Improving Box Office Result Predictions for Movies Using Consumer-Centric Models

Rui Paulo Ruhrländer
Cinuru Research GmbH
Hasso Plattner Institute
paulo.ruhrlaender@cinuru.com

Martin Boissier
Hasso Plattner Institute
martin.boissier@hpi.de

Matthias Uflacker
Hasso Plattner Institute
matthias.uflacker@hpi.de

ABSTRACT

Recent progress in machine learning and related fields like recommender systems open up new possibilities for data-driven approaches. One example is the prediction of a movie's box office revenue, which is highly relevant for optimizing production and marketing. We use individual recommendations and user-based forecast models in a system that forecasts revenue and additionally provides actionable insights for industry professionals. In contrast to most existing models that completely neglect user preferences, our approach allows us to model the most important source for movie success: moviegoer taste and behavior. We divide the problem into three distinct stages: (i) we use matrix factorization recommenders to model each user's taste, (ii) we then predict the individual consumption behavior, and (iii) eventually aggregate users to predict the box office result. We compare our approach to the current industry standard and show that the inclusion of user rating data reduces the error by a factor of 2× and outperforms recently published research.

CCS CONCEPTS

• **Applied computing** → **Forecasting**; • **Information systems** → *Data mining*; Recommender systems;

KEYWORDS

Box Office Predictions, User Ratings, Gradient-Boosted Trees, Logistic Regression, Recommender Systems, Motion Picture Industry

ACM Reference Format:
Rui Paulo Ruhrländer, Martin Boissier, and Matthias Uflacker. 2018. Improving Box Office Result Predictions for Movies Using Consumer-Centric Models. In *KDD '18: The 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, August 19–23, 2018, London, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3219819.3219840

KDD '18, August 19–23, 2018, London, United Kingdom

ACM ISBN 978-1-4503-5552-0/18/08…$15.00
https://doi.org/10.1145/3219819.3219840

1 PREDICTING BOX OFFICE RESULTS

In 2017, the global movie industry achieved an all-time high in revenues of USD $40.6 billion from the sales of cinema tickets (so-called *box office revenues*). Nonetheless, making movies is still a risky business: Every second movie does not break even [3, 9, 14]. To mitigate financial risks of movie productions, researchers and practitioners developed various strategies to estimate revenues. The methods vary from very intuitive – often manual – approaches (e.g., comparing movies deemed as 'similar' by genre and budget) to complex stochastic models. Most existing prediction models use movie metadata to predict the box office. This metadata includes e.g. genre, actors, and budget. Other important factors like a movie's story, the visual appeal, special effects, etc. are hard to quantify and seldom used. Although it is hard to explicitly quantify such variables, their effect is captured implicitly in *movie rating data sets*. In the domain of recommender systems, important progress has been made in inferring a user's taste and also implicit movie properties from large rating data sets.

Box office prediction is essentially the task of predicting how many moviegoers decide to buy a cinema ticket for a particular film. This decision is highly individual and depends on the movie taste of each moviegoer and the explicit and implicit properties of a movie. In this paper, we propose a moviegoer-centered approach to box office prediction by incorporating concepts and algorithms from the field of recommender systems. The goal of our work presented in this paper is mainly to improve prediction accuracy compared to existing models. Furthermore, our fine-granular approach allows us to leverage user-based information in order to mine and analyze target groups and to eventually formulate actionable recommendations.

Box office predictions are important in two distinct phases: First, before a movie is made and a decision on actors, budget, etc. has to be made. Second, in the movie's marketing, where it is crucial to attract the attention of the right moviegoers and track marketing progress and success. The latter task becomes increasingly important because of two factors: (i) the number of movies released to cinemas increases every year [17] and (ii) the time window in which a movie is shown in cinemas is decreasing. It is vital to reach the right people at the right time, e.g., via personalized advertisements.

We focus on prediction models for movie distributors organizing the marketing campaign of a film. Movie marketing accounts for about 30% of the total cost of a movie, on average USD 35.9 million per movie in the year 2007 [8]. Researchers claim that even not very promising films can be prevented from flopping if the distributor puts enough effort into the marketing of the movie [13].

Throughout this paper, we will make the following contributions:

- Introduce a novel, fine-granular modeling approach, centered on moviegoers.
- Showcase the use of results from recommender systems for revenue estimation.
- Leverage movie rating data sets for box office prediction.
- Show how the prediction process can be separated into three distinct stages.
- Demonstrate the superior accuracy of our approach by comparing it to both current industry standards as well as recent research publications.

The remainder of this paper is structured as follows. Section 2 lists related contributions in the field of box office predictions. Section 3 depicts our concept, argues for a user-based modeling approach and introduces the prediction pipeline. Section 3 presents the used data sets and describes our data cleansing for movies. The actual implementation of the prediction pipeline and its three stages are presented in Section 5. Section 6 shows the final results of our concept and we conclude in Section 7.

2 RELATED WORK

There has been a long history of research and industry approaches to tackle the challenge of predicting box office results. Historically many prediction approaches employed linear regression models, where the box office is taken as the dependent variable and movie characteristics, such as budget, actors, genre or number of screens are used as independent variables. Examples of these approaches are [19] and [20].

Eliashberg et al. [6] use movie scripts as an input for box office prediction. They pre-process the movie script in a semi-automated fashion to extract various variables about the story, setting, language, twists etc. In the first step, the authors preprocess the script and extract various variables. They use three different groups of variables: genre and Content variables, semantic variables, and bag-of-words variables. The quantification process is partially automated: Bag-of-words and semantic variables are calculated automatically, while human readers quantify the content variables. Content variables describe aspects of the movie's plot. Examples of these variables are whether a movie has a surprise ending, whether it is logical, or whether its hero has an inherent weakness. In total, there are 25 of these variables. The second group of variables contains semantic variables, such as the number of scenes, number of dialogues and their average length, which are extracted automatically. Additionally, they generate bag-of-words variables: First, the authors determine the 100 most important words and then calculate the importance of each script by using TF-IDF. Afterward, they feed the document-term matrix into a singular value decomposition algorithm (SVD). In the end, there are two remaining variables LS1 and LS2. A post-hoc analysis showed that these variables vaguely relate to settings (LS1) and degree of vulgarity (LS2). Besides the script-based variables, the movie's budget is used as additional information. Using the quantified variables, the authors calculate distances between the films in their sample. To predict the revenue, they use a weighted average of the box office of the other movies. The weight of a film is higher if it is similar regarding the variables generated before. To compare their results, they also implemented other methods, such as multiple regression, regression trees, and comparables-based methods, which resemble how industry professionals predict revenues. The data set contains a total of 300 movies from which the authors used 35 as a holdout test set. Their most accurate kernel-based approach (dubbed *Kernel-II*) reduces the mean squared error by 20% over the comparables approach.

