TMDB Box Office and IMDB Prediction

Enes YÜKSEKKOLAŞİN
TOBB University of Economics and Technology, Turkey
eyuksekkolasin@etu.edu.tr

Gizem ELÖVE
TOBB University of Economics and Technology, Turkey
gelove@etu.edu.tr

Mehmet Mustafa AYMAYAN
TOBB University of Economics and Technology, Turkey
maymayan@etu.edu.tr

Rabia ARSLAN
TOBB University of Economics and Technology, Turkey
rarslan@etu.edu.tr

ABSTRACT

For the audience the most important element that makes a film precious is appealing of the film. However, for the movie producers the most important thing is the revenue of the movie. IMDB score is one of the most important factors affecting the number of views of films. Revenues of films can be estimated by examining the characteristics of previously published films. We can estimate IMDB scores with the help of past films. Some attributes that affect the revenue and IMDB are the cast, crew, plot keywords, budget, posters, release dates, languages, production companies, and countries. These estimations can be done through typical machine learning algorithms. Hence in this study, revenue and IMDB estimation would be made by a machine learning model which uses all these attributes.

CCS Concepts

• **Mathematics of Computing** →Probability and Statistics
• **Computing methodologies**→ Artificial Intelligence

1. INTRODUCTION

Machine learning algorithms are used in data mining studies on metadata. In the sectors as marketing, banking, insurance and medicine there are various data mining techniques that are used for solving problems. Film industry is also a sector where big data could be processed. One of the popular researches of the film industry is about the profits of films. A successful movie generally has a strong scenario, accomplished producer, celebrated cast, famous director and strong visual effects. The more successful movie means the more amount of profit. This paper would cover the problem about cinema data. The cinema data contains information about the features of previous films such as the cast, director, budget and the box office revenues of these films. In this study an estimation would be made about box office revenues of new movies by using this data. For the producer companies, actors and movie directors, this estimation work is very important and informative since the risk analysis and budget planning could be made by using these informations. IMDB scores directly affect what people will follow. In this case, determines the importance of films in the sector. The majority of people do not want to watch movies with low IMDB scores. It is directly related to many issues in the sector. By using this study, the financial losses in the films can be largely prevented. The research questions of this study are “How to estimate revenue from available attributes?” and “How to estimate IMDB score from available attributes?”. The algorithms used in developing the model are a kind of Gradient Boosting (GBM) algorithm. Gradient Boosting (GBM) is a algorithm in which predictions are made sequentially, not independently. Gradient Boosting works sequentially and trains the model.Gradient Boosting (GBM) uses the logic that subsequent estimators learn from the mistakes of previous estimators[1].We used gbm based algorithms to train our model. The algorithms used are CatBoost, Light GBM and XGBoost. These algorithms were sequentially tried and the best results were calculated by adding the results to the average with different weights. In this study, we will examine how the attributes and algorithms added together with the developing model increase the performance of the model with quantitative data. The performance of the model is

shown step by step by comparing the data in the tests to be performed.

2. METHOD

2.1. Dataset

The dataset which is used in the project is taken from a competition in Kaggle[2]. There are 7399 movies and their metadata in the dataset. In training data; id, belongs to collection, budget, genres, homepage, imdb id, original language, original title, overview, popularity, poster path, production companies, production companies, release date, runtime, spoken languages, status, tagline, title, keywords, cast, crew and revenue are given columns. There is no revenue information in the test data and results will be controlled when the project is submitted. When the data set is examined, it is observed that there are missing and noisy data. Therefore, to increase prediction rate for revenue data will be preprocessed.

2.2. Data Preprocessing

Our dataset has missing values and we need to make some processes before develop a prediction model. We have used gradient boosting model "xgboost" to deal with missing values. This model has a feature about replacing missing values and use this feature in the training stage and we train an xgboost model to replace. Therefore, the dataset was more clear.

2.3. Data Analysis

To take information about correlations between attributes, we create a correlation matrix.

