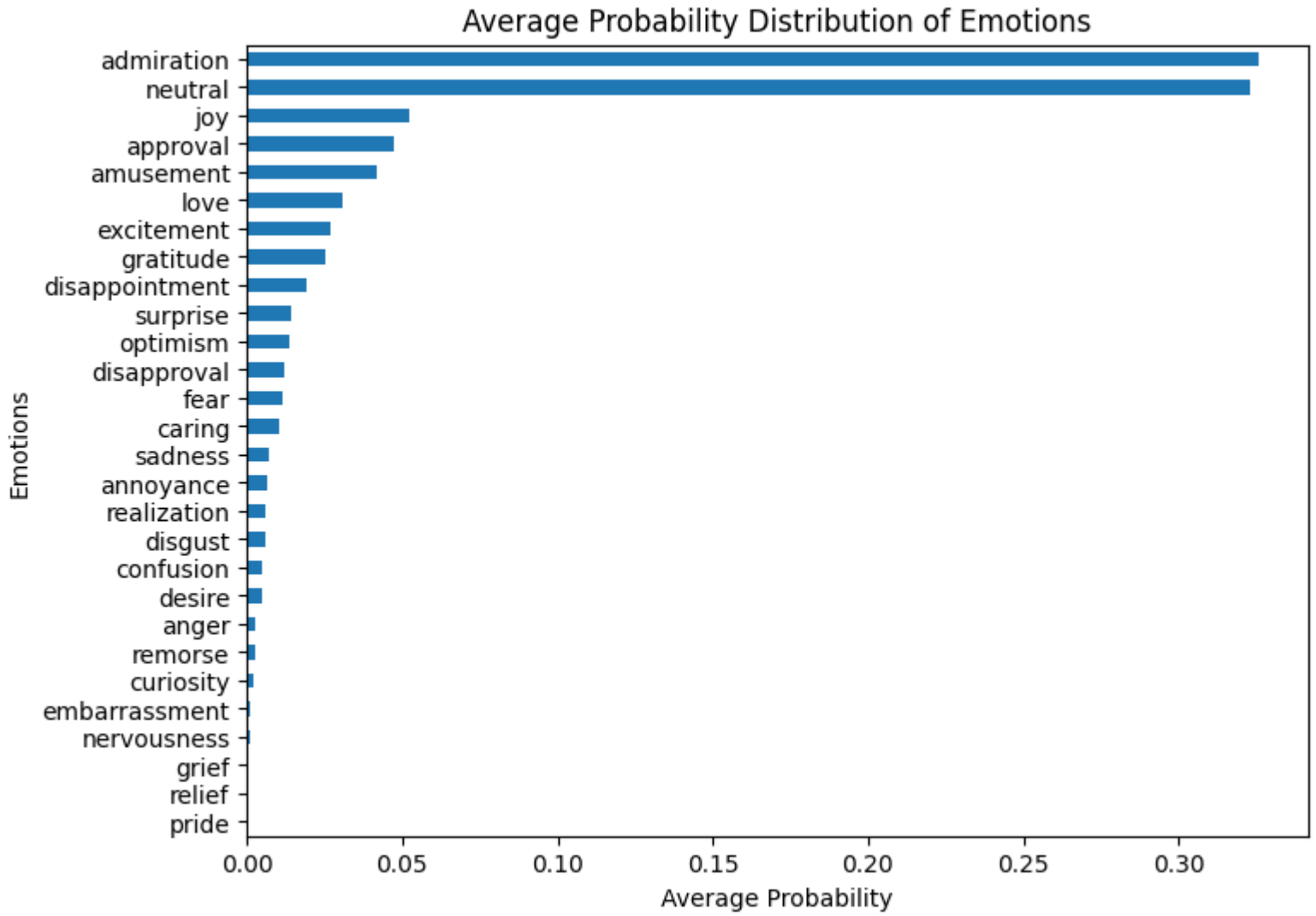
In [ ]:
```
#Delete unnamed columns
samp_sent_emo = samp_sent_emo.drop(columns=['Unnamed: 0.2', 'Unnamed: 0.1', 'Unnamed: 0'])
samp_sent_emo.info()
```

In [30]:
```
#Save to disk
samp_sent_emo.to_csv('C:/Users/grezs/Desktop/Rencool/MAster/Aarhus/BSS stuff/4th semester - thesis/data/samp_emoandsent_
```

In [46]:
```
emotions = pd.read_csv('C:/Users/grezs/Desktop/Rencool/MAster/Aarhus/BSS stuff/4th semester - thesis/data/samp_emoandse
```

