PREDICTING BOX OFFICE MOVIE REVENUE WITH MACHINE LEARNING METHODS

OĞUZ CAN KALKAN

AUGUST 2022

PREDICTING BOX OFFICE MOVIE REVENUE WITH MACHINE LEARNING METHODS

BAHÇEŞEHİR UNIVERSITY

THE GRADUATE SCHOOL OF

NATURAL AND APPLIED SCIENCES

COMPUTER ENGINEERING

OĞUZ CAN KALKAN

THE NECESSARY STUDIES FOR THE MASTER'S DEGREE IN COMPUTER ENGINEERING HAVE BEEN CARRIED OUT

AUGUST 2022

T.C.
BAHCESEHIR UNIVERSITY
GRADUATE SCHOOL

MASTER THESIS APPROVAL FORM

Program Name:	Computer Engineering
Student's Name and Surname:	Oğuz Can Kalkan
Name Of The Thesis:	Predicting Box Office Movie Revenue with Machine Learning Methods
Thesis Defense Date:	14/09/2022

This thesis has been approved by the Graduate School which has fulfilled the necessary conditions as Master thesis.

Prof. Dr. Ahmet ÖNCÜ
Institute Director

This thesis was read by us, quality and content as a Master's thesis has been seen and accepted as sufficient.

	Title/Name	**Signature**
Thesis Advisor's	Assoc. Prof. Tevfik Aytekin	
Member's	Assoc. Prof. Tamer Uçar	
Member's	Prof. Barış Bozkurt	

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname : Oğuz Can KALKAN

Signature :

ABSTRACT

PREDICTING BOX OFFICE MOVIE REVENUE WITH MACHINE LEARNING METHODS

Kalkan, Oğuz Can

Master's Program in Computer Engineering

Supervisor: Assoc. Prof. Tevfik Aytekin

August 2022, 42 pages

With the growth of the film industry in recent years, box office predictions have gained great importance. However, the process of prediction and being successful in this prediction is not easily obtained. To make an effective estimation, the movie data should be organized accurately, the machine learning method to be used should be chosen precisely and a properly working algorithm should be built.

Once the movies are published, people can comment and rate them on specific sites (e.g., IMDb and TMDb). Although this information is subjective, the large number of reviews and ratings of the movies make them usable. Since we decide to use people's reactions and votes, we must wait at least six months after the movie is released.

In this study, we attempt to get a more successful prediction that can be achieved by adding the ratings from IMDb and TMDb in addition to the movie features. Our results show that these rating additions increase the accuracy of predictions.

Keywords: Box Office, Machine Learning, Movie Revenue Prediction

ÖZ

MAKİNE ÖĞRENİMİ İLE FİLM HASILATI TAHMİNİ

Kalkan, Oğuz Can

Bilgisayar Mühendisliği Yüksek Lisans Programı

Tez Danışmanı: Doç. Dr. Tevfik Aytekin

Ağustos 2022, 42 sayfa

Son yıllarda film sektörünün büyümesiyle birlikte gişe tahminleri de büyük bir önem kazandı. Ancak bu tahminlerde başarılı olmak sanıldığı kadar kolay değildir. İnsanlar, filmler vizyona girdikten sonra bu filmlerle alakalı, belirli sitelerde (ör. IMDb ve TMDb) yorumlar yapıp bu filmlere puanlar verebilmektedirler. İnsanların yaptığı bu yorum ve puanlamalar her ne kadar subjektif olsalar da yapılan yorumların ve puanlamaların çokluğu bu bilgileri kullanılabilir kılmaktadır. Çalışmamızda bu yorum ve puanları kullanmaya karar verdiğimiz için, herhangi bir tahmin yapabilmek için bir film vizyona girdikten en az altı ay beklememiz gerekmektedir. Bu sayede insanlar bu filme puan verebilirler veya film hakkında yorum yapabilirler. Biz de bu çalışmamızda, insanların yaptıkları bu yorum ve puanlamaları daha önce yapılan hasılat tahmini çalışmalarında kullanılan film bilgileriyle birleştirerek daha iyi bir tahmin oluşturup oluşturulamayacağını inceledik. Aldığımız sonuçlar bu puanlamaların, oluşturulan tahminlerin doğruluk oranlarını arttırdığı ve hata payını düşürdüğü gözlemlenmiştir.

Anahtar kelimeler: Gişe, Makine Öğrenimi, Film Hasılat Tahmini

TEŞEKKÜR

Hayatımın her alanında destekleri ve sevgileri ile yanımda olan, bu günlere gelmemde çok emekler gösteren ve benim için çok kıymetli annem Songül KALKAN ve babam Mustafa Uğur KALKAN'a,

Tez sürecimde bilgileriyle önemli katkılar sağlayıp desteğini hiçbir zaman esirgemeyen ve öğrencisi olduğum için büyük mutluluk duyduğum danışman hocam Prof. Dr. Tevfik Aytekin'e,

Çalışma hayatım ve yüksek lisans sürecimde her daim desteğiyle yanımda olan ve her zaman olumlu tavırlarıyla cesaretlendiren, yoğun çalışmalarım sırasında büyük bir sabır gösterip bana katlandığı için sevgili eşim Madina KALKAN'a çok teşekkür ederim.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

DT	Decision Tree
XGB	Extreme Gradient Boosting
IMDb	Internet Movie Database
TMDb	The Movie Database
kNN	k-Nearest Neighbor Engine Marketing
RF	Random Forest
RMSE	Root Mean Squared Error
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error

Chapter 1

Introduction

As the movie industry grows, box office revenue predictions becoming are more and more important. Nevertheless, for good prediction, well organized data and delicately chosen models are required. Usually, these box office predictions are made using some machine learning methods and techniques as described in [1].