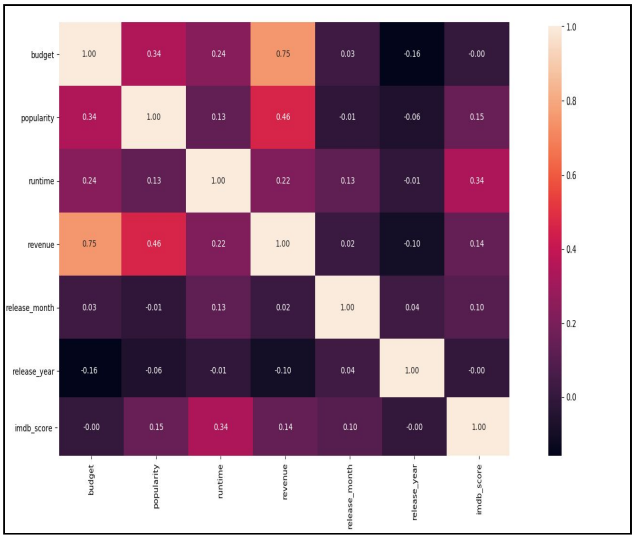

Figure 1 TMDB

When the correlation matrix is examined in figure 1, the correlation between budget and revenue is observed. The correlation is 75%. And the other important correlation is popularity-revenue correlation. This correlation matrix shows that the more popularity of movie makes more profit of movie.

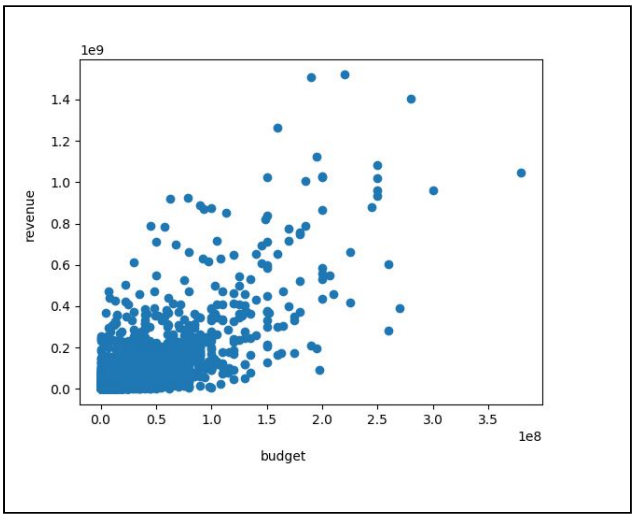

Figure 2 TMDB

Budget and revenue are highly correlated attributes. In figure 2, the relationship between the two attributes is seen. This situation shows that one variable could cause or depend on the values of another variable.

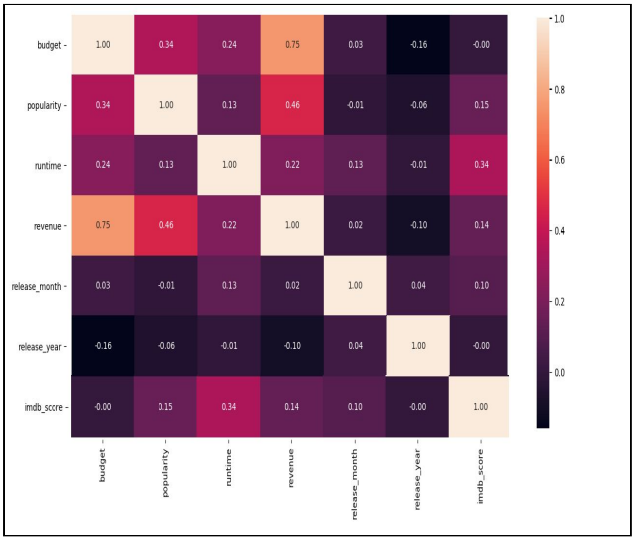

Figure 3 IMDB

When the correlation matrix is examined in figure 3, the correlation between all attributes are seen.

Production companies, production countries, language, cast and crew attributes have the most impressive effects on the revenues and IMDB scores of the movies.

