Enhancing the NPO Start Recommendation System with Metadata

submitted in partial fulfillment for the degree of master of science

Eileen Kapel
12369462

master information studies
data science
faculty of science
university of amsterdam

YYYY-MM-DD

	External Supervisor	External Supervisor	Second External Supervisor
Title, Name	Dr Maarten Marx	Arno van Rijswijk	Robbert van Waardhuizen
Affiliation	UvA, FNWI, IvI	NPO	NPO
Email	maartenmarx@uva.nl	arno.van.rijswijk@npo.nl	robbert.van.waardhuizen@npo.nl

Universiteit van Amsterdam Amsterdam Data Science nederlandse publieke omroep

ABSTRACT

...

KEYWORDS

Recommendation systems; Video recommendation systems; Metadata

1 INTRODUCTION

Humans are tasked with making thousands of decisions daily that may range from selecting what outfit to wear to which television series to watch. Recommendations can make the process of these decisions more efficient by sorting through potentially relevant information and making recommendations customised to the individual user [10, 13]. One system that employs recommendations is NPO Start[1]. It is a video-on-demand service from the NPO (Dutch Public Service Media) that gives people the ability to watch series, movies and documentaries as well as watch live television online. Personalised recommendations are available for registered users of the service, which is based on the collaborative filtering method. However, there is a lot of metadata available about the offered content that is unused in this current method. In this thesis, the metadata of broadcasts will be utilised to determine if it can improve the performance of the current video recommendation system.

This is achieved by answering the main research question: *'Can a hybrid recommendation system using metadata perform better than the current recommendation system of NPO Start which uses collaborative filtering?'*. The research is split into three sub-questions:

- **RQ.1** What is the performance of the NPO Start recommendation system?
- **RQ.2** Which metadata features improve the performance of the hybrid recommendation system the most?
- **RQ.3** Can the performance of the NPO Start recommendation system be improved by implementing a hybrid recommendation system?

The thesis is structured as follows: first, some background information on the NPO Start service is provided in section 2. Various related works of literature are outlined in section 3 that relate to the goal of the thesis. The methodology employed during the research is discussed in section 4 and is followed by the results in section 5. Subsequently, the conclusions of these results are presented in section 6 after which a discussion of the choices and possible future work follows in section 7.

2 BACKGROUND

NPO Start is a service that offers users the ability to watch video content on demand. This video content is displayed to users in so-called "ribbons" or rows that have a certain theme, like 'Populair', 'Nieuw' and 'Aanbevolen voor jou'. Each ribbon consists of a ranked list of several items and an item represents a series that can be streamed.

[1] www.npostart.nl

2.1 The NPO Start Recommendation System

Users of the service that have an account have the ability to receive several personalised ribbons that contain items that are recommended to a specific user. These recommendations are materialised on the front page of the service in the two ribbons 'Aanbevolen voor jou' en 'Probeer ook eens'. This thesis focuses on the 'Aanbevolen voor jou' ribbon. The recommendations for this ribbon are produced by a collaborative filtering approach that utilises the history of user interaction information with items. These user interactions are grouped on series level and evaluated by pairs of series that are frequently watched together, or coincide often, with the history of the user. Of these coincidences, the top 100 pairs are extracted after they are ordered based on their frequency. A sliding window technique of the past 21 days is used for the interaction information, which is based on the intuition that recent events are more relevant than older ones [1]. These recommendation lists are updated hourly at peak times and bi-hourly off-peak.

3 RELATED LITERATURE

Recommendations are based on ratings that are explicitly given by users or ratings that are implicitly inferred from users their actions [13], like a click or a certain watch duration. There are three main approaches for building a system that gives out recommendations. The first approach utilises information about items for a content-based system and recommends items that are similar to the well-rated items. The second approach utilises users their interaction information with items for a collaborative filtering system and recommends items that are well-rated by other users who have the same pattern of ratings. Lastly, there is a hybrid recommendation system, which is a combination of the two previous approaches, that exploits item information and interaction information to provide recommendations. The first two approaches each have their own shortcomings, like overspecialisation, rating sparsity and cold-start [1, 8], that hybrid systems aim to overcome to provide more accurate recommendations.

In this section, an overview of the current state of hybrid recommendation research is given. Furthermore, a few personalised services that employ recommendation systems are described and, lastly, the representation of features in recommendation systems is touched upon.

