Using content- and behavioural data for recommendations in the Norwegian news market

Peter Kolbeinsen Klingenberg

Supervisor: Assoc. Prof. Dr. Mehdi Elahi
Co-supervisor: Thomas Husken, MSc

Master's Thesis
Department of Information Science and Media Studies
University of Bergen

June 1, 2023

Scientific environment

This study is carried out at the Department of Information Science and Media Studies at the University of Bergen. The work of this thesis is a part of Work Package 2 (WP2) - User Modeling, Personalisation & Engagement at MediaFutures. The work is in collaboration with the media platform Bergens Tidende, owned by Schibsted, one of the leading Scandinavian media houses.

Acknowledgements

First and foremost, I would like to thank my supervisor, Assoc. Prof. Dr. Mehdi Elahi, who has been of great importance to me and the result of this master's thesis. It has been very instructive and inspiring that he is so genuinely interested and engaged in both the subject - and my development. He also has a large network which he willingly offers to us, students. For me, it gave me the opportunity to collaborate through MediaFutures to research recommender system with one of the largest newspapers in the Norwegian media industry, i.e., Bergens Tidende, where I got Thomas Husken as co-supervisor. Thomas is the other person I would like to thank in particular. He has given me access to lots of data, taught me a lot about statistics and interesting ways of analyzing findings, and, not least, took the time to discuss and analyze my various findings with me.

I would like to thank everyone at MediaFutures and Bergens Tidende/Schibsted for their cooperation and the opportunity to conduct this research. Mehdi and Thomas - it has been an honour to be under your guidance.

This work was supported by industry partners and the Research Council of Norway with funding to MediaFutures: Research Centre for Responsible Media Technology and Innovation, through The Centers for Research-based Innovation scheme, project number 309339.

Peter Kolbeinsen Klingenberg
Oslo, June 2023

Abstract

In the news domain, the technological development has contributed to always accessible platforms with frequent publishment of articles. This leads to a vast amount of articles, in addition to a multitude of choices. Interactions executed by users (clicks, views or ratings) and content of articles (text, section, etc.) on such platforms can be utilized to create personalised recommendations. However, there can arise a situation where there is a lack of data, i.e., when a new article is published or there is a new user, known as the cold start problem. In addition, personalised recommendations in the news industry, where user behaviour are the basis of recommendations, are more complex in comparison with, for example, the movie domain due to the timeliness with short duration and relevance of articles. Recommendations provided by content data may mitigate this issue.

This thesis investigates the viability of using both content- and user-behavioural data to generate recommendations in the Norwegian media market in collaboration with one of Norway's largest newspapers, i.e., Bergens Tidende (BT). The first technique investigated is collaborative-based filtering, built on the recommender models Alternating Least Squares, Bayesian Personalized Ranking and Logistic Matrix Factorization, which is specially tailored to use user behavioural data as input. The second technique investigated is content-based filtering, built on a state-of-the-art architecture named BERT, specially trained to draw the semantic content of sentences in the Norwegian language. A comprehensive offline evaluation of both techniques is executed using dimension reduction and a diverse selection of accuracy metrics. In addition, an online evaluation of a large-scale A/B test of the content-based filtering technique against a former recommendation technique in BT is performed. The evaluation methods used in the online experiment are descriptive data- and Bayesian analysis.

The results of the collaborative filtering technique have shown increased performance in different stages of filtering, in addition to the importance of hyperparameter tuning. The results have shown promising performance based on the content-based filtering technique, indicating that this attracts users' interest at a greater scale, even the groups that usually show less interaction.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The news media industry, similar to other industries, has experienced enormous technological development in recent decades. Readers are presented with an always accessible online platform of vast amounts of news, with frequent publishing. This leads us to the challenge of Choice Overload, where the consumers have a huge catalogue of items and a lack of relevant choices. This has been worsened due to the rapidly growing Volume, Variety and Velocity of data (*Beheshti et al.*, 2022). In addition, the media market today has far more competitors than before. The battle for attention is a major challenge that technology has many opportunities to contribute in a convenient way. Standing out has always been important in the media market, as minor differences could mean customers prefer their competitors. The digital environment has a considerable scale of information and content. To use and navigate through this vast amount, some form of filtering would be necessary to become more structured and satisfying for the consumers.

The many possibilities of technology have also helped to make it easier to find out what each consumer is interested in, what content is perceived as useful and interesting, and not least opposite. Recommender systems are techniques and software tools designed to highlight and suggest items that may be useful for a user (e.g. a news article). The purpose is to simplify the decision-making process, such as what item to buy, watch, listen or, in our case, what online news article to read (*Aggarwal et al.*, 2016; *Elahi et al.*, 2021; *Ricci et al.*, 2010, p. 1). A popular recommendation approach, in this case, is Collaborative Filtering (CF), which uses user preference to generate recommendations. User preferences can be collected by explicit feedback (ratings, likes/dislikes) or implicit feedback (clicks, watch, listen) (*Elahi et al.*, 2018). In the news domain, articles are usually frequently read, and consumers are typically not asked for ratings. Therefore, implicit feedback is often used within this domain. This approach has less certainty rather than explicit data because implicit data does not explain how satisfied the consumer is with the item. On the other hand, this requires less interaction, which is largely positive from a consumer perspective. This data will be there anyway. While recommendations of a CF approach have presented promising performance, it could suffer from cold start problems due to a lack of data. The cold start problem concerns that a subset of data must be present before providing accurate recommendations, for instance, for a new item or a new user (*Elahi et al.*, 2018; *Fernández-Tobías et al.*, 2016).

Another approach for recommendations is Content-Based Filtering (CBF) (*Lops et al.*, 2019). It collects input data from the descriptive attributes of items to make recommendations (*Aggarwal et al.*, 2016, p. 14). In the news domain, this is content information, such as the article's text. This approach has a good ability to produce relevant recommendations. A major advantage is that CBF only depends on the single current user's interaction as long as it has access to a certain amount of articles. It, therefore, needs the content of a range of other articles and would fall short when lacking data and suffering from the cold start problem (*Bakhshandegan Moghaddam and Elahi*, 2019).

CBF is an effective approach for providing related articles as recommendations of content information of an article currently being read, reflecting a user's current interest. However, CBF may not work when providing recommendations on a front page of a newsroom where multiple articles are presented. This is due to the fact that the user's specific preference for an article is not stated, while CBF relies on a specific item's content. On the other hand, CF would be more effective on a front page, requiring that the user has a certain number of previous interactions, as this is the basis for recommendations.

As stated, CBF and CF use different input data sources and have different strengths and weaknesses. Using the algorithmic power by combining these two would make the performance of the recommendations more robust, referred to as a hybrid approach (*Aggarwal et al.*, 2016, p. 199).

This thesis involves a collaboration with Bergens Tidende (BT), a major news medium in Norway with an average of over 100 thousand daily digital readers. As humans, we have a natural interest in keeping updated on what's happening in our society and around the world, making the news industry significant and highly important. With numerous competitors in the market and rapid technological advancements, it is crucial to remain updated and capitalize on emerging opportunities. Providing a positive user experience is a top priority for the employees in BT, making personalised recommendations crucial for retaining consumers' attention and ensuring their satisfaction. BT places great emphasis on recycling articles, recognizing that previously published articles could retain relevance and value over time.

1.2 Problem Statement

The involvement of recommender systems has increased greatly, and made progress to encounter the ever-growing amount of data. Thus, there are outstanding issues that need to be resolved. In the news media industry, issues can vary from cold start challenge to timeliness with short duration, recency, popularity, and trends as articles are frequently published. The consumption time of a news article is often quite short, meaning the time a user spends to complete it. PEW research center[1] states that articles containing 250 words would, on average, engage a reader in less than a minute. While a longer article containing 5000 words would, on average, engage a reader in

[1] https://www.pewresearch.org/journalism/2016/05/05/2-analysis-suggests-some-readers-willingness-tc

less than 5 minutes. In comparison, the consumption time of a movie is quite longer, where the average length is between 90-120 minutes. The lifespan/lifetime of an article is another challenge, as they could expire by relevance quite soon, maybe hours or some days. Compared to a movie, the interest and relevance would, to a greater extent, be more long-term, even over several years. Another challenge is the highly dynamic user behaviours, as short-term or long-term preferences could evolve over time (*Raza and Ding*, 2022).

When reading an article in Bergens Tidende (BT), there are recommendation lists for further readings at the bottom of the page. The focus areas here are “Most Popular”, “Most debated”, “More articles” and “Read more about this theme”. The first two slots will not involve a personalised touch due to the fact that they do not concern you particular or necessarily deal with a similar topic which is being read now or earlier. This will still be valuable to recommend, as it has a confirmed engagement by a majority. “More articles” are based on similarities with manual tags added by the journalists. This approach is perceived as providing mixed results by some BT employees, as it can sometimes provide recommendations about an article concerning a completely different theme. An example was a restaurant review that was tagged with the tag “restaurant” and received a suggestion for an article about a fire at a restaurant a couple of months ago. Therefore, this form of using tags will be demanding because it depends on what the journalist chooses as a tag and will be very dependent on a common understanding and structure to perform well. Lastly, “Read more about this theme” pops up as a slot when the journalist has manually entered articles they believe have a similarity. This requires a costly human interaction and would be an expensive process over time. However, both this and “More articles” can be automated using a feature based on sentence-embeddings of articles text that can be automatically retrieved and could help solve that issue. Another important point is that this will contribute to the recycling of articles, where older articles can be just as relevant a while after publication. In addition to this content-based approach, using user events to predict equal patterns would be an even more personalised approach. Combining these two techniques (CBF and CF) could strengthen the ability to predict correctly and interesting items, as each approach has different strengths and weaknesses.

This thesis aims to investigate using descriptive features and user patterns to automatically calculate and predict items for a more personalised experience for a user. In addition to this, the recirculation of articles is a highly prioritised field in BT, as previous articles often could still be relevant. Previous articles are less reachable due to the fact that newly published articles are mostly presented on the front page. These recommendation techniques will contribute to recirculation and make previously published articles more visible.

1.3 Research Questions

RQ1: How to effectively model the reading behaviours of the users in Norwegian media platforms and the content of the news articles in the Norwegian language?

RQ2: Which technique can be used to generate personalised recommendations for the users incorporating these two types of data, i.e. the users’ behaviour and the content of news?

1.4 Contribution

The main contributions of this thesis are listed here:

- Performing a comprehensive analysis of industry data with different exploratory analyses;
- Developing recommendation techniques based on popular collaborative filtering and content-based approaches;
- Conducting an offline evaluation of proposed recommendation approaches and comparing them in terms of a wide range of evaluation metrics, including precision, recall, MAP, NDCG, and AUC;
- Developing a unique content-based recommendation technique for the news domain utilizing the BERT model specifically trained on Norwegian language data;
- Deploying the proposed technique on one of the largest newspapers in Norway, i.e., Bergens Tidende (BT), and evaluating it in a large-scale A/B test;
- The implementation of the proposed techniques and the analysis made on the data is provided in a dedicated MediaFutures GitHub repository[2], except some missing concepts of BT.

1.5 Thesis outline

The following items outline the thesis's general structure:

- **Chapter 2: Background.** Outlines related work and concepts related to this thesis. Section 2.1 provides background concepts related to news recommender systems. Section 2.2 presents related approaches of the interaction between computers and human language. Section 2.3 summarises previous work and how this differs from this thesis.
- **Chapter 3: Methods.** Describes the techniques and procedures used to address the research questions outlined in the thesis. Section 3.1 describes a state-of-the-art method for contextual vector representation of textual content. Section 3.2 details the process of the degree of similarity between two documents used in the online evaluation. Section 3.3 describes the datasets extracted from the content of articles and user behaviour. Section 3.4 details the recommendation algorithms used in the offline evaluation. Section 3.5 describes a technique to optimize the performance of the recommendation algorithm in the offline evaluation. Section 3.6 describes a method utilised to perform the online evaluation. Section 3.7 details the technical details used to perform the offline evaluation. Finally, the design and metrics considerations used to evaluate the offline and online tests, in

[2] https://github.com/sfimediafutures/MA_Peter-Kolbeinsen-Klingenberg

addition to an overview of the chosen descriptive- and statistical analysis methods are outlined in Section 3.8.

- **Chapter 4: Results.** Outlines the analysis conducted on the content- and behavioural datasets for different recommender models and the online A/B testing. Section 4.1, Experiment A: Exploratory Analysis, presents an exploratory analysis of sentence-BERT results by articles text from the content datasets. Section 4.2, Experiment B: Offline evaluation, presents the offline evaluation of various recommendation algorithms in different scenarios of filters and hyperparameters using the user behavioural dataset. Section 4.3, Experiment C: Online Evaluation, presents descriptive data- and a Bayesian statistical analysis of the online test performed on the digital news service of Bergens Tidende.
- **Chapter 5: Conclusion and Future Work.** Summarises the research, discusses the results of the experiments, addresses limitations, suggests future work, and includes the relevant code used in this thesis.

Chapter 2

Background

This chapter outlines related work and concepts related to this thesis. Section 2.1 provides background concepts related to news recommender systems. Section 2.2 presents related approaches of the interaction between computers and human language. Section 2.3 summarises previous work and how this differs from this thesis.

2.1 News Recommender Systems

The massive increase in the number of choices has given users the opportunity to choose the most interesting items. The encounter with the digital world has thus made information more accessible, but it will also lead to the challenge of choice overload. This is a situation with unlimited choices without an assumption about what might capture that specific user's interest (*Elahi*, 2014). This can be handled by a recommender system (RS) by reducing the information overload by estimating relevance based on personal interests and preferences. The use of such algorithms has increased in the past years in the media, especially in the news industry. They have a great potential to handle this information overload due to much information that can cause confusion from a user's perspective (*Shin*, 2020).

Personalised news recommendations can be created using several approaches, with Content-Based Filtering (CBF) and Collaborative Filtering (CF) being the most frequent (*Abdollahpouri et al.*, 2021; *Karimi et al.*, 2018; *Raza and Ding*, 2022). CBF is an item-centric approach and attempts to pair users with items liked in the past. From a news perspective at Bergens Tidende, we don't have any basis or rating of whether a user liked an article or not. The use of CBF in this context is based on which article they are reading right now. The similarity is, therefore, not necessarily based on that particular user's ratings but on the basis of attributes of the object/article, which in this case is the article's textual content (*Aggarwal et al.*, 2016, p. 139; *Beheshti et al.*, 2022). You may be familiar with this approach if you have seen a recommender block saying "Similar to this" followed by a list of recommended items.

Table 2.1: Explicit feedback events

userId	itemId	rating
1	1	1
1	4	5
1	6	2
2	2	5
2	5	4
3	1	5
3	2	3
3	4	1
4	3	3
4	6	4
5	4	3
5	5	5
6	1	5
6	3	4

Table 2.2: Implicit feedback events

userId	itemId	views
1	1	1
1	4	1
1	6	1
2	2	1
2	5	1
3	1	1
3	2	1
3	4	1
4	3	1
4	6	1
5	4	1
5	5	1
6	1	1
6	3	1

Table 2.3: Explicit feedback matrix

	I_1	I_2	I_3	I_4	I_5	I_6
U_1	1			5		2
U_2		5			4	
U_3	5	3		1		
U_4			3			4
U_5				3	5	
U_6	5		4			

Table 2.4: Implicit feedback matrix

	I_1	I_2	I_3	I_4	I_5	I_6
U_1	1			1		1
U_2		1			1	
U_3	1	1		1		
U_4			1			1
U_5				1	1	
U_6	1		1			

Opposite to CBF, CF is a user-centric approach. CF models utilize the collaborative nature of ratings performed by multiple users to provide recommendations. The concept is to estimate the missing ratings of a user based on the observed rating, which frequently exhibits strong correlations across users and items. For example, consider two users who have respectively similar preferences; an underlying algorithm can identify this. In these cases, it is a high probability that the other user would rate the item quite equally (*Aggarwal et al.*, 2016, p. 8). The storage format of this is typically represented as action events, which can be observed in Table 2.1 and Table 2.2. Subsequently, it is commonly transformed into a matrix format, where users are listed in the rows and items by the columns, as demonstrated in Table 2.3 and Table 2.4. These ratings can be provided in two formats: explicit and implicit. Explicit ratings are a form of rating where a user rates specific items, i.e. a numerical representation of their confidence from 1 (strong dislike) to 5 (strong like). The missing fields indicate unspecified preferences. This is illustrated in Table 2.3. There are 6 users and 6 items, but this figure is oversimplified and would, in a real-world scenario, have a larger number of users and items to have a greater basis for predictions (*Aggarwal et al.*, 2016, p. 11). On the other hand, we have implicit ratings, and user preference is a more indirectly reflected option through observing user behaviour. This can be purchase history, browsing history, search pattern or mouse events (*Hu et al.*, 2008). Similarly to explicit feedback, the missing fields indicate unspecified preferences, and the rating of 1's would, for example, represent a click or purchase. This is depicted in Table 2.4.

When considering the implicit feedback, Table 2.4 illustrates that user 1 and user 3 may have quite similar preferences, as they both have observed interactions with item 1 and item 4. Based on this, a recommender algorithm might give user 1 a recommendation of item 2, while user 3 could receive a recommendation of item 6. However, if we look at Table 2.3 of the explicit feedback, the evaluation of users 1 and 2 for items 1 and 4 is a contradiction to each other. This discrepancy indicates that explicit feedback reflects user satisfaction, while implicit feedback reflects observed user behaviour. Therefore, implicit feedbacks are easier to collect as they are the user's natural interaction with the service. On the other hand, explicit feedback requires the users to rate items interactively. Thus, the collection process would be more challenging.

Recommender systems face a wide range of challenges. The cold start problem is a well-known issue that arises when recommender systems encounter either a new user or new items with a dearth of preference data. CF rely on these historical data to identify patterns of user-item interaction and provide recommendations accordingly. When such data in the start phrase is missing for a new user or items, the performance of CF is greatly hindered (*Bakhshandegan Moghaddam and Elahi*, 2019). To mitigate this issue, it is common to combine CF with CBF as a hybrid approach, which leverages existing content features to provide recommendations.

News Recommender Systems (NRS) has to provide timely recommendations for a large number of users by processing a vast amount of data (i.e., news articles) and processing the data in a responsible way (*Elahi et al.*, 2022). These NRS systems must handle large amounts of data due to the fact that articles are frequently published. This requires fast, real-time processing and substantial computations. To effectively accomplish this, it is important that an NRS consider characteristics such as popularity, recency, freshness, trends, uniqueness, and low latency to stay relevant and timely by the recommendations (*Raza and Ding*, 2022).

