American University of Armenia

Zaven & Sonia Akian College of Science and Engineering

CS 251/340, Machine Learning Course Project

Movie Rating Prediction through Topic Modeling

Hermine Grigoryan

Spring, 2022

Abstract

The movie industry is growing in a rapid pace with billions of dollars of investments. Therefore, it is of utmost importance to predict the success of a particular film. In this paper, we build machine learning models to predict the success of a film through IMDb ratings. Boosting methods, such as XGBoost and CatBoost are employed, and the results of the models are compared using ROC-AUC metric. Besides including the given features in the models, we create an additional feature based on the topic of the movie synopsis using BERTopic framework. The ROC-AUC scores of the models range from 0.79 to 0.82 with no additional power gained with the use of topics.

Keywords: IMDb, movie rating, classification, boosting, topic modeling, BERTopic

Contents

List of Figures

List of Tables

Introduction

In this digitally and socially connected world, many people leave a trace in the digital world to express their opinions and views on different products and services. The movie industry is among the spheres where users leave massive amounts of data, such as ratings and reviews. IMDb (Internet Movie Database) is one of the resources that provides information about movies and television programs [Imd]. This source can be used to analyze and predict the success or failure of a particular film using different attributes.

The objective of this research is to create a classification model which will be able to predict the rating of a movie (on a scale from 0 to 10) based on the factors, such as the cast, director, genre, duration of the film, number of votes, release date, synopsis, etc. This work can be used by movie creators to understand how successful a particular movie will be, and eventually find a mix of attributes that will provide a "formula for success" in the film industry.

In this paper, we use machine learning boosting methods to predict the film rating using the data scraped from imdb.com. Throughout the process of feature engineering, encoding, training and validating the models, we employ different techniques, such as topic modeling, to make use of the textual and categorical features.

The paper is organized as follows. The second section is devoted at discussing the available literature in predicting the movie ratings. Section 3 summarizes the data used, preprocessing steps and exploratory data analysis. The fourth section describes the theoretical framework and approaches that were employed in the paper. Finally, the last section summarizes the main results, along with concluding remarks.

Literature Review

Many authors have proposed different approaches and models for predicting the movie rating. Mhowwala et al. [MRD20] use the data from IMDb, YouTube and Wikipedia to predict the rating by Random Forest and XGBoost.

Dhir and Raj [DR18] performed a classification task and used a dataset from kaggle.com on IMDb ratings (5,043 observations). They have used classification models, such as k-nearest neighbors (KNN), Gradient Boosting, Random Forest, Support Vector Classifier, and AdaBoost. Among the models, Random Forest Classifier outperformed on the testing set with 61% general accuracy.

Hsu et al. [HSX14] have analyzed the data of 33,000 movies and developed the following models: linear combination model, multiple linear regression and neural networks. Among the models, the neural networks were able to achieve the highest accuracy. As an accuracy measure the authors have used Average PAE (Prediction Absolute Error - |*Predicted user rating - Actual user rating*|).

Another interesting paper by Karpov and Marakulin [Kar21] discusses the improvement of the movie rating prediction accuracy by introducing features of Social Network Analysis ("actor" - "casting director" - "talent agent" - "director" communication graph). The results have demonstrated that social network information can improve classification accuracy from 4 to 6 % depending on the classification method.

Data and Preprocessing

0.1 Dataset

The data for the analysis was obtained by scraping the imdb.com website. The data contains information on movies that are released between 1950-2022. As the overall number of movies released during that period is very large, and a lot of resources were required for scraping, it was decided to scrape the most popular 450 movies released for each month.

After a careful and detailed examination of the raw data, a sequence of preprocessing techniques was employed to manipulate and clean the data to use for predictive modeling. Non-relevant variables were discarded, new features were developed, and some of the existing variables were transformed. Distributions of variables were examined to identify outliers and understand the spread of the data. Also, missing values were filtered out from the final dataset.

The final analysis data includes around 180,000 observations with 11 variables. The next section describes the features and the preprocessing steps.

0.2 Feature Description, Feature Engineering and Preprocessing

As discussed in the previous subsection, 11 features are used throughout the analysis.

- **IMDb rating** *(Float)* - IMDb rating is a weighted average of an undisclosed calculation method given by users from a scale between 0 and 10. This variable is the target variable used for predictions. In order to use it in the classification setting, the rating was divided into classes: rating from 0-1, 1-2, 2-3, ... , 9-10.