Features used in box office predictions are mostly literal and fixed values. However, people's comments and ratings on these movies are completely subjective. Still, the ratings and interpretation of a large number of people is a feature that increases the usability of this information. In particular, the comments and ratings made in IMDb and TMDb are at a decisive point both for the success of the movies and whether people watch that movie or not.

In this thesis, we aimed to create a better box office revenue prediction by combining people's comments and ratings with movie-related data (e.g. genre, actors, producer). Our results are an indication that this additional information can be used to make better revenue predictions.

1.1 Overview of The Thesis

The next section contains background information on the machine learning methods which we used in this thesis. After that, there are previous studies in this field of box office prediction. The following section is the section that explains how we create the data and how it is used for this study. In the methodology section, how we do not examine the data, which features are in the prediction models which we created, and which machine learning methods we used. The following sections are the tests we have done on the models, the results we have obtained in these tests, and the future works with the conclusion section.

Chapter 2

Background

This chapter contains information about the box office and machine learning methods used in this field. How we used these methods is explained in detail after this chapter. The chapter starts with general information about box office, and it ends with explanations of the machine learning methods we used in this thesis.

2.1. Movie Industry and Box Office Numbers

In this episode, we tried to show how much the cinema industry has grown in recent years, based on the revenues of the box office movies. We specially selected our data from 2019, as 2020 and 2021 could not be used due to CoVID-19.

Table 1 shows the fifteen movies which have the largest revenue in the United States in 2019.

Table 1
Fifteen Movies with The Largest Revenue In 2019 Based On US Revenue (Mojo, 2019a)

RANK	Movie	Gross	Theaters	Release Date
1	Avengers: Endgame	$858,373,000	4,662	Apr 26
2	The Lion King	$543,638,043	4,802	Jul 19
3	Toy Story 4	$434,038,008	4,575	Jun 21
4	Frozen II	$430,144,682	4,440	Nov 22
5	Captain Marvel	$426,829,839	4,310	Mar 8
6	Star Wars: Episode IX- The Rise of Skywalker	$390,706,234	4,406	Dec 20
7	Spider-Man: Far from Home	$390,532,085	4,634	Jul 2
8	Aladdin	$355,559,216	4,476	May 24
9	Joker	$333,772,511	4,374	Oct 4
10	IT Chapter Two	$211,593,228	4,570	Sep 6
11	Jumanji: The Next Level	$192,094,536	4,227	Dec 13
12	Us	$175,084,580	3,743	Mar 22
13	Fast & Furious Presents: Hobbs & Shaw	$173,956,935	4,344	Aug 2
14	John Wick: Chapter 3- Parabellum	$171,015,687	3,850	May 17
15	How to Train Your Dragon: The Hidden World	$160,799,505	4,286	Feb 22

Table 2 shows the revenue of the movies in the United States only. Table 2.2 shows the top fifteen grossing films worldwide in 2019.

Table 2
Fifteen Movies with The Largest Revenue In 2019 Based On Worldwide Revenue (Mojo, 2019b)

RANK	Movie	WorldWide	Domestic	%	Foreign	%
1	Avengers: Endgame	$2,797,501,328	$858,373,000	30.7%	$1,939,128,328	69.3%
2	The Lion King	$1,656,943,394	$543,638,043	32.8%	$1,656,943,394	32.8%
3	Frozen II	$1,450,026,933	$477,373,578	32.9%	$1,450,026,933	32.9%
4	Spider-Man: Far from Home	$1,131,927,996	$390,532,085	34.5%	$1,131,927,996	34.5%
5	Captain Marvel	$1,128,274,794	$426,829,839	37.8%	$1,128,274,794	37.8%
6	Joker	$1,074,251,311	$335,451,311	31.2%	$1,074,251,311	31.2%
7	Star Wars: Episode IX - The Rise of Skywalker	$1,074,144,248	$515,202,542	48%	$1,074,144,248	48%
8	Toy Story 4	$1,073,394,593	$434,038,008	40.4%	$1,073,394,593	40.4%
9	Aladdin	$1,050,693,953	$355,559,216	33.8%	$1,050,693,953	33.8%
10	Jumanji: The Next Level	$800,059,707	$320,314,960	40%	$800,059,707	40%
11	Fast & Furious Presents: Hobbs & Shaw	$759,056,935	$173,956,935	22.9%	$759,056,935	22.9%
12	Ne Zha	$726,063,471	$3,695,533	0.5%	$726,063,471	0.5%
13	The Wandering Earth	$699,856,699	$5,971,413	0.9%	$699,856,699	0.9%
14	How to Train Your Dragon: The Hidden World	$521,799,505	$160,799,505	30.8%	$521,799,505	30.8%
15	Maleficent: Mistress of Evil	$491,730,089	$113,929,605	23.2%	$491,730,089	23.2%

Although we are analyzing the revenues of movies, it is necessary to look at the revenues of the movie producers to better understand this field. Of course, they are the ones who publish the movies. Table 3 shows the revenues of the top 15 movie producers.