3.1 Hybrid Recommendation Systems

Hybrid recommendation systems are able to deal with scenarios where there is little data on new users and items, also called the cold-start problem, or when there is sparse interaction data available. This is achieved by joining content and collaborative data in the system to produce recommendations that not only takes into account similar users but also personal interests.

Several techniques exist for combining content-based and collaborative filtering systems of which the weighted, mixed, switching and feature combination techniques [3] are most frequently used. A weighted hybrid recommendation system is one where the rating of an item is a combination of the content-based and collaborative rating. Alternatively, a mixed hybrid recommendation system outputs items from the different approaches together. The switching recommendation system uses a different approach dependent on the situation, for example a content-based system could be used when there is little interaction information and in other cases a collaborative filtering system is used. Finally, the feature combination technique combines both content-based and collaborative information into a single recommendation algorithm. This technique causes the recommendation system to rely less on the amount of ratings per item and allows for less-known but similar items to be recommended.

Furthermore, different algorithms can be employed in the hybrid recommendation techniques, however most techniques employ matrix factorisation for the collaborative filtering part of the system. This algorithm was popularised by the solution of the Netflix Prize competition that employed matrix factorisation using the alternating least square (ALS) algorithm [2, 6, 7]. Some works that have successfully employed matrix factorisation in their hybrid recommendation systems are Rubtsov et al. [14], Ludewig et al. [11] and Al-Ghossein et al. [1]. Rubtsov et al. used the feature combination technique by making use of the LightFM library, which is a Python implementation of a matrix factorisation model that can deal with user and item information [8], paired with a weighted approximate-rank pairwise loss. Ludewig et al. made use of matrix factorisation in their model by combining it with a k-nearest-neighbour technique and using the ALS algorithm. Content was incorporated into the model by weighing the matrix factorisation results with the IDF (inverse document frequency) score of titles to produce the final list of recommendations. Lastly, the feature combination hybrid recommendation system by Al-Ghossein et al. merged matrix factorisation with topics extracted using topic modeling for online recommendation.

Furthermore, neural networks have also been combined into hybrid recommendation systems due to its recent popularity. Volkovs et al. [16] produced a two stage model for automatic playlist continuation that first employs weighted regularised matrix factorisation to retrieve a subset of candidates and then uses convolutional neural networks and neighbour-based models for detecting similar patterns. In the second stage features of playlists and songs are combined with the items after which the final ranking of recommendations are produced. Another novel approach for automatic playlist continuation is the weighted hybrid recommendation system made up of a content-aware autoencoder and a character-level convolutional neural network (charCNN) by Yang et al. [18]. The content-aware autoencoder alternates in predicting artists fitting in a playlist and playlists fitting with an artist. The charCNN takes a sequence of characters as input, in this case a playlist title, and predicts the most fitting tracks with this sequence. The output of both components was linearly combined and produced the final recommendations.

3.2 Personalised services

Recommendation systems are frequently employed by services to provide a personalised experience to their users. This occurs in different domains, e.g. product recommendation by Amazon, music recommendation by Spotify and video recommendation by YouTube and Netflix.

Netflix is a service where the whole experience is defined by personalisation [4]. This is primarily showcased on its homepage which consists of rows of recommended videos with a similar theme, like ‘Top Picks‘ and ‘Trending Now‘, that are ranked by a personalised video ranker. Two of these rows, namely the genre and because you watched rows, take the content of the videos in account for the recommendations. The videos in the genre row are produced by a single algorithm that takes a subset of all videos that corresponds to a specific genre. Examples of such rows are ‘Suspenseful Movies‘ and ‘Romantic TV Movies‘. The because you watched row bases its recommendations on a single video that is watched by a user and uses a video-video similarity algorithm. This algorithm is not personalised, but the choice of which because you watched rows are offered to a user is personalised. Examples of this kind of row are ‘Because you watched Black Mirror‘ and ‘Because you watched Stranger Things‘.

The streaming service Spotify employs personalisation in several areas, like its homepage and the feature that allows for automatic playlist continuation. The homepage allows users to discover new playlists which are similar to the playlists and tracks a user has previously interacted. The automatic playlist continuation feature adds one or more tracks to a playlist that fits the original playlist of a user [19]. This takes into account the collaborative information of playlists and their corresponding tracks, but also their content in the form of titles and featured artists.

3.3 Representation of Features

Multimedia content, like songs, films and series, is often represented by a set of features. A feature is information that describes an attribute of an item, like its title, plot, genre, or release year.