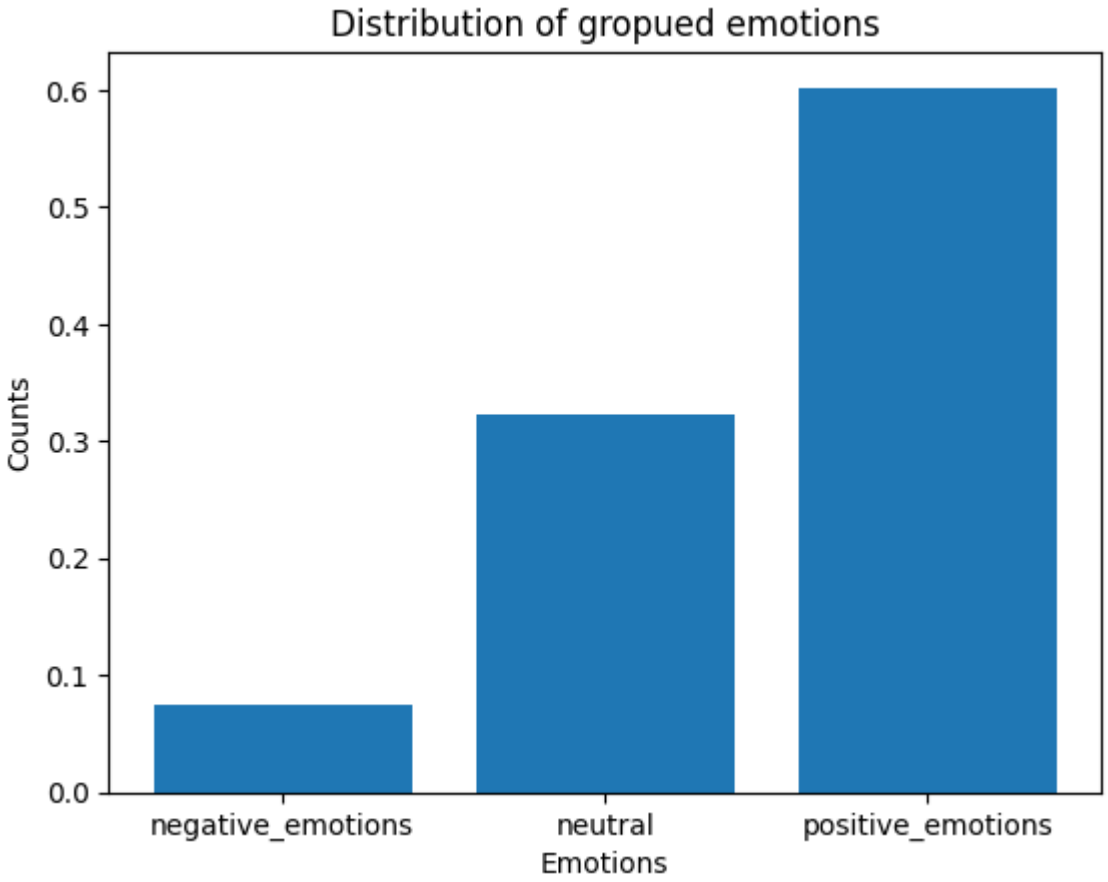
In [48]:
```
emotions = emotions[['neutral', 'admiration', 'gratitude', 'approval',
       'optimism', 'caring', 'joy', 'relief', 'excitement', 'realization',
       'amusement', 'surprise', 'disappointment', 'remorse', 'grief',
       'disapproval', 'desire', 'annoyance', 'love', 'confusion', 'pride',
       'curiosity', 'disgust', 'fear', 'sadness', 'anger', 'embarrassment',
       'nervousness']]
```

In [72]:
```
import seaborn as sns
import matplotlib.pyplot as plt

averages = emotions.mean()
averages = averages.sort_values()

plt.figure(figsize=(8, 6))  # Optional: adjust the figure size
averages.plot(kind='barh')

# Set plot labels and title
plt.ylabel('Emotions')
plt.xlabel('Average Probability')
plt.title('Average Probability Distribution of Emotions')

# Display the plot
plt.show()
```



In [94]:
```
negatives = emotions[['disappointment', 'remorse', 'grief',
       'disapproval',  'annoyance', 'confusion', 'disgust',
       'fear', 'sadness', 'anger', 'embarrassment', 'nervousness']]
positives = emotions [['admiration', 'gratitude', 'approval',
       'optimism', 'caring', 'joy', 'relief', 'excitement', 'realization',
       'amusement', 'surprise', 'desire', 'love', 'pride', 'curiosity',]]
neutral = emotions[['neutral']]

negative_emotions = negatives.mean().sum()
positive_emotions = positives.mean().sum()
```

```
neutral_emotions = neutral.mean().sum()

counts = [negative_emotions, neutral_emotions, positive_emotions]
grouped_em = ['negative_emotions', 'neutral', 'positive_emotions']

plt.bar(grouped_em, counts)

# Set plot labels and title
plt.xlabel('Emotions')
plt.ylabel('Counts')
plt.title('Distribution of gropued emotions')

# Display the plot
plt.show()
```



Delet unnecessary columns

In [ ]:
```
# Delete unnecessary columns such as text (we have the topics, emotions, sentiment polarity), word count(We have text l
reviews = reviews.drop(columns=['Unnamed: 0', 'review/text', 'word_cnt'])
reviews.info()
```

Target as last column

In [46]:
```
reviews.to_csv('C:/Users/grezs/Desktop/Rencool/MAster/Aarhus/BSS stuff/4th semester - thesis/data/semifinal_50k.csv')
```

In [ ]:
```
reviews = reviews[['Year', 'Month', 'Day', 'review/score', 'ratingsCount', 'txt_len',
       'Topic 1', 'Topic 2', 'Topic 3', 'Topic 4', 'Topic 5', 'Topic 6', 'Topic 7',
       'Topic 8', 'Topic 9', 'Topic 10', 'Topic 11', 'Topic 12', 'Topic 13', 'Topic 14', 'Topic 15',
       'Topic 16', 'Topic 17', 'Topic 18', 'Topic 19',
       'polarity', 'neutral', 'admiration', 'gratitude', 'approval',
       'optimism', 'caring', 'joy', 'relief', 'excitement', 'realization',
       'amusement', 'surprise', 'disappointment', 'remorse', 'grief',
       'disapproval', 'desire', 'annoyance', 'love', 'confusion', 'pride',
       'curiosity', 'disgust', 'fear', 'sadness', 'anger', 'embarrassment',
       'nervousness', 'target']]
```

Classification

Baseline model

In [78]:
```
reviews = pd.read_csv('C:/Users/grezs/Desktop/Rencool/MAster/Aarhus/BSS stuff/4th semester - thesis/data/semifinal_50k.
```

In [79]:
```
reviews=reviews.drop(columns=['Unnamed: 0'])
```

In [80]:
```
y = np.array(reviews['target'])
X = reviews.drop(columns=['target'])
```

Stratified sampling

In [81]:
```
from sklearn.model_selection import train_test_split
# Split the dataset into training and testing using stratified sampling
# Later we will perform a k fold cross validation, so we dont need a separate validation sample
# It is done in order to maximize the size of the training sample
# Let the ratio be 80-20

# First split train-remaining
X_train, X_test, y_train, y_test = train_test_split(X,y, train_size=0.8, random_state=42, stratify=y)


print(X_train.shape), print(y_train.shape)
print(X_test.shape), print(y_test.shape)
```

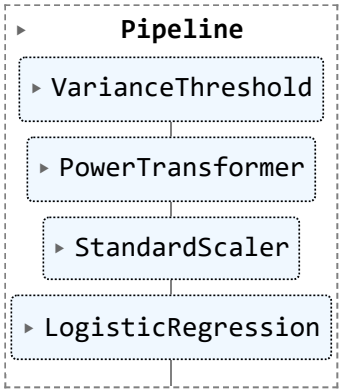
Out[81]:
```
(38527, 54)
(38527,)
(9632, 54)
(9632,)
(None, None)
```