Table 3
Fifteen Movie Production Companies with The Largest Revenue Based On Worldwide revenue (The-Numbers, 2021a)

RANK	Movie Production Company	Number Of Movies	Total Domestic Box Office	Total Worldwide Box Office
1	Warner Bros.	267	$19,539,743,337	$45,701,248,525
2	Universal Pictures	284	$18,979,460,098	$44,541,394,382
3	Columbia Pictures	256	$19,034,463,190	$42,181,951,630
4	Walt Disney Pictures	138	$16,276,918,238	$38,982,030,710
5	Marvel Studios	62	$12,998,885,559	$33,715,449,457
6	Paramount Pictures	212	$14,269,827,979	$31,830,173,613
7	20th Century Fox	112	$10,530,111,275	$25,939,183,209
8	Dune Entertainment	70	$6,307,177,998	$16,488,527,745
9	New Line Cinema	120	$7,114,556,024	$16,448,314,259
10	Legendary Pictures	62	$6,307,088,832	$16,169,119,454

Table 3 (Continue)

11	DreamWorks Animation	45	$5,712,427,290	$15,409,874,832
12	Relativity Media	117	$7,363,230,475	$15,378,964,622
13	Disney-Pixar	27	$6,076,943,959	$14,635,644,475
14	Amblin Entertainment	64	$5,955,094,398	$14,606,263,285
15	Village Roadshow Productions	81	$5,868,826,847	$13,945,245,835

2.2. Supervised Machine Learning Methods

In the classification part of this thesis, we use Multiple Linear Regression (MLG), Decision Tree (DT), Random Forest (RF), k-Nearest Neighbor (kNN), Extreme Gradient Boosting (XGB). This section presents a small review of the methods we use.

2.1.1 Multiple Linear Regression. Regression analysis is a collection of statistical techniques that serve as a basis for drawing inferences about relationships among interrelated variables. Since these techniques are applicable in almost every eld of study, including the social, physical, and biological sciences, business and engineering, regression analysis is now perhaps the most used of all data analysis methods. (Golberg and Cho, 2010)

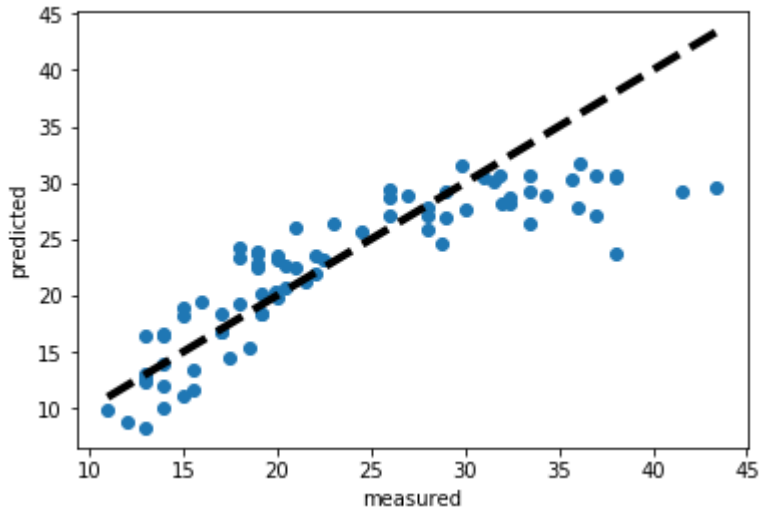

Figure 1. Linear Regression Model

Multiple linear regression (MLR) is a statistical method for predicting the result of a response variable by using a number of explanatory variables. Modelling the linear relationship between the explanatory (independent) factors and response (dependent) variable is the aim of multiple linear regression.

2.1.2 Decision Tree. Decision tree learning is a method for approximating discrete-valued target functions, in which the learned function is represented by a decision tree. Learned trees can also be represented as sets of if-then rules to improve human readability. These learning methods are among the most popular of inductive inference algorithms and they have been successfully applied to a broad range of tasks from learning to diagnose medical cases to learning to assess credit risk of loan applicants. (Mitchell, 1997)