2.4. Feature Engineering

In this section, we describe the preprocessing steps on the attributes in our dataset.

id : Since the attribute is unique for all sample it is dropped from the data set.

belongs_to_collection : In this attribute, the id, name, poster path and backdrop path information of the movie are kept as a json array. However, poster path and backdrop path are not distinctive features for our model. Therefore, we have created two new attributes named collection_name and has_collection instead of belongs_to_collection.

- **collection_name :** If our movie is included in a collection (if the json array is not hollow), our new attribute collection_name gives us the label encoded integer of name of the collection to which it belongs.
- **has_collection :** Because the attribute named belongs_to_collection corresponds to a json array, if its length is greater than 0, it takes the values 1 otherwise takes 0 as binary

budget : Feature includes budget information about movies. Range of this attribute is significantly large. Because of that, a new feature created called log_budget.

- **log_budget :** When we look at the distribution of revenue or budget we saw that those values were over a range covering many orders of magnitude, 1 to 1,000,000,000. There are also a lot more data points clustered at the low end of that scale. By using the log of revenue or budget we spread out those values much more evenly along the log scale. That makes it easier for use to visualise and easier for our models to fit.

genres : In this attribute, id and name information is kept. The type of subject of a movie will directly affect the revenue value as it will greatly affect the number of views.

- **num_genres :** This feature is created to keep number of genres that the movie belongs to.
- **all_genres :** This feature is created to keep label encoded format of sorted names of genres that the movie belongs to.
- **genre_<name> :** We extract these features to have a binary representation of the most common 15 genre names such as “drama”, “comedy”, “thriller”, “romance”, etc.

homepage : In this attribute, there is the official website address of the movie. When this feature is examined, it is seen that having or not having a website address rather than the website address has an impact on revenue. For this reason, we created a new feature named has_homepage.

- **has_homepage** : If there exist any home_page link, it takes the values 1 otherwise takes 0 as binary

imdb_id : Since the attribute is unique for all sample it is dropped from the data set.

original_language : This feature contains information about main language of movies. We encoded the original language string with an integer.

original_title : Since the attribute is not informative for our model it is dropped from the data set.

overview : The short description about the movie is kept in this feature. We used the vector which includes the words of overview to use in our predictor model.

popularity: Feature includes popularity score about movie. This information is not valuable when predicting revenues of movies.

poster_path : Contains the URL of the movie poster. We have seen that this information does not have effect on the value as a result of our experiments, and therefore we have removed it from the data.

production_companies : In the attribute id and name information is kept. The attribute informs about the movies’ production companies. The first 30 movies with the most revenue are examined and it is seen that some production companies dominate over the other companies. By making inferences, it is thought that producer company would have an impact on the revenue value of the movies. Therefore, we created new features.

- **num_companies :** This feature has the number of companies that has role in the movie.
- **all_production_companies :** Used to keep all of the company names with alphabetical order.
- **production_company_<name> :** We extract these features to have a binary representation of the most common 30 company names such as “Walt Disney Pictures”, “Paramount Pictures”, etc.

production_countries: : In this attribute iso 3166 country code and country name information is kept. The attribute informs about production country of the movies. The first 20 movies with the most revenue are examined and it is seen that some production countries dominate over the other countries. We extract two feature from this feature.

- **num_countries :** The number of production countries of the movie.
- **all_countries :** Used to keep all of the production country names with alphabetical order.

release_date : The release date of movie is kept in this attribute. Generally in some specific months and years, the revenues of movies change upward/downward. Yet the day of month is not informative. Therefore we extract two attribute named "release_month" and "release_year" from release_date.

- **release_month :** The release month is kept as an integer between 1 and 12 in this feature.
- **release_year :** The release year is kept as an integer value in this feature.

runtime : The attribute contains the duration time of the movie as a double value.

spoken_languages : This feature includes the languages that are spoken in the movie with iso_639 language code and language name. We use the number of languages in the movie by num_languages attribute and all of the language names in alphabetical order by all_languages attribute.

status : Since the attribute is not informative, it is dropped.

tagline : This feature has a tag that includes general information about movie. Since this is a textual attribute, words are weighted by using TF-IDF. At the beginning, we kept the number of words yet, this number is not informative about revenue prediction. Therefore, we do not keep the number.

title : This feature is title of the movie. Since this is a textual attribute, words are weighted by using TF-IDF. At the beginning, we kept the number of words yet, this number is not informative about revenue prediction. Therefore, we do not keep the number.

keywords : The keywords about the movie and their ids are kept in this feature.

