Enhancing the Recommendations of NPO Start 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

2019-06-28

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

Amsterdam
Data Science

nederlandse
publieke
omroep

ABSTRACT

The current recommendation system of the NPO Start service uses a collaborative filtering approach for serving out video recommendations. However, a lot of available metadata is unused in this system that can be utilised for providing recommendations. This thesis aims to determine whether a hybrid recommendation system that utilises this metadata can perform better than the NPO Start recommendation system. Based on experiments where interaction information and different combinations of metadata features were supplied to a hybrid LightFM model, the model that utilised the metadata features broadcaster, description, genres and title performed similar but slightly higher than the model utilising no metadata features. After the hyperparameters of the former model were optimised and compared to the current model of the NPO Start recommendation system, it was concluded that a hybrid recommendation system using metadata can perform better than the current recommendation system of NPO Start.

KEYWORDS

Recommendation systems; Hybrid 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 a collaborative filtering approach. 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 content 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 is presented in section 7.

[1] www.npostart.nl

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. A ribbon typically displays about five items and allows for exploring more items that fit with that particular theme.

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 users their 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 of a user. 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 is linearly combined and produces 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. An example of this kind of row is 'Because you watched Black Mirror'.

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 with. The automatic playlist continuation feature adds one or more tracks to a playlist that fits the original playlist of a user [19]. This feature takes not only the collaborative information of playlists and their corresponding tracks into account but also playlists 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.

Features of content 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, meaning that 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 can be 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. [14] that used the top-2000 most frequent words in titles as a feature in their recommendation system opposed to all the words.

4 METHODOLOGY

This section describes the methodology employed for answering the research questions. First, the hybrid recommendation model. This is followed by a description of the data that is provided to the recommendation systems. Furthermore, the metrics for evaluating the performance of both recommendation systems 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 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 LightFM documentation 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, 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 the 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. The interaction information spans a period of 22 days of which the first 21 days are intended for training the model and the last day for testing. A period of 21 days was used since that is the sliding window used by the current NPO Start recommendation system (see section 2.1). The event data of the period 1 to 22 March, 2019 was used as interaction information for the NPO Start recommendation system and the collaborative filtering part of the hybrid recommendation system. This event data was pre-processed to only gather interaction information about the watched series of users, on the condition that one episode of a series needs to be watched for at least half its duration in order to be included. The total interaction information consists of 1,235,728 interactions and the distribution of these interactions is shown in Figure 1. A weekly pattern of interactions is visible in the distribution and the average amount of interactions per day is about 56,000.

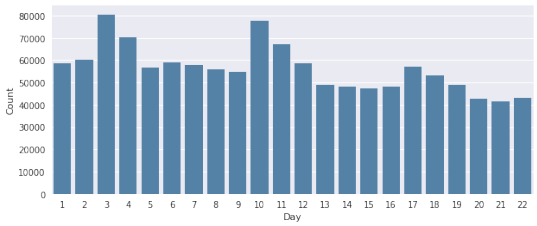
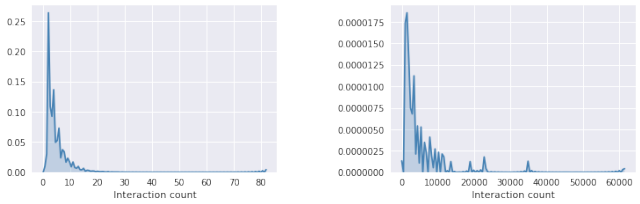


Figure 1: Distribution of the Amount of Interactions per Day

A total of 181,357 users were identified and on average about 42,000 unique users had at least one interaction per day. The distribution of the amount of unique series each user interacted with is shown in 2a. This distribution is skewed to the right, indicating that the majority of users has only watched a couple of series in this period. However, there is a long tail of a few users who have had tens of interactions.

A total of 1446 unique series were watched and on average about 808 series were watched per day in this period. The distribution of the amount of unique users that watched a series is shown in 2b. This distribution is also skewed to the right, indicating that the majority of series was watched by a couple to 5000 users. A big portion was watched by 5000 to 10,000 users. The distribution has got a long tail, indicating that there are a couple of series which were watched by a great number of users, thus were well-rated. The tail does contain a few spikes of series that were watched by roughly the same amount of users.