Create preprocessing pipeline for logistic regression

In [82]:
```
# Pipeline
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import PowerTransformer
from sklearn.linear_model import LogisticRegression
from sklearn.feature_selection import VarianceThreshold


# Create the steps to be performed
steps = [('variance', VarianceThreshold(threshold=0.01)),
         ('yeo-johnson', PowerTransformer()), # default is yeo-johnson transformation
         ('scale', StandardScaler()),
         ('LR', LogisticRegression(multi_class='ovr')) ]

# create pipeline object
pipe = Pipeline(steps)
```

In [83]:
```
pipe.fit(X_train, y_train)
```

Out[83]:



In [66]:
```
# kfold cross validation
from sklearn.model_selection import cross_validate

# Pass pipeline and training and test data set
base_logreg = cross_validate(pipe, X=X_train, y=y_train, cv=5, n_jobs=1)

print(f"Accuracy scores for baseline logreg: {base_logreg['test_score']}")
print(f"Mean accuracy score for baseline logreg: {base_logreg['test_score'].mean()*100}")
```

```
Accuracy scores for baseline logreg: [0.6067999  0.60070075 0.59428942 0.60129786 0.60713822]
Mean accuracy score for baseline logreg: 60.204523035304746
```

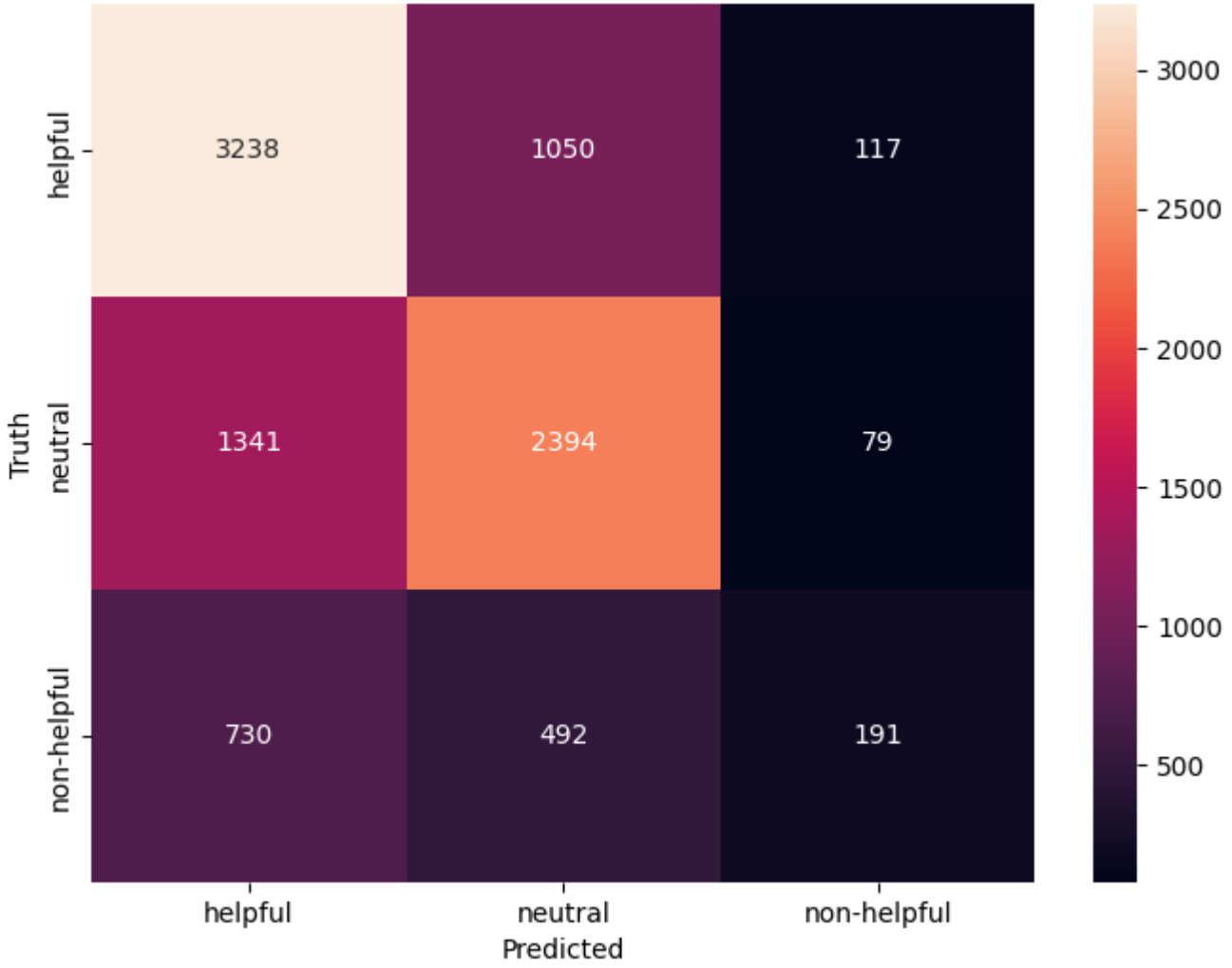
In [84]:
```
base_acc = pipe.score(X_train, y_train)
base_acc
```

Out[84]:
```
0.6023308329223661
```

In [68]:
```
#Visualize where did it all go wrong
from sklearn.metrics import confusion_matrix
# Make predictions on the test data
y_pred = pipe.predict(X_test)

# Calculate the confusion matrix
cm = confusion_matrix(y_test, y_pred)

import seaborn as sn
import matplotlib.pyplot as plt
plt.figure(figsize = (8,6))
sn.heatmap(cm, fmt='d', annot=True, xticklabels=pipe.classes_, yticklabels=pipe.classes_)
```

```
plt.xlabel('Predicted')
plt.ylabel('Truth')
```