These features are used in content-based recommendation systems and thus also in hybrid recommendation systems. The performance of such systems is predicated on the quality of the features, so features derived from high-quality metadata lead to a better performance [8, 14].This is evident in the work of Soares & Viana [15] where the version of their recommendation system that used more granular metadata as features, e.g. genres and sub-genres, resulted in recommendations of a higher quality. If high-quality metadata is not available then good quality metadata can be obtained from item descriptions, like actor lists and synopses [8]. However, metadata of a lower quality, e.g. by being sparse, may result in overfitting and causes models to not make use of content in an effective way [19].

Feature selection is frequently used to improve the quality of metadata. It is a method where rather than using all the features only a subset of the features is used [12]. By carefully selecting this subset of features a better effectiveness of the system is achieved. For example, the hybrid recommendation system by Soar et al. that only employed the director as feature, instead of all the features, resulted in recommendations that were more precise [15]. This is likely explained by the fact that a director can provide specific information on the potential quality of content that cannot be described with another set of metadata elements, e.g. actors may participate in movies with different ratings but ratings of movies by the same directors are more similar. A single feature can also be made more precise by taking advantage of mutual information [12], e.g. using frequent words or information retrieval methods like TF (term frequency), DF (document frequency), and TF-IDF (term frequency-inverse document frequency). This was employed by the hybrid recommendation system of Rubtsov et al. which used the top-2000 most frequent words in titles as a feature in their recommendation system opposed to all the words [14].

4 METHODOLOGY

This section describes the methodology employed for answering the research questions. First, the hybrid recommendation model is outlined after which a description of the data that is provided to this model follows. Furthermore, the metrics for evaluation the performance of the recommendation system are presented and the experimental setup is described.

4.1 The Hybrid Recommendation Model

The hybrid recommendation model uses a feature combination technique and consists of a matrix factorisation model that incorporates item information. The model is implemented using the LightFM library [8], which is a Python implementation of a matrix factorisation model that can deal with user and item information. This model acts as a standard matrix factorisation model when no user or item information is provided.

The LightFM model represents each user and/or item as a combination of features their latent representations. For example, the representation of the series Beste Zangers is a combination of the representation of the genre music, the genre amusement and the broadcaster AVROTROS. The latent representation approach is utilised in the hybrid model for each item, so if the genre music and the genre amusement are both liked by the same users then their embeddings will be close together; if both genres are never liked by the same users then their embeddings will be far apart. The dimensionality of these latent feature embeddings can be optionally adjusted in the model.

The LightFM library offers two types of loss functions for implicit feedback learning-to-rank: the WARP (Weighted Approximate-Rank Pairwise) loss [17] and BPR (Bayesian Personalised Ranking) loss. The documentation of the LightFM model states that the WARP loss typically performs better than BPR so this function has been chosen for the implementation of the hybrid recommendation model. The WARP loss samples a negative item for each user and positive item pair and computes predictions for both positive and negative items. A gradient update is performed if the prediction of the negative item is valued higher than that of the positive item, otherwise negative items are continuously sampled until a higher negative prediction does occur.

The execution of the LightFM model can be sped up by making use of the offered multi-threading during training, prediction and evaluation of the model [9]. This can however lead to a decrease in accuracy when the interaction matrix is dense, but does not lead to a measurable loss of accuracy when a sparse data set is trained.

4.2 The Data

The input data for the recommendation systems consists of interaction information and content features.

4.2.1 Interaction Information. The first data set consists of interaction information that is provided by the event data of the NPO. The event data describes all interactions that users have had with the NPO Start service, e.g. clicks, stream starts and refreshes. This data was pre-processed to only gather interaction information of users that have watched at least half the duration of episodes of a series.

4.2.2 Content Features. The Publieke Omroep Media Service (POMS) contains information about all content that is offered by the NPO, which ranges from broadcasts and movies to podcasts, and amounts to a total of about 1.5 million media items. Each item consists of 37 columns describing metadata, e.g. a media id (mid), age rating, broadcaster, credits, descriptions, genres, images, locations, etc. These items were processed to only gain broadcasts which are available to stream on the NPO Start service, resulting in about 84 thousand broadcasts. Each broadcast has its own media id and a series reference which refers to the series this broadcast is a part of. A total of 2490 series were identified and a series may consist of a lot of individual broadcasts, a couple broadcasts or a single broadcast. The series NOS Journaal is an example of a series that consists of a lot of individual broadcasts since it has 10.297 broadcasts. Out of all the metadata, six metadata features were selected to be used in the content-based part of the recommendation system, namely broadcaster, credits, description, genres, subtitles and title. These features are described in Appendix A Table 3. The POMS data is grouped per series and aggregated based on unique values per item. All metadata is provided by program makers and can differ in completeness and detail. The percentage series with missing values for the content features is displayed in Figure 1 to investigate the completeness. This shows that the features broadcaster and title are complete for all series, however some are missing information about its description and genres. Furthermore, 40% of the series do not have information about credits and subtitles, which amounts to about 1000 series.