(a) Amount of Unique Series Interactions per User

(b) Amount of Unique User Interactions per Series

Figure 2: Distribution of the Amount of Interactions for Users and 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. This 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, etc. These items were pre-processed to only gain broadcasts which are available to stream on the NPO Start service, resulting in about 84,000 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 can 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. The content features broadcaster, credits, description, genres and title were selected because they are commonly used as features in content-based recommendation systems (see section 3.3). The feature subtitles is often not provided in multimedia content and has been included since it gives more in-depth information about the content in a broadcast than the title or description. The content features are described in Appendix A Table 6. 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 3 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.

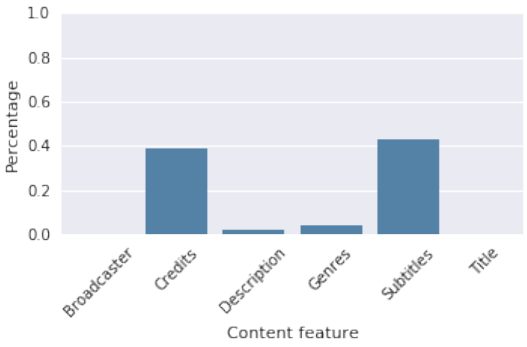


Figure 3: Percentage Series with Missing Values for the Content Features

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

Categorical features. A total of 30 unique broadcasters were identified for the series and each series has at least one or several broadcasters associated with it. The percentage of how often a broadcaster is associated with a series is shown in Figure 4. Most series are broadcasted by the VPRO and other frequent broadcasters are the NTR, AVTR (AVROTROS) and BNVA (BNNVARA).

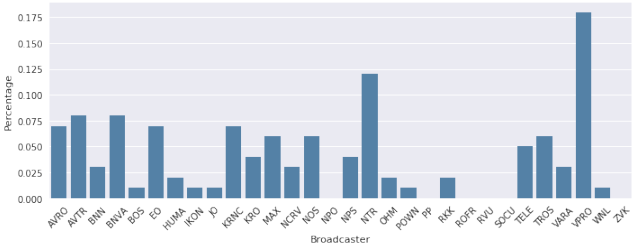


Figure 4: Percentage Series with Broadcaster

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.

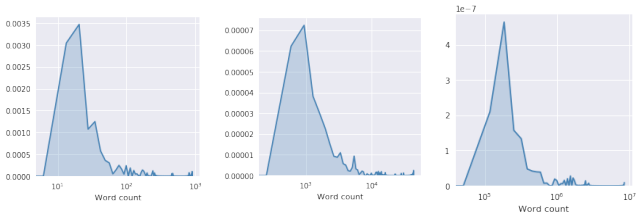
Series either had multiple genres, one genre or no genre assigned to them. A total of 53 different genres were identified and the percentage of how often a genre is associated with a series is shown in Appendix A Figure 10. Genres have a main type indicated by an id of four integers, e.g. 3.0.1.7 'Informatief', 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. As mentioned before, all content features are grouped per series and all unique values are aggregated. This means that all unique broadcast values for the textual features were concatenated for each series. The average word count of these features and their median per series is shown in Table 1. It should be noted that 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 5 on a logarithmic x-scale. The three distributions all indicate a skewed distribution to the right, meaning that a big portion of the series has textual features with a low word count. The long tail of the distributions indicate that a few series have high word counts for the textual features.

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



(a) Title **(b) Description** **(c) Subtitles**

Figure 5: Distributions of Word Count for the Textual Features

4.3 Feature Encoding

The interaction information and content features were prepared into the right format 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 was employed to improve the quality of these content features. For the title, the three words with the highest TF-IDF were extracted per series, the top ten for the description and the top 20 for the subtitles.