Out[68]:
```
Text(70.72222222222221, 0.5, 'Truth')
```



In [69]:
```
#precision, recall
from sklearn.metrics import precision_score, recall_score
precision_base = precision_score(y_test, y_pred, average='macro')

recall_base = recall_score(y_test, y_pred, average='macro')
```

```
print(f' Precision score baseline: {precision_base}')
print(f' Recall score baseline: {recall_base}')
```

```
 Precision score baseline: 0.570559820965233
 Recall score baseline: 0.4993115456573854
```

Fine-tuning

In [70]:
```
reviews = pd.read_csv('C:/Users/grezs/Desktop/Rencool/MAster/Aarhus/BSS stuff/4th semester - thesis/data/semifinal_50k.
```

In [71]:
```
reviews=reviews.drop(columns=['Unnamed: 0'])
```

In [72]:
```
# get rid of non-helpful reviews
no_bad = reviews[reviews['target'].str.contains('non-helpful')==False]
no_bad['target'].value_counts()
```

Out[72]:
```
target
helpful    22023
neutral    19068
Name: count, dtype: int64
```

In [73]:
```
# THe data is a bit imbalanced
# Upsample neutral class
from sklearn.utils import resample

# Separate classes, minority: Neutral, majority: Helpful
no_bad_majority = no_bad[no_bad['target'].str.contains('neutral')==False]
no_bad_minority = no_bad[no_bad['target'].str.contains('helpful')==False]

# Upsample minority class
no_bad_minority_upsampled = resample(no_bad_minority,
                                     replace=True,     # sample with replacement
                                     n_samples=22023,    # to match majority class
                                     random_state=123) # reproducible results

# Combine majority class with upsampled minority class
no_bad_upsampled = pd.concat([no_bad_majority, no_bad_minority_upsampled])

# Display new class counts
no_bad_upsampled['target'].value_counts()
```

Out[73]:
```
target
helpful    22023
neutral    22023
Name: count, dtype: int64
```

In [74]:
```
# Separate target from features
y = np.array(no_bad_upsampled['target'])
X = no_bad_upsampled.drop(columns=['target'])
```

In [75]:
```
# Split the dataset into training and testing using stratified sampling
# Later we will perform a k fold cross validation, so we dont need a separate validation sample
# It is done in order to maximize the size of the training sample
# Let the ratio be 80-20
from sklearn.model_selection import train_test_split

# First split train-remaining
X_train, X_test, y_train, y_test = train_test_split(X,y, train_size=0.8, random_state=42, stratify=y)

print(X_train.shape), print(y_train.shape)
print(X_test.shape), print(y_test.shape)
```

Out[75]:
```
(35236, 54)
(35236,)
(8810, 54)
(8810,)
(None, None)
```

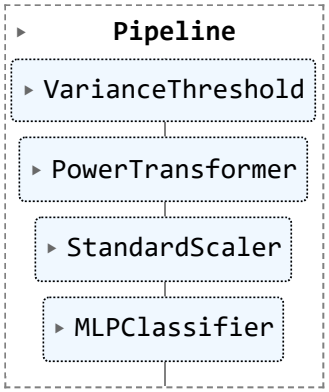
In [34]:
```
# Pipeline for Logreg
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import Normalizer
from sklearn.preprocessing import PowerTransformer
from sklearn.linear_model import LogisticRegression
from sklearn.feature_selection import VarianceThreshold
from sklearn.feature_selection import RFE

rfe = RFE(estimator=LogisticRegression(), n_features_to_select=40)
# Create the steps to be performed
steps = [
         ('yeo-johnson', PowerTransformer()), # default is yeo-johnson transformation
         ('scale', StandardScaler()),
         ('rfe', rfe),
         ('LR', LogisticRegression()) ]
```

```
# create pipeline object
pipe_log = Pipeline(steps)
```

In [56]:
```
# Pipeline for Neural net
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import Normalizer
from sklearn.preprocessing import PowerTransformer
from sklearn.linear_model import LogisticRegression
from sklearn.feature_selection import VarianceThreshold
from sklearn.neural_network import MLPClassifier

mlp = MLPClassifier(hidden_layer_sizes=(56), activation='relu', solver='adam', max_iter=500)

# Create the steps to be performed
steps = [('variance', VarianceThreshold(threshold=0.01)),
         ('yeo-johnson', PowerTransformer()), # default is yeo-johnson transformation
         ('scale', StandardScaler()),
         ('mlp', mlp) ]
pipe_nn = Pipeline(steps)
```

In [57]:
```
# fit the pipelines
pipe_log.fit(X_train, y_train)
pipe_nn.fit(X_train, y_train)
```

Out[57]:



In [58]:
```
# Accuracy scores
logreg_acc = pipe_log.score(X_train, y_train)
nn_acc = pipe_nn.score(X_train, y_train)

print(f'Accuracy score for Logistic regression: {logreg_acc}')
print(f'Accuracy score for Neural nets: {nn_acc}')
```

```
Accuracy score for Logistic regression: 0.6983766602338517
Accuracy score for Neural nets: 0.7757690997843115
```