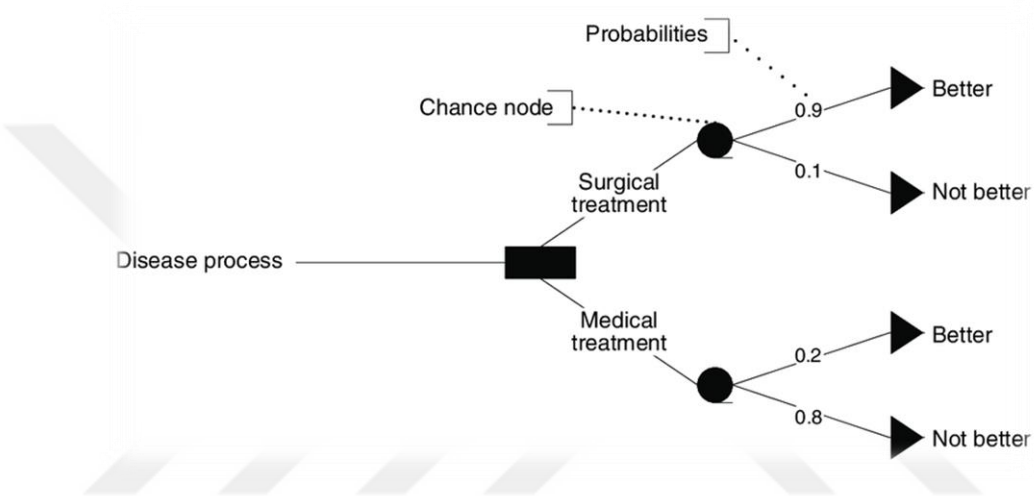

Figure 2. Schematic diagram of a Decision Tree

2.1.3 Random Forest. A classification algorithm made up of numerous decision trees is called the random forest. It attempts to create an uncorrelated forest of trees whose prediction by the committee is more accurate than that of any individual tree by using bagging and feature randomness when building each tree.

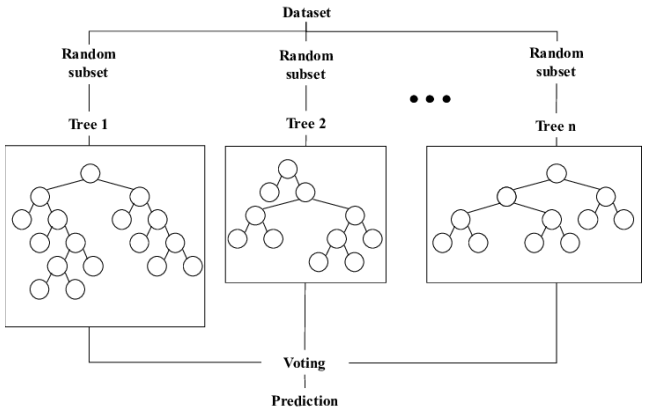

Figure 3. Schematic diagram of a Random Forest

As the name of the algorithm suggests, it somehow randomly creates a forest. The decision trees that make up the "forest" that it has created were all trained using the "bagging" technique. The bagging method's general premise is that combining learning models improves the result. While growing trees, Random Forest increases the model's randomness. When dissecting a node, it searches for the best feature among a chosen subset of features rather than the most crucial one. This leads to a wide variety, which typically produces a better model.

2.1.4 Extreme Gradient Boosting. Extreme Gradient Boosting (XGB) is one of the most popular and efficient implementations of the Gradient Boosted Trees algorithm. It is a supervised learning method based on function approximation by optimizing specific loss functions along with applying several different techniques.

Extreme Gradient Boosting is a specific implementation of the Gradient Boosting method that uses more precise approximations to find the best tree model. It employs several neat tricks that make it incredibly successful, especially with structured data. XGB has additional advantages such as training. It is extraordinarily fast and can be parallelized and spread across clusters.

2.1.5 k-Nearest Neighbor. kNN is an instance-based learning algorithm and it is one of the simplest among machine learning methods. It assumes that all instances in the data belong to an n-dimensional space where n denotes the number of attributes. Using the standard Euclidean distance measure, k nearest neighbors of a data point are specified. (Mitchell, 1997b)

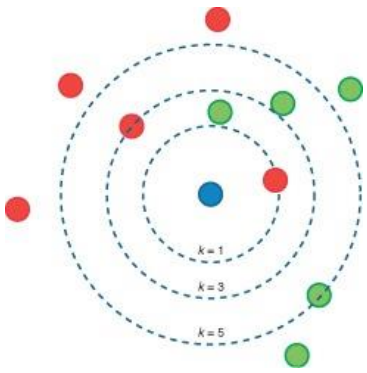

Figure 4. Illustration of a kNN classification model. (Mitchell, 2014a)

For k = 1, the model will classify the blue query instance as a member of the red class; for k = 3, it will again be assigned to the red class, this time by a 2–1 vote; however, since the fourth and fifth nearest neighbors are both green, a k = 5 model would classify it as part of the green class by a 3–2 majority.(Mitchell, 2014b)

Chapter 3

Related Work

Box office predictions have been very popular in recent years, and it is an area that has been studied a lot. Many studies in this field have been done using different methods. Especially the number of studies on the comments made by people on social media is quite high.