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

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

4.4 Evaluation

The performance of a recommendation system is assessed by the quality of recommendations. The quality was evaluated by the two 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 per user 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 recommendation lists that a model produces. This metric evaluates how successful the model is in ranking the highest relevant item to users, so it measures how many non relevant recommendations users have to skip in

their ranked list until the first relevant recommendation. The mean reciprocal rank is calculated by dividing the best possible rank by the actual rank of the first relevant item 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 performance 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. As mentioned in section 4.2.1, a period of 21 days was used for the train set and the following day after the train period was used as test set for the recommendation systems. The train set consists of a total of 1,192,556 interactions and the test set of 41,538 interactions. The interactions of the train set were performed by 179,714 unique users and the test set by 31,127 users. An additional test set was constructed from the original test set, called the "recommended test set", which consists of the interactions that occurred on the top-k series that were actually recommended in the 'Aanbevolen voor jou' ribbon on the NPO Start service. The recommended test set was included since this is the subset of interactions whereof information is available about the top-k precision and rank of these items. A k of 5 was used for the recommended test set, since that is the typical amount of items that is visible on a ribbon of the NPO Start service (see section 2.1). The test set ended up with 149 interactions that were performed by 124 users. It should be noted that interactions of users that were present in the test sets but not in the train set were removed from the experiment. The sparsity of the used interaction information is 0.22%, thus this data is sparse.

4.5.1 RQ1. The experimental setup of the NPO Start recommendation system is shown in Figure 6a. It starts with supplying the train set to the NPO Start model as described in section 2.1. Afterwards, the predictions of this model are evaluated against the test sets using the performance metrics.

4.5.2 RQ2. The experimental setup of the hybrid recommendation system is shown in Figure 6b. The same train and test sets are used in this system as the NPO Start recommendation system. The interaction matrix of the train set and the feature matrix of the content features was supplied to the hybrid recommendation model as 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. Multi-threading was used during the training, prediction and evaluation of the model to speed up the execution and it should not lead to a measurable loss of accuracy in this case since the interaction information is sparse (see section 4.1).

The described experimental setup was used for performing 64 different experiments on the hybrid recommendation model. Each experiment used the same interaction information and a different set of content features while training the hybrid model. The first experiment acted as a baseline wherein no content features were supplied to the model and the other 63 different experiments experiments used a different combination of the six content features (see Table 6 for all combinations). The 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. The experiment model that accomplished the highest performance was compared to the baseline model afterwards.

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

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

The hybrid recommendation system used 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 the scikit-optimize library [5], which allows for finding the optimal hyperparameters to maximise model performance. The optimal hyperparameters were then used for producing recommendations with this model.

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

5 RESULTS

5.1 RQ1: What is the Performance of the NPO Start Recommendation System?

The NPO Start recommendation system offers recommendations to a user several times a day (see section 2.1) and since the performance metrics are evaluated per user, the precision@5 and reciprocal rank results of an offer were first averaged per user before producing the final results. Since there was no precision@5 and reciprocal rank information available for all the interaction information of the full test set, all user interactions who were not included in the recommended test set were rewarded a precision@5 and reciprocal rank of 0. The results of the performance metrics for both test sets are summarised in Table 3.

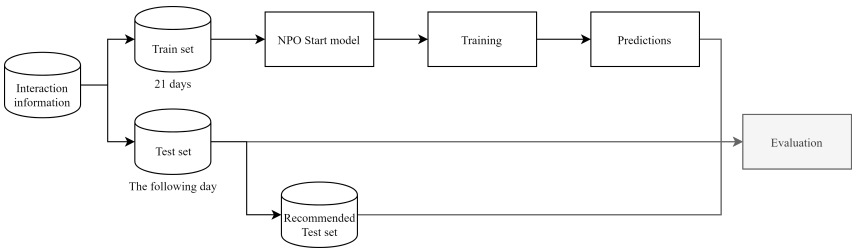
Table 3: Results of the NPO Start Recommendation System