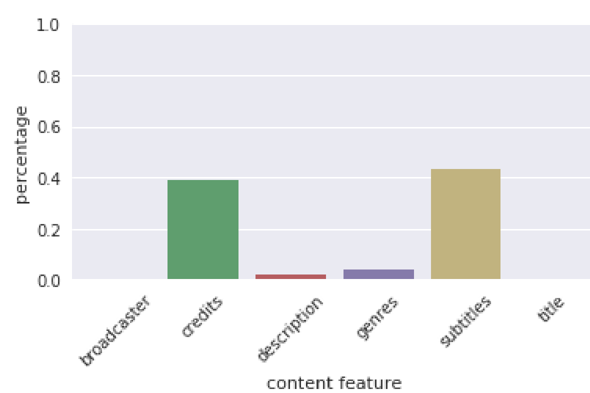


Figure 1: Percentage Series with Missing Values for the Content Features

Half of the six content features consists of categorical features and the other half are textual features. The three content features broadcaster, credits and genres are categorical. The other features title, description and subtitles are textual.

Categorical features. A total of 30 unique broadcasters are identified for the series and each series may have one or several broadcasters associated with it. The percentage of how often a broadcaster is associated with a series is shown in Figure 2. Most series are broadcasted by the VPRO and other frequent broadcasters are the NTR, AVTR (AVROTROS) and BNVA (BNNVARA).

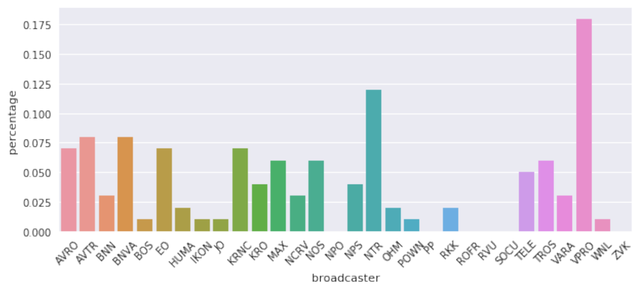


Figure 2: Percentage Series with Broadcaster

5383 unique credits were identified and a series either has multiple credits, one credit or no credits associated with it. A single person is often accredited in a single series and sometimes in several. However, there are a few people, like Sophie Hilbrand, Tom Egbers and Astrid Kersseboom, that are accredited more than ten times. It should be noted that the credits are dirty, since a big portion of the series does not include this feature and the amount of accredited people per series can range from hundreds of people to only one person.

Series either have multiple genres, one genre or no genre assigned to them. A total of 53 different genres were identified and the percentage of how often a genre is associated with a series is shown in Appendix A Figure 5. Genres have a main type indicated by an id of four integers, e.g. 3.0.1.7 'Informatief', and may have sub-types which are indicated by an identifier of five integers, e.g. 3.0.1.6.26 'Informatief, Religieus'. About 30% of the series have the genre 'Informatief' which amounts to about 750 series. Other frequent genres for series are 'Amusement', 'Jeugd' en 'Documentaire', which each occur about 13%.

Textual features. Additionally, there are three textual features, namely title, description and subtitles. The average word count of these features and their median per series is described in Table 1. The word count only includes series that have data for that particular feature. The feature title has the lowest mean word count, followed by description and subtitles has the highest. All the medians of the features lie below the mean, indicating that the length of that particular feature is not evenly distributed. The distribution of the title, description and subtitles word count is displayed in Figure 3 on a logarithmic scale. The three distributions all indicate a skewed distribution to the left, which means that a big portion of the series has textual features with a low word count. However there are a couple of series that have big values. Also, as mentioned before, the data is grouped per series and aggregated based on unique value per item. Since some series are made up of a big amount of broadcasts and metadata can differ in detail this can mean that these series may contain more words than other series. This is particularly apparent in the spikes that are visible in the description plot of Figure 3.