- **num_keywords :** This feature has the number of keywords for each movie.
- **keyword_<keyword> :** For all most common keywords we create new attributes.

cast : This feature contains information about the actors who play in the movies. Name and gender information of the players are kept. Most common 15 cast members were selected and a new attribute has been created from this attribute.

- **num_cast :** The number of cast is kept this feature
- **genders_<1-2>_cast :** We create a feature that keeps gender for each cast person.
- **cast_name_<real name> :** This feature keeps the names of most popular actors.
- **cast_character_<character name> :** The names of characters that are impersonated by actors are kept for each character.

crew : This feature includes a team working behind the camera in movies and contains credit id, department, directing id, job, director name, profile path and gender information. Most common 15 crew members were selected and a new attribute has been created from this attribute. Gender is not necessary for prediction so this information was not used.

- n**um_crew :** This feature keeps the number of crew of the movie.
- **crew_name_<name> :** The names are kept in this feature for each crew person.
- **jobs_<name> :** The mission in the movie of each crew colleague are kept in this feature.
- **departments_<name> :** This feature keeps department names for each department that belongs to crew.

revenue : This feature keeps information about the revenue of movie. This is the value that is tried to be estimated by the model.

2.5. Models

In this study, we use the Gradient Boosting Decision Tree algorithm to train a model that estimates revenues and imdb scores of the movies. The most important reason for us to use this algorithm is efficiency, accuracy, and interpretability of it. As we show below in Figure 3, the GBM(Gradient Boosting Machines) algorithms have an error handling mechanism and each iteration decreases the error rate.

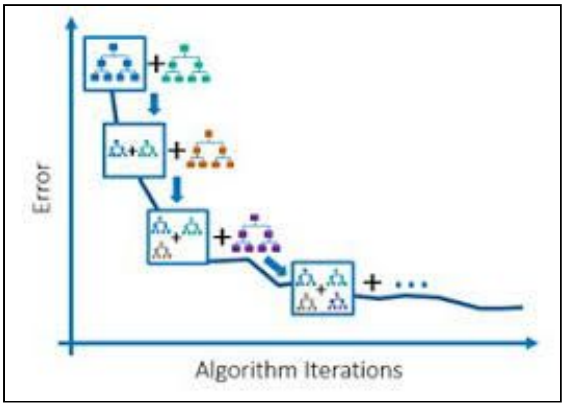

Figure 4 Error depends on iterations[5]

We use 3 different gradient boosting frameworks to train prediction models and we calculate a final decision by weighting these models' estimations. The first framework that we used is Light GBM. Light GBM uses a tree based learning algorithm. The most impressive attribute of this algorithm is speed. Light GBM has fast training, high

efficiency and large scale data handling capability. XGBoost is the other algorithm that we used. XGBoost helpt us to handle missing values and deal with them. Additionally, XGBoost has easy deploying and is compatible with many languages. In the Figure 4 there is an historical development from decision tree to XGBoost.

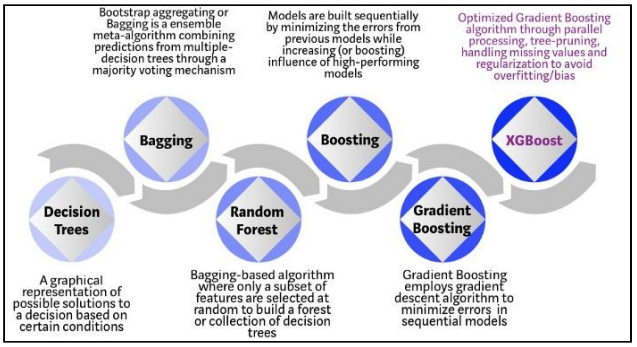

Figure 5 The Evolution of DT [6]

Lastly, we used the youngest gradient boosting framework named CatBoost. CatBoost is developed by Yandex in 2017. This framework is good at analysing categorical features. We use catboost also while doing label encoding on the textual attributes such as "all_genres".