The tables 2.1, 2.2, 2.3 and 2.4 is inspired by *Aggarwal et al.*, 2016, p. 12.

2.2 Embeddings in Natural Language Processing

The use of human language and text is naturally a large part of an information system, as natural language is the language humans use to communicate. Natural language processing (NLP) is a field in technology that involves the use of computers and their applications to achieve specific goals. The essence of NLP is to give the computer the ability to comprehend, interpret and utilize human language (*Chen et al.*, 2018; *Nwagwu*, 2022). Without this technique, the computer does not have the ability to have an underlying understanding of what a sequence of natural language concerns. A method for achieving this is the use of an approach named TF-IDF, which stands for Term Frequency-Inverse Document Frequency. This is a statistical measure to determine the importance of a term in a single or a group of documents. It creates a weight to each term based on how often it appears in a document, together with a balance of how often it occurs across the rest of the documents (*Bafna et al.*, 2016). A weakness is that the basis of the calculation of vectors is only on the documents that are fitted at that specific time and has no form of semantic understanding.

Another method in this field, and possibly better, is a state-of-the-art model named BERT, released in 2018 by Google, which stands for Bidirectional Encoder Representation for Transformers. This is designed to pre-train deep bidirectional representations from plain text by conditioning the context in both directions (left and right) in all layers. The pretraining process includes two unsupervised tasks, such as the Masked Language Model (MLM) and Next Sentence Prediction (NSP). MLM is the process of randomly masking some tokens while the goal is to predict the original tokens in their context. This is the task that enables the ability of BERT to learn in both directions, making it into a bidirectional Transformer. NSP is the task to pre-train the representation of text pairs. As a state-of-the-art model, BERT has been proven to have a major advance in many Natural Language Processing tasks as a context-based embedding model (*Devlin et al.*, 2018; *Koroteev*, 2021; *Ravichandiran*, 2021). A disadvantage of BERT is that it does not provide independent sentence embeddings, which leads to the difficulty of not having a contextual understanding of a whole sentence or sequence. In contrast, a solution that takes this into account is an approach named Sentence-BERT, which adapts BERT's architecture. The Sentence-BERT fine-tunes BERT by producing a siamese and triplet network to update the model's weights and ensure semantic meaningful sentence embeddings. In addition to this, the sentence embeddings would be generated as fixed-size vectors for a whole sentence or sequence of text, independent of the length and amount of words (*Reimers and Gurevych*, 2019).

2.3 Previous work and differences

The main difference between this thesis and related previous work is the execution of an online A/B test of a recommender's approach. Most of the previous work has focused on only offline evaluation. The few research works in this field that performed an online evaluation tend to include few users, typically consisting of less than a hundred users. This thesis has been conducted in collaboration with one of the largest Norwegian newspapers, i.e., Bergens Tidende (BT), owned by one of the leading media firms in Scandinavia, i.e., Schibsted. The A/B testing is an online experiment conducted by thousands of users daily. A second difference is that very few similar works are tailored to the Norwegian market and the Norwegian language, which is the aim of this thesis. Most are tailored to different languages, especially English. Another difference is that this thesis uses a state of art technique named BERT. Many, but especially the more traditional or early works, primarily focused on TF-IDF, binary representation and bag of words, which are simpler techniques. BERT creates dense vector representations with more contextual understanding. There exist papers using BERT for recommendations, but this research is based on the Norwegian language. Finally, another notable difference is that this thesis provides an analysis of Bayesian statistics of the A/B testing results in addition to a descriptive statistical evaluation.

In summary, the majority of related work focused on an Offline evaluation of BERT using languages such as English, and the few who did an Online evaluation of an A/B test tend to have few users included in the test. This thesis performed an online experiment of an A/B test tailored to the Norwegian market using the Norwegian language, tested on a larger scale of users.

Chapter 3

Methods

This chapter describes the techniques and procedures used to address the research questions outlined in the thesis. Section 3.1 describes a state-of-the-art method for contextual vector representation of textual content. Section 3.2 details the process of the degree of similarity between two documents used in the online evaluation. Section 3.3 describes the datasets extracted from the content of articles and user behaviour. Section 3.4 details the recommendation algorithms used in the offline evaluation. Section 3.5 describes a technique to optimize the performance of the recommendation algorithm in the offline evaluation. Section 3.6 describes a method utilised to perform the online evaluation. Section 3.7 details the technical details used to perform the offline evaluation. Finally, the design and metrics considerations used to evaluate the offline and online tests, in addition to an overview of the chosen descriptive- and statistical analysis methods are outlined in Section 3.8.

3.1 Sentence Embeddings

The sentence-Bert uses a siamese and triplet BERT network to produce sentence embeddings (*Reimers and Gurevych*, 2019). This makes it possible to generate fixed-size dimensions of vectors for a whole sentence/sequence of text. The Sentence Transformers[1] library in Python is utilized for implementing the Sentence-Bert approach, wherein a pre-trained model is fed as input (*Wolf et al.*, 2020). Many different pre-trained models are available, which are trained on specific problems. These are trained on a large number of texts to create a good basis for estimating vectors for a contextual understanding of a sentence or a sequence of text. This thesis is based on a pre-trained Sentence Transformers model called “NB-SBERT-BASE”[2].

Like all other Sentence Transformers pre-trained models, the basis of “NB-SBERT-BASE” is built on the general BERT Cased multilingual model, which is a MASK token prediction. Further, the digital collection from the National Library in Norway has trained this model on a large selection of Norwegian text in both Bokmål and Nynorsk over the past 200 years. Through this training process, we get the model called “NB-

[1] https://www.sbert.net/

[2] https://huggingface.co/NbAiLab/nb-sbert-base

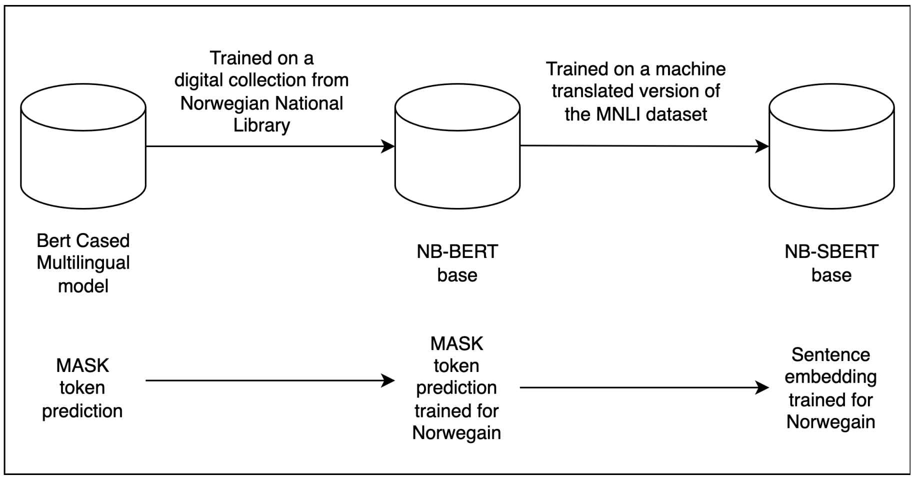

Trained on a digital collection from Norwegian National Library
Trained on a machine translated version of the MNLI dataset
Bert Cased Multilingual model
NB-BERT base
NB-SBERT base
MASK token prediction
MASK token prediction trained for Norwegain
Sentence embedding trained for Norwegain

Figure 3.1: How "NB-SBERT-BASE" is built

BERT-BASE", developed by the Norwegian National Library's AI Lab[3], which is also a MASK token prediction model of BERT. This is referenced to be the best-performing model for Norwegian and Scandinavian languages until now. The difference between this and the BERT Cased multilingual model is that this is fine-tuned for the Norwegian language. Furthermore, the "NB-BERT-BASE" is then trained on a machine-translated version (to Norwegian by Google Translate) of the MNLI dataset (*Williams et al.*, 2018). This finally gives us a model by BERT named "NB-SBERT-BASE" that provides us sentence embeddings as results. The model is trained and developed by *Braaten et al.*, 2022 in the Norwegian National Library's AI Lab. This model can be fed as input by the string "NbAiLab/nb-sbert-base" in the Sentence Transformers library in Python. The MNLI dataset consists of 433.000 sentence pairs annotated with textual entailment information and then provides a good basis for numerical representations of sentences, paragraphs and sequences of text (*Williams et al.*, 2018). The process of how "NB-SBERT-BASE" is built can be seen in the Figure 3.1.

The essence of "NB-SBERT-BASE" is that it provides a 768-dimensional dense vector space. Tasks such as semantic search and clustering can be accomplished by utilizing this vector. Calculating the cosine distance is a way to measure the similarity between two sentences/sequences (explained in Section 3.2). The similar meaning will have a small cosine distance and a similarity value close to 1. The model is designed to ensure that sentences with similar meanings in different languages will also be positioned close to each other. In an ideal scenario, an English-Norwegian sentence/sequence would have a high similarity score (*Braaten et al.*, 2022).

3.2 Similarity metric

In the process of estimating how similar two sequences of texts are, cosine distance is a widely used mathematical formula. This formula takes two inputs of text documents in the format of vectors of terms. The degree of similarity is executed by calculating the cosine distance between two document term vectors (*Rahutomo et al.*, 2012). As

[3]https://www.nb.no/sprakbanken/ressurskatalog/oai-nb-no-sbr-72/

shown in Figure 3.2, computing the cosine distance of the angle θ between documents A and B determines the degree of similarity. The output would be a value between 0 and 1, where 0 denotes the absence of similarity, while 1 signifies a high degree of similarity. *Manwar et al.*, 2012 defines the formula of cosine similarity between two documents, A and B, as:

$$Sim(A,B) = cos(\theta) = \frac{A \cdot B}{||A|| \cdot ||B||}$$

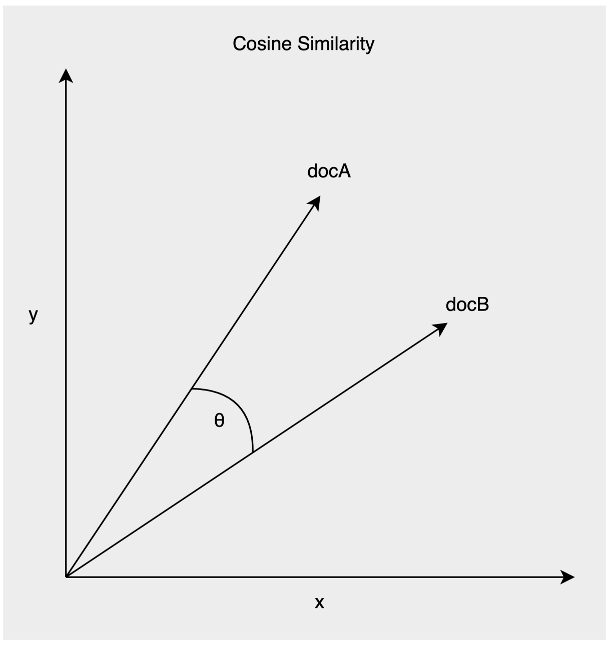


Figure 3.2: How Cosine Similarity works

3.3 Datasets

This thesis uses data provided by one of the largest Norwegian newspapers, i.e., Bergens Tidende (BT). The first dataset at hand contains descriptive features/attributes that describe the article's content, i.e., title, section, main text, date published, etc. The second dataset is the behavioural dataset which contains user events (clicks) of articles provided by anonymized logged-in users. A description of these datasets can be seen in Table 3.1 and Table 3.2. The time perspective spans over a duration of 56 days, or eight weeks, encompassing the time frame between the 12th of September and the 7th of November, 2022. These datasets are collected directly from BT's internal data warehouse.

Table 3.1: Dataset of article features

Dataset	Features	Observations
Article Features	59	10 134

Table 3.2: Dataset of user behaviour

Dataset	Interactions	Users	Items
User Behaviour	18 093 375	179 568	48 213

3.4 Recommendation Algorithms

The task of a recommendation algorithm is to predict the usefulness of an item for a user and recommend those that are considered useful. As detailed in Section 1.1 and Section 2.1, algorithms come in three categories, Content-based filtering (CBF), Collaborative filtering (CF) and, lastly, a hybrid approach, which combines CBF and CF (*Jalili et al.*, 2018). Helping users sort out what items they are looking for can play a huge role in their satisfaction. Examples of CF techniques specified for this are the algorithms named Alternating Least Squares (ALS), Bayesian Personalized Ranking (BPR) and Logistic Matrix Factorization (LMF). These algorithms would require a certain amount of past user behaviours/feedback to have the ability to predict possible further choices. These can come in many forms, such as clicks, page views, and media streaming counts (*Johnson*, 2014). User behaviour data of clicks provided by registered users on articles at Bergens Tidende is used in this thesis, i.e., implicit feedback.

All machine learning (ML) algorithms, in a CF approach used in this thesis, are implemented in the Python recommendation library Implicit[4], developed by Ben Frederickson.

ALS, BPR and LMF are all matrix factorization methods in CF that decompose a user-item matrix into two lower-dimensional metrics. These metrics represent user preference and item attributes (*Hu et al.*, 2008, *Rendle et al.*, 2012, *Johnson*, 2014). By optimizing the values of these metrics, ALS aims to minimize the difference between the predicted and observed values in the user-item matrix, which reflects the implicit feedback structure of user behaviour (*Hu et al.*, 2008). On the other hand, we have BPR, which functions by optimizing a pairwise ranking objective to learn user preferences and item attributes. The goal is to maximize the likelihood of whether a user prefers a positive over a negative item (*Rendle et al.*, 2012). LMF employs a probabilistic approach to model user preferences and item attributes. The probability of a user interacting with an item is performed using a logistic function, which is parameterized by the sum of the inner product of the user and item latent factor vectors, as well as the user-item biases (*Johnson*, 2014). The difference between these three algorithms is that BPR and LMF are specially designed for implicit feedback as input, whereas ALS also works for explicit feedback.

3.5 Hyperparameter Tuning

Machine learning (ML) algorithms are capable to handle different problems and datasets. The process of finding and building an effective ML algorithm is complex because they could contain different hyperparameters, and the most optimal selection

[4]https://benfred.github.io/implicit/

of these could be individually for the data or problem at hand (*Yang and Shami*, 2020). It was established back in the 1990s that different combinations of these configurations tend to work better for different datasets. Tuned hyperparameters improve the performance rather than using the default machine learning libraries provide. Hyperparameter tuning, generally explained, is an approach for tailoring the machine learning models to the problem at hand (*Feurer and Hutter*, 2019).

3.6 A/B testing

Internet connectivity has provided an unprecedented opportunity to try out and quickly receive results of the performance using a controlled experiment. This is called A/B testing, a method used to compare two approaches, A and B. Towards technological development, it is important for an industry to adapt and develop its services. When performing a change, a hypothesis is frequently formulated regarding the expected outcomes. Such experiments or A/B testing can be tested and evaluated on real users to understand whether the hypothesis was correct or not. In the 1990s, with the growth of the Internet, this type of method's attention increased. Big-tech industries, such as Facebook, Amazon, Google and LinkedIn, run thousands of controlled experiments yearly. Typical issues they investigate are changes in User Interface, enhancements of machine learning algorithms (search, advertisement, personalisation, recommendation, etc.), changes to apps, etc. (*Kohavi and Longbotham*, 2017).

3.7 Technical details

To conduct the offline evaluation, all experiments are executed using Python 3.9.0. The hardware used was 2GHz Quad-core Intel Core i5 CPU, 16 GB RAM and an Intel Iris Plus Graphics 1536 MB GPU.

3.8 Experiment design

This section will introduce the design and metrics considerations used to evaluate the recommender systems' performance. The evaluation process comprises two components, namely, the offline evaluation and the online evaluation. The offline evaluation includes collaborative filtering and content-based filtering, whereas the online evaluation solely relies on content-based filtering.

3.8.1 Offline evaluation

In the execution of offline evaluation, two datasets are used. Both timespan of the datasets is between the 12th of September and the 7th of November, 2022. The first dataset concerns article features and contains 59 features and 10134 observations. This means that there are 59 features describing all 10134 articles in the defined timespan. The second dataset is about user behaviour and reflects user clicks provided on articles in the time span. All clicks on articles outside the timespan would be filtered out.

EXPERIMENT A: Exploratory analysis

Visualisation of high-dimensional data is a well-known problem and deals with a wide variety of dimensions. The representation of the dimensions of images, word counts and sentence embeddings are examples where their documents typically have hundreds to thousands of dimensions. t-SNE is a widely used technique to map these high-dimensional data into low-dimensional space, such as two or three-dimensions, while preserving their pairwise similarities. This means that dimensionality reduction aims to maintain the significant structure of the high-dimensional data as possible when creating the lower-dimensional data representation. The results of these lower-dimensional data are then visualised using scatterplots or other graphical techniques (*Van der Maaten and Hinton*, 2008). *Van der Maaten and Hinton*, 2008 states that the visualisation of t-SNE is significantly better than other dimension reduction techniques, such as Sammon mapping, Isomap, etc.

In this experiment, several techniques are used to visualize how semantically equal a set of articles is to each other. To semantically represent the article's text, a method called Sentence-BERT ("NB-SBERT-BASE") is used to create a 768-dimensional numerical vector space of sentence embeddings (*Braaten et al.*, 2022). For plot visualization, t-SNE was used to reduce these 768 dimensions to x and y coordinates. Each article belongs to one of the 24 sections available at Bergens Tidende, which is used to categorize them from each other in the plot. Lastly, Matplotlib was used to create the visualization of semantic equality by the articles.

EXPERIMENT B: Accuracy metrics

For evaluation of the performance of the recommendation models ALS, BPR and LMF, the accuracy metrics used to predict the accuracy of the recommendations are Precision@K, Recall@K, MAP@K, NDCG@K, ROC_AUC and PR_AUC. The Recometrics[5] library in Python was used to split train/test and predict the accuracy of the models to execute this evaluation.

Precision@K is referred to as Precision at top K recommendations (P@K). This is a metric for evaluating the effectiveness of a recommendation system in suggesting relevant items. To calculate P@K, the system identifies the top K recommended items per user, and only those with ratings that also appear in the test set T are considered. To compute P_u@K for each user u, the following formula is used (*Schedl et al.*, 2018):

$$P_u@K = \frac{|L_u \cap \hat{L}_u|}{|\hat{L}_u|}$$

The set L_u represents the relevant items for user u in the test set T, where $\hat{L}_u$ denotes the set of recommended items consisting of the K highest ratings predicted for user u in T. To calculate the overall P@K, the P_u@K values for all users in the test set are averaged.