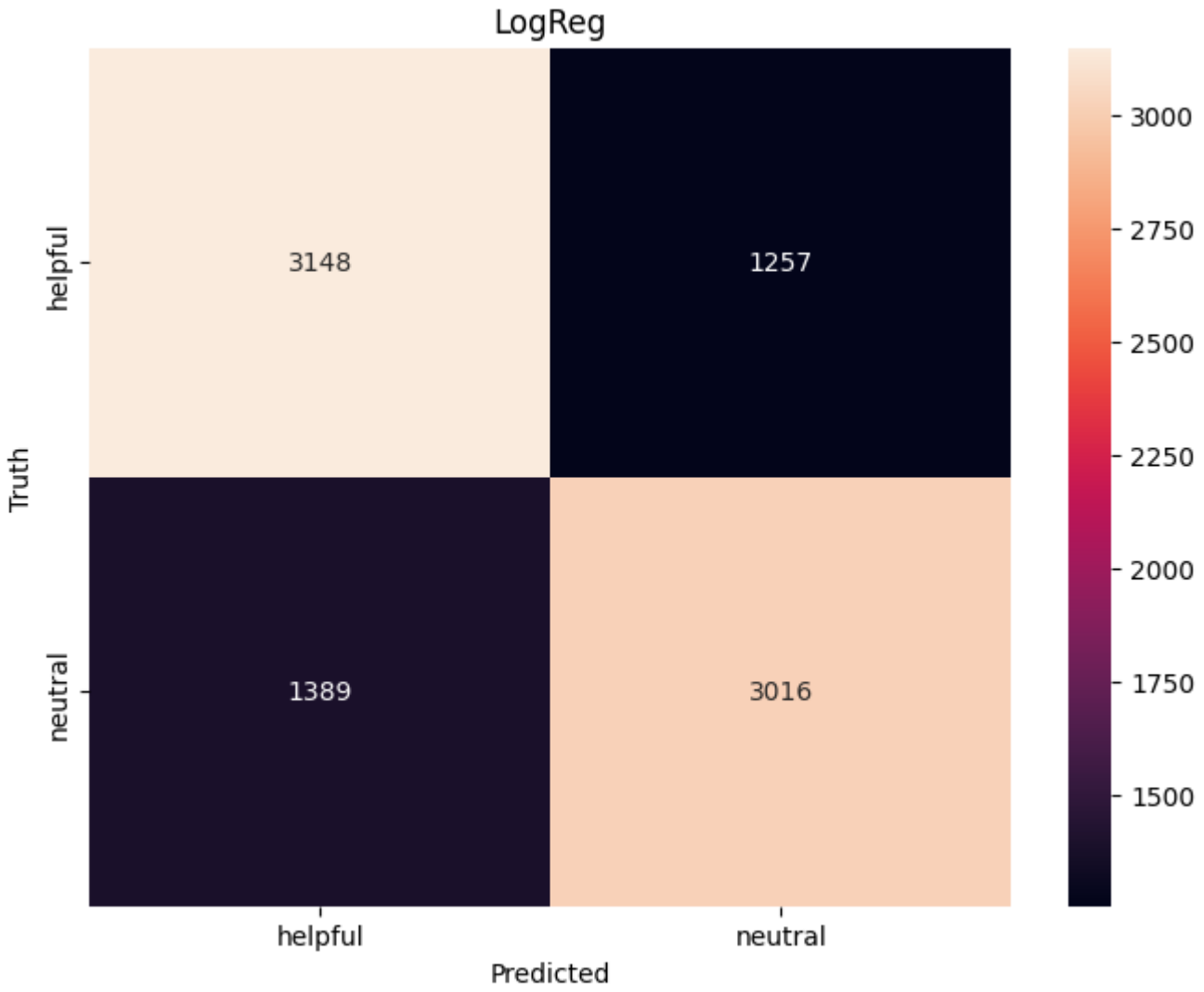
In [59]:
```
# kfold cross validation for both
from sklearn.model_selection import cross_validate

# Pass pipeline and training and test data set
results_logreg = cross_validate(pipe_log, X=X_train, y=y_train, cv=5, n_jobs=1)
results_nn = cross_validate(pipe_nn, X=X_train, y=y_train, cv=5, n_jobs=1)

print(f"Accuracy scores for logreg: {results_logreg['test_score']}")
print(f"Mean accuracy score for logreg: {results_logreg['test_score'].mean()*100}")

print(f"Accuracy scores for neural net: {results_nn['test_score']}")
print(f"Mean accuracy score for neural net: {results_nn['test_score'].mean()*100}")
```

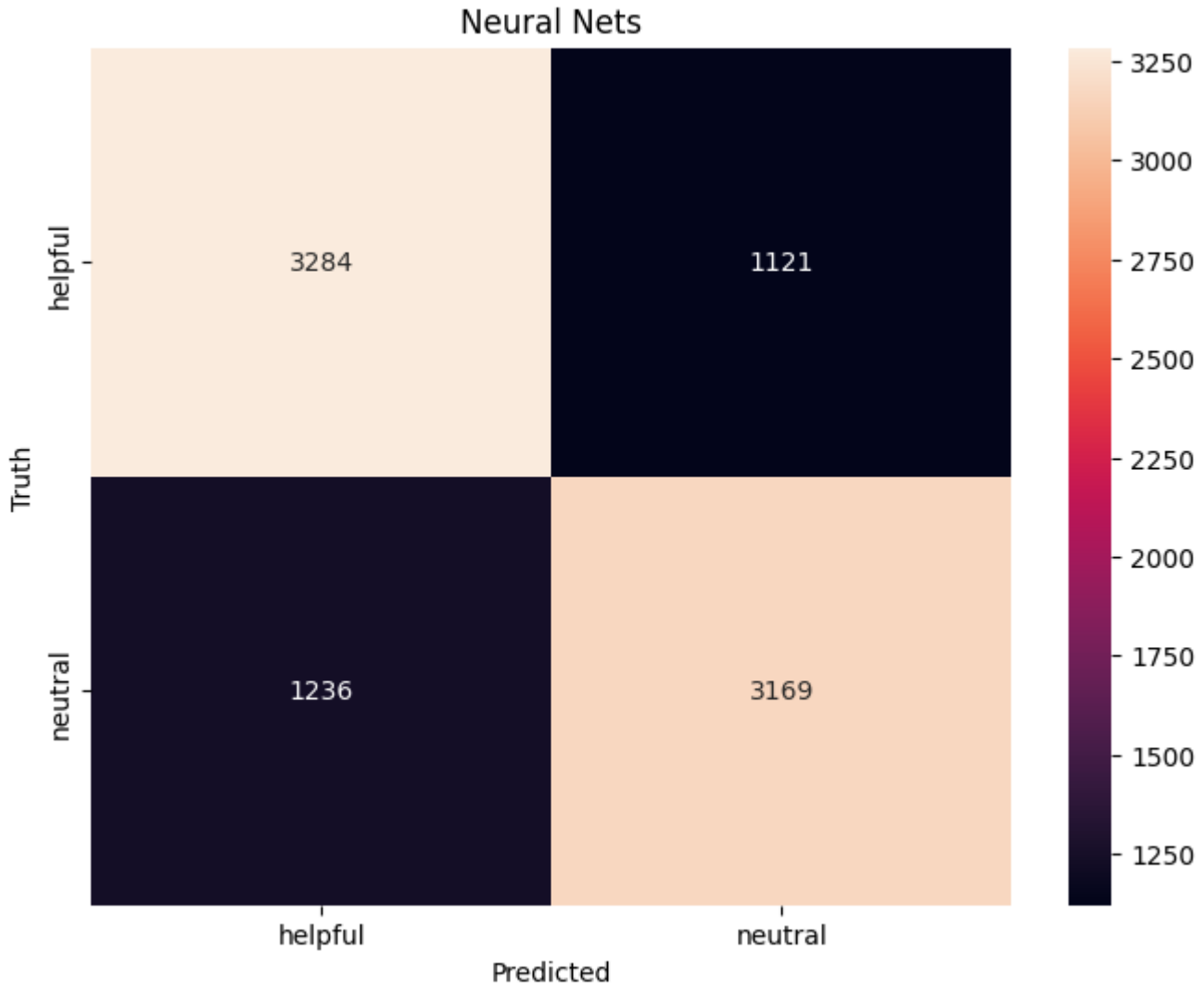
```
Accuracy scores for logreg: [0.68998297 0.7066837  0.68880375 0.70299418 0.69632468]
Mean accuracy score for logreg: 69.6957854889346
Accuracy scores for neural net: [0.73254824 0.74159217 0.73123315 0.73151696 0.73534838]
Mean accuracy score for neural net: 73.44477778277101
```