- **Actors** *(List[Str])* - top 4 actors in a movie. This variable is transformed into 4 columns (*actor 1*, *actor 2*, ..., *actor 4*), assuming that the order and the importance of the actors is preserved in the list. Then, label encoder is applied to the variables so that it is possible to use them in the models.

- **Director** *(Str)* - movie director. Label encoder is applied to the variable.

- **Genre** *(Bool)* - genre of a movie. One movie can fall into several categories. Multi-label binarizer was used to encode this variable.

- **Movie duration** *(Int)* - movie duration in minutes.

- **Number of votes** *(Int)* - number of people that have voted.

- **Release year** *(Int)* - release year of the movie in the range from 1950 to 2022.

- **Release month** *(Int)* - release month of the movie.

- **TV Series** *(Bool)* - a variable indicating whether the movie falls in the category of TV series or not.

- **Synopsis** *(Str)* - a short description of the movie.

- **Title** *(Str)* - the movie title.

In order to be able to incorporate the synopsis of a movie in the models, we use the labels obtained by topic modeling. Topic modeling is an unsupervised learning method, similar to clustering, which collects the texts into similar groups. The description of the algorithms will be provided in the next sections.

0.3 Exploratory Data Analysis

Exploratory data analysis is conducted using visual analysis. Figure 1 represents the distribution of IMDb scores. We can notice that the scores are almost-normally distributed with $\mu = 6.23$ and $\sigma = 1.31$.

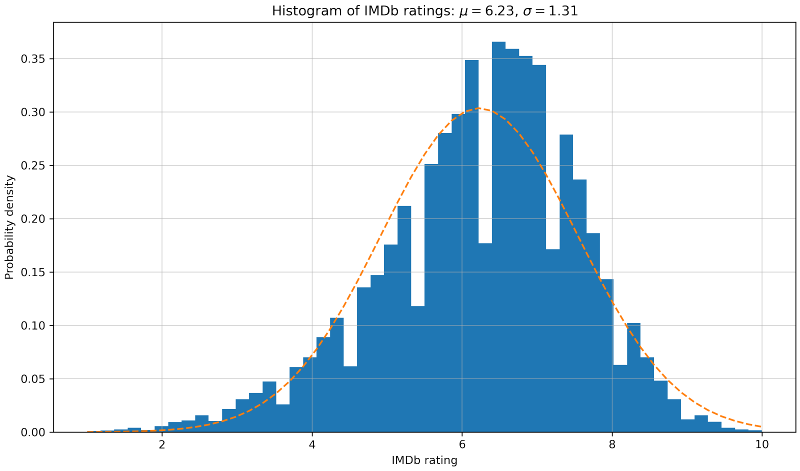

Figure 1: Distribution of IMDb ratings

The next figure represents the relationship between the number of votes and IMDb rating. It can be observed, that the features are not highly correlated with each other.

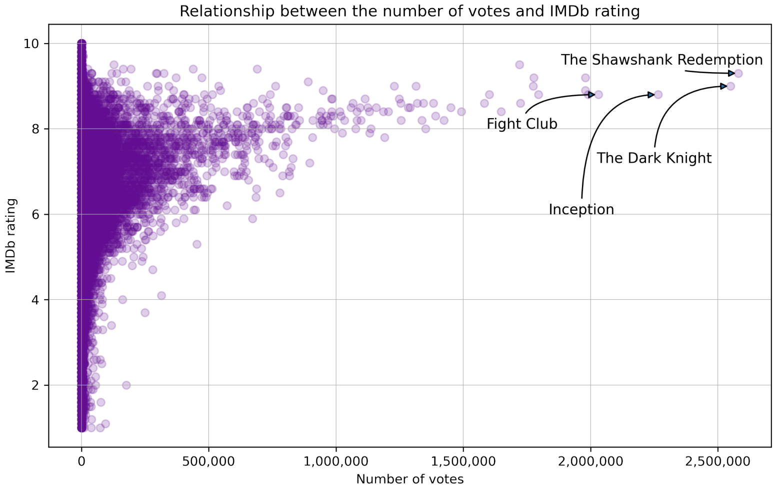

Figure 2: Votes vs IMDb ratings

Algorithms and Models

0.4 Algorithms and Models

In most of the machine learning competitions, ensemble methods, such as random forests and gradient boosting algorithms demonstrate impressive results. In order to predict the IMDb ratings, we will also employ ensemble learning methods. Particularly, we will use XGBoost and CatBoost frameworks.

XGBoost (eXtreme Gradient Boosting) [CG16] is an ensemble learning method since it combines weak learners and uses the output of many models in the final prediction.