In this study, Vasu J. (2013) made an analysis through the tweets people sent about movies on Twitter between 2009 and 2012. 200 tweets were used for each movie in their analysis. As a result of these analyzes, they stated their success rate as 64.4% in their estimates for 8 movies. In the other study, Andrei O., Mathias B., Manos T., and Maarten de R. (2012) used comments on social media. Using the tweets posted on Twitter for the movies and the comments on the trailers of the movies on YouTube, they achieved an 89% r^2 score in their prediction.

Another study using social media data is Krushikanth R. A., Merin J., Supreme M., Chan C-C, Kathy J. L., and Federico de G. (2013). Data collected from Twitter, YouTube, and IMDB sites were used in this study. As a result of the estimation, they got a successful prediction performance in 13 of the 35 movies on their list.

In another study, Mestyán M., Yasseri T., and Kertész J. (2013) conducted an analysis of people's activities about movies on the Wikipedia site and tried to predict the revenues of the films from these activities. In this study, they used the used multivariate linear regression model and obtained an r^2 score of 94.0%. In 2012, Reddy A.S.S, Kasat P., and Jain A. (2012) conducted a box office prediction study by doing a hype analysis on tweets posted on Twitter.

Of course, box office predictions are not made only with data from social media. Zhou Y., Zhang L. & Zhang Y. (2017) presented a different method. In their studies, they used the multi-modal deep neural networks method on movie posters. They achieved a success rate of 52.2% on 3807 sample movies, which they divided into 6 categories according to their revenue.

Another Neural Network Approach used by Zulkernine F. H. and Rhee T.G. (2016). They offered a back-propagation neural network model for predicting the box-office success by classifying them as “flop” or “bomb”. They validated the approach using cross-entropy validation and their prediction accuracy was %91.

Surely Box office predictions are not only made on Hollywood movies, Gaikar D.D, Marakarkandy B., and Dasgupta C. (2015) have a study of a box office prediction on Bollywood movies released in 2014 with the data they received from Twitter.

Chong O., Roumani J., Nwankpa J. K., and Hu H.F (2017), in their study, used consumer participation behavior and user activities on Twitter, Facebook, and YouTube to predict the box office. For this study, From Facebook, the number of likes of the movie and the number of mentions of the page of the movie. From Twitter, the number of tweets posted for each movie and the number of tweets posted about the movie. From YouTube, they received the number of views of the trailer posted by the movie and the number of comments posted by everyone. In the findings of the study, the tweets about movies positively related to the movie's performance. However, this relationship is weakened when Facebook and YouTube features are added to the model. They also propose that YouTube and Facebook-based features are key indicators of the movie's future economic performance.

Chapter 4

Data Set

Although there are too many open-source datasets for box office prediction, these datasets were not to meet our needs. That's the reason why we have created our dataset by ourselves. The data set includes movie master data (such as genre, producer, cast), financial figures (budget and revenue of movie), and IMDb and TMDb ratings, total votes, and the movie's popularity. In the rest of this section, the main components of our dataset are explained in detail.

4.1 Movie Master Data

IMDb (Internet Movie Database) is an online database where movies and TV series information are stored and listed. The information of almost all movies and tv series that have been released is published on this site freely.

Another online database similar to IMDb is The Movie Database (TMDb). TMDb, just like IMDb, is an online database that shows all the information of movies on their website.

We have collected the master movie features from these two online databases through the service which we created to use the IMDb Application Programming Interface (IMDb API) and the TMDb Application Programming Interface (TMDb API). Both of these APIs have a daily search limit for developers to make calls. The movie information we receive due to these APIs, genre, homepage, original language, poster, Production Companies, Production Countries, Release Date, Runtime, spoken languages, title, keywords, cast, and crew.

Both TMDb and IMDb show the information we receive through APIs on their sites. Figure 5 and Figure 6 are examples of these search screens from each site. Besides, this information is seen not only on the search screens but also on the pages of the movies.

Figure 5. Advance title search sample results of IMDb

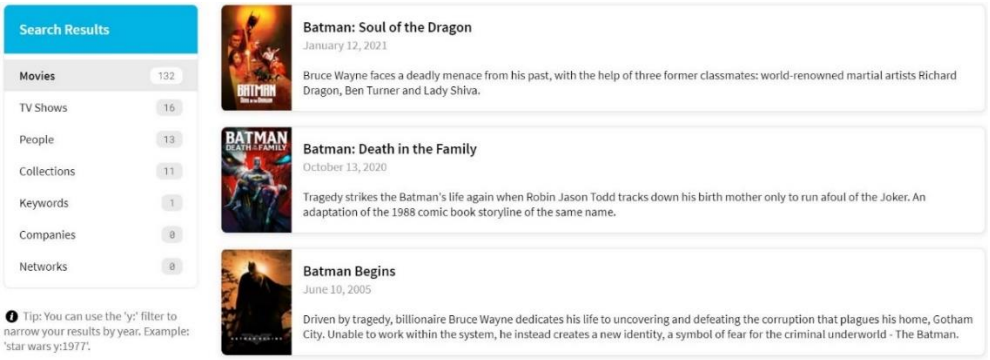

Figure 6. Advance title search sample results of TMDb

Figure 7 is the page of a movie on the IMDb site and Figure 8 is the page of a sample movie on the TMDb site.