[5] https://recometrics.readthedocs.io/

Recall@K is referred to as Recall at top K recommendations (R@K). This is a metric to show completeness and shows the fraction of relevant items recommended at K positions. To compute R$_u$@K for each user u, the following formula is used:

$$R_u@K = \frac{|L_u \cap \hat{L}_u|}{|L_u|}$$

The set L$_u$ represents the relevant items for user u in the test set T, where $\hat{L}_u$ denotes the set of recommended items consisting of the K items with the highest predicted ratings for the user u in T. To calculate the overall R@K, the R$_u$@K values for all users in the test set are averaged (*Schedl et al.*, 2018).

Mean Average Precision at top K recommendations (MAP@K) is a metric based on ranking that measures the precision of a system at varying lengths of recommendation lists. To measure MAP@K, it calculates the average precision of the system across all users in the test set and takes the arithmetic mean of these values. The formula for Average Precision for the top K recommendations (AP@K) is:

$$AP@K = \frac{1}{N} \sum_{i=1}^{K} P@i \cdot rel(i)$$

In the given equation, rel(i) indicates whether the i$^{\text{th}}$ recommended item is relevant or not, where rel(i) is 1 for relevant items and 0 for irrelevant ones. N represents the total number of relevant items (*Schedl et al.*, 2018).

Normalized Discounted Cumulative Gain (NDCG) is a metric used for ranking the performance of recommendations. If the predicted rating values for user u are arranged in decreasing order, the formula of the metric DCG$_u$@K is:

$$DCG_u@K = \sum_{i=1}^{K} \frac{r_{u,i}}{log_2(i+1)}$$

In the given equation, $\mathrm{r_{u,i}}$ indicates the actual rating of the item at rank i for user u, as found in the test set T. K indicates the length of the recommendation list. As the rating distribution is dependent on the user's behaviour, the DCG values are not directly comparable between different users. To address this, the cumulative gain for each user should be normalized. This is achieved by calculating the ideal DCG value, denoted as IDCG$_u$, which represents the best achievable DCG value for user u. The ideal DCG is achieved by sorting the items in descending order based on their true ratings (*Schedl et al.*, 2018). The formula for the Normalized Discounted Cumulative Gain for user u is:

$$NDCG_u@K = \frac{DCG_u@K}{IDCG_u@K}$$

To calculate the overall Normalized Discounted Cumulative Gain at K (NDCG@K), the NDCG$_u$@K values for all users in the set are averaged (*Schedl et al.*, 2018).

ROC-AUC is referred to as Area Under the ROC Curve, where ROC stands for Receiver Operator Characteristics. This is a metric that aims to quantify a recommender system's ability to effectively differentiate the relevant objects (items a user would appreciate) and irrelevant objects (all other items) (*Lü et al.*, 2012). The method to compute ROC-AUC involves comparing the probability of recommending relevant objects to that of irrelevant objects. If we conduct n independent comparisons, where each comparison involves selecting one relevant and one irrelevant object, and we observe n' instances where the relevant object has a higher score than the irrelevant one, and n'' instances where the scores are equal, ROC-AUC is defined as *Zhou et al.*, 2009:

$$ROC\_AUC = \frac{n' + 0.5n''}{n}$$

A ROC-AUC score of 1 indicates a perfect recommendation list, while 0.5 indicates a random ranking. A score above 0.5 measures a recommendation algorithm's ability to identify relevant objects (*Lü et al.*, 2012).

PR_AUC is referred to as Area Under the Precision-Recall Curve. In order to calculate this, we need the Precision and Recall for the whole set. Precision and Recall is defined as *Manning*, 2009, p. 155:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

In the given equations, TP stands for True Positives, which is the retrieved recommendations that were relevant. FP stands for False Positives, which is the retrieved recommendations that were not relevant. Lastly, FN stands for False Negatives, which is the not retrieved recommendations that were relevant (*Manning*, 2009, p. 155). Using the Precision and Recall with the trapezoidal rule, the formula for PR_AUC is as follows:

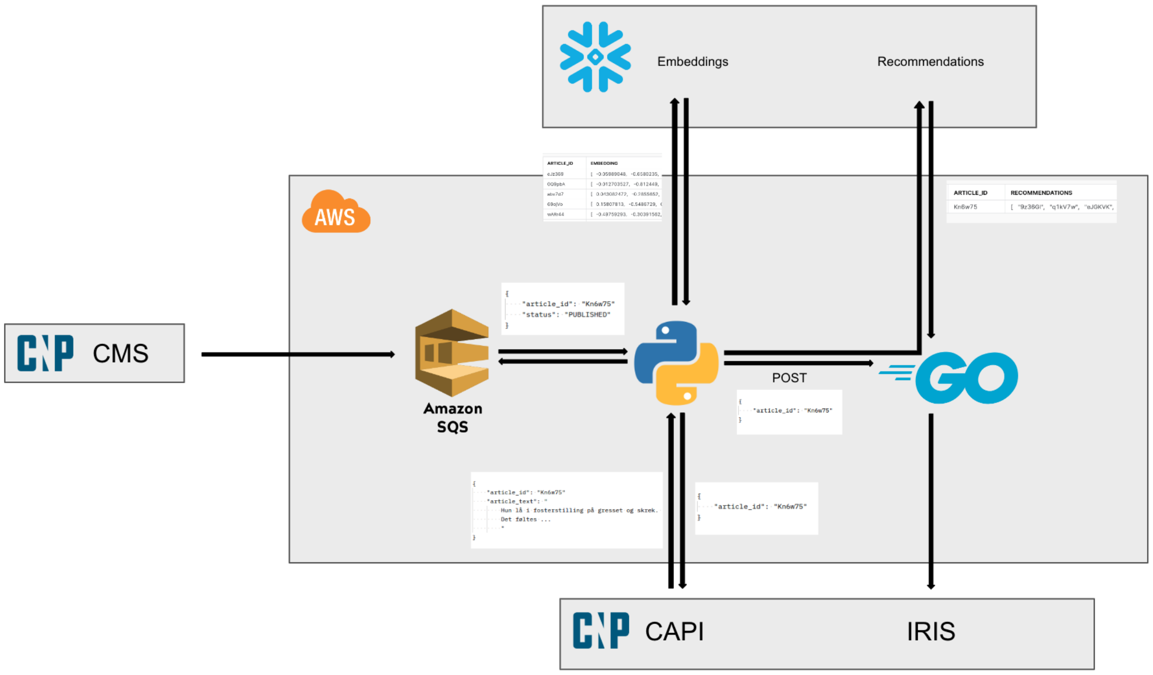
$$PR\_AUC = \int_a^b f(x)\,dx$$

3.8.2 Online evaluation

Implementation

For the online evaluation, the feature extraction process is deployed on Bergens Tidende's internal servers and automatically generated recommendations for all new articles based on cosine similarity. A simple figure of the process is illustrated in Figure 3.3. The process starts when an Amazon SQS queue triggers an event from CMS (Content Management System) when an article's status is either published or updated. An AWS (Amazon Web Services) program that runs Python extracts that particular article's text from CAPI, embeds the text and stores this embedding on SnowFlake.

Subsequently, it loops through all candidate articles' sentence embeddings, computes the cosine similarity, extracts the top six most semantically similar article IDs, and stores this on Snowflake. All articles that are published in the last six months and high lifetime articles from the last 14 months are considered candidate articles. Article lifetime is a value (low/medium/high) journalists pick to indicate how long an article stays relevant. If an article concerning a weather report or a traffic accident is considered, these are typically articles with a lower lifetime. On the other hand, a restaurant review would, to a greater extent, have a higher probability of being relevant longer, thus having a higher lifetime. Lastly, as a fast computational programming language, GO is used to fetch these recommendations and expose them to the frontend rendering service (IRIS), making it visible on the Bergens Tidende web page. Recommendation lists were generated and presented to users as titled list/slot saying "Les videre". This is located at the bottom of the page when reading an article, related to the top 6 most similar articles to this particular article.



ARTICLE_ID	EMBEDDING
eJz369	[-0.05989048, -0.6580235,
OO9pbA	[-0.012703527, -0.812449,
abv7d7	[0.043082472, -0.2855852,
69ojVo	[0.15807813, -0.5486729,
wARr44	[-0.49759293, -0.30391562,

ARTICLE_ID	RECOMMENDATIONS
Kn6w75	["9z36GI", "q1kV7w", "eJGKVK",

```
{
    "article_id": "Kn6w75"
    "status": "PUBLISHED"
}
```

```
{
    "article_id": "Kn6w75"
}
```

```
{
    "article_id": "Kn6w75"
    "article_text": "
        Hun lå i fosterstilling på gresset og skrek.
        Det føltes ...
        "
}
```

```
{
    "article_id": "Kn6w75"
}
```

Figure 3.3: The architecture of the recommendations in Bergens Tidende

Experiment design and evaluation

An online experiment was conducted on Bergens Tidende between the 15th and the 29th of March, where the methodology of the A/B test was used, with a 50/50 split. The method described above was tested against an existing, more generic widget based on tags. The recommendation lists/slots are shown at the bottom when reading an article. The interactions of user clicks are the measure of evaluation. To evaluate the results, a descriptive data analysis is applied. In addition to this, a statistical approach of Bayesian statistics was used, and here we briefly explain it. Statistics with a Bayesian approach is the principle of probability statements and updating the probabilities after collecting new data. Bayes theorem as a model has the ability to describe such updating using this formula (*Gelman et al.*, 2013; *Kamalbasha and Eugster*, 2021):

$$P(\lambda|X) = \frac{P(\lambda)P(X|\lambda)}{P(X)}$$

Bayes theorem explains how the posterior distribution P(λ|X) of the parameter λ, after observing the data of X. P(λ) of the parameter λ is the prior distribution, while P(X|λ) is the likelihood of the data X given the parameter λ, and the marginal likelihood P(X) of the data X.

The binomial distribution is a discrete distribution that describes the probability fraction of successes and trials. In terms of a test on a website on clicks, success is the number of actual clicks, while trials are the number of views. The results of the success probability of p $\in$ [0,1] in each trial. The continuous Beta prior distribution Beta(a,b) is defined in the interval [0,1]. A and b as parameters represent the shape of the distribution, while a shape of (1,1) defines a standard uniform distribution. The conjugate prior for a binomial distribution is received by the beta distribution (*Gelman et al.*, 2013; *Kamalbasha and Eugster*, 2021).

The advantage of using a Bayesian approach is that with a higher sample size of users, as this A/B test has in terms of clicks, a traditional statistic would probably have too much statistical power. There is a great chance that every value would be traditionally significant. Using a 94% credible interval makes it more uniform and understandable for non-technicians as well. An exact p-value is not stated in Bayesian statistics, but rather a 94% credible interval where the probability would be between and an assumption of the mean value.

Chapter 4

Results

This chapter outlines the analysis conducted on the content- and behavioural datasets for different recommender models and the online A/B testing. Section 4.1, Experiment A: Exploratory Analysis, presents an exploratory analysis of sentence-BERT results by articles text from the content datasets. Section 4.2, Experiment B: Offline evaluation, presents the offline evaluation of various recommendation algorithms in different scenarios of filters and hyperparameters using the user behavioural dataset. Section 4.3, Experiment C: Online Evaluation, presents descriptive data- and a Bayesian statistical analysis of the online test performed on the digital news service of Bergens Tidende.

4.1 Experiment A: Exploratory analysis

The first set of experiments involves an exploratory analysis of the dataset containing article features in the time span between the 12th of September and the 7th of November, 2022. This is done to understand better how semantically equal the embeddings of the article's textual content "NB-SBERT-BASE" classifies them to be. The dataset contains 10134 observations (articles) and 59 features describing each article.

The techniques sentence-BERT, t-SNE dimension reduction, and Matplotlib were adopted to accomplish this data visualisation. Each article belongs to one of the 24 available sections. These sections are used to create clusters to differentiate the articles in the plot from each other.

In this section, the result of t-SNE is explained. Figure 4.1 illustrates the results of a dimension reduction of all articles. As it can be seen, there are clearly defined clusters, especially in the clusters of Bedriftsroboten, Hyttesalg and Boligsalg. This means that they are located close to each other in groups. This indicates that the model "NB-SBERT-BASE" can identify semantic similarity because these are written by a robot on a defined structure, which means that they have a fairly similar textual structure. These are also sections in which articles are frequently published, which can be seen by the number of plots of these sections in this given time span. Another section that is often located close to the robot-written articles is the Økonomi section. Bedriftsroboten is a section where the robot, with the help of APIs, automatically retrieves data about annual results, tax figures, etc. Boligsalg and hyttesalg are also sections where data about

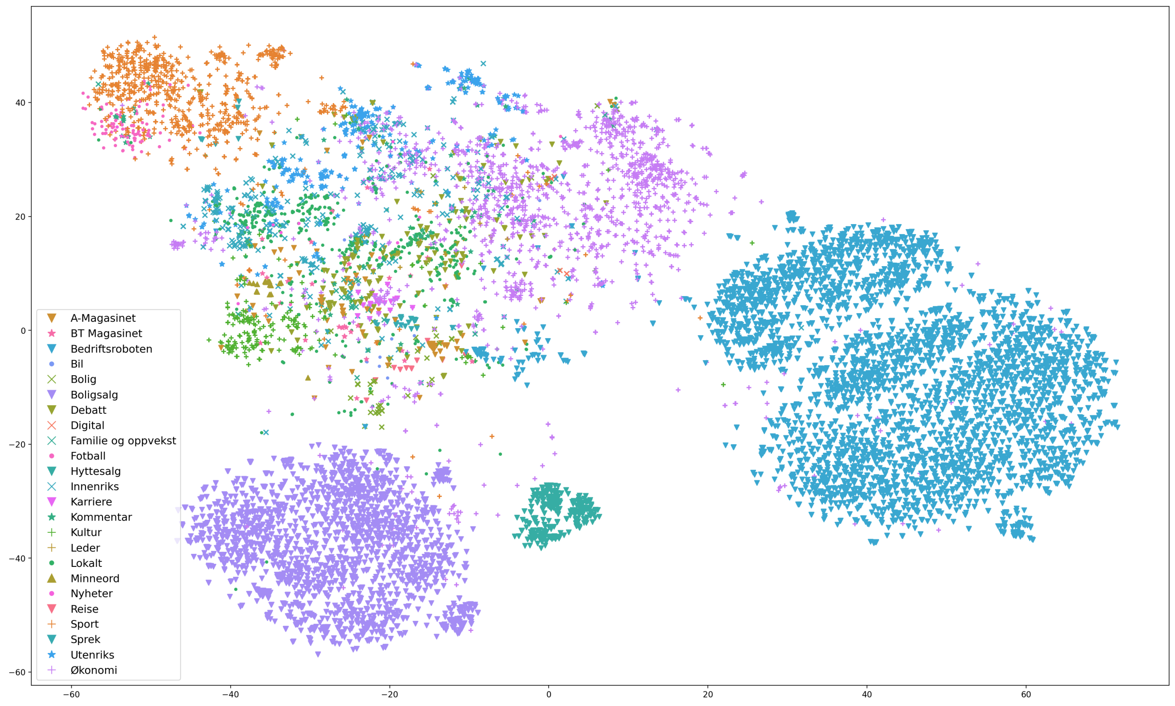


Figure 4.1: tSNE dimension reduction of all articles

property sales are automatically retrieved. Økonomi is a section written by human journalists where they don't have any particular template to follow. This strengthens the credibility of NB-SBERT-BASE when we see that the Økonomi section is often close to these.

In two cases, we see that the Sports section is located near Boligsalg. These are located at approximately x and y positions (-15, -30) and (-18,-23), identified as an orange plus sign in Figure 4.1. This may immediately seem a little strange that these two sections have a similarity without knowing the content of the articles. In the case of a manual check, these are articles about the fact that the Norwegian sports profiles Erling Braut Haaland and Johannes Høsflot Klæbo have each bought a property. This further strengthens the credibility/reliability of the "NB-SBERT-BASE" model, where we can see similarities in content across different sections.

In the following experiment, robot-written articles in the sections Bedriftsroboten, Hyttesalg, Boligsalg and Bolig are removed. As previously mentioned, these are sections where articles are frequently published. By removing these, only 4282 articles are left, which means that around 6000 of all articles in the dataset were robot-written articles. In this case, the result of the t-SNE visualisation can be seen in Figure 4.2. It turns out that the trend for the sections/clusters of articles is, to some extent, more spread out. All of these articles are written by humans, which means that the variation of textual structure will to a greater extent, differ. This will affect that the articles may have a slightly greater distance from each other but are usually just as relevant and have similarities. Articles located closest to each other are the ones we are most interested in, representing similarity in content. We can see that the majority of the articles in the Økonomi section are located at the top right. It also stretches a bit scattered around among the other articles, which makes sense in the form that this section or topic of-

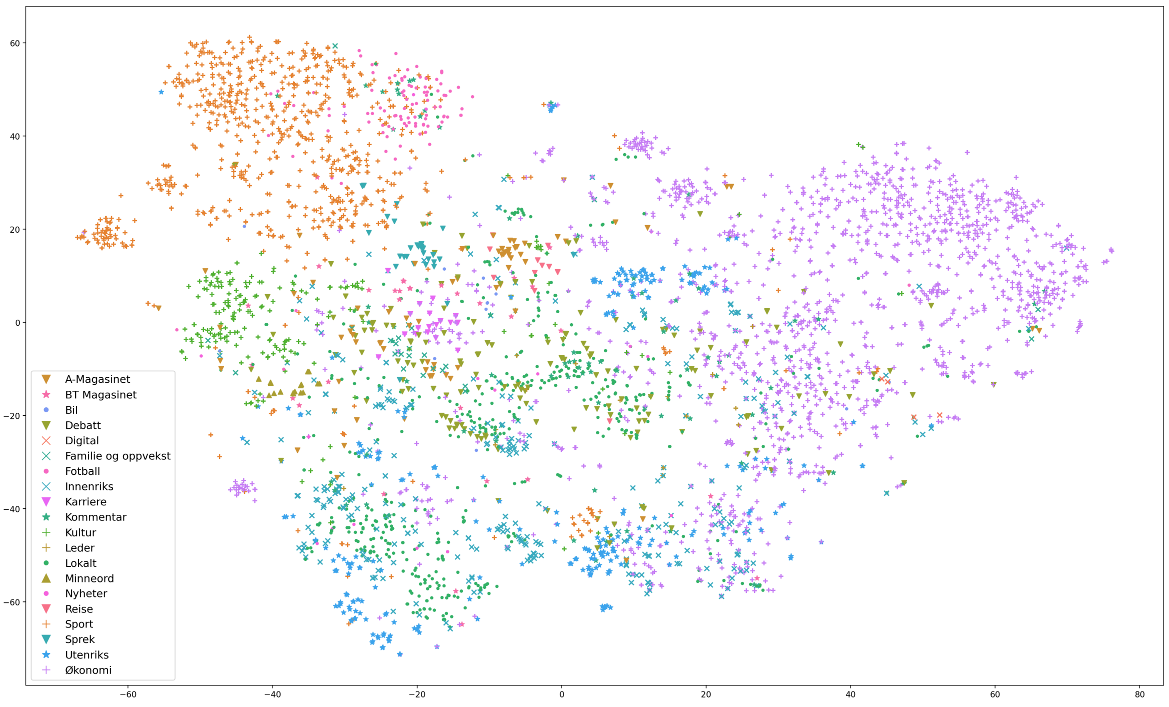


Figure 4.2: tSNE dimension reduction of all non-robot written articles

ten has a similarity or connection with other sections and topics. Furthermore, we see a trend that similar sections tend to be located close to each other.