Table 1: Mean and Median Word Count for the Textual Features

Feature	Mean	Median
Title	8.2	4.0
Description	391.9	76.0
Subtitles	58951.1	12434.0

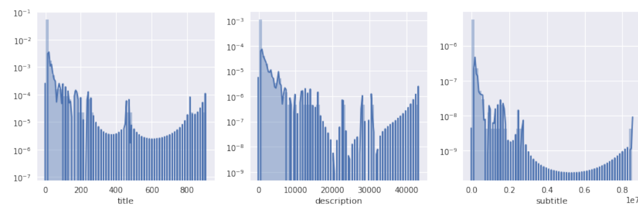


Figure 3: Distributions of Word Count for the Textual Features Length

4.3 Feature Encoding

The interaction information and content features were pre-processed before being provided to the hybrid recommendation model.

4.3.1 Interaction information. The interaction information was processed into (user, series) pairs and transformed into a user interaction matrix.

4.3.2 Content features. For the textual features, some language processing was employed during pre-processing. This pre-processing consisted of lowercasing and tokenising the words for the features. Afterwards, the punctuation, tokens smaller than four letters and dutch & english stopwords were removed. Finally, TF-IDF was performed where each series was regarded as a text and the whole set of series as a document. Feature selection is employed to improve the quality of these content features. For the title, the three words with the highest TF-IDF were extracted per series. For the description, the top ten and for the subtitles, the top 20.

Afterwards, all content features were exploded into (series, feature value) pairs and one-hot-encoded using scikit-learn's DictVectorizer class [5]. This produces a dictionary for each series, where the key is the feature value and the weights are the values, and this is transformed into an item information matrix. The amount of unique feature values for each content features is displayed in Table 2.

4.4 Evaluation

The performance of a recommendation system is assessed by the quality of recommendations. The quality is evaluated by the two

Table 2: Amount of Unique Feature Values per Content Feature

Content Features	Amount of Unique Feature Values
broadcaster	28
genres	53
credits	5689
title	3252
description	13396
subtitles	14210

metrics mean precision@k (mean p@k) and mean reciprocal rank (MRR).

4.4.1 Mean Precision@k. Mean precision@k is a metric that evaluates the average proportion of top-k recommended items that are relevant to users. A relevant item is an item that was chosen by a user when it was offered in a ribbon. Relevant items are denoted as a true positive (TP) which are positive predicted values. The precision is thus denoted as the total number of predicted positives out of all predicted items. The equation for the precision@k is shown in equation 1.

$$P@k = \frac{|\{i \in TP \mid i \text{ ranked in top k}\}|}{k} \tag{1}$$

The precision@k is evaluated over all recommendations and averaged into the mean precision@k (see below) to evaluate the overall quality of the system (see equation 2).

$$MeanP@k = \frac{\sum_{n=1}^{N} P@k(n)}{N} \tag{2}$$

4.4.2 Mean Reciprocal Rank. Mean reciprocal rank is a metric that evaluates the average ranking quality of lists that a model produces. It evaluates how successful the model is in ranking the highest relevant item to users. It is calculated by dividing the best possible rank by the actual rank of the first relevant items and averaging it (see equation 3).

$$MRR = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{rank_i} \tag{3}$$

The higher the value of the metrics, the better. The version with the highest mean precision has the most success of recommending items that users are interested in, and the version with the highest MRR is most successful in ranking the highest relevant item in a personalised manner.

4.5 The Experimental Setup

The experimental setup consists of three parts that each correspond to a research question.

Interaction information was split into a train and test set for the experimental setup. For the train set, interaction information for a period of 21 days was used, which corresponds to the sliding window described in section 2.1. The following day after the train period was used for testing of the recommendation systems. The period 1 to 21 March 2019 was used as train set, which corresponds to a total of 1.192.556 interactions. The following day (22 March 2019) was used for testing and consists of 41.538 interactions. The sparsity of the total interaction information is 0.22%. An additional test set was constructed out of the original test set, called the "recommended test set", which consists of the interactions that happened on the top-k series that were actually recommended in the 'Aanbevolen voor jou' ribbon on the NPO Start service. The evaluation is performed on a k of 5, since that is the mean amount of items that is typically visible on a ribbon of the NPO Start service. Interactions of users that were present in the test sets but not in the train set were removed from the experiment.

4.5.1 RQ1. The experimental setup of the NPO Start recommendation system is shown in Figure 4a. It starts with the above described train set and the recommended test set, since there is only rank information available about series that were actually offered in the 'Aanbevolen voor jou' ribbon. The train set was supplied to the current model as described in section 2.1 and predictions were evaluated against the test set using the performance metrics.