Figure 7. IMDb - Example movie page

Figure 8. TMDb – Example movie page

4.2 Financial Data

Since our motivation is to predict movies revenue, it is crucial to have robust information on the financial facts about the movies. Again, IMDb provides some financial numbers, but they are not available for all of the movies; that's why we looked for additional sources.

We retrieved the movies financial information such as movie budget and gross revenue from The-Numbers.com. They track box-office numbers in a systematic way and then publish these financial performance records (The-Numbers, 2021b).

As they note on their website, it is hard to find reliable budget data since filmmakers might be reluctant to share this information or use some tricks to inflate or reduce numbers. Regarding this matter, we have accepted this website as a reliable source since it is used in other studies as well (Zhou and Ghiassi & Lio & Moon, 2017, 2015).

To extend the number of movies for which we have the financial data, we combined the numbers coming from IMDb, TMDb and The-Numbers.com. We have chosen IMDb as the primary source, meaning that, if these sources have different values for a movie, we used the ones coming from IMDb. There were some movies that exist in one and not in the other; however, the numbers were close for the ones that exist in both.

4.3 Rating Data

IMDb is a movie database in which movies are listed as we mentioned in the previous section, but it is also a website where people can give scores to the movies on the website. The scale of the score is 1-10 (1 is the lowest and 10 is the highest). This scoring is calculated over the scores of all members registered in the system. However, they do not use the arithmetic mean, that is, the sum of all votes divided by the number of votes. This is done by a method called weighted average. The number of registered members of IMDb is around eighty-three million in 2022.

TMDb is a movie database just like IMDb, people can score movies on the website. They use the same scoring scale (1-10, 1 is the lowest and 10 is the highest). The number of registered members of TMDb is around three million in 2022.

The next four figures (9, 10, 11, 12) are examples of scoring from IMDb and TMDb websites.

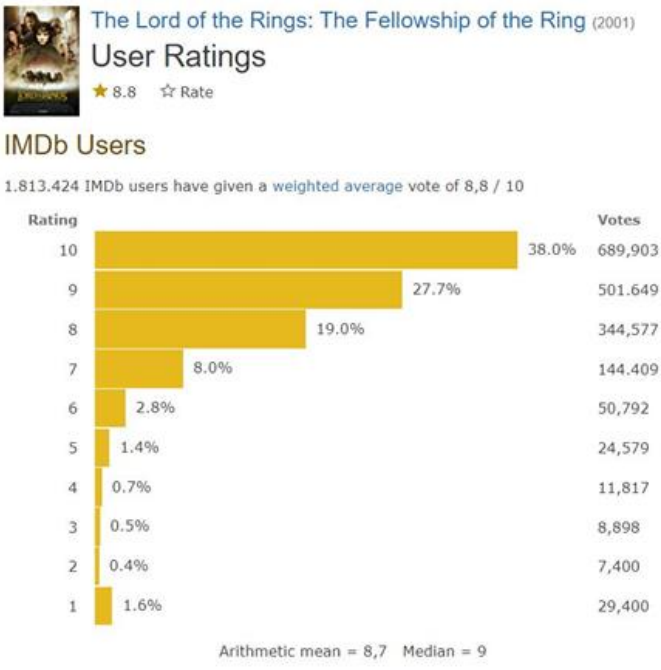

Figure 9. IMDb – Movie rating page example (high rating)

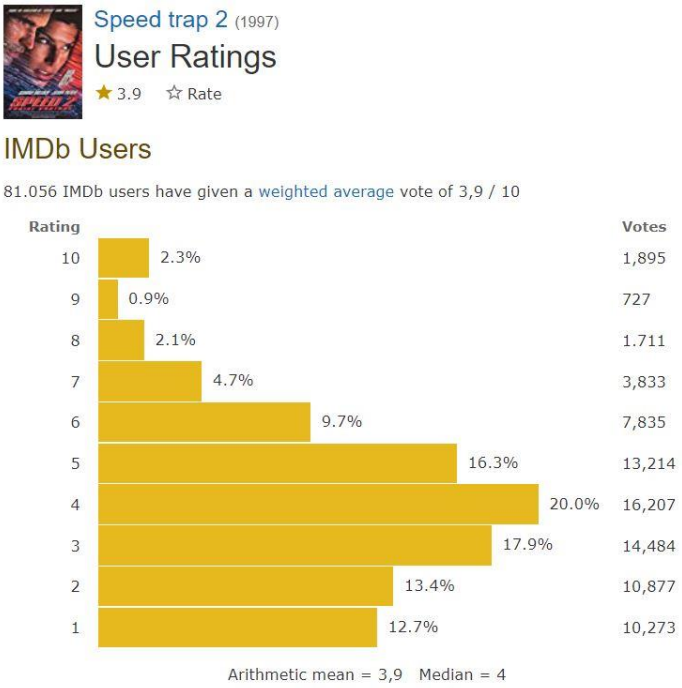

Figure 10. IMDb - Movie rating page example (low rating)