In 2006, Sharda and Delen developed a neural network to predict the box office of movies [26]. They used a data set of 834 movies released between 1998 and 2002 and variables representing MPAA rating, competition, star power, genre, special effects, sequel, and the number of screens. It is interesting to note that, in contrast to most other studies, the movie's budget was not used as an input variable. The neural network classifies movies into one of nine categories, ranging from flop (revenue < 1M) to blockbuster (revenue >200M). They correctly classify 36.9% of the movies in their data set, and in 75.2% of the cases, they predict either the exact category or one category off.

Several published models use a *behavioral-based* approach that introduces various stages of the customer's decision process. Zufryden [29] models this process as an awareness/intention/behavior hierarchical process. First, a moviegoer has to become aware of the film; then he may get interested in it and later decides to buy a cinema ticket. Another example of this kind of modeling is by Sawhney and Eliashberg [25]. To predict the success of a movie based on demand and supply (here: ticket sales and movie screens), modeling based on coupled differential equations has been proposed by Jones and Ritz [15] and by Elberse and Eliashberg [5].

In 2000, Eliashberg et al. implemented *MOVIEMOD* [7], a pre-release market evaluation model for the movie industry. It is a behavioral model that tries to express the consumer's intentions and actions. Instead of using film and market data to calibrate the model, a so-called "consumer clinic" is used to collect the data needed to predict the movie's box office. The consumer clinic is a test screening session with moviegoers: At first, the audience answers a questionnaire regarding their movie-going behavior, then they are exposed to different advertisement stimuli and their respective impact on the intentions to see the movie are measured. Then, they see the movie and afterward reviews and word-of-mouth intentions are measured. In contrast to other models, the MOVIEMOD system yields actionable results. In this case, advice on how to optimize the marketing strategy. Unfortunately, the paper does not contain directly comparable results, as the authors tested the model only on a few movies before publication. The authors state that "its prediction was better in all cases" compared to other models, resulting in an estimated 50% box office revenue increase for one movie in which the distributor followed recommendations.

In 2005, Delen and Sharda [4] implemented a system called *Forecast Guru*, which was a decision support system for Hollywood managers. It employed various techniques and merged their results afterward. It used neural networks, decision trees, logistic regressions, discriminant analyses, and an information fusion algorithm to combine the results of the individual models.

Two approaches often deployed in the movie industry are the so-called *comparables* and *tracking studies*. The most common approach to getting an early estimate for a movie's commercial potential is

to find similar movies that have already been released and then averaging their revenues. These similar movies are called *comparables*. This process is manual and results highly depend on the people selecting the comparables. Most often they are chosen on the basis of budget, lead actors, genre or topic. As predictions based on comparables are the industry's standard, researchers often use this approach as a benchmark. Although comparables seem to be a simplistic model, comparisons show that it can outperform other prediction models like linear regression or regression trees [6].

To estimate a movie's box office briefly before its release movie studios further use tracking surveys [27], which measure the awareness and interest of customers. Such surveys are telephone interviews that track if moviegoers have heard of the movie and whether they plan to see it. Based on these tracking studies movie studios have developed own prediction models to forecast the box office revenue. Normally these models are not published, but in 2014 the Sony Pictures tracking models were leaked through a hacker attack. These models are simple – mostly linear – combinations of the individual variables coming from the tracking studies. Besides the leaked models, it is also interesting to study the leaked e-mails: Sony Pictures top executives show their mistrust in these models and complain about their inaccuracy. One e-mail by Abe Recio, Senior VP of Strategic Marketing and Research summarizes: "I think at times there is more art than science going into these predictions".

For the interested reader, we recommend [8] and [11], which give an in-depth analysis of many prediction methods.

3 CONCEPT

In this section, we explain the conceptual overview of our model and the different steps in our modeling pipeline.

3.1 Use-Case and Data Sets

Our model is aimed at movie distributors and estimates the potential of an unreleased movie. It is usable after the movie has been finished, but before any marketing activity has started. We use historical data about already released movies to train our model. This data consists of metadata about the movies[1], and the MovieLens 20M rating data set (see Section 4). Our metadata set consists of complete information for 2 964 movies released between 2005 and 2016.

In a real-world use case, where the box office of an unreleased movie should be predicted, the movie is shown to a test audience to collect their ratings of the movie. Additionally, the test audience is requested to rate other movies that are already released. Afterward, their ratings and the movie's meta properties are added to the data set of existing movies.

3.2 The Prediction Pipeline

Our prediction pipeline consists of three stages: **(i)** The first stage is the taste space modeling, where we use matrix factorization techniques that have been developed for recommender systems to calculate movie profiles and user taste profiles. **(ii)** The second stage is to train a model for each user that predicts their cinema-going behavior. **(iii)** In the third stage, we build an aggregation model that predicts the box office revenue. Figure 1 shows the conceptual architecture of our pipeline.

3.2.1 Stage One: Taste Modeling. We use the rating data set to calculate movie profiles and users' taste profiles. First, we use our rating data set to build a `user` × `movies` matrix, where each non-missing entry represents a rating attributed to the particular user to the respective movie. We then use matrix factorization techniques as described in [16, 21, 24] to transform this high dimensional sparse matrix into lower dimensional approximations. The decision for matrix factorization has been driven by its wide successful application in recommendation systems and its runtime performance. The outcomes of this step are user taste vectors and movie profile vectors. A multiplication of the i-th user vector with j-th movie vector yields the estimated rating for movie j by user i. Even though rating data is usually highly sparse, we found further stratification to be unnecessary. In Section 5.1, we will describe how we use these technologies. Our contribution is to use the concepts in the context of box office revenue estimation. To our best knowledge, such an approach has not been undertaken before.