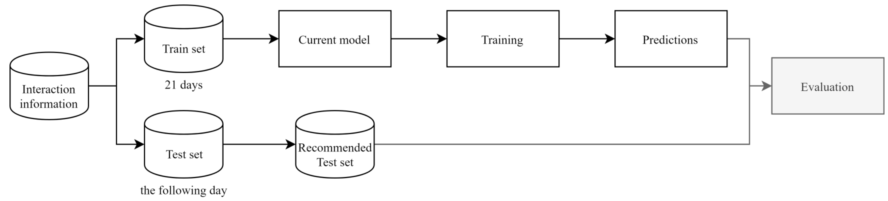
4.5.2 RQ2. The experimental setup of the hybrid recommendation system is shown in Figure 4b. Similar to the NPO Start recommendation system, the hybrid recommendation system starts with the same train and test set, however it uses both test sets for evaluation. The interaction matrix of the train set and the feature matrix of the content features waas supplied to the hybrid recommendation model described in section 4.1. The model used these two matrices for training and serving out ranked item predictions for each user present in the test sets. Lastly, the performance of the hybrid recommendation system was evaluated against the test sets of interaction information using the performance metrics.

The described experimental setup was used for performing 64 different experiments on the hybrid recommendation. Each experiment used a different set of content features that were incorporated into the hybrid model and then trained to produce predictions. The first experiment acts as a baseline that incorporates no content features with the train set and the other 63 different experiments use a different combination of the six content features (see Table 3). These combinations start with a single content feature, go to combinations of two content features and end with a combination that incorporates all content features. For example, experiment 16 incorporates the description and genres features into the hybrid model. Each experiment model was trained on a range from 0 to 100 epochs with a step size of 10 on standard settings and its predictions were evaluated against both test sets to investigate the learning curve of each model. Afterwards, the experiment model that accomplished the highest performance is compared to the baseline model.

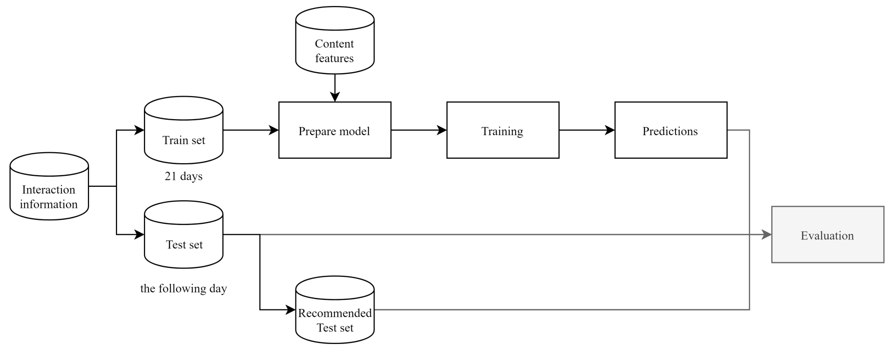
4.5.3 RQ3. The last part of the experimental setup compares the performance of the NPO Start to that of the hybrid recommendation system.

The current recommendation system uses the same experimental setup described above in section 4.5.1.

The hybrid recommendation system uses the experimental setup for the experiment model that accomplished the highest performance as described above in 4.5.2. The hyperparameters of this model were optimised using a tree based regression model from scikit-optimize library [5], which allows for finding the optimal



(a)



(b)

Figure 4: (a) The NPO Start Recommendation System (b) The Hybrid Recommendation System

hyperparameters to maximise model performance. The optimal hyperparameters were then used for the execution of the model.

Afterwards, the performance metrics of both systems are compared to one another.

5 RESULTS

5.1 RQ1: What is the performance of the current recommendation system?

5.2 RQ2: Which metadata features improve the performance of the hybrid recommendation system the most?

5.3 RQ3: Can the performance of the current recommendation system be improved by implementing a hybrid recommendation system?

6 CONCLUSIONS

7 DISCUSSION

7.1 The Data

7.2 Methodology

7.3 Results

7.4 Conclusions

ACKNOWLEDGMENTS

...

REFERENCES

[1] Marie Al-Ghossein, Pierre-Alexandre Murena, Talel Abdessalem, Anthony Barré, and Antoine Cornuéjols. 2018. Adaptive collaborative topic modeling for online recommendation. *Proceedings of the 12th ACM Conference on Recommender Systems* (2018), 338–346.