CatBoost [Ver18] is another gradient boosting open-source framework. The advantage of the CatBoost frameworks is that among other features, it tries to solve for categorical and textual features using automated feature encoding techniques.

In line with using the supervised classification algorithms, we use unsupervised learning methods, particularly topic modeling, to make use of the textual features in the data set. One of the methods to find latent topics in the text is by Latent Dirichlet Allocation (LDA) algorithm. Using this model on our data did not produce any meaningful results, that is why we have decided to use BERTopic [Gro22]. BERTopic is a topic modeling method that uses BERT embeddings and a class-based TF-IDF to create dense clusters. BERTopic produces interpretable topics and keeps only important words in the topic descriptions.

In this work, we use BERTopic to find clusters of movies, and then use those cluster labels as an additional feature for the classification task.

0.5 Performance Measurement

In order to measure the performance of the classification models, we will use ROC-AUC as the main accuracy metric. Four models will be compared to each other using this metric:

1. XGBoost with the encoded features
2. XGBoost with the encoded features with the addition of topic labels
3. CatBoost with the original textual and categorical features
4. CatBoost with the original textual and categorical features with the addition of topic labels

Results and Conclusion

0.6 Experiments and Results

For training the models, we have randomly divided the data into training and testing sets (80-20%). To choose the hyper-parameters of models, we use grid search with 5-fold cross validation.

As an experiment, we have decided to incorporate the synopsis of movies in the models as well. The first option was to train models without the inclusion of the synopsis. Secondly, as mentioned in the previous section, we have used BERTopic to get clusters of movies that are similar to each other, and incorporate the labels as an additional feature in the models. After a sequence of text preprocessing steps, such as the removal of stopwords and lemmatization, we run the BERTopic model with default parameters. As a result, for each movie, we obtain a particular interpretable topic. Overall, we have got 142 unique topics. One of the topics has label -1 indicating outlier movies, i.e., movies that do not fall into any other topic. The first 12 topics are summarized in the figure below.

One of the features of the BERTopic model is finding a topic by providing a search term. For example, inputting a term *movie about animals*, the model outputs that with 0.412 probability, the search term "*movie about animals*" falls into "*elephant animal lion zoo*" topic.

By visually observing the topics and comparing the topic labels with the synopsis, we can see that sometimes the topics describe the movies excellently. However, the accuracy of the model is not always high.