3.2.2 Stage Two: User Models. There is a difference between a user's predicted rating for a movie and their likelihood of seeing the film in the cinema. While we seldom go to a film we assume to dislike, we go only to a small percentage of movies we will probably like. Additionally, we might even watch movies with a low rating when we watch with a group of friends. Many other factors, such as marketing, availability in a cinema next to the user, and the user's movie-going frequency, have an influence on their likelihood of seeing a movie in the cinema. Therefore, estimating how much a user might like a film is only a preliminary step in predicting if they will see the movie in the cinema.

While well-established prediction approaches for movie ratings exist, modeling individual cinema attendance behavior poses a new challenge. We have implemented and evaluated different modeling techniques, such as neural networks, logistic regression, and boosted tree ensembles. Section 5.2 describes the implementation and evaluation details of these models. In this stage, we use the following input data: a movie's meta-characteristics (such as genre or budget), the movie profile, and the user's taste profile.

3.2.3 Stage Three: The Aggregation Model. The aggregation model receives predictions from all user models as inputs and processes them to output a box office revenue prediction.

The biggest challenge for the aggregation model is to mitigate overfitting, as the dimensionality of the input data is much higher than the number of training cases. Although our movie sample is significantly larger than in most of the related studies, the number of movies is still significantly smaller than the number of input dimensions, which is the number of users in our rating data set.

[1] We use the following metadata for each movie: The popularities of the main actors and of the director, the budget, the revenue, the MPAA age rating, the release date, the popularity of their source material (e.g. book), the popularity of the series (e.g. James Bond).

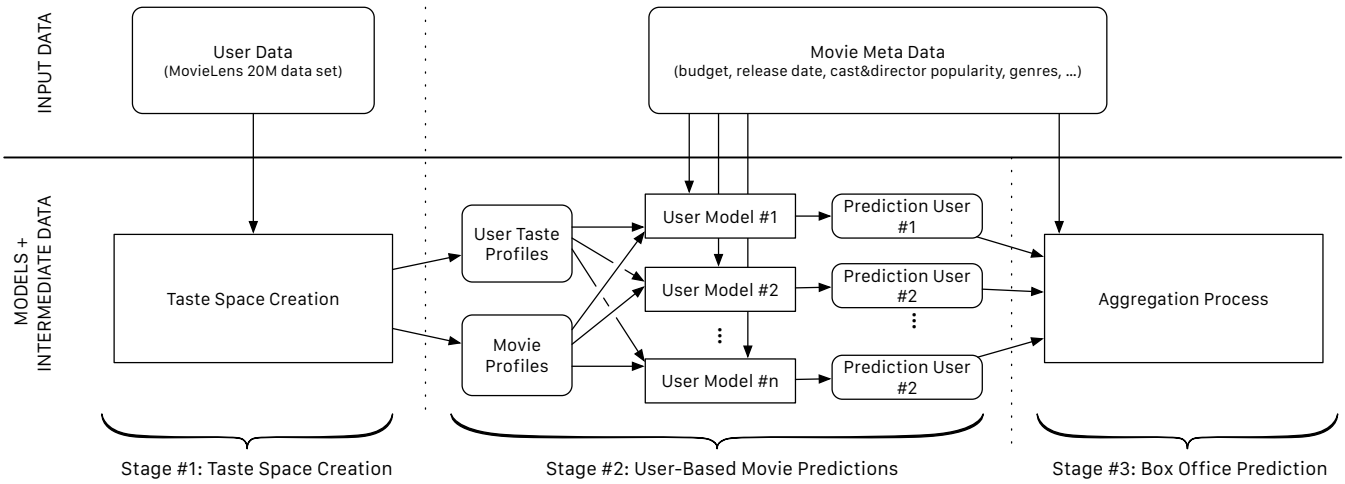

Figure 1: Our prediction pipeline. First, we process the rating data and compute movie and taste profiles. With this and additional movie metadata a user model for each user is trained. Finally, the predictions from all users are aggregated into a box office revenue prediction.

3.3 Individualized Models

Existing box office prediction models vary broadly from a technological point-of-view, but conceptually most of them are movie-centered. They work with movie metadata, perform analyses on it and output an estimate for the box office.

We approach the box office prediction problem from a different angle. While factors like budget, or a well-known cast have a correlation with the box office of a movie, we advocate that this correlation is indirect. A film is successful at the box office if many individual moviegoers decide to buy a cinema ticket. The movie's meta properties impact the box office indirectly through an influence on the moviegoers.

Behavioral Models have followed an audience-centered approach, but they treat moviegoers as a homogeneous population. With the advent of big data and large-scale parallelism, the simplification of assuming that all users are similar is obsolete. We claim that each step of the *awareness-intention-action* chain can differ depending on the individual user. While a TV commercial for a film raises the general awareness, it does not do so for all users. To make a movie successful, distributors only need to get the attention of the users who are potentially interested in the film.

Why do we think that our user-centered modeling approach is superior to success-factor-based, movie-centered models? Let us take a look at the following example: Many people love to go to blockbuster movies with a high budget, but some people prefer low-budget independent productions. While many people like to watch movies starring Tom Cruise, others do not. Intuitively we would assume that the proportion of users who like Tom Cruise is higher in the population of "blockbuster-goers" than it is in the population of "independent-film-goers." Here comes the point where most models fail: Booking Tom Cruise would drastically increase the revenue prediction for a movie. While this is true for most movies, it certainly is not for all. It is easily possible to integrate simple constructs like the example above into a movie-centered model, but in reality, the interactions between variables are much more complex, and the number of combinations is growing exponentially with the number of variables.

We can reduce this complexity by modeling at the level of individual users. For example, you might like or dislike a particular actor. In rare cases, it might depend on another variable like the genre of the movie ("I like Tom Cruise in action films, but dislike his romantic comedies"). Having more complex conditions is unlikely ("I like Tom Cruise if the budget is below $50 million, and the film starts around Christmas, or its director is very unpopular"). For this reason, modeling a behavior of one user leads to simpler models than modeling the whole population's behavior. Instead of learning one model with complex dependencies and rules, we train many simpler models.