After numerous manual tests, the “NB-SBERT-BASE” model seems to draw semantic similarities quite well. Furthermore, in Section 4.3: Online Evaluation, an A/B test of the techniques by BERT against a former, more generic model based on tags on real users in Bergens Tidende is conducted.

4.2 Experiment B: Offline evaluation

The purpose of this experiment is to train and evaluate the performance of the machine learning (ML) models - Alternating Least Squares (ALS), Bayesian Personalized Ranking (BPR) and Logistic Matrix Factorization (LMF) - using the dataset of user behaviour from Bergens Tidende (BT). When training models, it is important to find the balance between insufficient and excessive amounts of data that enhances the ML models’ ability to provide correct predictions. Different filtering scenarios are considered for the dataset, as well as a hyperparameter tuning for the models, while the performance of the ML models is compared against a random and most popular recommender. The random recommender chooses random K items to recommend, while the most popular recommend K most popular items.

The dataset consists of the interaction of click events, where each row contains information about a user ID, an article ID, the number of views (which indicates how many times the user clicked on that particular article), as well as the date and time of the first click. The number of views greater than one is reduced to one, as we are only interested in whether a user clicked on an article or not. This type of data is referred to as implicit feedback, as it’s an observed user behaviour, namely clicks, unlike explicit feedback, which indicates a stated user satisfaction through a numerical value (*Hu et al.*, 2008; *Aggarwal et al.*, 2016, p. 11). An example of implicit feedback can be seen in Table 2.2 (as a dataset of events), where the only difference is that we have an additional column of datetime. As these experiments aim to use ML algorithms like ALS, BPR and LMF to predict further user choices based on past click behaviour, the dataset only includes data from registered users to link clicks to each user. The chosen train-test split for all models in the experiments is 80% for training and 20% for testing.

The dataset contains a total of 18 093 375 clicks made by 179 568 users on 48 213 articles within an eight-week period ranging from the 12th of September to the 7th of November, 2022. The choice of this timeframe is by the fact that recently executed clicks will often have higher relevance than clicks executed more than eight weeks ago. This is because users' interests can change over time, and by considering clicks carried out in the past eight weeks, we get a better insight into their current interests.

4.2.1 Experiment B1: General Filters

Totally the dataset consists of articles belonging to 25 unique sections. All articles in the section "Direkte" in BT refer to the same page, and we've determined that articles in this section will not be relevant in a recommendation process and are excluded from the dataset. Additionally, the dataset might include clicks on articles that were published several years back in time. Still, we are only interested in articles published within the same eight-week period of time. Consequently, we exclude all articles outside this period. With these filters in place, we are left with 16 657 690 clicks made by 161 092 users on 7 769 articles.

Table 4.1: Dataset for experiment B1

Dataset	Interactions	Users	Items
User Behaviour	16 657 690	161 092	7 769

In the first sub-part of this experiment, we use the dataset of click events seen in Table 4.1. To measure the performance, a comparison of ALS, BPR and LMF against a random and most popular recommender is executed through evaluation metrics like P@K, R@K, MAP@K, NDCG@K, ROC_AUC and PR_AUC. The chosen K value is 5 due to the fact that the recommenders lists in BT are typically containing 5 or 6 items. A recommenders list in BT can be seen in Figure 4.4. In this case, no hyperparameters for the algorithms are specified, so the default for each is chosen. For ALS, factors, iterations and regularizations are displayed. In addition to these hyperparameters, learning_rate is displayed in BPR and LMF as well. The default hyperparameters per algorithm can be seen in Table 4.2.

Table 4.2: Default hyperparameters used to train ALS, BPR and LMF in Experiment B1

Algorithms	**Hyperparameters**			
	factors	iterations	regularization	learning_rate
ALS	100	15	0.01	N/A
BPR	100	100	0.01	0.01
LMF	30	30	0.6	1.0

Table 4.3: Evaluations of Experiment B1

Algorithms	**Evaluation metrics**					
	P@5	R@5	MAP@5	NDCG@5	ROC_AUC	PR_AUC
Random	0.0046	0.0005	0.0021	0.0046	0.5006	0.0061
Most Popular	**0.1679**	**0.0295**	**0.1276**	**0.1829**	**0.9326**	**0.0977**
ALS	0.0868	0.0135	0.0566	0.0927	0.6168	0.0416
BPR	0.0721	0.0139	0.0470	0.0773	0.6392	0.0300
LMF	0.0271	0.0080	0.0152	0.0289	0.8685	0.0298

As seen in Table 4.3, Most Popular performs best in terms of P@5 with a score of 0.1679, followed by ALS, BPR, LMF and Random with scores of 0.0868, 0.0721, 0.0271 and 0.0046. In terms of R@5, Most Popular achieved the best score of 0.0295, followed by BPR, ALS, LMF and Random with scores of 0.0139, 0.0135, 0.0080 and 0.0005. According to the measure of MAP@5, again, Most Popular is the recommender that achieved the best score at 0.1276, followed by ALS, BPR, LMF and Random with scores of 0.0566, 0.0470, 0.0152 and 0.0021. In terms of NDCG@5, Most Popular has the best performance with a score of 0.1829 and almost twice as good a score as ALS with 0.0927. BPR, LMF and Random follow with scores of 0.0773, 0.0289 and 0.0046. In terms of ROC_AUC, LMF has achieved a good score of 0.8685, while Most Popular is better here as well, with a score of 0.9326. ALS, BPR and Random achieved the lowest scores of 0.6392, 0.6168 and 0.5006. The ROC_AUC score of Random makes sense in terms of a close score of 0.5, indicating a random ranking (*Lü et al.*, 2012). Finally, regarding PR_AUC, Most Popular gets the best score of 0.0977, followed by ALS, with a score of 0.0416. BPR and LMF achieved the third and fourth positions with almost equal scores of 0.0300 and 0.0298. Random also accomplished the worst performance at this stage, with a score of 0.0061.

Summing up, the Most Popular recommender outperforms all other recommenders using these types of filters. The filters applied for this experiment were the exclusion of articles published outside the specified time period ranging from the 12th of September to the 7th of November, 2022. Additionally, a filter of the exclusion of articles in the section “Direkte”, whereas all articles in this section refer to the same page and were misleading from a recommenders perspective.

4.2.2 Experiment B2: Specific Filters

In the second sub-part of this experiment, our starting point is the data from Experiment B_1, as described in Table 4.1, including the applied filters. It is a well-known phenomenon that the methods the recommender systems obtain can contain data that is considered irrelevant for a user, referred to as noise. The data is obtained either through explicit feedback, where the user states the satisfaction of an item in terms of a rating, or implicit feedback, i.e. through the interaction of clicks. In the case of human activity, errors may occur in the form of noise, where an explicit rating can be considered a heavy or tiring process. On the other side, with implicit feedback, where the interaction of a click is potentially not of interest, there will perhaps be an even greater potential for errors or noise to occur in the data (*O'Mahony et al.*, 2006). This is the basis of data and would be the source for training the machine learning algorithms. Thus, it is important to analyse and filter out potential noise.

In our case, the dataset is the type of implicit feedback from clicks made by users on articles. This means that we have no degree of satisfaction whether the user liked the article or not. In this experiment, we perform more specific filters for removing data that can be considered as noise.

An article recently published, or has not been read by a single user, will be considered to be in a cold start situation. This situation cannot be considered from a collaborative filtering perspective, as the recommendations are based on user behaviour. Each line in the dataset represents that a user has performed an event of a click on an article. This means that each article represented in the dataset will have at least one interaction. A single click on an article indicates that the interest at that time has not been great yet, or the fact that the article is not that relevant anymore. The decision to exclude those articles has been executed, as this can potentially be a form of noise. The minimum for an article to be included in the dataset is two or more clicks. The reason why only two clicks were considered is by the fact of not excluding too much data.

A user having inconsistent behaviour in the form of periodically low or high activity is another scenario considered. Including users having continuity in their activity would contribute to a certain diversity of data. The cold start scenario is also an aspect among users, where lack of data of a user will potentially affect algorithms' ability for correct predictions. As mentioned, the dataset has a time period ranging from the 12th of September to the 7th of November, 2022. A filter that requires a user to read at least one article a week is applied. Additionally, this means that the requirements for a user to be included in the dataset and to be eligible for recommendations of this approach is to have read a minimum of 8 articles during these 8 weeks. After these filters, it remains 15 531 878 clicks provided by 69 762 users on 6 324 articles. Applying these filters results in a reduction of around a million interactions, around ninety thousand users and around thousand articles.

Table 4.4: Dataset for experiment B2

Dataset	**Interactions**	**Users**	**Items**
User Behaviour	15 531 878	69 762	6 324

In this experiment, we use that dataset of clicks displayed in Table 4.4. The default hyperparameters used can be seen in Table 4.5. A similar evaluation process as in

Experiment B1 is applied.

Table 4.5: Default hyperparameters used to train ALS, BPR and LMF in Experiment B2

Algorithms	**Hyperparameters**			
	factors	iterations	regularization	learning_rate
ALS	100	15	0.01	N/A
BPR	100	100	0.01	0.01
LMF	30	30	0.6	1.0

Table 4.6: Evaluations of Experiment B2

Algorithms	**Evaluation metrics**					
	P@5	R@5	MAP@5	NDCG@5	ROC_AUC	PR_AUC
Random	0.0086	0.0006	0.0038	0.0084	0.5012	0.0120
Most Popular	**0.2845**	**0.0220**	**0.2037**	**0.2950**	**0.9293**	**0.1388**
ALS	0.1789	0.0136	0.1134	0.1825	0.7495	0.0839
BPR	0.1135	0.0086	0.0698	0.1183	0.6356	0.0406
LMF	0.0504	0.0035	0.0258	0.0500	0.8601	0.0491

As seen in Table 4.6, Most Popular performs best in terms of P@5 with a score of 0.2845, followed by ALS, BPR, LMF and Random with scores of 0.1789, 0.1135, 0.0504 and 0.0086. Secondly, in terms of R@5, Most Popular and ALS receive the best scores of 0.0220 and 0.0136. BPR, LMF and Random follow with scores of 0.0086, 0.0035 and 0.0006. According to the measure of MAP@5, again, Most Popular receives the best score of 0.2037, followed by ALS, BPR, LMF and Random with the scores 0.1134, 0.0698, 0.0258 and 0.0038. In terms of NDCG@5, the best scores achieved are by Most Popular and ALS of the scores 0.2950 and 0.1825. BPR has more than twice as high a score as LMF, where the scores are 0.1183 and 0.0500. LMF achieved the lowest score of 0.0084. In terms of ROC_AUC, LMF archives a good score of 0.8601, while Most Popular at this time as well has better performance and receives a score of 0.9293. ALS, BPR and Random achieved the lowest scores of 0.7495, 0.6356 and 0.5012. Finally, in terms of PR_AUC, the best performance was achieved by Most Popular and ALS with scores of 0.1388 and 0.0839. LMF, BPR and Random follow with the scores 0.0491, 0.0406 and 0.0120.

Summing up, the baseline Most Popular outperforms all other recommenders using these types of filters. The filters from Experiment B1 were applied, excluding articles published outside the period ranging from the 12th of September to the 7th of November, 2022, in addition to the exclusion of articles in the section “Direkte”. Additionally, a filter of articles only read by one user was applied. A filter of inconsistent user activity that required the user to read at least one article per week through the period of 8 weeks to be qualified was also applied.

Moreover, the scores of all algorithms in this experiment have increased compared to Experiment B1. This could mean that the dataset provided in Experiment B1 had an overload of data considered as noise, where the performance of recommenders was affected.

4.2.3 Experiment B3: Hyperparameter Tuning

In the third sub-part of this experiment, our starting point is the data from Experiment B2: Specific Filters, as described in Table 4.4, including the applied filters. There are no fixed answers of which hyperparameters to apply, as this is individual for each algorithm. In addition to that, the dataset or the problem at hand causes an influence. This experiment aims to do a manual hyperparameter tuning to get a better insight into which parameters to use on this dataset to enhance the ML algorithm's performance. For Alternating Least Squares (ALS), the chosen hyperparameters to test are factors, iterations, and regularizations. For Bayesian Personalized Ranking (BPR) and Logistic Matrix Factorization (LMF), the same hyperparameters are chosen in addition to the learning rate. In order to perform this experiment, each machine learning model ALS, BPR and LMF are trained with an identical training set. Each combination of the selections of hyperparameters is evaluated according to the same metrics used in Experiments B1 and B2. This means the number of times each model is evaluated is by the product of the length of all combinations of hyperparameters. The method for selecting the best hyperparameters is chosen by the evaluation metric of the highest P@5 value. For each model, the top 5 results are displayed in Table 4.7, 4.8 and 4.9.

Listing 4.1: ALS hyperparameters chosen to tune

```
        factors = [5,10,15,30,60,100]
     iterations = [5,10,15,30,60,100]
regularizations = [0.001,0.01,0.1]
```

The list of the chosen values for the hyperparameter to test for ALS is displayed in Listing 4.1. This means that for ALS, all combinations of these three sets of hyperparameters are evaluated, totalling the product of the length of all three. This corresponds to 108 rows of evaluations of the metrics for this algorithm.

Table 4.7: Top 5 results of hyperparameter tuning of ALS, measured by the best P@5.

Hyperparameters			**Evaluation metrics**					
fac	iter	reg	P@5	R@5	MAP@5	NDCG@5	ROC_AUC	PR_AUC
5	**5**	**0.01**	**0.2304**	0.0178	0.1590	0.2381	0.8826	0.1166
5	30	0.01	0.2294	0.0176	0.1591	0.2375	0.8838	0.1157
5	10	0.001	0.2292	0.0176	0.1589	0.2370	0.8838	0.1156
5	5	0.1	0.2276	0.0177	0.1571	0.2362	0.8832	0.1159
5	15	0.001	0.2267	0.0174	0.1569	0.2346	0.8847	0.1151

It is evident that the optimal selection of factors for ALS is 5, as indicated in Table 4.7 by the fact that all top five results contain this value. In terms of iterations, a greater range of values is observed, ranging between the variants between 5 and 30, i.e., 4 out of 6 available values. The best score of P@5 is achieved using 5 iterations. But as seen in P@5, these scores have minor differences considering the spread of iterations. In terms of regularizations, all available combinations are displayed in the top 5 of P@5, meaning that these minor differences of P@5 do not affect the results significantly. The best score is measured using a regularization of 0.01.

Listing 4.2: BPR hyperparameters chosen to tune

```
      factors = [5,10,15,30,60,100]
   iterations = [5,10,15,30,60,100]
regularizations = [0.001,0.01,0.1]
 learning_rate = [0.001,0.01,0.1]
```

The list of the chosen values for the hyperparameter to test for BPR is displayed in Listing 4.2. This means that for BPR, all combinations of these four sets of hyperparameters are evaluated, totalling the product of the length of all four. This corresponds to 324 rows of evaluations of the metrics for this algorithm.

Table 4.8: Top 5 results of hyperparameter tuning of BPR, measured by the best P@5.

Hyperparameters				**Evaluation metrics**					
fac	iter	reg	l_rate	P@5	R@5	MAP@5	NDCG@5	ROC_AUC	PR_AUC
30	**5**	**0.1**	**0.001**	**0.2869**	0.0222	0.2049	0.2966	0.6266	0.1028
100	100	0.1	0.001	0.2866	0.0222	0.2045	0.2960	0.7252	0.1055
5	15	0.1	0.001	0.2866	0.0222	0.2044	0.2961	0.6565	0.0908
60	10	0.01	0.001	0.2866	0.0222	0.2043	0.2960	0.7223	0.1175
100	5	0.01	0.001	0.2866	0.0222	0.2041	0.2957	0.7331	0.1202

The optimal selection of factors for BPR is not straightforward, considering that factors between 5 and 100 appear in the top 5 of P@K. Four out of six possible combinations of the factors appear at the top. Note that the P@5 scores in the top 5 at P@K are almost identical across different factors. Anyway, the best score is achieved using 30 factors. Additionally, the optimal values for BPR are not entirely clear in terms of iterations. In the top 5 of P@5, four out of six possible combinations of iterations also appear. Anyway, the best score is achieved using 5 iterations. Regarding regularization, we observe that the value of 0.1 is the three highest scores based on the list of top 5 P@5. However, it is worth noting that the available combination for regularizations is only three, with two of them appearing in the top 5 P@K. Finally, the learning_rate of 0.001 is the only combination that appears in the top 5, indicating that this selection achieves the best performance for BPR on this dataset.

Listing 4.3: LMF hyperparameters chosen to tune

```
      factors = [5,10,15,30,60,100]
   iterations = [5,10,15,30,60,100]
regularizations = [0.001,0.01,0.1,0.3,0.6,1.0]
 learning_rate = [0.001,0.01,0.1,0.3,0.6,1.0]
```

The list of the chosen values for the hyperparameter to test for LMF is displayed in Listing 4.3. This means that for LMF, all combinations of these four sets of hyperparameters are evaluated, totalling the product of the length of all four. This corresponds to 1296 rows of evaluations of the metrics for this algorithm. The decision to expand values in regularizations and learning_rate is by the fact that LMF has higher defaults for those hyperparameters compared to ALS and BPR.

Table 4.9: Top 5 results of hyperparameter tuning of LMF, measured by the best P@5.