Metric	Mean	Std
Mean p@5	0.00	0.01
MRR	0.00	0.04

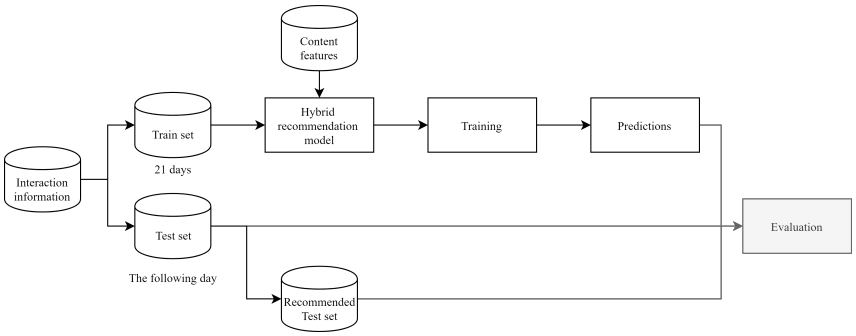
(a) Test Set

Metric	Mean	Std
Mean p@5	0.19	0.06
MRR	0.61	0.34

(b) Recommended Test Set



(a) The NPO Start Recommendation System



(b) The Hybrid Recommendation System

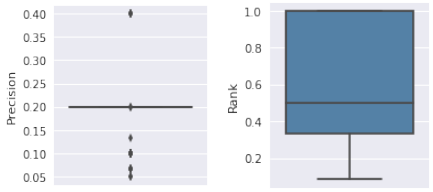
Figure 6: The Experimental Setup

Mean Precision@5. The mean p@5 of 0.00 and standard deviation of 0.01 on the test set indicates that there were almost no relevant items for a user in the recommendations provided by the NPO Start recommendation system. This indicates that hardly any series that were recommended to a user were found being relevant.

However, the mean p@5 of 0.19 on the recommended test set indicates that on average 1 in 5 recommendations is a relevant item for a user. This indicates that on average a user needs to see a ranked list of 5 items once in order to find a series that suits him or her. The boxplot of this metric (see Figure 7a) indicates that a majority of users have a mean p@5 of 0.20 and there are a few occasions where on average 2 in 5 recommendations is relevant or about 1 in 10 recommendations is relevant.

Mean Reciprocal Rank. The mean p@5 of 0.00 and standard deviation of 0.04 on the test set indicates that there were almost no relevant items for a user in the recommendations, since a reciprocal rank of 0 indicates that there was no highest relevant item in the offered recommendations. This illustrates that hardly any series that were recommended fit the user.

Since a reciprocal rank of 1.0 indicates that the highest relevant item was the first item and a reciprocal rank of 0.5 indicates it being being the second item, a MRR of 0.61 on the recommended test set indicates that on average the highest relevant item is placed as the second item in a ranked list for a user. The MRR does have a high standard deviation and the boxplot (see Figure 7b) indicates that 50% of users their MRR ranges from a score of 1.0 to 0.33, thus being ranked as either the first, second or third item.



(a) Mean P@5 **(b) MRR**

Figure 7: Boxplots of the NPO Start Recommendation System Results per User

5.2 RQ2: Which Content Features Improve the Performance of the Hybrid Recommendation System the Most?

The results of the 64 different experiment models of the hybrid recommendation system for the test set and the recommended test set is shown in Appendix B Figure 11 & 12.

The results of the experiment models on the test set, which contains all watched interactions, show smooth learning curves for both performance metrics. The learning curves show an exponential rise with increasing epochs.

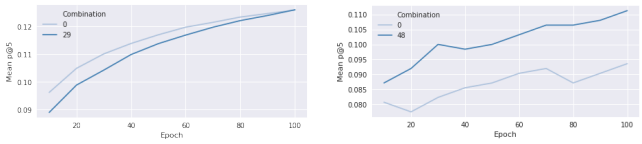
The results of the experiment models on the recommended test set, which contained all watched interactions that were recommended, show learning curves that are erratic. However, the learning curves do show an increase in metrics with increasing epochs.

Mean Precision@5. The top-10 mean p@5 of content feature combinations is shown in Table 8. The mean p@5 and standard deviation results for the top-10 combinations of each test set are

very similar. Overall, the precision of the test set results are slightly better than that of the recommended test set.