This method needs much more data and computational resources because it has to train thousands of models. While such an approach was technologically unfeasible some years ago, improvements in massive parallelization and machine learning algorithms nowadays allow solving this problem efficiently.

3.4 Actionable Insights

Although not the main focus of this paper, we want to briefly discuss our model's capabilities to create actionable insights. A prediction model is more valuable to movie business professionals if it allows deriving actionable insights [10]. However, most existing approaches do not allow to do so as the level of granularity of predicted insights it too coarse. Although we focus on improving the prediction accuracy, the intermediate results of our model can be used to generate additional insights. Our model outputs a predicted rating and a probability of watching for each user. These values can be used to gather further insights about the target group of a film,

about the marketing potential and about strategies on how to reach the audience. In the following, we will present small examples on how to use these intermediate results.

By bucketing the users by their likelihoods of watching, and analyzing the size of each bucket, we could assess the span of possible box office results. While some movies might have a small but strong fan base, other might appeal to a larger but less enthusiastic group. This is helpful to estimate upside potential and downside risk. Similarly, one could evaluate the difference between users that have a high predicted rating but a low probability of watching. If similarities between these users exist, this could be a hint on how to optimize the marketing strategy.

An important mean to create awareness for a movie is trailering in cinema. By analyzing the target group's movie taste, we can find other movies with a similar target group and show our movie's trailer before these movies. It could be especially interesting to look at users with a high predicted rating but a low probability of seeing the film, which allows us to address user groups that are not recognizable without manual work using existing approaches.

With a richer data set that contains the users' media consumption, we could analyze, which marketing channel is best to reach the film's target group. Our model allows to determine a movie's target group much earlier as before and allows further analysis on a per-user level. This fine-grained data yields the potential for improvements in the marketing processes of distributors.

4 DATA SOURCES

This section briefly discusses which data sources we have used for our evaluation. Further, we disclose to which extent we filtered outliers to improve transparency and reproducibility. Both of these aspects fall often short in previous studies. Note, the acquired data sets are all publicly available and mostly directly downloadable.

We use two distinct data sets: *movie metadata* and *rating data*. The movie metadata contains information like the movie's budget and revenue, its actors, or genre. The rating data set consists of film ratings given by moviegoers, represented as a large sparse matrix.

4.1 Ratings Data Set

We use the MovieLens data set [12], which is a popular rating data set (cf. Harper and Riedl [12]). It is a family of openly available data sets, from which we use the largest one, containing 20 million movie rating tuples. Each tuple consists of a user ID, a movie ID, a rating from half a star to five stars and a timestamp. The rating data set contains 27 278 movies rated by 138 493 users. The first ratings in the MovieLens data set are from 1998, and the data set is continuously updated. While the number of ratings per month varies over time, there is a sufficiently large amount of ratings per month to cover all time periods. We expect ratings that have been gathered before the movie was released to home cinema as an indicator that the movie has been seen in the cinema. For application in production, a data set collected on purpose for this aim would presumably increase prediction accuracy substantially.

As the data set was not collected for the purpose of box office prediction it has "missing-value" issues: We assume a missing rating signifies the user has not seen the movie. Of course, this is not true in all cases. A data set without this problem would most likely increase prediction accuracy, as the user models could be trained with clean data.

4.2 Movie Metadata

We created the movie metadata set by aggregating information from different sources. The data set contains information for most major movies released in the USA since 2004, in total 2 964 movies with complete information and around 17 000 with partial information.

We used the following data sources to obtain movie metadata:

Wikidata was initiated by the Wikimedia Foundation to create a free and open knowledge base[2], whose data can be queried and accessed through a public API. We use Wikidata to get information about a movie's source material and to examine whether it is part of a series. If it has a source material, we extract the link to the corresponding Wikipedia article.

The Numbers[3] is a movie information website. We use movie budget and gross revenue information from The Numbers. In contrast to box office revenues, a project's budget is most often not reported publicly. Therefore, it is often an estimate or information leaked by business insiders. Researchers and practitioners widely use the budget information from The Numbers as a trusted data source (Google Scholar lists 408 publications on box office predictions referring to The Numbers).

The Internet Movie Database (IMDb)[4] is one of the most popular resources on movie business information. It contains information about 360 759 feature films and to over 7.6 million people involved in the movie industry. Unfortunately, there is no official API to access the information systematically. Hence, we use IMDb mainly as an additional source to verify our data. Additionally, we use box office and budget information from IMDb.

The Open Movie Database (OMDb)[5] is a is a free database to access movie information. Similar to IMDb, information is gathered by the user community, but in contrast to IMDb, it is a non-profit organization with a well-documented and freely available API. The Open Movie Database is the primary source for our movie metadata.

Google Trends and Wikipedia Access Statistics. Known success factors of movies are a famous cast, well-known source material and being the sequel to a popular series. It is difficult to measure these factors reliably, often proxies such as news coverage or star meter rankings, or academy awards are used as proxies. The task of estimating popularity gets even more complicated because the movies in our sample range over more than a decade, so it is not sufficient to measure the current popularity, but a trend over time is needed. In [23], Ripberger analyzes how well Google's search trends approximate more sophisticated attentiveness measures – collected in a manual and time-consuming way – and conclude that they converge over time.

Although Google Trends is a good popularity measure, the raw data has a calibration problem: Instead of showing absolute search

[2]Wikidata - Website: https://www.wikidata.org
[3]The Numbers - Website: http://www.the-numbers.com
[4]The Internet Movie Database - Website: http://www.imdb.com
[5]Open Movie Database - Website: http://www.omdbapi.com

volume, Google Trends displays search volume over time on a relative scale, making it impossible to compare the raw data of two search terms. Each one is relative to the highest search volume for the particular search term, and we do not know the absolute value.

To calibrate the Google Trends data and make it comparable between the individual entities we use Wikipedia page view statistics. These statistics show how often each Wikipedia article is accessed on a particular day. We calculate an average of the recent period and multiply this with the relative data we get from Google Trends.