Hyperparameters				**Evaluation metrics**					
fac	iter	reg	l_rate	P@5	R@5	MAP@5	NDCG@5	ROC_AUC	PR_AUC
5	**100**	**1.0**	**1.0**	**0.1285**	0.0092	0.0755	0.1295	0.8918	0.0720
5	100	0.6	1.0	0.1117	0.0082	0.0667	0.1159	0.8792	0.0625
5	100	1.0	0.6	0.1053	0.0083	0.0700	0.1220	0.8506	0.0534
5	60	1.0	1.0	0.1014	0.0078	0.0608	0.1090	0.8720	0.0585
10	60	1.0	1.0	0.0978	0.0075	0.0586	0.1038	0.8775	0.0612

It is worth mentioning that the number of combinations for regularizations and learning_rate is both six, rather than only three for BPR. The optimal selection of factors for LMF is approximately clear due to the fact that four out of the top 5 P@5 contain a factor of 5. Considering iterations, it tends to have high values, where the choice of 60 and 100 factors occurs at the top 5. The three values in the top 5 of P@5 have 100 iterations, indicating that this is the most optimal value for LMF and this dataset. In terms of regularizations, this also tends to perform best with the choices of high values, where 0.6 and 1.0 occur in the top 5 of P@5. These values are the two largest of the possible combinations of regularizations. The values of 1.0 occur most often and have the best-observed performance. These findings indicate that the best performance is achieved using high values of regularizations. Similarly, in terms of learning_rate, the two highest combinations of possible values, 0.6 and 1.0, appear on the top 5 of P@5. The value of 1.0 also occurs most often and is that selection that receives the top two of best performance. This indicates that 1.0 is the most optimal choice, while an overall indication that a selection of high values achieves the best performance.

In Figure 4.3, a boxplot of all P@5 results of hyperparameter tuning for ALS, BPR and LMF can be seen. Regarding the P@5 for ALS, the spread of scores ranges from about 0.175 as a minimum to about 0.23 as a maximum. There are only 108 combinations of factors, iterations, and regularizations available. In terms of P@5 for BPR, the spread is even greater and ranges from around 0.175 as a minimum and around 0.287 as a maximum. There are 324 combinations of factors, iterations, regularizations and learning rates available. The range of the definition area is greater, which may indicate that the choice of hyperparameters has a greater impact. Regarding P@5 for LMF, the spread is smaller and more compact than BPR. The circles/dots indicate outliers, which means values that deviate significantly compared to the rest of the results. There are 1296 combinations of factors, iterations, regularizations and learning rates available. The minimum value is close to the score of 0, while the maximum value in terms of the diversity of the scores is around 0.08. But considering the outliers as well, the best score is achieved of around 0.128. These outliers may indicate that some selections of parameters affect the performance significantly for this algorithm and the dataset at hand.

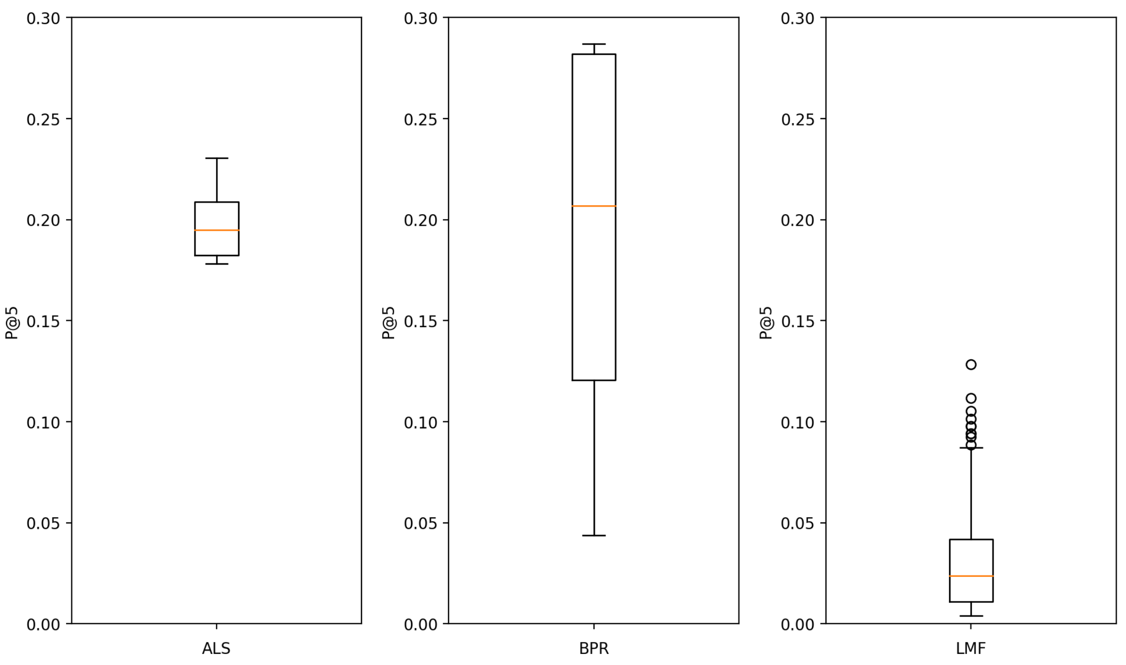


Figure 4.3: Boxplot of all results of P@5 for hyperparameter tuning for ALS, BPR and LMF.

Table 4.10: Most optimal hyperparameters in terms of best P@5 scores for ALS, BPR and LMF

Algorithms	**Hyperparameters**			
	factors	iterations	regularization	learning_rate
ALS	5	5	0.01	N/A
BPR	30	5	0.1	0.001
LMF	5	100	1.0	1.0

The most optimal hyperparameters that received the best performance in the score of P@5 on each algorithm ALS, BPR and LMF can be seen in Table 4.10. The dataset used for this extraction of most optimal hyperparameters is the dataset used in Experiment B2: Specific Filters of 15 531 878 interactions, provided by 69 762 users on 6 324 items, seen in Table 4.4.

Table 4.11: Best scores achieved in terms of P@5 for ALS, BPR and LMF, compared to baseline recommenders Random and Most Popular

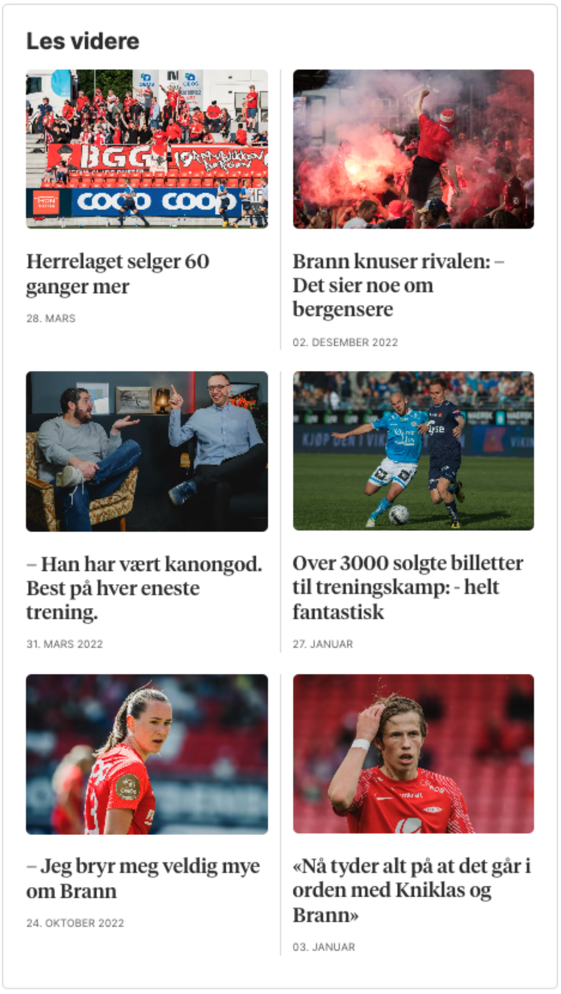
Algorithms	**Evaluation metrics**					
	P@5	R@5	MAP@5	NDCG@5	ROC_AUC	PR_AUC
Random	0.0086	0.0006	0.0038	0.0084	0.5012	0.0120
Most Popular	0.2845	0.0220	0.2037	0.2950	**0.9293**	**0.1388**
ALS	0.2304	0.0178	0.1590	0.2381	0.8826	0.1166
BPR	**0.2869**	**0.0222**	**0.2049**	**0.2966**	0.6266	0.1028
LMF	0.1285	0.0092	0.0755	0.1295	0.8918	0.0720

In summary, the overall performance of ALS, BPR and LMF has increased compared to the evaluation of the same dataset using default hyperparameters (see Table 4.6). Compared to the baseline recommender Most Popular, BPR has a slightly better performance of P@5 with a score of 0.2869, while Most Popular has a score of 0.2845. ALS, LMF and the baseline recommender Random follow with scores of 0.2304, 0.1285 and 0.0086. Regarding R@5, BPR receives the best score with 0.0222, tightly followed by Most Popular with a score of 0.0220. ALS, LMF and Random receive the lowest scores of 0.0178, 0.0092 and 0.0006. According to the measure of MAP@5, again, BPR receives the best score of 0.2049, followed by Most Popular, ALS, LMF and Random with scores of 0.2037, 0.1590, 0.0755 and 0.0038. Regarding NDCG@5, Most Popular receives a quite high score of 0.2950, while BPR currently performs better and receives a score of 0.2966. ALS, LMF and Random receive the lowest scores of 0.2381, 0.1295 and 0.0084. In terms of ROC_AUC, Most Popular performs best, with a score of 0.9293. LMF follows with a score of 0.8918, tightly followed by ALS with a score of 0.8826. The two lowest scores are achieved by BPR and Random, with 0.6266 and 0.5012. Finally, in terms of PR_AUC, the best performance was achieved by Most Popular with a score of 0.1388, followed by ALS, BPR, LMF and Random with scores of 0.1166, 0.1028, 0.0720 and 0.0120.

ALS, BPR and LMF have lower performance than the baseline model based on Most Popular in two out of the total three experiments. BPR has slightly better performance than Most Popular in the third experiment, including the most optimal hyperparameters. Collaborative filtering (CF) in the news domain is, as stated in Section 1.2, more complex due to articles' timeliness with short duration, limited relevancy (lifespan/lifetime) and more activity due to low consumption time of articles. In addition, the implementation of CF requires more effort to solve. At this stage, the decision to test the technique based on content data has a higher degree of reliance and is easier to implement, where therefore taken.

4.3 Experiment C: Online evaluation

An online experiment has been conducted to evaluate the quality of a content-based recommendation based on a BERT model in the Norwegian language on the textual content of articles. This online experiment was conducted on Bergens Tidende's digital newspaper for 14 days, from the 15th to the 28th of March, 2023. The recommendation is based on the BERT model in the Norwegian language sentence embeddings and cosine similarity to calculate the degree of similarity between 0-1. Each article is iterated through all candidate articles in the last 6 months, in addition to articles with a high lifetime value in the last 14 months. The top 6 articles with the highest cosine score, thus considered most similar, are selected as recommendations for that specific article. This model by BERT is A/B tested against an existing and more generic model based on a tag structure. It is, therefore, a 50/50 split, where the user is either presented with the new model of BERT or the former model by the generic tags. Users are presented with a list of recommendations at the bottom when reading an article under the slot called “Les Videre”, as shown in Figure 4.4.


Les videre

Herrelaget selger 60 ganger mer
28. MARS

Brann knuser rivalen: – Det sier noe om bergensere
02. DESEMBER 2022

– Han har vært kanongod. Best på hver eneste trening.
31. MARS 2022

Over 3000 solgte billetter til treningskamp: - helt fantastisk
27. JANUAR

– Jeg bryr meg veldig mye om Brann
24. OKTOBER 2022

«Nå tyder alt på at det går i orden med Kniklas og Brann»
03. JANUAR

Figure 4.4: Screenshot of the recommendation widget the user is presented with, located at the bottom when reading an article in Bergens Tidende. Translated to English, "Les Videre" is "Read Further".

There is no precise data on how many users have viewed the recommendation widget "Les videre", seen in Figure 4.4, or which recommenders (BERT and Tags) they are presented with. On the other hand, exact data of all clicks performed on both recommenders is available. A method to receive how many users viewed the recommender widget is by tracking how far a user scrolled on an article before performing a "Leave" event. A "Leave" event indicates that they either opened a new tab or moved along to another article. A user who executed a scroll rate of 85% or greater on an article is assumed to have viewed the recommendation widget. All views and clicks data are provided by the participants of the A/B test in BT by registered/paying users.

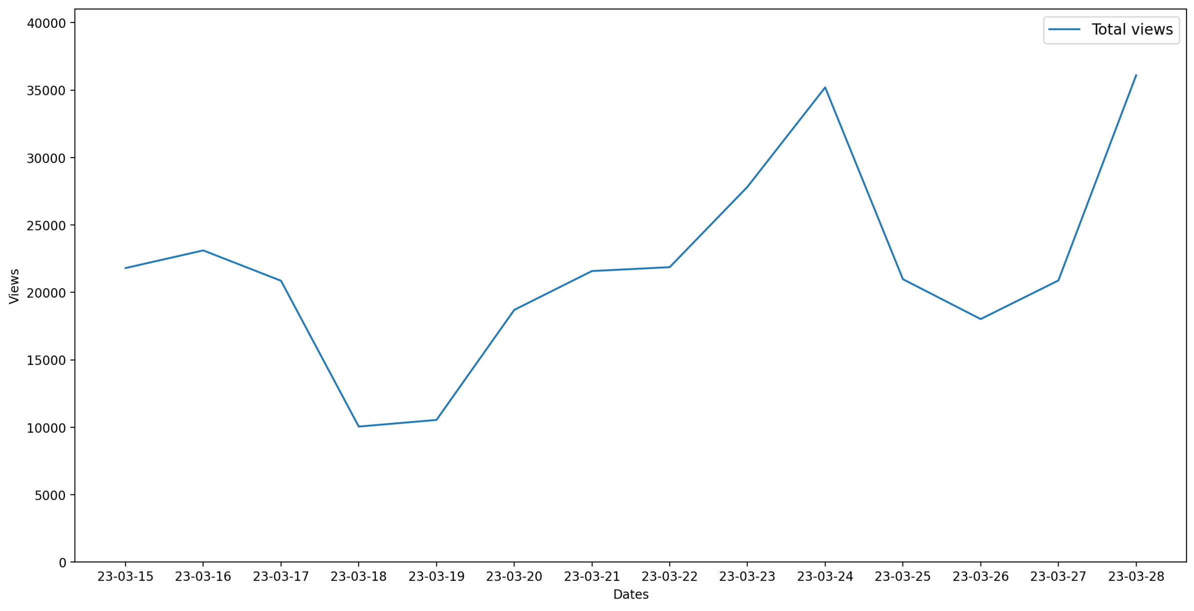


Figure 4.5: Total views of A/B test recommenders widget

In Figure 4.5, the blue line indicates all views of the recommendation widget from the 15th to the 28th of March, 2023. Given that the A/B test is an equal 50/50 split, we can assume that each recommender is viewed by half of the total number of views.

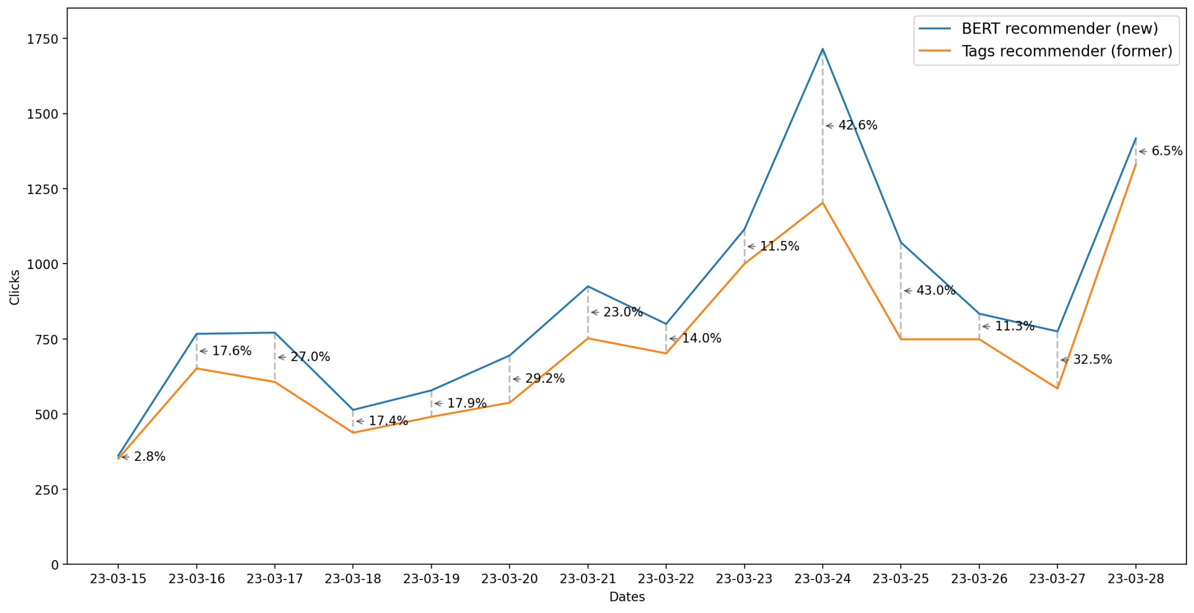


Figure 4.6: Clicks performed on A/B test of recommenders widget

From day one, Figure 4.6 shows that the BERT model has a greater engagement in terms of clicks than the previous model based on Tags. The data is based on clicks on the UI elements made by the users in the time span between the 15th and the 28th of March, 2023. The x-axis represents the dates, whereas the y-axis represents the number of clicks made. The blue line shows the new recommendation model of BERT, while the orange line shows the former recommendation model based on Tags. The dashed lines between each point by dates show a percentage difference between the models. All percentages are positive, meaning that the BERT model has more clicks every day for these two weeks of the experiment. We also see a strong overall correlation between the lines, despite some situations where one increases while the other remains flat or slightly decreases. The overall strong correlation makes sense by the fact that there is a 50/50 split of which of the models the users are presented with. In total, the BERT model has 21.6% more engagement.