The experiment model that incorporated the 29th combination of content features accomplished the highest mean p@5 for the test set. This combination used the content features broadcaster, genres and title. The results of this experiment model and that of the baseline model, which uses no content features, is shown in Figure 8a. A two sample right-tailed z-test ($\alpha = 0.05$) was conducted to compare the mean p@5 of experiment model 29 and the baseline model. There was no significant difference in the scores for experiment model 29 ($\bar{x} = 0.13$, $s = 0.12$, $n = 31127$) and the baseline model ($\bar{x} = 0.13$, $s = 0.12$, $n = 31127$); $z = 0.09$, $p = 0.46$. These results suggest that the best experiment model that used content features does not perform better than the baseline model based on the mean p@5 of the test set.

The experiment model that incorporated the 48th combination of content features accomplished the highest mean p@5 for the recommended test set. This combination used the content features broadcaster, description, genres and title. The results of this experiment model and that of the baseline model, which uses no content features, is shown in Figure 8b. A two sample right-tailed z-test ($\alpha = 0.05$) was conducted to compare the mean p@5 of experiment model 48 and the baseline model. There was no significant difference in the scores for experiment model 48 ($\bar{x} = 0.11$, $s = 0.10$, $n = 124$) and the baseline model ($\bar{x} = 0.09$, $s = 0.10$, $n = 124$); $z = 1.36$, $p = 0.09$. These results suggest that the best experiment model that used content features does not perform better than the baseline model based on the mean p@5 of the recommended test set.



(a) Test Set **(b) Recommended Test Set**

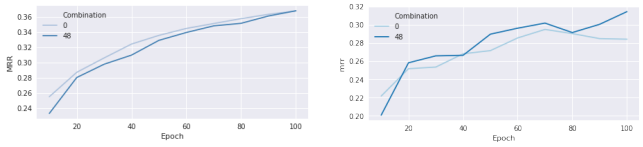
Figure 8: Mean P@5 Results of the Baseline and Top Combination

The experiment model that achieved the highest mean p@5 on both tests used similar content features. Experiment model 48 included the description feature which was not used in experiment model 29. The results of these experiment models indicate that incorporating content features into the hybrid recommendation model does not necessarily improve the mean p@5.

Mean Reciprocal Rank. The top-10 MRR of content feature combinations is shown in Table 9. The MRR and standard deviation results for the top-10 combinations of each test set are very similar. Overall, the precision of the test set results are a little higher than that of the recommended test set

The experiment model that incorporated the 48th combination of content features accomplished the highest MRR for the test set. This combination used the content features broadcaster, description, genres and title. The results of this experiment model and that of the baseline model, which uses no content features, is shown in Figure 9a. A two sample right-tailed z-test ($\alpha = 0.05$) was conducted to compare the MRR of experiment model 48 and the baseline model. There was no significant difference in the scores for experiment model 48 ($\bar{x} = 0.37$, $s = 0.35$, $n = 31127$) and the baseline model ($\bar{x} = 0.37$, $s = 0.35$, $n = 31127$); $z = 0.06$, $p = 0.48$. These results suggest that the best experiment model that used content features does not perform better than the baseline model based on the MRR of the test set.

The experiment model that incorporated the 48th combination of content features accomplished the highest MRR for the recommended test set. This combination used the content features broadcaster, description, genres and title. The results of this experiment model and that of the baseline model, which uses no content features, is shown in Figure 9b. A two sample right-tailed z-test ($\alpha = 0.05$) was conducted to compare the MRR of experiment model 48 and the baseline model. There was no significant difference in the scores for experiment model 48 ($\bar{x} = 0.31$, $s = 0.32$, $n = 124$) and the baseline model ($\bar{x} = 0.28$, $s = 0.31$, $n = 124$); $z = 0.76$, $p = 0.22$. These results suggest that the best experiment model that used content features does not perform better than the baseline model based on the MRR of the recommended test set.



(a) Test Set **(b) Recommended Test Set**

Figure 9: MRR Results of the Baseline and Top Combination

Experiment model 48 achieved the highest MRR on both tests. The results of this experiment model indicates that incorporating content features into the hybrid recommendation model does not necessarily improve the MRR.

5.3 RQ3: Can the Performance of the NPO Start Recommendation System be Improved by Implementing a Hybrid Recommendation System?