Evaluation

In [39]:
```
from sklearn.metrics import confusion_matrix
import seaborn as sn
import matplotlib.pyplot as plt

# Make predictions on the test data
y_pred_logreg = pipe_log.predict(X_test)
y_pred_nn = pipe_nn.predict(X_test)

# Calculate the confusion matrix
cm_log = confusion_matrix(y_test, y_pred_logreg)
cm_nn = confusion_matrix(y_test, y_pred_nn)


#plt.subplot(1, 2, 1)
plt.figure(figsize = (8,6))
sn.heatmap(cm_log, fmt='d', annot=True, xticklabels=pipe_log.classes_, yticklabels=pipe_log.classes_)
plt.xlabel('Predicted')
plt.ylabel('Truth')
plt.title('LogReg')

#plt.subplot(1, 2, 2)
```

```
plt.figure(figsize = (8,6))
sn.heatmap(cm_nn, fmt='d', annot=True, xticklabels=pipe_nn.classes_, yticklabels=pipe_nn.classes_)
plt.xlabel('Predicted')
plt.ylabel('Truth')
plt.title('Neural Nets')

plt.show()
```





In [40]:
```
#precision, recall
from sklearn.metrics import precision_score, recall_score
precision_log = precision_score(y_test, y_pred_logreg, pos_label='helpful')
precision_nn = precision_score(y_test, y_pred_nn, pos_label='helpful')

recall_log = recall_score(y_test, y_pred_logreg, pos_label='helpful')
recall_nn = recall_score(y_test, y_pred_nn, pos_label='helpful')

print(f' Precision score logreg: {precision_log}')
print(f' Recall score logreg: {recall_log}')
```

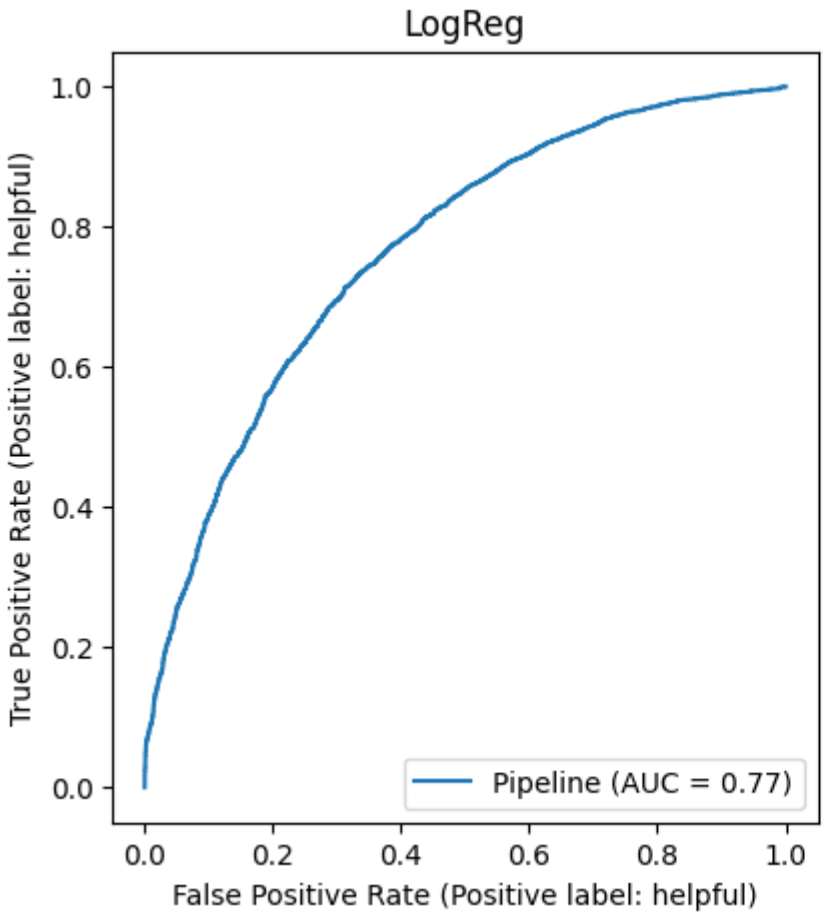
```
print(f' Precision score neural network: {precision_nn}')
print(f' Recall score neural network: {recall_nn}')
```