3. EXPERIMENTAL RESULTS

3.1 Settings

In this study; pandas, numpy, sklearn, xgboost, lgboost, catboost libraries are used in Python 3. To put data in regular format, pandas and numpy libraries are applied. By using pandas, data analysis is facilitated. The data is visualized by using Matplotlib and Seaborn libraries.

Some of the revenue and budget values are corrected from an external resource.

Imdb scores are collected from IMDB's official site by using html parser library named BeautifulSoup.

Since the prediction values are continuous, there are not certain labels. The error rate must be calculated. The quality measure of predictions are received from root mean squared error rates.

$$RMSE = \sqrt{(\hat{y} - y)^2}$$

RMSE is calculated by using measure of the distance between the regression line and data points. y^ is the prediction value and y is the actual value.

The main TMDB data contains 23 columns with revenue value. Some of the features are presented in dictionary lists. Categorical attributes are created from a number of main attributes. Textual values like title, overview, tagline, original title are converted to weighted TF-IDF vectors by finding best parameters. After expanding the data set, over 250 columns are formed. In the experiment XGB, LGB and CAT models are created. XGB model's mean score, LGB model's mean score and CAT model's are 2.08, 2.06 and 2.05, respectively. According to result we get, an external data set is added into our data. To improve our performance, some values of revenue and budget are updated because of the noise of data. New attributes were created according to budget, run time, popularity, and year values that interacted most with revenue value. After these changes, error rates are decreased.

To deal with overfitting, unique attributes like imdb_id, id are removed. Generated features are examined and their effect of error rate are investigated. In cast attribute, there were three types of gender. 0 was assigned as unspecified. Therefore, number of cast member specified gender tagged 0 is dropped.

To minimize the error rate, results of algorithms are obtained by multiplying prediction value and constant weights. By changing weights, we controlled the effect of each algorithm to our result. With this method, we obtained a min squared error rate as 1.68 in the test data in TMDB and 0.629 in IMDB.

3.2 Results

Algorithm	Average Min Squared Error
XGBoost	1.668
LGBoost	1.697
CatBoost	1.701
XGBoost + LGBoost + CatBoost	**1.688**

Figure 6 TMDB prediction results

Algorithm	Average Min Squared Error
XGBoost	0.615
LGBoost	0.630
CatBoost	0.643
XGBoost + LGBoost + CatBoost	**0.629**

Figure 7 IMDB prediction results

Models are demonstrated in Figure 6 and Figure 7 are the final models about TMDB and IMDB prediction. The best results are given with weighted averages on test data.

4. CONCLUSION

In our study, we train models that predict the revenues and imdb rating values of movies. The data that is provided us includes 23 attributes about movies. Firstly, we analyze the data to have information and infer which attributes could be useful for estimation. The most important work on this study is the feature engineering, since feature extraction and reasonable usage of features take part in prediction. Revenues and imdb rating values of movies are numerical attributes. Therefore, although we have used logarithmic values of these attributes the values are sparse and it is hard that making prediction about them. This condition encourages us to train models that are compatible with numeric values. After the data preprocessing based on features, we train the models that use Gradient Boosting Decision Tree algorithms. We used XGBoost, Light GBM and CatBoost models with tuned weighting to have an optimized estimation model.

5. REFERENCES

[1] How to Win Kaggle Competitions, Zeeshan-ul-hassanUsmani, https://www.kaggle.com/getting-started/44997

[2] Understanding Gradient Boosting Machines, Harshdeep Singh, Nov 3, 2018, https://towardsdatascience.com/understanding-gradient-boosting-machines-9be756fe76ab

[3] Kaggle https://www.kaggle.com/c/tmdb-box-office-prediction/data

[4] Kamal Chhirang https://www.kaggle.com/kamalchhirang/tmdb-competition-additional-features

[5] Error depends on iterations http://www.cse.chalmers.se/~richajo/dit866/lectures/l8/gb_explainer.pdf

[6] The Evolution of DT https://medium.com/@ahmetkuzubasli/neden-ilk-adresiniz-xgboost-75010416a583