Box Office Mojo is a is a box office revenue tracking website[6] widely used in the film industry. Box Office Mojo records the box office on a weekly basis for the USA and Canada (Domestic box office) and additionally for up to 50 other territories for the most popular films.

4.3 Data Cleansing

As described before, we built our data set in a semi-automatic way that leads to several sources of error. While we tried to eliminate most of them, others cannot be compensated for automatically. Therefore, we chose the following criteria for removing movies from our data set:

Non-US movies: Our model predicts the domestic box office of a movie[7]. Although many international productions are also released in the US, their commercial potential is systematically different. Often, they are targeted at the audiences of their country of origin. Let us take the example of the German movie "Suck Me Shakespeer"[8]. Although it was a major hit in Germany, it was never officially released in US cinemas. Still, at some point, the film might be shown in an American cinema, e.g., because of a special German language program. At this point, it would have, for example, a US Box Office result of USD 1 000. Having a budget of more than five million, but a neglectable domestic box office the movie would appear as a major flop. In our data set, we removed 313 movies for the reasons described in this part.

Movies with a budget of less than five million US Dollar: Generally, the available data quality is higher for bigger productions. While it is relatively easy to automatically gather information about movies with more than USD 20 million, it gets increasingly difficult below. We started with all movies with a budget of more than a million USD, but while manually verifying them we noticed that the error rate was too high. Although our model's predictions outperform benchmarking models on these movies we decided to remove them, because the errors dominate and render many evaluation metrics useless. This step removed 137 movies.

Movies with a too low revenue: We removed movies that grossed less than a twentieth of their production budget because we assume these numbers are due to data error. It is difficult to set a threshold to which extent a movie can flop and at which point we have to assume it is actually a data error. We researched major flops, and no reported movie came close to our threshold. One trustworthy list[9] depicts the biggest flops, based on worldwide loss. Most of these movies still grossed half of their expense, and only one grossed less than a seventh of the expense. Our data set contains only the films' production budgets that are only one part of the total expense. All in all, we believe that our threshold is very conservative and justifiable. In total, we remove 76 movies with a too low revenue in this step.

Movies with erroneous revenue data: We removed movies manually because their data is obviously false. In this step, we removed four movies.

We transformed categorical features using hot-one encoding.

5 IMPLEMENTATION

In this section, we show, which methods we have implemented and evaluated for each stage of our prediction pipeline.

5.1 Modeling Stage One: Taste Space

As described previously, we use a movie rating data set to infer the movie taste of individual users and to compute movie profiles for each movie. These profiles are n-dimensional latent topic vectors. For the user profiles each entry indicates a user's affinity to this topic and for movie profiles, it indicates how much the topic is present in the particular movie. This factorization approach to collaborative filtering has been proposed by [24] and gained popularity during the Netflix Prize competition [16].

5.1.1 Implementation & Evaluation. We implemented our matrix factorization approach by using Vowpal Wabbit[10] to compute a factorization of rank 10[11]. The resulting matrices serve as input into the user models and also allow the prediction of non-existing ratings. Figure 2 shows the accuracy of our recommender compared to other implementations. Please note that these numbers originate from studies with different data sets: The Netflix Prize data and the MovieLens data (see Section 4). Although the data sets are different, a baseline predictor, that predicts an average movie rating to each user has a very similar performance on both data sets. The historic Netflix Cinematch Recommender performs better than the baseline, the goal of the Netflix Prize was to increase the accuracy by 10% compared to Cinematch. Our implementation is better than Cinematch but worse than the winner of the Netflix Prize [1]. This performance is sufficient for our model, as the output of the recommender is only input to the user models.

5.2 Modeling Stage Two: User Models

For each user in our data set, we train a personal model that predicts how likely this user will see each movie in the cinema. The input for this modeling stage is the movie's metadata, the user's taste profile, and the movie profiles. The output is the estimated probability that the user will see the movie. As a first step, we used the user's profile and the movie profile to calculate an estimated rating of the movie and combine this rating with rating movie meta data to estimate the user's likelihood of the user seeing this movie. We implement

[6] Box Office Mojo - Website: http://www.boxofficemojo.com
[7] Domestic Box Office = US + Canada
[8] The German title of the movie is "Fack ju Göhte"
[9] The Numbers - Biggest Flops: http://www.the-numbers.com/movie/budgets/

[10] Vowpal Wabbit - Repository: https://github.com/JohnLangford/vowpal_wabbit
[11] We decided to use rank 10 as it provided the best balance between accuracy and dimensionality (potentially negative impact on following stages).

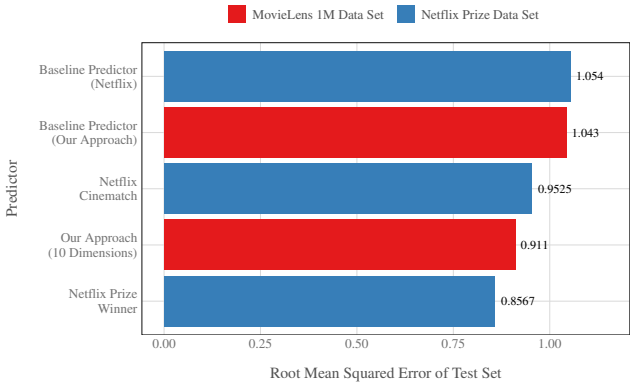

Figure 2: Comparison of various recommenders for MovieLens data set and Netflix Prize data set (cf. [28]).

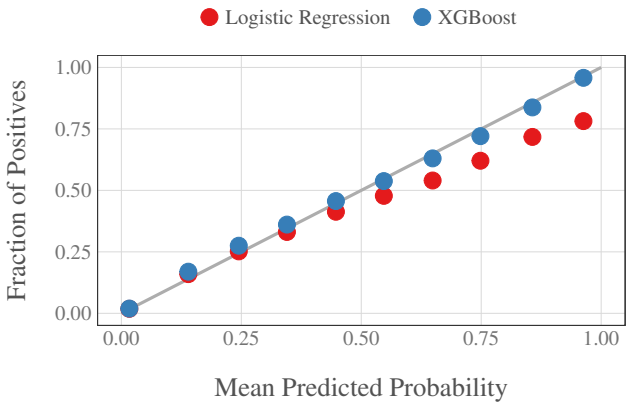

Figure 3: Calibration graph of the XGBoost (parameters: max-depth = 5, η = 0.05, rounds = 100) and logistic regression models. The gray line depicts a perfect calibration.

two models and evaluate them with the following evaluation metrics: logarithmic loss (i.e., LogLoss), McFadden's Pseudo-R^2, and precision/recall (using a cut-off of 0.5).