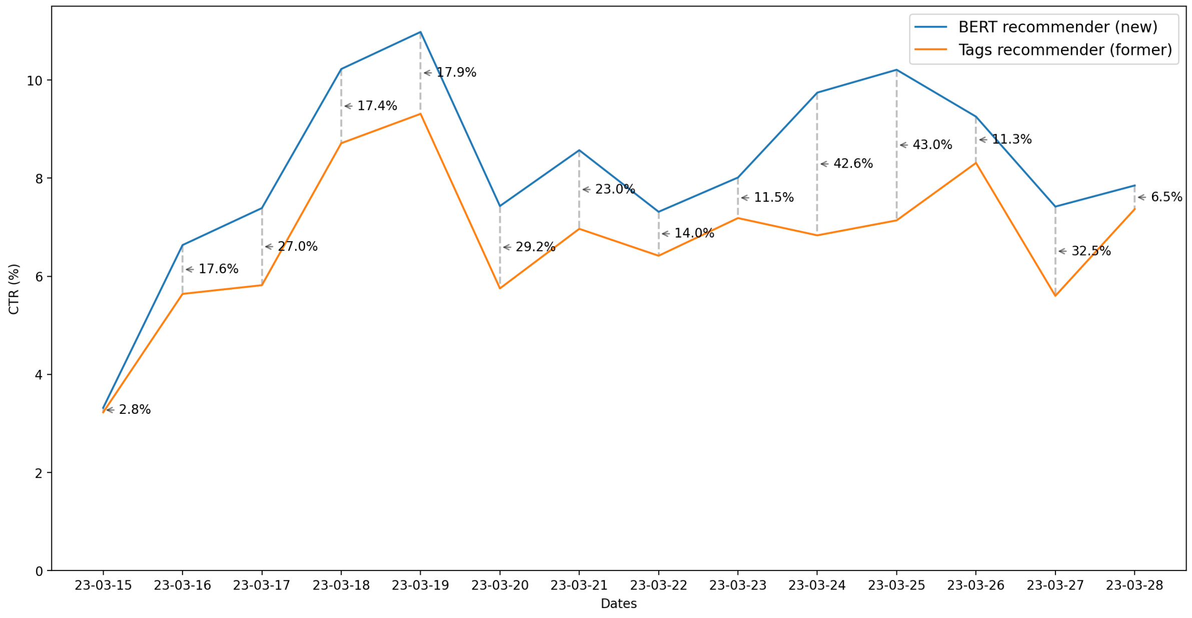


Figure 4.7: CTR on A/B test of recommenders widget

In terms of the CTR of BERT and Tags recommenders, the CTR is displayed in Figure 4.7 for each day along the experiment of two weeks. Similarly to Figure 4.6, the percentage difference between BERT (blue-line) and Tags (orange-line), as the dashed line between them, are equal. This makes sense, as the CTR is a calculation of clicks performed divided by views. As stated, it is a 50/50 split, and the assumption of views is measured for each recommender as half of the total number of views of the recommendation widget, i.e. the views for both are equal. The views, clicks, and CTR statistics can be summarised in Table 4.12.

Table 4.12: Statistics from A/B testing

Metric	BERT recommender	Tags recommender
#Views	153 794	153 794
#Clicks	12 340	10 148
#CTR	0.0802	0.0659

It is important to note that the position of this recommendation list/slot was at the very bottom when reading an article. Furthermore, there are other lists of recommendations that appear above this one. Many users may not scroll all the way down and see this list of recommendations. Taking this into account, the number of views and clicks per day gives us a good basis for insight into how it performs.

In order to get a better understanding of which users are interested in such recommendations of the A/B test, we have carried out an analysis in terms of the user's gender and age for each click performed. This means if we have an active reader who performed ten clicks on the recommendation lists, the age and gender of this person will be counted ten times. Note that in some cases, we don't have data on the gender and age of users. This is, e.g., users invited as a sub-user of a paying subscriber. In Figure 4.8, there are three categories female, male and unknown. In Figure 4.9, we have categorised age into three categories 45 years or younger, over 45 years and unknown. In Bergens Tidende, 45 years is a threshold they often use to figure out if they hit the younger or older target group.

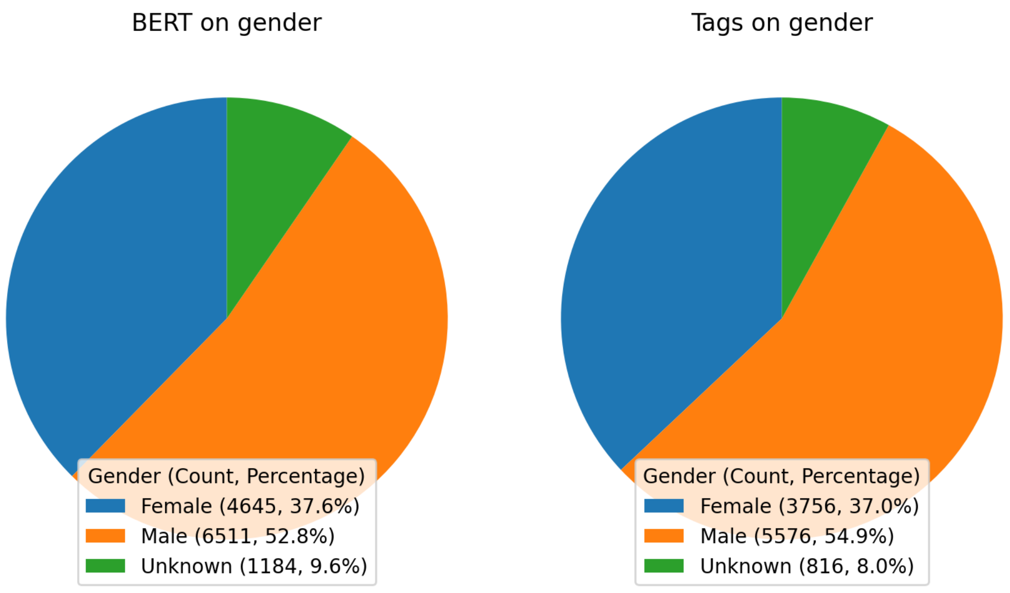


Figure 4.8: Distributions of the gender in the A/B test of the BERT model and the generic tags model

On the left-hand side of Figure 4.8, the distribution of BERT is illustrated. It is evident that the majority of clicks, totalling 6511 or 52.8%, were made by males. Females made the second highest number of clicks, with 4645, representing 37.6% of the total clicks. The remaining clicks of 1184, accounting for 9.6% of the total, were made by subscribers of unknown gender. On the right-hand side of Figure 4.8, the distribution of Tags is illustrated, and similarly to BERT, the majority of clicks were made by males, with 5576 clicks or 54,9%, followed by females with 3756 clicks, corresponding to 37%. The remaining 816 clicks, or 8%, were made by subscribers whose gender was unknown.

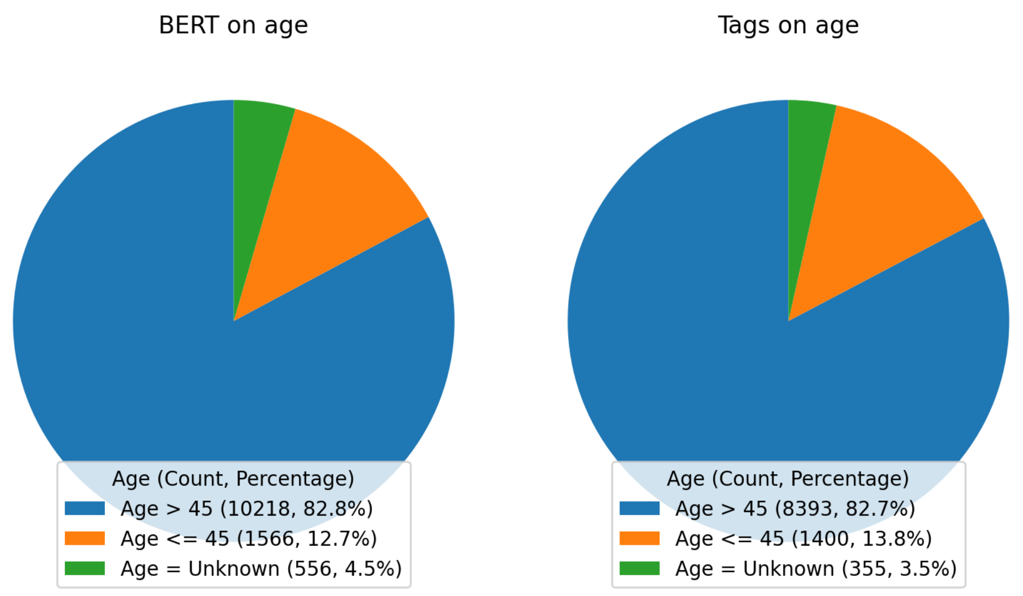


Figure 4.9: Distributions of the age in the A/B test of the BERT model and the generic tags model

On the left-hand side of Figure 4.9, the distribution of BERT is illustrated. It is evident that the majority of clicks, totalling 10218, or 82.8% of the total, were made by subscribers with age over 45 years. Subscribers less than or equal to 45 made the second-highest number of clicks, corresponding to 1566, or 12.7% of the total clicks. The remaining clicks of 556, accounting for 4.5% of the total, were made by subscribers of unknown age. On the right-hand side of Figure 4.9, the distribution of Tags is illustrated, and similarly to BERT, the majority of clicks were made by subscribers with age over 45, consisting of 8393 or 82.7% of the total, followed by the age group of less than or equal to 45 with 1400 clicks, corresponding to 13.8% of the total. The remaining 355 clicks, or 3.5% of the total, were made by subscribers whose age was unknown.

As seen in the two previous analyses (Figure 4.8 and Figure 4.9), the distribution of the gender and age of the user clicks on BERT and Tags were highly correlated. It is evident that the majority of users providing these clicks on the A/B test in terms of gender is Male, and in terms of age group, the users are mostly over 45 years old. There is a well-known phenomenon in BT that these two groups constitute the majority of subscribers. BT aims to reach out to the female and younger groups, as those often show less interaction and attention. In the following experiment, a further investigation of the user's gender and age using Bayesian statistics of the recommenders is conducted. The recommender based on Tags will be referred to as baseline, while the recommender based on BERT is referred to as challenger.

The chosen beta prior is (1,1), as we assume this is a uniform distribution. This is a standard flat prior to use for a lot of analysis. The binomial distribution for each group is created using the number of trials, which is the total number of views, the probability of the Beta prior and the successes, indicating the total number of clicks.

Firstly, an investigation of the gender distribution of the A/B test is conducted. Users with unspecified gender are excluded from this analysis. The analysis is divided into four groups, where we display the distribution of successes and trials in the baseline and challenger for both males and females.

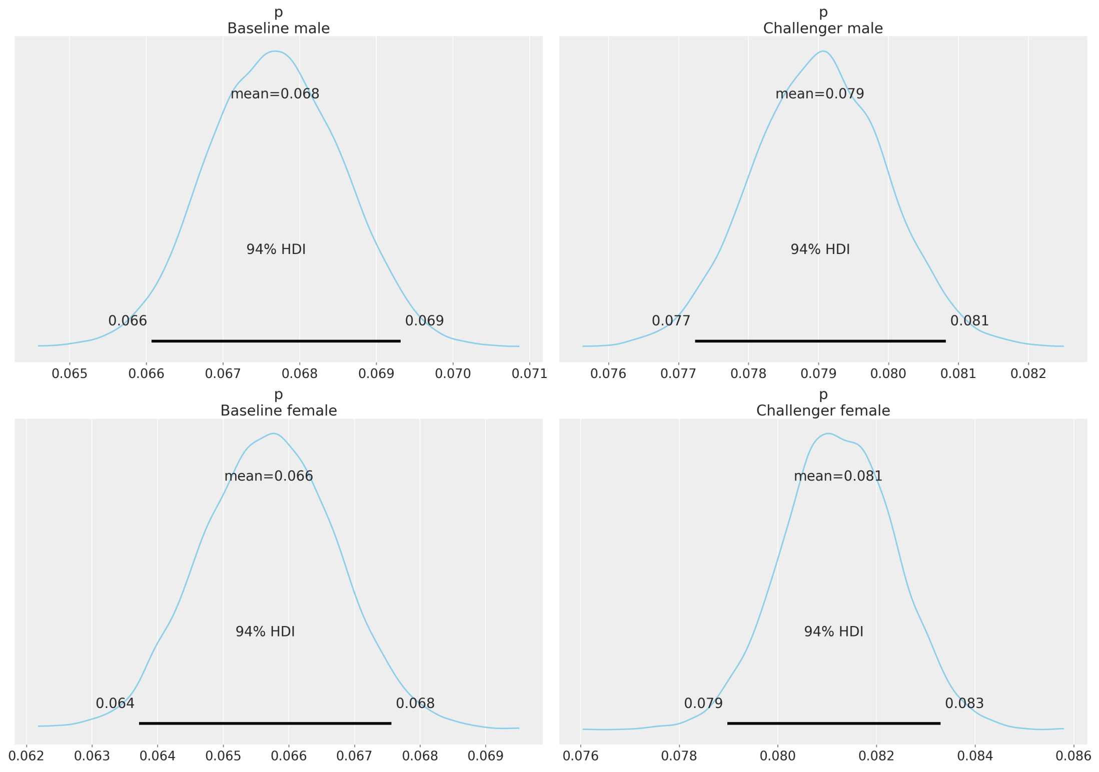


Figure 4.10: The 94% credible intervals of the posterior distributions for both the baseline and challenger recommender on male and female genders

The upper two figures in Figure 4.10 illustrate the posterior distributions for the baseline and the challenger of male users. The baseline has a 94% credible interval between 0.066 and 0.069, with a mean of 0.068. The challenger has a 94% credible interval between 0.077 and 0.081, with a mean of 0.079. The lower two figures in Figure 4.10 illustrate the posterior distributions for the baseline and challenger for female users. The baseline has a 94% credible interval between 0.064 and 0.068, with a mean of 0.066. The challenger has a 94% credible interval between 0.079 and 0.083, with a mean of 0.081.

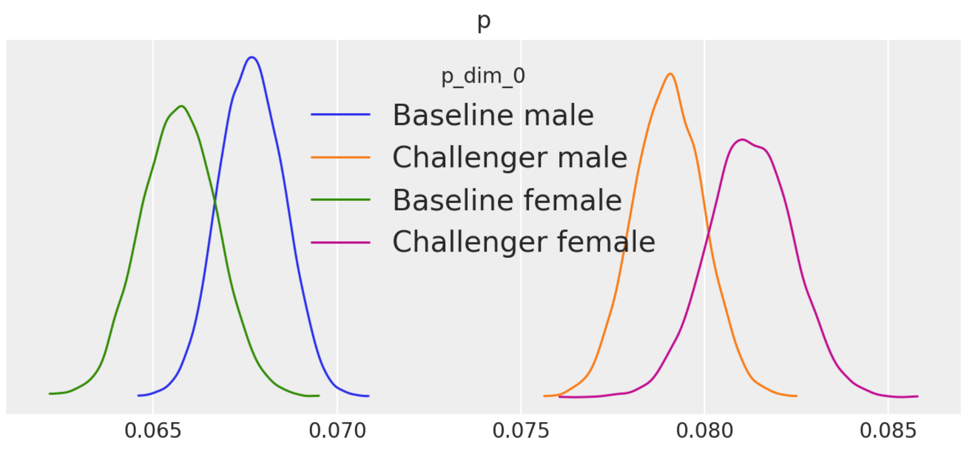


Figure 4.11: Summary of posterior distributions for both the baseline and challenger recommender on male and female genders

To determine that the challenger performs better than the baseline, a comparison of the posterior distribution samples for males and females is conducted. In terms of the challenger compared to the baseline of males, it turns out that the posterior distribution samples for the challenger have a probability of performing better in all cases, i.e. 100%. Additionally, in terms of the challenger compared to the baseline for females, it turns out that the posterior distribution samples of the challenger also perform best in all cases, i.e. 100%. This makes sense due to the fact that the posterior distribution samples of the challenger and baseline for each gender group are not overlapping each other, as seen in Figure 4.11.

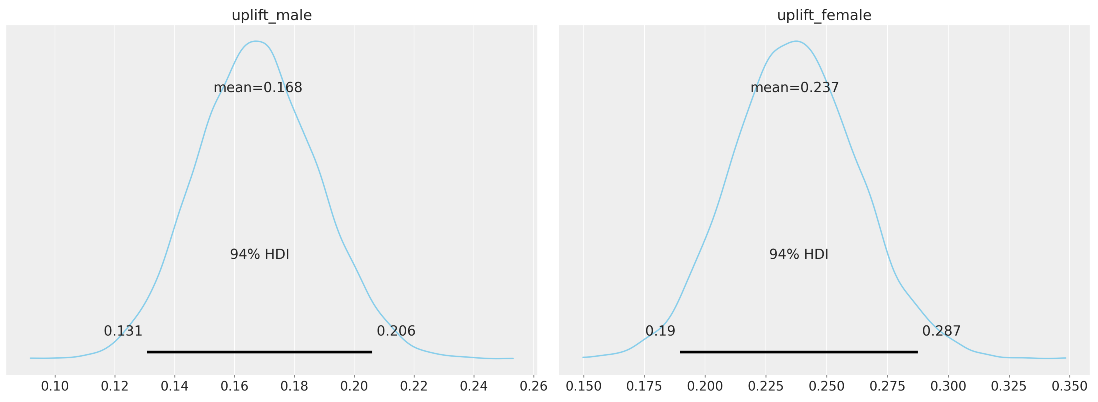


Figure 4.12: Uplift posterior distributions on male and female genders

By the posterior distributions of the baseline and challenger, for male and female, the uplift can be calculated. The uplift states the relative increase of the challenger by the baseline, which is calculated by the challenger by subtraction and division of the baseline. The uplift distribution of males has a 94% credible interval between 0.131 and 0.206, with a mean of 0.168. This can be seen on the left-hand side of Figure 4.12. The uplift distribution of females has a 94% credible interval between 0.190 and 0.287, with a mean of 0.237. This can be seen on the right-hand side of Figure 4.12.

The mean uplift for females is 0.237, while 0.168 for males. This is a finding that indicates that the BERT model had a greater impact on females in terms of the percentage growth of clicks. As mentioned earlier, BT has a greater number of male subscribers and therefore aims to engage females. This analysis of the A/B test proves that the BERT model has increased female engagement. The uplift for both is positive, which also indicates that the BERT model compared to the former model of Tags, had an impact on the engagement. A summary of the 94% credible intervals for posterior distributions and uplifts for gender can be seen in Table 4.13.

Table 4.13: Summary of all 94% credible intervals for gender groups

Groups	Mean	Sd	Hdi_3%	Hdi_97%
Baseline male	0.068	0.001	0.066	0.069
Challenger male	0.079	0.001	0.077	0.081
Baseline female	0.066	0.001	0.064	0.068
Challenger female	0.081	0.001	0.079	0.083
Uplift male	0.168	0.020	0.131	0.206
Uplift female	0.237	0.026	0.190	0.287

The second group of investigation is the age distribution of the A/B test. Users with unspecified ages are excluded from this analysis. The analysis is divided into four groups, where we display the distribution of successes and trials in the variants of the baseline and challenger model for users less than or equal to 45 and users over 45.