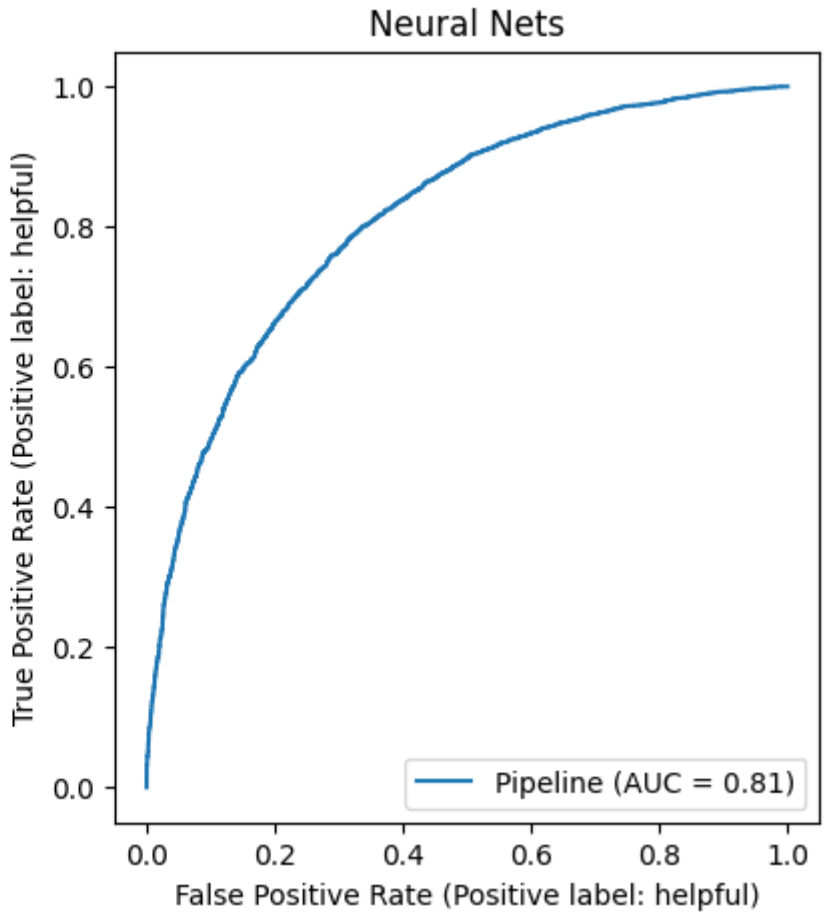
```
Precision score logreg: 0.6938505620454044
Recall score logreg: 0.7146424517593644
Precision score neural network: 0.7265486725663717
Recall score neural network: 0.7455164585698071
```

In [ ]:
```
#ROC>
from sklearn.metrics import RocCurveDisplay
import matplotlib.pyplot as plt

roc_log = RocCurveDisplay.from_estimator(pipe_log, X_test, y_test, pos_label='helpful')
plt.title('LogReg')

roc_nn = RocCurveDisplay.from_estimator(pipe_nn, X_test, y_test, pos_label='helpful')
plt.title('Neural Nets')
```

In [42]:
```
fig, axs = plt.subplots(1, 2, figsize=(10, 5))

roc_log.plot(ax=axs[0])
axs[0].set_title('LogReg')

roc_nn.plot(ax=axs[1])
axs[1].set_title('Neural Nets')
```