Logistic Regression: Logistic regression is a rather simple and robust regression-based model. We decided for logistic regression as it is a very robust model for predicting probabilities, and especially for its simplicity, which eases the interpretation of both the resulting model and the prediction results. In our use case, where we predict the likelihood of a particular user seeing a given movie in the cinema, a label of one represents that they have seen the film and a zero if not.

Gradient-Boosted Trees: For gradient boosted trees, we use XGBoost [2] and train a `binary:logistic` predictor, that uses binary training data to predict outcome probabilities. XGBoost offers different parameters to control the training and mitigate overfitting, while for our use case the maximum depth of individual trees, the number of trees in the ensemble, and the η parameter, which defines how much a single tree can influence the final prediction, are most important. We evaluated varying configurations of XGBoost. We achieved the best results for a maximum tree depth of five levels, an ensemble of 100 trees and a maximum weight per tree of 0.05.

5.2.1 Implementation & Evaluation. The logistic regression model performs well (precision = 0.614, recall = 0.428) with a LogLoss of 0.311 and a McFadden Pseudo R^2 of 0.181. Comparing the gradient-boosted trees to logistic regression shows both a lower LogLoss of 0.253 and an improved Pseudo R^2 value of 0.369 (recall = 0.36). These results show that the boosted trees are the best-performing approach of our user models. Figure 3 shows the calibration of the predictions made by the XGBoost models with the highest *PseudoR*^{2} score and the logistic regression. The calibration of XGBoost is slightly better than the logistic regression calibration.

Remark 1. *Gradient-boosted trees (e.g., using XGBoost) outperform all other evaluated approaches for the user models of stage two. Nonetheless, logistic regression is a valuable addition for two reasons: (i) it is very fast and easily applicable, and (ii) easier to understand and evaluate than other models. Especially the latter is important when gathering actionable insights. Furthermore, we evaluated feed-forward neural networks using resilient backpropagation with weight backtracking [22] and a logistic activation function. Both its accuracy as well as runtime performance have not been competitive.*

5.3 Modeling Stage Three: The Aggregation Model

In an ideal world, where we would have a perfect user model for each existing moviegoer we would not need a box office prediction model: We could simply sum up the probabilities, multiply them by the ticket price and get the box office estimate.

In the real world, we do not have a model for each moviegoer, and the moviegoer models are not perfect. To solve the first problem, we have to evaluate our data set. If we had a perfectly representative data set, we could still base our box office prediction on our sum of probabilities: We could simply multiply it by a constant. This constant symbolizes the factor our sample is smaller than the general population. In our case the data set is not representative: We must assume that our data set is heavily skewed.

We attribute a weight w_i to each user u_i in our data set. The weight is higher for users who are underrepresented in our data set and lower for users who are overrepresented. Assume that user data is heavily skewed towards young movie enthusiasts frequently watching movies. Therefore, we weight the influence of individuals towards the global prediction result. This approach is common in polling, but to attribute the weights, additional information about the users is needed, generally this is demographic data. Unfortunately, we do not have any demographic data about our moviegoers. As a consequence, we try to learn the weights from the data.

This process non-trivial to perform, because the data set contains more users than movies, so models have more input variables than training cases. Simply fitting a model to the data results in overfitting, so our box office aggregation models need strategies to mitigate overfitting. We implemented various strategies but measured the best results for non-negative least squares regression and

gradient-boosted trees. Both models have been strongly regularized in order to handle the vast number of features. Conceptually, instead of being dominated by frequently rating movie enthusiasts, we extract users that best represent the global prediction result.

Non-Negative Least Squares Regression: Positive coefficient regression is similar to ordinary least squares regression, but with the additional constraint that all coefficients have to be positive. Conceptually, it does not make sense to allow users to have a negative impact on the box office. We implemented the positive coefficient regression using the Lawson-Hanson algorithm [18] to compute the weights. An examination of the trained weights showed that it is zero for most users. Thus, only a few users' predictions are used for the box office predictions. While it might be a mathematically correct solution, it does not use all available information and is possibly affected by overfitting.

Gradient-Boosted Trees: We further implemented two models based on gradient-boosted regression trees. The high dimensional tree model uses the movie × user probability matrix as an input. The low dimensional tree model uses only the sum of the user probabilities, but additionally movie metadata to compensate for the non-representativity of the data set.

Our *High Dimensional Tree* model has several thousand input features: the estimated movie watching probability for each user. As the dimensionality is more than ten times larger than the number of training cases, it is crucial for this model to mitigate overfitting. We experienced the best results by limiting the depth of each tree to a maximum of five levels and reducing the maximum weight to $\frac{1}{20}$ and performing 200 rounds of boosting. The *Low Dimensional Tree* model uses movie metadata to compensate for the non-representativeness in the rating data set. To determine how much influence the user-based features have on the overall performance of this model, we implemented the exact same model but trained it only on movie metadata – this reduced benchmarking model is called *movie metadata tree*.

5.4 Runtime Performance

We use Vowpal Wabbit to factorize our rating matrix. The matrix decomposition with Vowpal Wabbit takes five to thirty minutes, depending on the number of users and movies. The training time of the user models depends on the algorithm: The logistic regression model is fast to train, it takes under five minutes to train all user models on a single core. The tree ensemble user models took between 5 to 15 minutes on five cores in parallel for the whole data set, depending on the parameters. The computation of the positive coefficient regression is fast, taking under a minute to train. The training of the low and high dimensional tree-based aggregation models is slower, it took 5 to 10 minutes to train them.