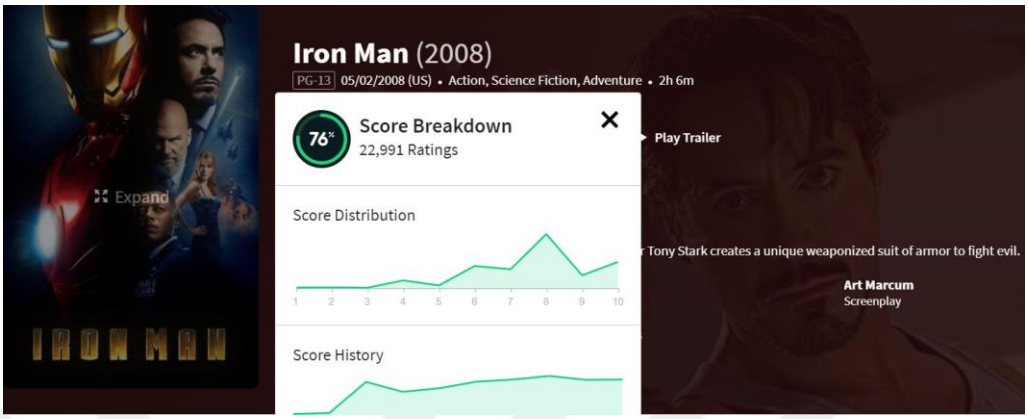

Figure 11. TMDb – Movie rating page example (high rating)

Figure 12. TMDb – Movie rating page example (low rating)

Before we collected the data from two databases, we thought that it would be difficult to combine those two datasets but after getting some example data from both databases we observed that TMDb holds movies with IMDb's unique keys.

Therefore, we collect data from both databases through their API's and combine them afterwards. We have removed the movies that did not have some necessary information (genre, producer company etc.) so in the end, there is no missing values in our data set. After these processes, our data preparation step was complete. Eventually, we had 2,154 movies with their IMDb and TMDb ratings, total votes, and weekly popularities in our final data set.

Chapter 5

Methodology

On this section, we explained the experiments done with the dataset and algorithms introduced in chapter 2.1. We did not include our rating features in 'Model Selection' section to select the best forecasting model without those features.

5.1 Programming Languages and Tools

Following programming languages, tools and the libraries were used for the implementation of the thesis.

- Python: Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Python is commonly used for data processing by data analysts. On this study, we used version 3.7.4.

- NumPy: NumPy is a library containing a large collection of high-level mathematical functions to operate multi-dimensional arrays and matrices for the Python programming language.

- Pandas: Pandas is a library for analyzing, cleaning, exploring, and manipulating data for the Python programming language.

- Scikit-Learn: Scikit-Learn is a popular machine learning library for the Python programming language that is used for classification, regression, clustering, predictive analytics, and other machine learning tasks.

- Seaborn: Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

- PyCharm: PyCharm is a popular source-code editor by JetBrains that supports development for multiple programming languages with plugin extensions.

The computers, which we used for the experiments, hardware specifications are:

- CPU: Intel Core i7-15000K CPU @3.15 GHz
- RAM: 16 GB DD4 4500MHz RAM
- SSD: 1 TB SSD 6500 RPM

5.2 Feature Analysis

In this section, we go over all the features that we include in our study in order to increase the prediction accuracy. In the first part, we explore the features that are directly related to movies, which we call movie master features. These features have been previously used in different studies (Wasu and Andrei and Krushikanth, 2013, 2012, 2013)

In the second part, we explain all the features we got from IMDb and TMDb rating, total votes, and movie popularities in detail. And the last part, we explore the features which we created using master movie features and ranking features together.

After conducting the experiments, we will see that only some of them have a positive effect on the forecasting results.

5.2.1 Movie Master Features. In this section, we examine movie features, IMDb rankings and TMDb rankings features and the new features which we created using movie features and ranking features together.

5.2.1.1 Budget(budget). This feature shows the production budget of the movies. Budget is a crucial factor in the determination of whether a movie makes a high revenue or not. Film producers who put a high budget on a movie expect higher revenue.

Figure 13 shows the link between budget and revenue values, as can be seen on the figure, low budget movies get low revenue and high budget movies get high revenue.

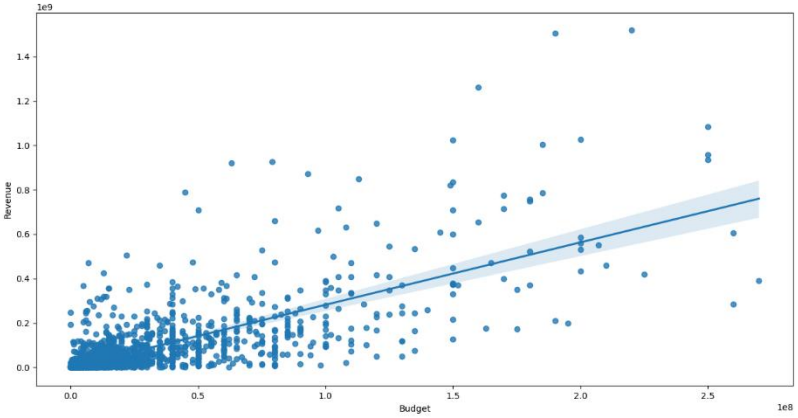

Figure 13. The link between movie budgets and movie revenues

5.2.1.2 Homepage (homepage). This feature shows the movie's own homepage. We created a new feature from this feature 'HasHomePage'. Only %64 of the movies in our dataset have their own homepage.