Out[42]:
```
Text(0.5, 1.0, 'Neural Nets')
```

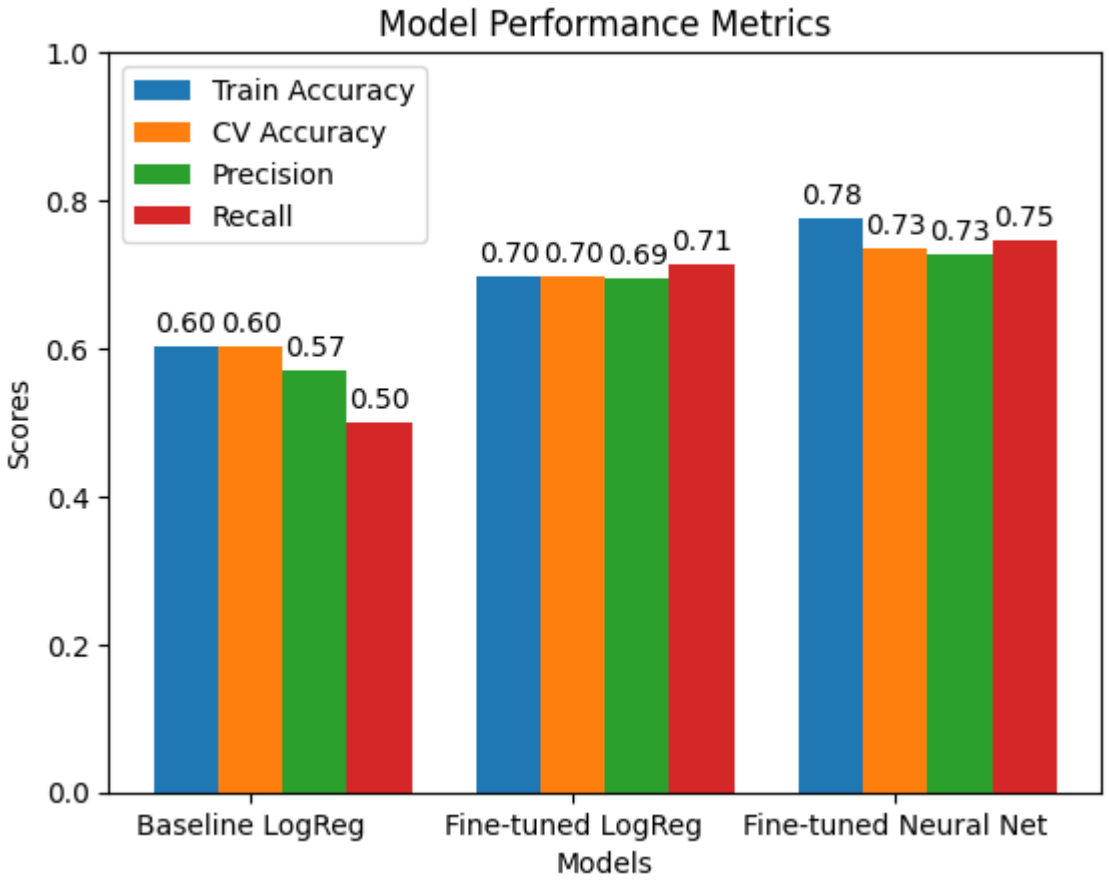




Plot all 3 models

In [88]:
```
import numpy as np
import matplotlib.pyplot as plt

models = ['Baseline LogReg', 'Fine-tuned LogReg', 'Fine-tuned Neural Net']

cv_base = base_logreg['test_score'].mean()
cv_logreg = results_logreg['test_score'].mean()
cv_nn = results_nn['test_score'].mean()

train_accuracy = [base_acc, logreg_acc, nn_acc]
cv_accuracy = [cv_base, cv_logreg, cv_nn]
precision = [precision_base, precision_log, precision_nn]
```

```
recall = [recall_base, recall_log, recall_nn]

bar_width = 0.2

index = np.arange(len(models))

# Plotting
fig, ax = plt.subplots()
train_acc = ax.bar(index, train_accuracy, bar_width, label='Train Accuracy')
cv_acc = ax.bar(index + bar_width, cv_accuracy, bar_width, label='CV Accuracy')
prec = ax.bar(index + 2 * bar_width, precision, bar_width, label='Precision')
rec = ax.bar(index + 3 * bar_width, recall, bar_width, label='Recall')

# Set the labels and title
ax.set_xlabel('Models')
ax.set_ylabel('Scores')
ax.set_title('Model Performance Metrics')
ax.set_xticks(index + bar_width)
ax.set_xticklabels(models)
ax.legend()

# Set the y-axis limits
ax.set_ylim([0, 1])

# Add data labels
def add_labels(rects):
    for rect in rects:
        height = rect.get_height()
        ax.annotate('{:.2f}'.format(height),
                    xy=(rect.get_x() + rect.get_width() / 2, height),
                    xytext=(0, 3),
                    textcoords="offset points",
                    ha='center', va='bottom')

add_labels(train_acc)
add_labels(cv_acc)
add_labels(prec)
add_labels(rec)

# Display the plot
plt.show()
```

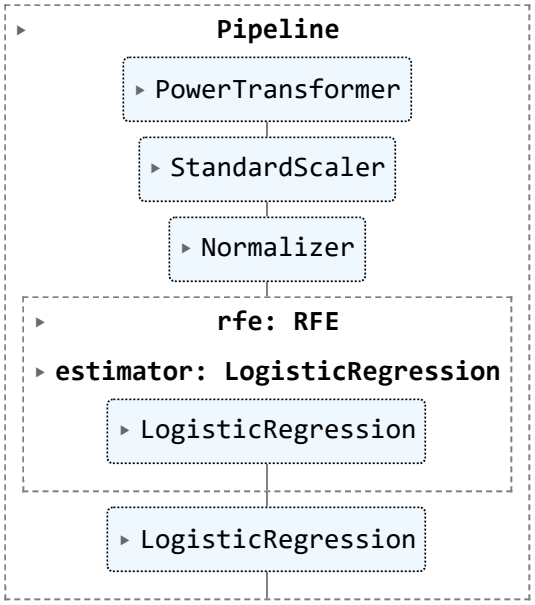


Feature ranking with RFE

In [24]:
```
# Pipeline
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import Normalizer
from sklearn.preprocessing import PowerTransformer
from sklearn.linear_model import LogisticRegression
from sklearn.feature_selection import RFE

rfe = RFE(estimator=LogisticRegression(), n_features_to_select=40)
# Create the steps to be performed
steps = [('yeo-johnson', PowerTransformer()), # default is yeo-johnson transformation
```

```
        ('scale', StandardScaler()),
        ('normalize', Normalizer()),
        ('rfe', rfe),
        ('LR', LogisticRegression()) ]

# create pipeline object
pipe = Pipeline(steps)
pipe.fit(X_train, y_train)
```

Out[24]:



In [31]:

```
# Print out the selected features
rf_df = pd.DataFrame(rfe.ranking_,index=X_train.columns,columns=['Rank']).sort_values(by='Rank',ascending=True)
rf_df.head(40)
```

Out[31]:

	Rank
nervousness	1
Topic 18	1
Topic 19	1
polarity	1
embarrassment	1
admiration	1
approval	1
optimism	1
caring	1
joy	1
relief	1
excitement	1
realization	1
amusement	1
surprise	1
disappointment	1
annoyance	1
grief	1
Topic 17	1
disapproval	1
Topic 16	1
Topic 14	1
disgust	1
curiosity	1
confusion	1

	Rank
Topic 1	1
love	1
Topic 3	1
Topic 4	1
Topic 15	1
Topic 6	1
Topic 7	1
Topic 8	1
Topic 9	1
Topic 10	1
Topic 11	1
Topic 12	1
Topic 13	1
fear	1
anger	1

In [32]:
```
# Feature selection with order of importance
feature_rankings = pd.DataFrame(rfe.ranking_, index=X_train.columns, columns=['Rank'])

sorted_rankings = feature_rankings.sort_values(by='Rank', ascending=True)
sorted_rankings['Importance Order'] = range(1, len(sorted_rankings) + 1)

print(sorted_rankings)
```

	Rank	Importance Order
nervousness	1	1
Topic 18	1	2
Topic 19	1	3
polarity	1	4
embarrassment	1	5
admiration	1	6
approval	1	7
optimism	1	8
caring	1	9
joy	1	10
relief	1	11
excitement	1	12
realization	1	13
amusement	1	14
surprise	1	15
disappointment	1	16
annoyance	1	17
grief	1	18
Topic 17	1	19
disapproval	1	20
Topic 16	1	21
Topic 14	1	22
disgust	1	23
curiosity	1	24
confusion	1	25
Topic 1	1	26
love	1	27
Topic 3	1	28
Topic 4	1	29
Topic 15	1	30
Topic 6	1	31
Topic 7	1	32
Topic 8	1	33
Topic 9	1	34
Topic 10	1	35
Topic 11	1	36
Topic 12	1	37
Topic 13	1	38
fear	1	39
anger	1	40
gratitude	2	41
Topic 5	3	42
desire	4	43
neutral	5	44

Topic 2	6	45
Month	7	46
sadness	8	47
remorse	9	48
txt_len	10	49
pride	11	50
review/score	12	51
Year	13	52
ratingsCount	14	53
Day	15	54

Appendix 7. – Source of dataset

https://www.kaggle.com/datasets/mohamedbakhet/amazon-books-reviews?select=Books_rating.csv