[2] Robert Bell, Yehuda Koren, and Chris Volinsky. 2007. Modeling relationships at multiple scales to improve accuracy of large recommender systems. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 95–104.

[3] Robin Burke. 2002. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction* 12, 4 (2002), 331–370.

[4] Carlos A Gomez-Uribe and Neil Hunt. 2016. The netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems (TMIS)* 6, 4 (2016), 13.

[5] Tim Head, MechCoder, Gilles Louppe, Iaroslav Shcherbatyi, fcharras, Zé Vinícius, cmmalone, Christopher Schröder, nel215, Nuno Campos, Todd Young, Stefano Cereda, Thomas Fan, rene rex, Kejia (KJ) Shi, Justus Schwabedal, carlosdanielcsantos, Hvass-Labs, Mikhail Pak, SoManyUsernamesTaken, Fred Callaway, Loïc Estève, Lilian Besson, Mehdi Cherti, Karlson Pfannschmidt, Fabian Linzberger, Christophe Cauet, Anna Gut, Andreas Mueller, and Alexander Fabisch. 2018. scikit-optimize/scikit-optimize: v0.5.2. (March 2018). https://doi.org/10.5281/zenodo.1207017

[6] Yehuda Koren. 2009. The bellkor solution to the netflix grand prize. *Netflix prize documentation* 81, 2009 (2009), 1–10.

[7] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009), 30–37.

[8] Maciej Kula. 2015. Metadata Embeddings for User and Item Cold-start Recommendations. In *Proceedings of the 2nd Workshop on New Trends on Content-Based Recommender Systems co-located with 9th ACM Conference on Recommender Systems (RecSys 2015), Vienna, Austria, September 16-20, 2015. (CEUR Workshop Proceedings)*, Toine Bogers and Marijn Koolen (Eds.), Vol. 1448. CEUR-WS.org, 14–21. http://ceur-ws.org/Vol-1448/paper4.pdf

[9] Maciej Kula. 2016. Welcome to LightFM's documentation! (2016). https://lyst.github.io/lightfm/docs/index.html

[10] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. 2011. Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*. Springer, 73–105.

[11] Malte Ludewig, Iman Kamehkhosh, Nick Landia, and Dietmar Jannach. 2018. Effective Nearest-Neighbor Music Recommendations. In *Proceedings of the ACM Recommender Systems Challenge 2018*. ACM, 3.

[12] Christopher Manning, Prabhakar Raghavan, and Hinrich Schütze. 2010. Introduction to information retrieval. *Natural Language Engineering* 16, 1 (2010), 100–103.

[13] Michael J Pazzani. 1999. A framework for collaborative, content-based and demographic filtering. *Artificial intelligence review* 13, 5-6 (1999), 393–408.

- [14] Vasiliy Rubtsov, Mikhail Kamenshchikov, Ilya Valyaev, Vasiliy Leksin, and Dmitry I Ignatov. 2018. A hybrid two-stage recommender system for automatic playlist continuation. In *Proceedings of the ACM Recommender Systems Challenge 2018*. ACM, 16.
- [15] Márcio Soares and Paula Viana. 2015. Tuning metadata for better movie content-based recommendation systems. *Multimedia Tools and Applications* 74, 17 (2015), 7015–7036.
- [16] Maksims Volkovs, Himanshu Rai, Zhaoyue Cheng, Ga Wu, Yichao Lu, and Scott Sanner. 2018. Two-stage model for automatic playlist continuation at scale. In *Proceedings of the ACM Recommender Systems Challenge 2018*. ACM, 9.
- [17] Jason Weston, Samy Bengio, and Nicolas Usunier. 2011. Wsabie: Scaling up to large vocabulary image annotation. In *Twenty-Second International Joint Conference on Artificial Intelligence*.
- [18] Hojin Yang, Yoonki Jeong, Minjin Choi, and Jongwuk Lee. 2018. Mmcf: Multimodal collaborative filtering for automatic playlist continuation. In *Proceedings of the ACM Recommender Systems Challenge 2018*. ACM, 11.
- [19] Hamed Zamani, Markus Schedl, Paul Lamere, and Ching-Wei Chen. 2018. An Analysis of Approaches Taken in the ACM RecSys Challenge 2018 for Automatic Music Playlist Continuation. *Proceedings of the 12th ACM Conference on Recommender Systems* (2018), 527–528.