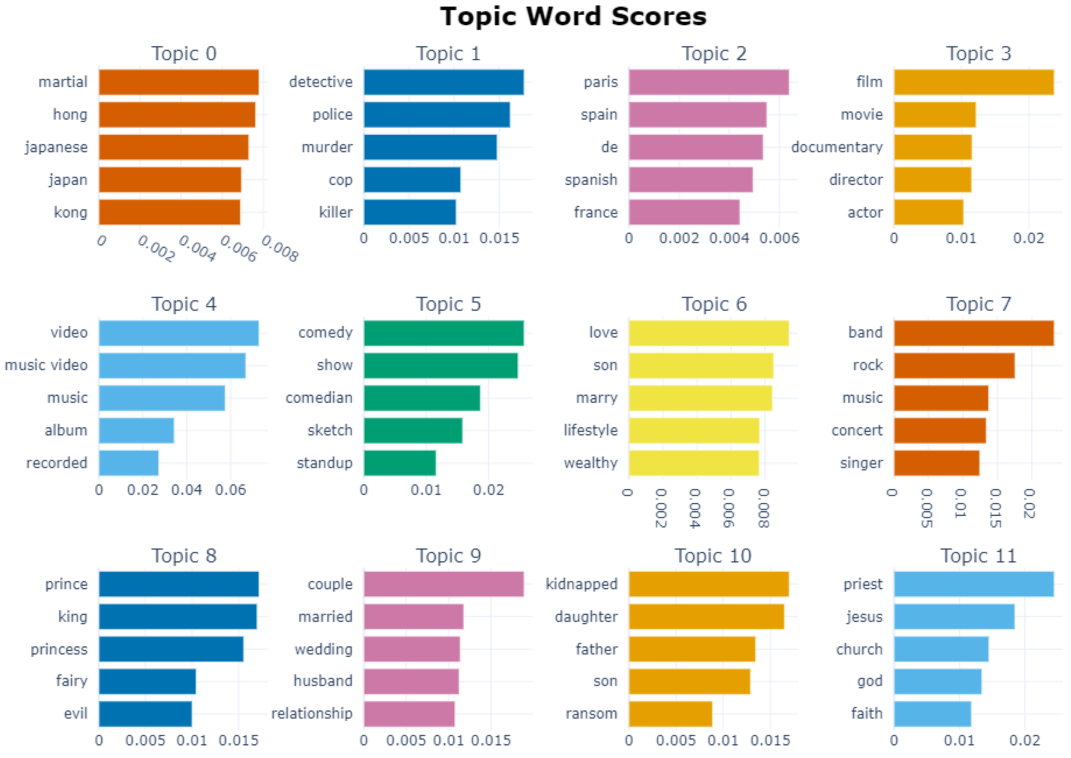

Figure 3: Barchart of important words in each topic

After getting the movie topics, we run classification models (XGBoost and CatBoost) with and without the feature of the topic. Because of computation cost, we decided to tune only 2 parameters both for XGBoost and CatBoost. The best parameters for each model are summarized in the table below.

Parameter	**Parameter Description**	**Model**	**Best value**
n_estimators	Number of gradient boosted trees. Equivalent to number of boosting rounds.	XGB with topics	100
n_estimators	Number of gradient boosted trees. Equivalent to number of boosting rounds.	XBG without topic	100
max_depth	Maximum tree depth for base learners.	XGB with topics	5
max_depth	Maximum tree depth for base learners.	XGB without topics	4
learning_rate	The learning rate. Used for reducing the gradient step.	CatBoost with topics	0.5
learning_rate	The learning rate. Used for reducing the gradient step.	CatBoost without topics	0.5
depth	Depth of the tree.	CatBoost with topics	5
depth	Depth of the tree.	CatBoost without topics	4

Table 1: Best hyper-parameters for each model

As an accuracy measure, we have used ROC-AUC metric. As can be noticed from table 2, the results of models do not vary much. Particularly, for the XGBoost and CatBoost models, the inclusion of topics did not increase the model accuracy, however, CatBoost provides slightly better results.

ROC-AUC for models		
	Topic model included	Topic model excluded
XGBoost	0.799	0.799
CatBoost	0.818	0.815

Table 2: Cross-validated ROC-AUC scores for different models

0.7 Conclusion

The goal of this research project was to predict the success of a film through IMDb ratings using machine learning models. The data for the analysis was scraped from imdb.com and 180,000 observations with 11 variables were considered. We have used gradient boosting methods, particularly XGBoost and CatBoost. In addition to using the given features, we have used BERTopic, a topic modeling framework, to group similar movies (based on the synopsis) and included the labels as an additional feature. As a result, the topics did not add additional power to models.

For the future work, it would be appropriate to further explore the topic modeling methods and use different parameters for training. This could help achieving better separated topics, hence higher classification accuracies for the models.

In addition, the training of the models can be completed using a more powerful computer. It will allow to increase the number of iterations, do hyper-parameter tuning using more parameters, etc.

Appendices

0.8 GitHub Repository

The scraping, exploratory data analysis and modeling codes are written in Python and shared in GitHub[1].

[1] https://github.com/HermineGrigoryan/imdb

Bibliography

[HSX14] Ping-Yu Hsu, Yuan-Hong Shen, and Xiang-An Xie. “Predicting movies user ratings with Imdb attributes”. In: *Rough Sets and Knowledge Technology* (2014), 444–453.

[CG16] Tianqi Chen and Carlos Guestrin. “XGBoost: A Scalable Tree Boosting System”. In: KDD ’16 (2016), pp. 785–794.

[DR18] Rijul Dhir and Anand Raj. “Movie success prediction using machine learning algorithms and their comparison”. In: *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)* (2018).

[Ver18] Ershov Gulin Veronika Dorogush. “CatBoost: gradient boosting with categorical features support”. In: *arXiv preprint arXiv:1810.11363* (2018).

[MRD20] Zahabiya Mhowwala, A. Razia, and Sujala D. “Movie rating prediction using ensemble learning algorithms”. In: *International Journal of Advanced Computer Science and Applications* 11.8 (2020).

[Kar21] Marakulin Karpov. “Social Network Analysis of the Professional Community Interaction - Movie Industry Case”. In: *arXiv preprint arXiv:2109.01722v1* (2021).

[Gro22] Maarten Grootendorst. “BERTopic: Neural topic modeling with a class-based TF-IDF procedure”. In: *arXiv preprint arXiv:2203.05794* (2022).

[Imd] *Ratings, reviews, and where to watch the best movies amp; TV shows.*