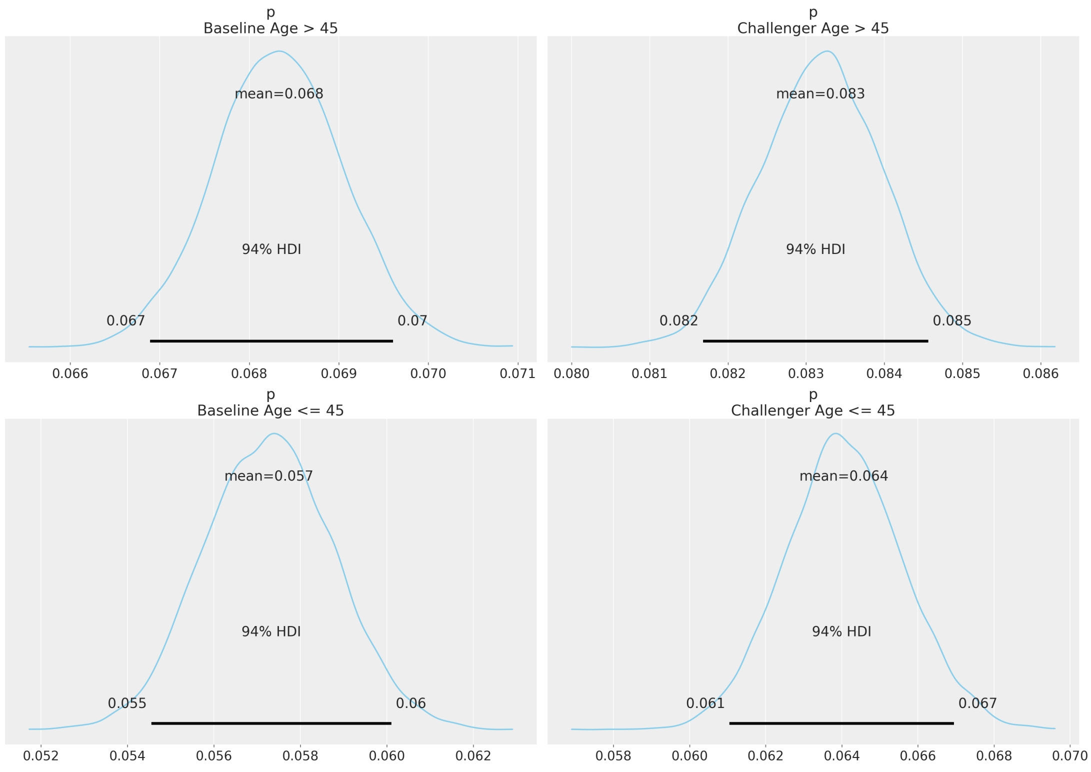


Figure 4.13: The 94% credible intervals of the posterior distributions for both the baseline and challenger of Age>45 and Age<=45

The upper two figures in Figure 4.13 illustrate posterior distributions for the baseline and challenger for users with an age over 45. The baseline has a 94% credible interval between 0.067 and 0.07, with a mean of 0.068. The challenger has a 94% credible interval between 0.082 and 0.085, with a mean of 0.083. The lower two figures in Figure 4.13 illustrate the posterior distributions for the baseline and challenger for users with an age of less than or equal to 45. The baseline has a 94% credible interval between 0.055 and 0.06, with a mean of 0.057. The challenger has a 94% credible interval between 0.061 and 0.067, with a mean of 0.064.

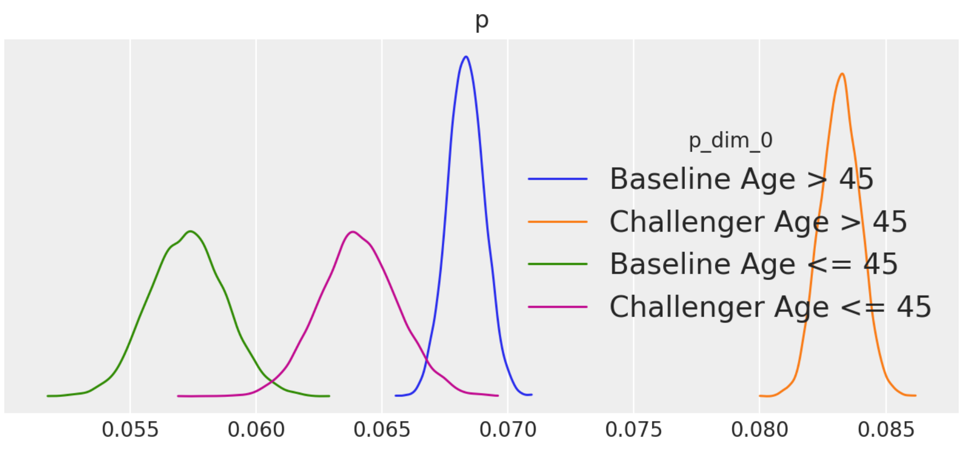


Figure 4.14: Posterior distributions for both the baseline and recommender on Age>45 and Age<=45

To determine that the challenger performs better than the baseline, a comparison of the posterior distribution samples for ages over 45 and ages less than or equal to 45 is conducted. In terms of the challenger compared to the baseline of age over 45, it turns out that the posterior distribution samples for the challenger have a probability of performing better in all cases, i.e. 100%. In terms of the posterior distribution of samples for the challenger and the baseline for age less than or equal to 45, it turns out that the posterior distribution samples for the challenger have a probability of performing better in almost all cases, with 99.9%. This makes sense due to the fact that the posterior distribution samples of the challenger and baseline for age over 45 are not overlapping each other, while for age less than or equal to 45, the overlap is minimal, as seen in Figure 4.14.

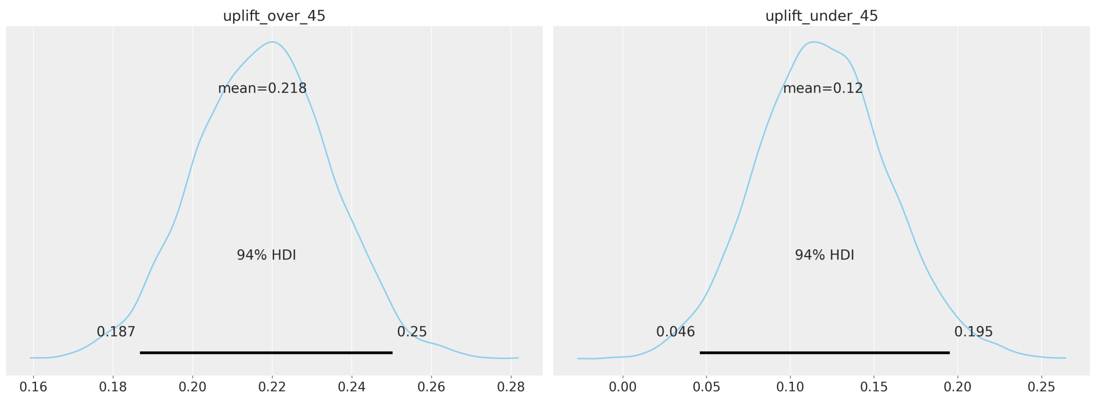


Figure 4.15: Uplift posterior distributions on Age>45 and Age<=45

The uplift distribution can be calculated by the posterior distributions of the baseline and the challenger for ages over 45 and ages less than or equal to 45. The uplift states the relative increase of the challenger by the baseline, which is calculated by the challenger by subtraction and division of the baseline. The uplift distribution of age over 45 has a 94% credible interval between 0.187 and 0.250, with a mean of 0.218. This can be seen on the left-hand side of Figure 4.15. The uplift distribution of age less than or equal to 45 has a 94% credible interval between 0.046 and 0.195, with a mean of 0.120. This can be seen on the right-hand side of the Figure 4.15.

The mean uplift for ages over 45 is 0.218, while 0.120 for ages less than or equal to 45. This is in some way expected due to the fact that, as mentioned earlier, most subscribers in total and the participants in the A/B test are over 45. The uplift for both is positive, which also indicates that the BERT model compared to the former model of Tags, had an impact on the engagement. A summary of the 94% credible intervals for the posterior distributions and uplifts can be seen in Table 4.14.

Table 4.14: Summary of all 94% credible intervals for age groups

Groups	Mean	Sd	Hdi_3%	Hdi_97%
Baseline Age > 45	0.068	0.001	0.067	0.070
Challenger Age > 45	0.083	0.001	0.082	0.085
Baseline Age <= 45	0.057	0.002	0.055	0.060
Challenger Age <= 45	0.064	0.002	0.061	0.067
Uplift Age > 45	0.218	0.017	0.187	0.250
Uplift Age <= 45	0.120	0.041	0.046	0.195

Chapter 5

Conclusion and Future Work

5.1 Summary

This thesis addresses how to improve recommendation generation for Norwegian news articles to enhance recycling of articles and to promote user engagement. This is done by using both content- and user-behavioural data provided by Bergens Tidene, one of the largest newspapers in Norway. The thesis proposes two recommendation techniques using content-based filtering and collaborative filtering.

The work includes:

- A presentation of the basic concepts related to the research, together with the related work within recommender systems (detailed in Chapter 2).
- A comprehensive description of the methods used for performing the study, including the experiment design of the two techniques. The experiment design includes the architecture, implementation, evaluation methods, and a description of the data used (described in Chapter 3).
- An evaluation of both proposed techniques in offline experiments (described in Chapter 4).
- An evaluation of a large-scale A/B test in an online experiment with over 20000 interactions on the media platform of Bergens Tidende (described in Chapter 4).

5.2 Main contributions

This thesis provides contributions to the field of news recommender systems by the following conditions:

- Performing a comprehensive analysis of industry data: Chapter 3 proposed a unique technique of sentence-embeddings by BERT, specially trained in the Norwegian language. In addition, a method for dimensionality reduction is detailed. In Chapter 4, an analysis of a plot visualisation is performed to evaluate the ability to cluster semantic equal articles.

- Conducting an offline evaluation of a proposed recommendation approach: Chapter 3 proposed evaluation metrics utilized to measure the performance of the collaborative filtering recommendation techniques through the recommender models ALS, BPR and LMF. These are compared to two common baseline models, i.e., random and most popular. In Chapter 4, a comprehensive offline evaluation is conducted in various data filtering stages and hyperparameter tuning optimisations.

- A comprehensive evaluation of a unique content-based recommendation technique with an online experiment: Chapter 3 proposed the methodology of A/B testing utilized on one of Norway's largest newspapers, i.e., Bergens Tidende, of a unique technique of sentence-embeddings by BERT, specially trained on the Norwegian language. In Chapter 4, a comprehensive descriptive- and Bayesian analysis is performed, focusing on the age and gender groups of the participants of the A/B test.

5.3 Conclusion

The exploratory analysis in Section 4.1 provides a dimension reduction of sentence embeddings by BERT, specially trained in the Norwegian language. The articles (plots) located close to each other indicate the technique's estimate of similar textual content. It appears that the BERT model has the ability to cluster articles with the same section closely, especially robot-written articles. Articles in these sections are written on the same structure, which strengthens the reliability of the model that such articles are closely located in clusters. Non-robot-written articles have a greater variety of textural structures, explaining why the cluster is more spread out in such sections. Regarding RQ1, this technique seems to have the ability to effectively draw semantic similarities of content.

The offline evaluation in Section 4.2 provides a comprehensive evaluation of the recommender models ALS, BPR and LMF using user behavioural data. These recommender models are compared to baseline recommender models of random and most popular. From the experiments B1 to B2 in Section 4.2, it is evident that the filters had an impact and improved the performance of all recommenders, including the baselines. This highlights the importance of identifying and the exclusion of data that affect the performance of the models, referred to as noise. For Experiment B1, ALS has the best scores compared to BPR and LMF on most of the evaluation metrics, except R@5 and ROC_AUC. For Experiment B2, ALS receives the best scores on all except ROC_AUC compared to BRP and LMF. But the baseline recommender Most Popular outperformed all of them in total in both experiments. After performing a hyperparameter tuning in Experiment B3 for ALS, BPR and LMF and utilizing the most optimal hyperparameters for each model, the performance has increased for all. This also highlights the importance of the selection of hyperparameters in these models, where the best choice for each model can be individual for different inputs of datasets. While ALS performed better than BPR and LMF on most of the metrics in the two previous experiments, it turns out that BPR outperforms them in terms of almost all metrics except ROC_AUC and PR_AUC. The best performance in terms of ROC_AUC and PR_AUC is achieved by the baseline model Most Popular. The performance of BPR in terms of the metrics P@5, R@5, MAP@5 and NDCG@5 is just slightly better than the baseline model Most Popular. Regarding RQ1, the proposed technique of recommender models has shown increased performance and the ability to effectively model the user behavioural data. Additional filters and optimisations could be researched further to increase performance even more for greater reliance on the recommendations. On the other hand, regarding RQ2, this technique produces personalised recommendations, as the basis of the recommender models is provided by user behavioural data.

The online evaluation in Section 4.3 provides descriptive- and Bayesian analysis on a large-scale A/B test based on BERT, specially trained in the Norwegian language, tested against a former model used in the media platform of Bergens Tidende. The descriptive results show that the BERT model has a greater engagement, with a 20.6% increase compared to the former model. In terms of gender and age groups of the participants, it turns out that males and ages over 45 are the majority of clicks performed on these recommendations. The results of the Bayesian analysis state that females have a greater increase/uplift of clicks performed on the BERT model rather than the former model compared to males' increase/uplift. Moreover, all uplifts for both the gender and age groups are positive, indicating that the BERT model had an impact on engagement. The posterior distribution of the gender groups showed that the probability is 100% that the BERT model is better than the former model. For the age group's posterior distribution, the probability is also 100% to be better of BERT for ages over 45, while 99.9% for ages less than or equal to 45. This supports RQ1, while this recommendation technique is an effective model for providing recommendations. Regarding RQ2, this recommendation technique is not personalised by itself due to that the recommendation list for each article would be equal for all users. But it could mitigate a cold start scenario, in addition to providing more robust recommendations together with the technique based on user behavioural data.

5.4 Limitations and future work

The research addressed in this thesis includes limitations and the potential for future work. This thesis's first limitation concerns the hyperparameter tuning process, which required several days to evaluate all combinations for each model. Therefore, a possible future work could be to test the models' combinations of hyperparameters to a greater extent, as this thesis has demonstrated that the choice of hyperparameters notably affected the performance.

Another limitation is collaborative filtering in the news domain, which is more challenging than recommending items to buy or movies to watch. The reason behind this is the temporal relevance of news articles due to timeliness. A news article often has a short-term relevance, i.e., a short lifetime. Compared to buy and watch habits, these belong more to long-term relevance, and the importance of applying filters to exclude damaging data for a model is not that important. This is part of the decision why only content-based filtering was applied as a technique in the online experiment of this thesis due to the fact that the reliance on the relevance of the recommendation of collaborative filtering in the news domain is more challenging. As seen in Section 4.2, the performance of the models states that they, to some extent, have the ability to predict correctly but need to be investigated more for greater reliance.

The Bayesian analysis of the A/B testing showed that the challenger recommender based on BERT had a greater impact on females and ages over 45 than the former model. A goal of BT is to attract females and the younger age group more, as these are the two groups that typically show less amount of interactions compared to their opposite groups. A suggestion for future work is to analyse further why such recommendations by BERT are contributing to more clicks in general, as well as for the groups. This could make the decision-making process of finding the most optimal recommenders' choices to tailor to each group clearer. This may result in more interactions and contribute to a greater range of different users.

5.5 Open Science

To ensure openness and the possibility of future work, the code used to address the research of this thesis is accessible in a dedicated MediaFutures GitHub repository[1]. The repository contains code for all experiments, except for some missing concepts of Bergens Tidende due to confidentiality reasons, i.e., datasets and the implementation of the A/B test.

[1]https://github.com/sfimediafutures/MA_Peter-Kolbeinsen-Klingenberg

Appendix A

Appendix Experiment B3

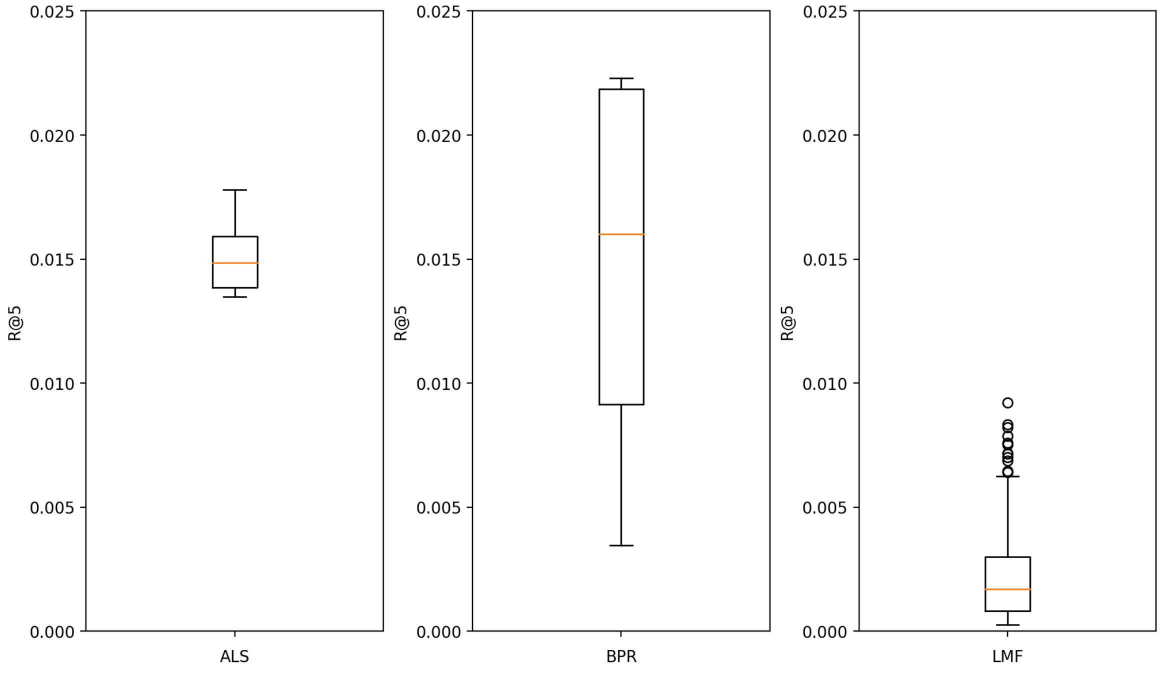


Figure A.1: Boxplot of all results of R@5 for hyperparameter tuning for ALS, BPR and LMF.