6 EVALUATION OF BOX OFFICE PREDICTIONS

In this section, we present the eventual results of our prediction pipeline: the predicted box office results. We use a ten-fold cross-validation to evaluate our models. Also, we kept a holdout for the final model evaluation which has not been used for any training. For each round of the cross-validation, we create a separate rating

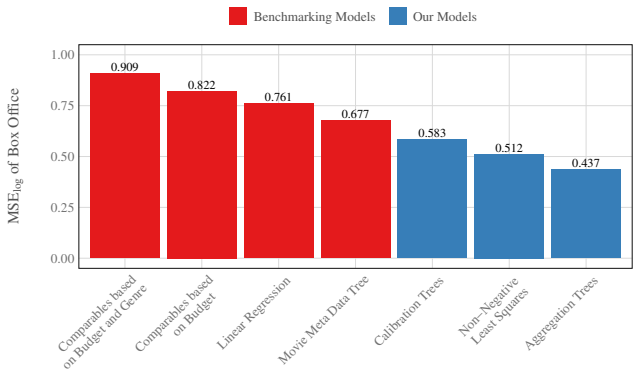

Figure 4: MSE_{log} of the box office revenues (millions). Benchmarking models work solely on movie metadata. Using user rating data improves prediction performance substantially.

data set. This data set consists of all ratings for the movies from the training set, and 80 ratings for each movie from the test data set. This resembles the situation of a test screening: The movie is shown to a small group of people, from their ratings we want to estimate the movie profile.

After creating the test and training data sets, we perform the matrix factorization, train the user models, and the aggregation models. Finally, we compare the predicted results of the aggregation models with the true outcome of the movies in the test set.

To evaluate our models, we will use the mean squared error of the logarithm (MSE_{log}) of the domestic box office (as used in [6] and [20]). We are aware that this metric is debatable, but decided for it over other alternatives such as *mean absolute percentage error (MAPE)* in order to be comparable with previous work.

6.1 Benchmarking Models

We implemented several benchmarking models. The industry standard is the so-called *Comparables* model that is used to get an early estimate of a movie's box office potential. We implemented the same two approaches based on comparables as described in [6]. The first one estimates a movie's budget using the ten closest movies and averaging their box office revenues, the second only considers movies with the same genres and proceeds as described before.

The second benchmarking model is a basic *Linear Regression* model using the movie metadata as input variables and the box office revenue as the target. This model resembles early linear regression models (cf. [19, 20]).

The third benchmarking model is a *tree ensemble*, more precisely an XGBoost model using movie metadata as input. This tree ensemble is the same as the low dimensional tree model, but it is trained without user data, only on movie metadata. By comparing the two models, we can analyze, whether the performance increases as user-based information is included.

6.2 Results

Figure 4 shows the predictive performance of all evaluated approaches. The best predictor is the high dimensional tree model that works purely on the user-based data achieving an MSE_{log} of 0.437 on the box office (in millions USD). In comparison, the best model without user data amounts to an MSE_{log} of 0.677. It is also interesting to note that the low dimensional tree, a model that works on movie metadata and aggregated user data, marks the transition between metadata-based models and user-based models, conceptually and performance-wise. The non-negative least squares approach achieves an MSE_{log} of 0.512. The low dimensional tree ensemble achieves the worst performance of all user data models. This model is trained on the movie metadata, movie profiles and the summed up watching probabilities of all users.

Looking at the three benchmarking models we implemented, the tree-based approach performed best. Notably, the comparables approach performs worse than expected. Eliashberg, et al. [6] reported that comparables based on the budget and genre outperform linear regression models. We have not been able to confirm this finding. We observed that comparables based on the budget perform well for larger movies but they are not able to handle the high volatility of smaller productions.

The linear regression results depend heavily on a good feature selection. Fitting the model with all features results in an inferior performance. To achieve our current result, we used only the most significant features, namely budget, age ratings, and popularities. To reduce the number of features, we built aggregates, e.g., we represented the cast popularity as a single number by calculating a weighted sum. We also summed up the popularities of the series, the source material, etc.

The tree ensemble trained solely on movie metadata was the best performing benchmark predictor. For this model, we did not pre-aggregate or remove features, as the underlying implementations seem to cope well with the task of feature selection. We did not measure large differences between the training set performance and test set performance, which shows that the model does not overfit and generalizes well.

Remark 2. *The performance of models that use user rating data is superior to the performance of models that solely use movie metadata. Every single model using user rating data performs better than even the best-performance benchmarking model.*

6.3 Comparison to Published Results

The kernel-based approach by Eliashberg et al. (cf. Section 2) is one of the most recent publications [6]. Their results are the best we found for an early box office prediction model. It is very difficult to compare results from one study to another, especially without knowing the exact data used. A good example is comparables as a benchmarking predictor. While it achieves an MSE_{log} of 0.822 on our whole data set, it improves drastically if we only take movies with a budget above USD 100 million into account. In this group of movies, *Comparables based on Budget* achieves an MSE_{log} of 0.54. The accuracy of our data set is very different from the accuracy Eliashberg et al. reported: The comparables based only on the budget predictor achieved an MSE_{log} of 0.68 in their data set [6]. This could indicate that our data set is more diverse.

Although the comparison is difficult, we relate our result to theirs by calculating the percentage of MSE_{log} decrease over the comparables by budget benchmark. Eliashberg et al. achieved a decrease of 43.63% over the comparables-based on the budget approach. We achieved a decrease of 46.84% over the same approach using our data set which we deem to be comparable.

Sharda and Delen classified movies into one of nine classes, from flop to blockbuster with a neural network (cf. Section 2, [26]). They correctly classify 36.9% of the movies in their data set, and in 75.2% of the cases, they predict either the exact class or one above or below. We have been evaluating our models in the same fashion: We classify both, the true revenue and our prediction into the categories used by Sharda and Delen. Then we calculate the distance between both. Without optimizing our models towards classification, we achieve better results. The high dimensional tree model classifies 38.7% of the movies into the right category and 81.6% of the movies have been classified either correctly or one above or below. Unfortunately, we have not been able to compare our results with further published studies due to missing benchmarking models or insufficient published information.

7 CONCLUSION

Predicting the box office is an economically important but challenging task where existing prediction models lack sufficient accuracy.