The majority of the results of the previous section indicate that the experiment model that incorporated the 48th combination performed similar but slightly better than the baseline, so the hyperparameters of the model that used the content features broadcaster, description, genres and title were optimised. The resulting hyperparameters after the optimisation is shown in Table 4 and the accompanying results for the performance metrics is shown in Table 5.

Mean Precision@5. Two sample right-tailed z-tests ($\alpha = 0.05$) were conducted to compare the mean p@5 results of the optimised model to the NPO Start model.

When evaluated on the test set, there was a significant difference in the scores for the optimised model ($\bar{x} = 0.19$, $s = 0.11$) and the NPO Start model ($\bar{x} = 0.00$, $s = 0.01$); $z = 296.8$, $p = 0.00$. These results suggest that a recommendation system using the optimised

Table 4: Hyperparameter Values of the Optimised Model

Hyperparameter	Value
Epochs	89
Learning rate	0.01
Number of components	168
Item alpha	0.00
Scaling	0.06

Table 5: Results of the Optimised 48th Experiment Model

Metric	Mean	Std
Mean p@5	0.19	0.11
MRR	0.53	0.34

(a) Test Set

Metric	Mean	Std
Mean p@5	0.13	0.10
MRR	0.39	0.32

(b) Recommended Test Set

model performs better than a recommendation system using the NPO Start model based on their mean p@5 of the test set.

When evaluated on the recommended test set, there was no significant difference in the scores for the optimised model ($\bar{x} = 0.13$, $s = 0.10$) and the NPO Start model ($\bar{x} = 0.19$, $s = 0.06$); $z = -5.17$, $p = 1.00$. These results suggest that a recommendation system using the optimised model does not perform better than a recommendation system using the NPO Start model based on their mean p@5 of the recommended test set.

The results of both test sets indicate that the performance of the hybrid recommendation system in comparison to the NPO Start recommendation system based on the metric mean p@5 is dependent on the set of interaction information used for evaluation.

Mean Reciprocal Rank. Two sample right-tailed z-tests ($\alpha = 0.05$) were also conducted to compare the MRR results of the optimised model to the NPO Start model.

When evaluated on the test set, there was a significant difference in the scores for the optimised model ($\bar{x} = 0.53$, $s = 0.34$) and the NPO Start model ($\bar{x} = 0.00$, $s = 0.04$); $z = 268.5$, $p = 0.00$. These results suggest that a recommendation system using the optimised model performs better than a recommendation system using the NPO Start model based on their MRR of the test set.

When evaluated on the recommended test set, there was no significant difference in the scores for the optimised model ($\bar{x} = 0.39$, $s = 0.32$) and the NPO Start model ($\bar{x} = 0.61$, $s = 0.34$); $z = -4.48$, $p = 1.00$. These results suggest that a recommendation system using the optimised model does not perform better than a recommendation system using the NPO Start model based on their MRR of the recommended test set.

The results of both test sets indicate that the performance of the hybrid recommendation system in comparison to the NPO Start recommendation system based on the metric MRR is dependent on the set of interaction information used for evaluation.

6 CONCLUSIONS

In this thesis, a hybrid recommendation system that utilises metadata was presented and compared to the current recommendation system of the NPO Start service which uses collaborative filtering. The hybrid recommendation system serves out predictions using a hybrid LightFM model to which interaction information and content features are supplied. The content features consist of the six metadata features broadcaster, credits, description, genres, subtitles and title.

Based on experiments where different combinations of the content features were supplied to the hybrid model, the results indicated that the model that utilised the broadcaster, description, genres and title features performed similar but slightly higher than the model that utilised no content features. This concludes that incorporating content features into the hybrid recommendation model does not necessarily improve the performance.

Based on the comparison of the optimised best performing hybrid recommendation model and the current model of the NPO start recommendation system, the results indicated that the performance of the hybrid recommendation system is better than that of the NPO Start recommendation system when based on a broader evaluation set. This concludes that a hybrid recommendation using metadata can perform better than the current recommendation system of NPO Start.

7 DISCUSSION

This section discusses the results and the limitations of the employed methodology. Also, possible future work is presented that could be taken to overcome these limitations.