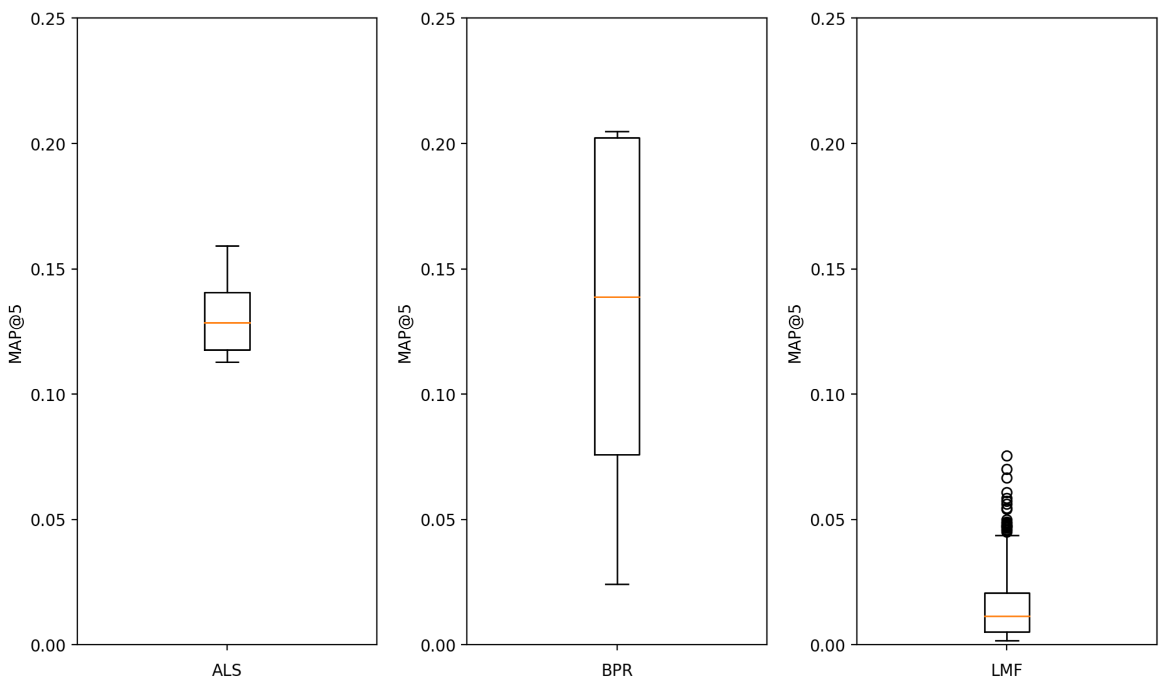


Figure A.2: Boxplot of all results of MAP@5 for hyperparameter tuning for ALS, BPR and LMF.

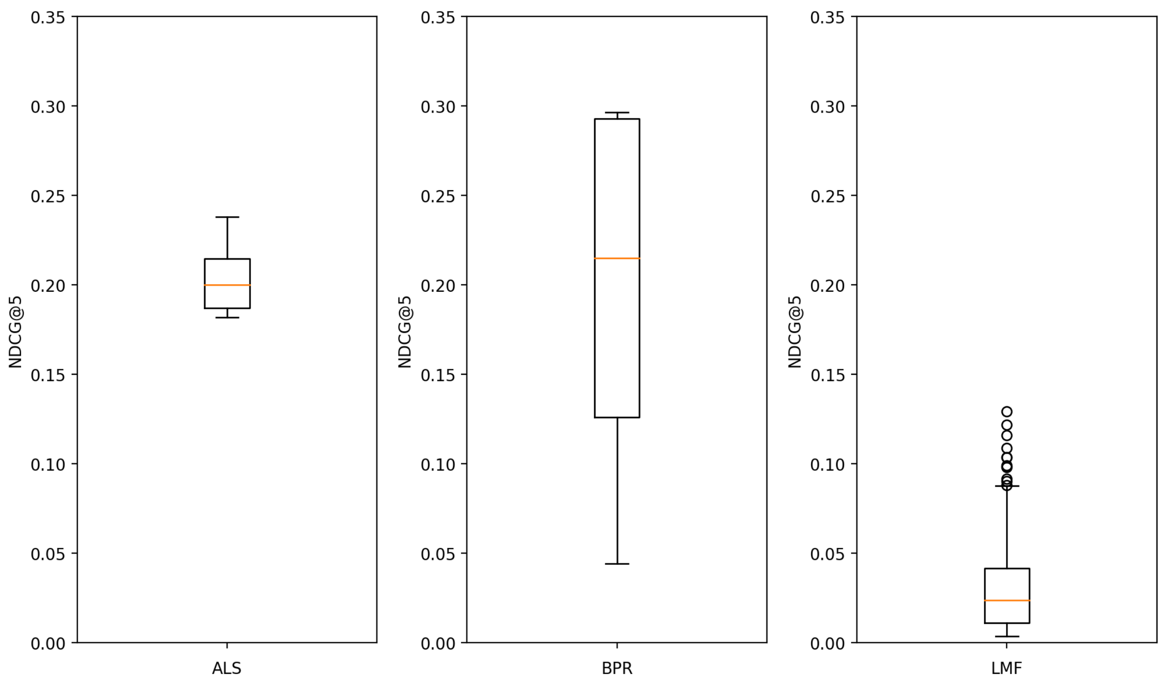


Figure A.3: Boxplot of all results of NDCG@5 for hyperparameter tuning for ALS, BPR and LMF.

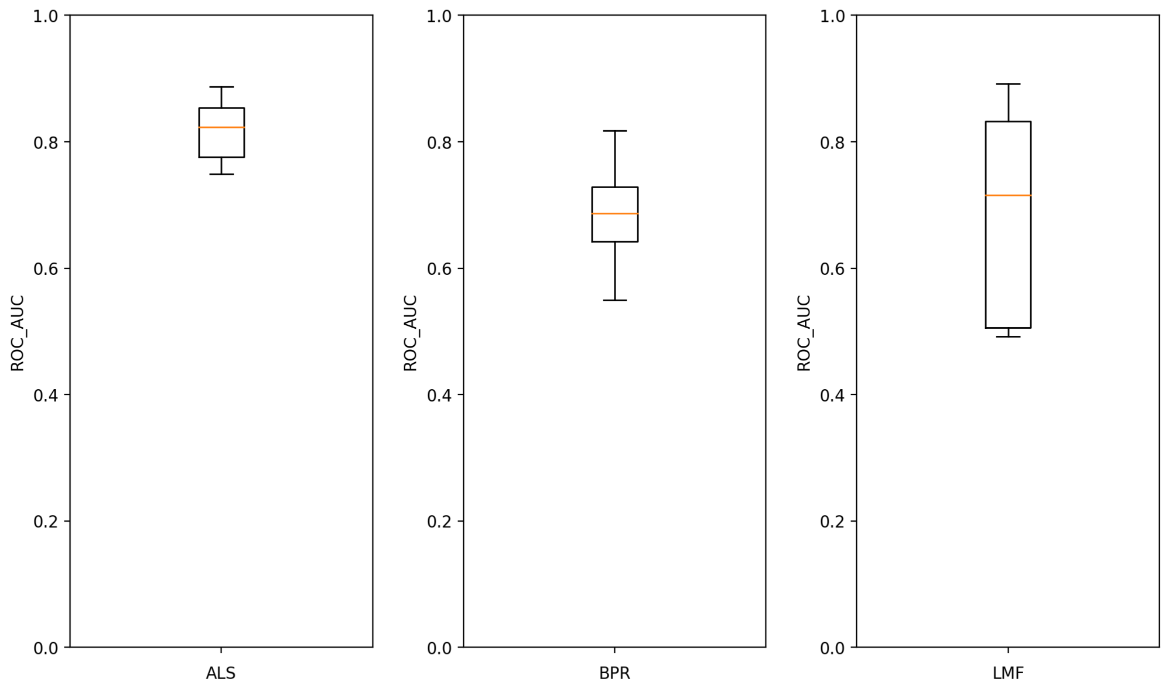


Figure A.4: Boxplot of all results of ROC_AUC for hyperparameter tuning for ALS, BPR and LMF.

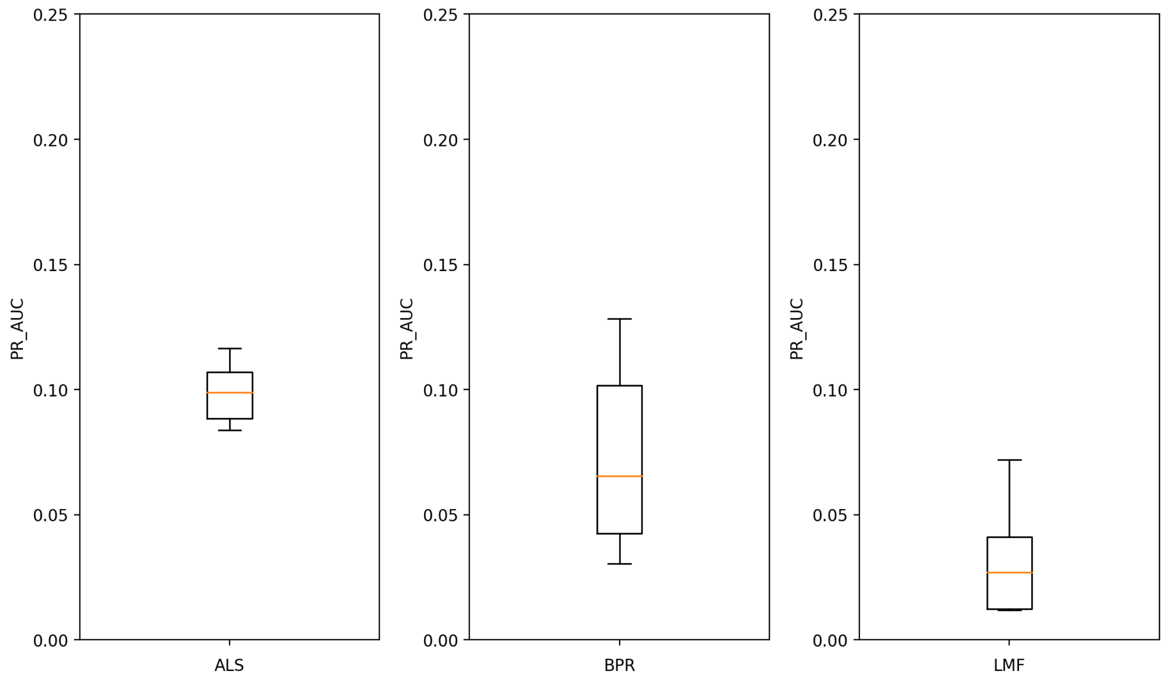


Figure A.5: Boxplot of all results of PR_AUC for hyperparameter tuning for ALS, BPR and LMF.

Bibliography

Abdollahpouri, H., E. C. Malthouse, J. A. Konstan, B. Mobasher, and J. Gilbert (2021), Toward the next generation of news recommender systems, in *Companion proceedings of the web conference 2021*, pp. 402–406. 2.1

Aggarwal, C. C., et al. (2016), *Recommender systems - The Textbook*, Springer. 1.1, 2.1, 2.1, 4.2

Bafna, P., D. Pramod, and A. Vaidya (2016), Document clustering: Tf-idf approach, pp. 61–66, doi:10.1109/ICEEOT.2016.7754750. 2.2

Bakhshandegan Moghaddam, F., and M. Elahi (2019), Cold start solutions for recommendation systems, in *Big Data Recommender Systems: Recent Trends and Advances*, p. 23, IET. 1.1, 2.1

Beheshti, A., S. Ghodratnama, M. Elahi, and H. Farhood (2022), *Social Data Analytics*, CRC Press. 1.1, 2.1

Braaten, R.-A., P. E. Kummervold, and J. d. l. Rosa (2022), Nb-sbert-base. 3.1, 3.1, 3.8.1

Chen, X., H. Xie, F. L. Wang, Z. Liu, J. Xu, and T. Hao (2018), A bibliometric analysis of natural language processing in medical research, *BMC medical informatics and decision making*, *18*(1), 1–14. 2.2

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2018), Bert: Pre-training of deep bidirectional transformers for language understanding, *arXiv preprint arXiv:1810.04805*. 2.2

Elahi, M. (2014), Empirical evaluation of active learning strategies in collaborative filtering, Ph.D. thesis, Ph. D. thesis, Ph. D. Dissertation. Free University of Bozen-Bolzano. 2.1

Elahi, M., M. Braunhofer, T. Gurbanov, and F. Ricci (2018), User preference elicitation, rating sparsity and cold start. 1.1

Elahi, M., A. Beheshti, and S. R. Goluguri (2021), Recommender systems: Challenges and opportunities in the age of big data and artificial intelligence, *Data Science and Its Applications*, pp. 15–39. 1.1

Elahi, M., D. Jannach, L. Skjærven, E. Knudsen, H. Sjøvaag, K. Tolonen, Ø. Holmstad, I. Pipkin, E. Throndsen, A. Stenbom, et al. (2022), Towards responsible media recommendation, *AI and Ethics*, pp. 1–12. 2.1

Fernández-Tobías, I., M. Braunhofer, M. Elahi, F. Ricci, and I. Cantador (2016), Alleviating the new user problem in collaborative filtering by exploiting personality information, *User Modeling and User-Adapted Interaction*, *26*, 221–255. 1.1

Feurer, M., and F. Hutter (2019), Hyperparameter optimization, *Automated machine learning: Methods, systems, challenges*, pp. 3–33. 3.5

Gelman, A., J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin (2013), *Bayesian data analysis*, CRC press. 3.8.2

Hu, Y., Y. Koren, and C. Volinsky (2008), Collaborative filtering for implicit feedback datasets, in *2008 Eighth IEEE international conference on data mining*, pp. 263–272, Ieee. 2.1, 3.4, 4.2

Jalili, M., S. Ahmadian, M. Izadi, P. Moradi, and M. Salehi (2018), Evaluating collaborative filtering recommender algorithms: a survey, *IEEE access*, *6*, 74,003–74,024. 3.4

Johnson, C. C. (2014), Logistic matrix factorization for implicit feedback data, *Advances in Neural Information Processing Systems*, *27*(78), 1–9. 3.4

Kamalbasha, S., and M. J. Eugster (2021), Bayesian a/b testing for business decisions, in *Data Science–Analytics and Applications: Proceedings of the 3rd International Data Science Conference–iDSC2020*, pp. 50–57, Springer. 3.8.2

Karimi, M., D. Jannach, and M. Jugovac (2018), News recommender systems–survey and roads ahead, *Information Processing & Management*, *54*(6), 1203–1227. 2.1

Kohavi, R., and R. Longbotham (2017), Online controlled experiments and a/b testing., *Encyclopedia of machine learning and data mining*, *7*(8), 922–929. 3.6

Koroteev, M. (2021), Bert: a review of applications in natural language processing and understanding, *arXiv preprint arXiv:2103.11943*. 2.2

Lops, P., D. Jannach, C. Musto, T. Bogers, and M. Koolen (2019), Trends in content-based recommendation: Preface to the special issue on recommender systems based on rich item descriptions, *User Modeling and User-Adapted Interaction*, *29*, 239–249. 1.1

Lü, L., M. Medo, C. H. Yeung, Y.-C. Zhang, Z.-K. Zhang, and T. Zhou (2012), Recommender systems, *Physics reports*, *519*(1), 1–49. 3.8.1, 4.2.1

Manning, C. D. (2009), *An introduction to information retrieval*, Cambridge university press. 3.8.1

Manwar, A., H. S. Mahalle, K. Chinchkhede, and V. Chavan (2012), A vector space model for information retrieval: a matlab approach, *Indian Journal of Computer Science and Engineering*, *3*(2), 222–229. 3.2

Nwagwu, W. (2022), The rise and rise of natural language processing research, 1958-2021, doi:10.21203/rs.3.rs-2265814/v1. 2.2

O'Mahony, M. P., N. J. Hurley, and G. C. Silvestre (2006), Detecting noise in recommender system databases, in *Proceedings of the 11th international conference on Intelligent user interfaces*, pp. 109–115. 4.2.2

Rahutomo, F., T. Kitasuka, and M. Aritsugi (2012), Semantic cosine similarity, in *The 7th international student conference on advanced science and technology ICAST*, vol. 4, p. 1. 3.2

Ravichandiran, S. (2021), *Getting Started with Google BERT: Build and train state-of-the-art natural language processing models using BERT*, Packt Publishing Ltd. 2.2

Raza, S., and C. Ding (2022), News recommender system: a review of recent progress, challenges, and opportunities, *Artificial Intelligence Review*, pp. 1–52. 1.2, 2.1, 2.1

Reimers, N., and I. Gurevych (2019), Sentence-bert: Sentence embeddings using siamese bert-networks, *arXiv preprint arXiv:1908.10084*. 2.2, 3.1

Rendle, S., C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme (2012), Bpr: Bayesian personalized ranking from implicit feedback, *arXiv preprint arXiv:1205.2618*. 3.4

Ricci, F., L. Rokach, and B. Shapira (2010), Introduction to recommender systems handbook, in *Recommender systems handbook*, pp. 1–35, Springer. 1.1

Schedl, M., H. Zamani, C.-W. Chen, Y. Deldjoo, and M. Elahi (2018), Current challenges and visions in music recommender systems research, *International Journal of Multimedia Information Retrieval*, *7*, 95–116. 3.8.1

Shin, D. (2020), How do users interact with algorithm recommender systems? the interaction of users, algorithms, and performance, *Computers in Human Behavior*, *109*, 106,344. 2.1

Van der Maaten, L., and G. Hinton (2008), Visualizing data using t-sne., *Journal of machine learning research*, *9*(11). 3.8.1

Williams, A., N. Nangia, and S. Bowman (2018), A broad-coverage challenge corpus for sentence understanding through inference, in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122, Association for Computational Linguistics. 3.1

Wolf, T., L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. Rush (2020), Transformers: State-of-the-art natural language processing, in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Association for Computational Linguistics, Online, doi: 10.18653/v1/2020.emnlp-demos.6. 3.1

Yang, L., and A. Shami (2020), On hyperparameter optimization of machine learning algorithms: Theory and practice, *Neurocomputing*, *415*, 295–316. 3.5

Zhou, T., L. Lü, and Y.-C. Zhang (2009), Predicting missing links via local information, *The European Physical Journal B*, *71*, 623–630. 3.8.1