We have introduced a novel modeling approach that adopts concepts and algorithms developed for recommender systems. We focus less on the sophistication of prediction or learning algorithms and employ robust and well-known techniques, which explicitly leave room for tuning and further improvements. The focus lies on transporting recent developments to an important domain – box office predictions – and introduce an user-centered pipeline to improve accuracy by using movie ratings as a starting point.

Our approach incorporates a publicly available source of information: large data sets of movie ratings from moviegoers. It allows us to model movie-going behavior on a per-user level, which increases prediction accuracy significantly and additionally allows fine-granular target group analyses that can lead to actionable insights for movie business professionals.

We have evaluated our user-based models against metadata-based benchmarking models and have shown that prediction performance improves by adding user-centered data to box office prediction models. While our pipeline adds a significant computation overhead with the creation of user taste and movie profiles, we think this overhead is justified by the substantially improved accuracy. The best model – the *high dimensional trees* – have a 46.8% lower MSE_{log} than the *comparables-based* benchmarking model, a standard approach used as an early estimator in industry.

REFERENCES

[1] Robert M. Bell and Yehuda Koren. 2007. Lessons from the Netflix prize challenge. *SIGKDD Explorations* 9, 2 (2007), 75–79.

[2] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 785–794.

[3] Arthur De Vany and W. David Walls. 1999. Uncertainty in the Movie Industry: Does Star Power Reduce the Terror of the Box Office? *Journal of Cultural Economics* 23, 4 (1999), 285–318.

[4] Dursun Delen, Ramesh Sharda, and Prajeeb Kumar. 2005. Movie forecast Guru: A Web-based DSS for Hollywood managers. *Decision Support Systems* 43, 4 (2005), 1151–1170.

[5] Anita Elberse and Jehoshua Eliashberg. 2003. Demand and Supply Dynamics for Sequentially Released Products in International Markets: The Case of Motion Pictures. *Marketing Science* 22, 3 (2003), 329–354.

[6] Jehoshua Eliashberg, Sam K. Hui, and Z. John Zhang. 2014. Assessing Box Office Performance Using Movie Scripts: A Kernel-Based Approach. *IEEE Transactions on Knowledge and Data Engineering* 26, 11 (2014), 2639–2648.

[7] Jehoshua Eliashberg, Jedid-Jah Jonker, Mohanbir S Sawhney, and Berend Wierenga. 2000. MOVIEMOD: An implementable decision-support system for prerelease market evaluation of motion pictures. *Marketing Science* 19, 3 (2000), 226–243.

[8] Jehoshua Eliashberg, Charles B. Weinberg, and Sam K. Hui. 2008. *Decision Models for the Movie Industry*. Springer US, Boston, MA, 437–468.

[9] Stephen Follows. 2016. Do Hollywood movies make a profit? https://stephenfollows.com/hollywood-movies-make-a-profit/. Accessed: 2018-05-24.

[10] Jannis Funk. 2013. *On Predicting Box Office Academic and Business Approaches*. Master's thesis. Film University Babelsberg Konrad Wolf.

[11] Allègre L. Hadida. 2009. Motion picture performance: A review and research agenda. *International Journal of Management Reviews* 11, 3 (2009), 297–335.

[12] F Maxwell Harper and Joseph A Konstan. 2015. The MovieLens data sets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4 (2015), 19:1–19:19.

[13] Thorsten Hennig-Thurau, Mark B Houston, and Shrihari Sridhar. 2006. Can good marketing carry a bad product? Evidence from the motion picture industry. *Marketing Letters* 17, 3 (2006), 205–219.

[14] Thorsten Hennig-Thurau and Oliver Wruck. 2000. Warum wir ins Kino gehen - Erfolgsfaktoren von Kinofilmen. *Marketing ZFP* 22, 3 (2000), 241–258.

[15] J. Morgan Jones and Christopher J. Ritz. 1991. Incorporating distribution into new product diffusion models. *International Journal of Research in Marketing* 8, 2 (1991), 91–112.

[16] Yehuda Koren, Robert M. Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *IEEE Computer* 42, 8 (2009), 30–37.

[17] Brent Lang. 2015. Variety – Is Hollywood Making Too Many Movies? http://variety.com/2015/film/news/hollywood-making-too-many-movies-1201526094. Last accessed: 2018-05-24.

[18] C. Lawson and R. Hanson. 1995. *Solving Least Squares Problems*. Society for Industrial and Applied Mathematics.

[19] B R Litman and H Ahn. 1998. Predicting Financial Success of Motion Pictures: The Early 90's Experience. In *The Motion Picture Mega-Industry*.

[20] S Abraham Ravid. 1999. Information, blockbusters, and stars: A study of the film industry. *The Journal of Business* 72, 4 (1999), 463–492.

[21] Steffen Rendle. 2012. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)* 3, 3 (2012), 57.

[22] Martin Riedmiller and Heinrich Braun. 1993. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In *IEEE International Conference On Neural Networks*. IEEE, 586–591.

[23] Joseph T. Ripberger. 2011. Capturing Curiosity: Using Internet Search Trends to Measure Public Attentiveness. *Policy Studies Journal* 39, 2 (2011), 239–259.

[24] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2000. Application of Dimensionality Reduction in Recommender System - A Case Study. (2000).

[25] Mohanbir S Sawhney et al. 1996. A Parsimonious Model for Forecasting Gross Box-Office Revenues of Motion Pictures. *Marketing Science* 15(2) (1996), 113–131.

[26] Ramesh Sharda and Dursun Delen. 2006. Predicting box-office success of motion pictures with neural networks. *Expert Systems with Applications* 30, 2 (2006), 243–254.

[27] Jason E. Squire. 2017. *The Movie Business Book* (fourth ed.). Routledge.

[28] Andreas Töscher, Michael Jahrer, and Robert M Bell. 2009. The BigChaos Solution to the Netflix Grand Prize. *Netflix Prize Documentation* (2009), 1–52.

[29] F Zufryden. 1996. Linking Advertising to Box Office Performance of New Film Releases. *Journal of Advertising Research* 36, 4 (1996), 29–41.