The results indicated that the hybrid recommendation model does not perform better when content features are used opposed to when no content features are used. This does not fit with previous research stating that incorporating content into a collaborative filtering approach provides more accurate recommendations when ratings are sparse [1, 8]. One possible cause for this result is the completeness of the used metadata for the content features. Previous research has shown that features derived from high-quality metadata lead to a better performance of content-based recommendation systems [8, 14, 15]. As mentioned in section 4.2.2, the used metadata differed in completeness, detail and had missing values for a portion of the content features. This indicates that a better performance could have been achieved in the hybrid recommendation experiment models that used content features when the metadata was of a better quality. Further research is needed to establish if the quality of the metadata was a limitation of evaluating the performance of content features in the hybrid recommendation model.

Furthermore, the results demonstrated that the hybrid recommendation system performs significantly better than the NPO Start recommendation system when evaluated on the full test set opposed to the recommended test set. The recommended test set consists of watched series that were recommended to users and assumes that users would have interacted with the same series, regardless of which model was used to generate the recommendations. This assumption is a major drawback of offline experiments [4], since the recommended test set does not take into account how different the hybrid recommendation model is compared to the NPO Start model. The NPO Start recommendation system recommends well-rated series to users, opposed to the hybrid recommendation system that recommends a mix of well-rated and similar series to users. This

is apparent in the huge loss of performance when the NPO Start recommendation system was evaluated on the full test set instead of the recommended test set. The full test set gives a broader view of relevant items for users and is thus more suited when comparing the two recommendation systems to each other. However, the most reliable performance results are achieved when both recommendation systems are compared in an online setting, because this evaluates the recommendations on actual user behaviour.

Lastly, the generalisability of the results is limited by the interaction information used in the experimental set-up, since only one specific time period was used. Different performance results could be achieved for the recommendation systems in different time periods because of the temporality of interactions. A more collaborative approach could be favoured in one time period based on a higher occurrence of well-rated series, e.g. the series 'Poldark' generated a high amount of interactions in a short time because it was heavily promoted inside the NPO Start service and on social media. Alternatively, a more content-based approach could be favoured based on events happening in the world, e.g. users watch more content about the Dutch royal families close to Kingsday. Future research is needed to establish the performance of the recommendation systems in different time periods and how temporality could be incorporated to improve the hybrid recommendation model.

ACKNOWLEDGMENTS

I would like to thank the Marketing Intelligence Team at the NPO for entrusting me with this project and for providing a welcoming and supportive environment to work in. I would especially like to thank Robbert van Waardhuizen for supervising me internally in the company. Additionally, I am grateful for the helpful observations provided by Dr Maarten Marx.

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 6: 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.