A THE DATA

Table 3: Overview of the Content Features

Feature	Type	Description
broadcaster	categorical	Broadcaster of the broadcast, e.g. NOS.
credits	list	The people accredited in the broadcast, such as presenters or guests.
description	string	Description of the broadcast. This is either the main description, otherwise the short description or the kicker.
genres	list	Genres of the broadcast denoted by a genre id and name, e.g. (3.0.1.6, [Amusement]).
subtitles	string	The subtitles of the broadcast, which were extracted using the POMS subtitles API.
title	string	The main title of the broadcast.

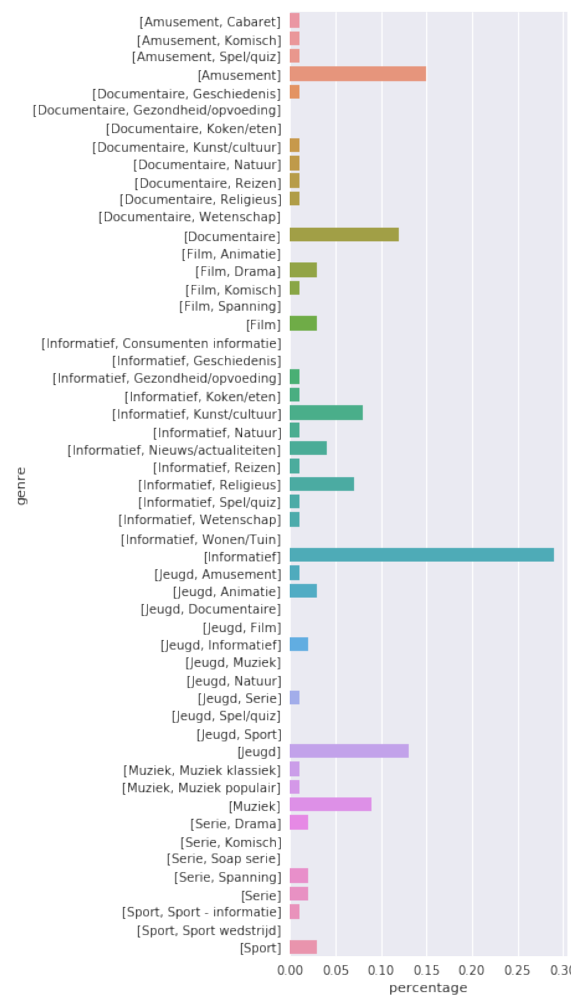


Figure 5: Percentage Series with Genres

Table 4: The Content Features Combinations

Index	Feature
0	none
1	broadcaster
2	credits
3	description
4	genres
5	title
6	subtitles
7	broadcaster, credits
8	broadcaster, description
9	broadcaster, genres
10	broadcaster, title
11	broadcaster, subtitles
12	credits, description
13	credits, genres
14	credits, title
15	credits, subtitles
16	description, genres
17	description, title
18	description, subtitles
19	genres, title
20	genres, subtitles
21	title, subtitles
22	broadcaster, credits, description
23	broadcaster, credits, genres
24	broadcaster, credits, title
25	broadcaster, credits, subtitles
26	broadcaster, description, genres
27	broadcaster, description, title
28	broadcaster, description, subtitles
29	broadcaster, genres, title
30	broadcaster, genres, subtitles
31	broadcaster, title, subtitles
32	credits, description, genres
33	credits, description, title
34	credits, description, subtitles
35	credits, genres, title
36	credits, genres, subtitles
37	credits, title, subtitles
38	description, genres, title
39	description, genres, subtitles
40	description, title, subtitles
41	genres, title, subtitles
42	broadcaster, credits, description, genres
43	broadcaster, credits, description, title
44	broadcaster, credits, description, subtitles
45	broadcaster, credits, genres, title
46	broadcaster, credits, genres, subtitles
47	broadcaster, credits, title, subtitles
48	broadcaster, description, genres, title
49	broadcaster, description, genres, subtitles
50	broadcaster, description, title, subtitles
51	broadcaster, genres, title, subtitles
52	credits, description, genres, title
53	credits, description, genres, subtitles
54	credits, description, title, subtitles
55	credits, genres, title, subtitles
56	description, genres, title, subtitles
57	broadcaster, credits, description, genres, title
58	broadcaster, credits, description, genres, subtitles
59	broadcaster, credits, description, title, subtitles
60	broadcaster, credits, genres, title, subtitles
61	broadcaster, description, genres, title, subtitles
62	credits, description, genres, title, subtitles
63	broadcaster, credits, description, genres, title, subtitles