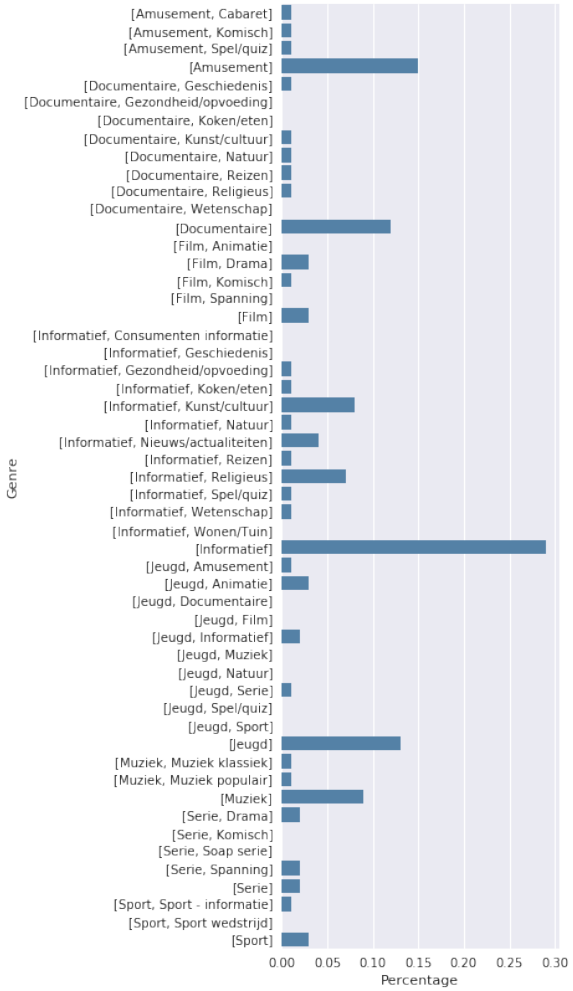


Figure 10: Percentage Series with Genres

Table 7: 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

B RESULTS

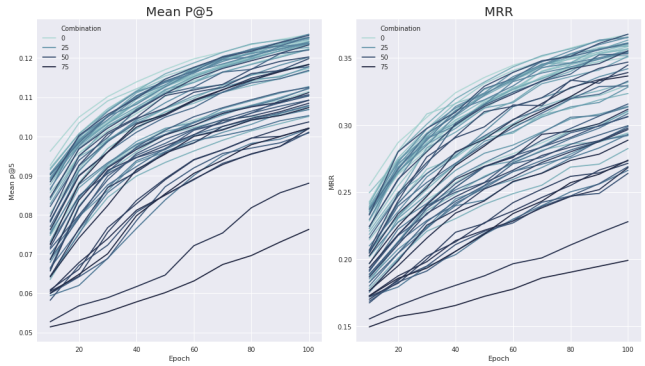


Figure 11: Results for the Content Feature Combinations on the Test Set

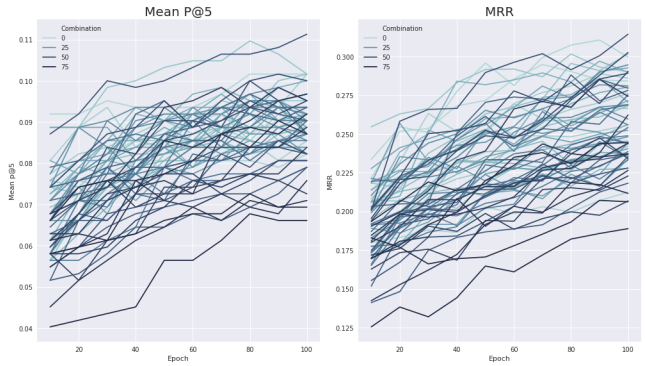


Figure 12: Results for the Content Feature Combinations on the Recommended Test Set

Table 8: The Top-10 Mean P@5 Content Feature Combination Results

Rank	Combination	Epoch	Mean p@5	Std
1	29	100	0.13	0.12
2	0	100	0.13	0.12
3	48	100	0.13	0.12
4	38	100	0.13	0.12
5	8	100	0.13	0.12
6	16	100	0.13	0.12
7	19	100	0.13	0.12
8	26	100	0.13	0.12
9	17	100	0.12	0.12
10	3	100	0.12	0.12

(a) Test Set

Rank	Combination	Epoch	Mean p@5	Std
1	48	100	0.11	0.10
2	8	80	0.11	0.10
3	9	100	0.10	0.10
4	1	80	0.10	0.10
5	5	100	0.10	0.10
6	50	90	0.10	0.10
7	49	80	0.10	0.10
8	18	100	0.10	0.10
9	16	70.0	0.10	0.11
10	29	90.0	0.10	0.10

(b) Recommended Test Set

Table 9: The Top-10 MRR Content Feature Combination Results

Rank	Combination	Epoch	Mean p@5	Std
1	48	100	0.37	0.35
2	0	100	0.37	0.35
3	16	100	0.37	0.34
4	29	100	0.37	0.34
5	26	100	0.37	0.35
6	19	100	0.36	0.34
7	27	100	0.36	0.34
8	38	100	0.36	0.34
9	9	100	0.36	0.34
10	3	100	0.36	0.34

(a) Test Set

Rank	Combination	Epoch	Mean p@5	Std
1	48	100	0.31	0.32
2	1	90	0.31	0.32
3	51	100	0.30	0.32
4	9	100	0.30	0.32
5	8	80	0.30	0.30
6	16	90	0.30	0.32
7	0	70	0.30	0.33
8	29	100	0.30	0.32
9	20	100	0.29	0.32
10	50	100	0.29	0.31

(b) Recommended Test Set