5.2.1.3 Production Companies (ProductionCompanies). This feature shows the producers of the movies. Although the movie producers seem to be budget-related, people can see the producers of the movies and decide whether to watch those movies or not. This assumption means that the effect of movie producers on revenue is not minor at all.

5.2.1.4 Release Date (ReleaseDate). This feature shows the release date of the movies on our dataset. We extracted the year from the release dates and saved it as a new feature because movie box office revenue changes from year to year, and this feature can be useful to capture this yearly trend in our prediction model.

5.2.1.5 Runtime (Runtime). Run time is the duration of the movie as minutes. It may seem useless for a prediction, but movies of long or short duration may help people decide whether to watch those movies or not.

5.2.1.6 Spoken Languages (SpokenLanguages). This feature shows which languages have been spoken in the movies. %24,4 of our movies have more than one spoken language.

5.2.1.7 Cast (Cast). This feature shows the actors and the actresses of the movies. Cast feature is the same as movie producers. It could be related to budget but again, when people see the famous actresses or actors, they could decide to see that movie.

Table 5.1 shows five famous actors and actresses' movies average revenues.

Table 4

Five famous actors' and actresses' movies average revenues. (Dollars in millions)

Actor / Actress	Revenue
Tom Cruise	88.2
Johnny Depp	87.4
Morgan Freeman	86.8
Jennifer Aniston	86.5
Julia Roberts	87.2

5.2.1.8 Crew (Crew). This feature shows the crew members of movies such as Scriptwriter, Director, Production Designer or Casting Director. We believe one of the most important data from this feature is the director. People can decide to watch the movie after seeing the director.

5.2.1.9 Keywords (Keywords). This feature shows the keywords about the movies.

5.2.1.10 Genre (Genre). A movie genre is a stylistic or thematic category for motion pictures based on similarities either in the narrative elements, aesthetic approach, or the emotional response to the movie. The list of movie genres in our data set is as follows: 'Action', 'Adventure', 'Animation', 'Biography', 'Comedy', 'Crime', 'Drama', 'Family', 'Horror', 'Music', 'Musical', 'Mystery', 'Romance, 'Sci-Fi', 'Thriller', 'War', 'Western'. A movie has commonly more than one genre.

5.2.2 IMDb and TMDb Rating Features. We have used seventeen unique features from IMDb and TMDb in our experiments. Twelve features from IMDb and three features (popularity, movie rank and total votes) from TMDb. We listed first IMDb features and then three TMBb features in this section.

5.2.2.1 Total Votes (TotalVotes). Total vote shows how many votes were given to the movies by members of IMDb.

5.2.2.2 Average Votes (AverageVotes). Average Vote is the average of the total votes. IMDb calculates this value by using a weighted average method.

5.2.2.3 Mean Votes (MeanVotes). Shows the mean of the total votes.

5.2.2.4 Total Male Votes (TotalMaleVotes). Shows the votes given by the male members.

5.2.2.5 Average Male Votes (AverageMaleVotes). Shows average value of male total votes.

5.2.2.6 Total Female Votes (TotalFemaleVotes). Shows the votes given by the female members.

5.2.2.7 Average Female Votes (AverageFemaleVotes). Shows average value of female total votes.

5.2.2.8 Total Votes US (TotalVotesUS). Shows the votes given by members from the United States of America.

5.2.2.9 Average Votes US (Average Votes US). Shows average value of US total votes.

5.2.2.10 Total Votes Non-US (TotalVotesNonUS). Shows the votes given by the members not from the United States of America.

5.2.2.11 Average Votes Non-US (AverageVotesNonUS). Shows average value of non-US total votes.

5.2.2.12 Popularity (Popularity). Shows the popularity of the movie on the IMDb website. Popularity, calculated by using the amount of people who visit the movie page, movie ranking score and total votes given by members.

5.2.2.13 User Review Counts (UserReviewCounts). Shows the number of counts in which IMDb members write reviews on the movie.

5.2.2.14 Critics Review Counts (CriticsReviewCounts). Shows the number of counts in which IMDb critics write reviews on the movie.

5.2.2.15 Total Votes TMDb (TotalVotesTMDb). Total votes show how many votes were given to the movies by members of TMDb.

5.2.2.16 Average Votes TMDb (AverageVotesTMDb). Average Votes is the average of the total votes given by the members of TMDb.

5.2.2.17 Popularity TMDb (PopularityTMDb). Shows the popularity of the movie on the TMDb website. This value is calculated as same as IMDb popularity.

5.2.3 Generated Features. These features are created by using master movie features and rating features together. We listed the features which we used in our forecasting models.

5.2.3.1 Genre List (). Genre List does not represent only one feature, we generated bool genre features like 'genreAction', 'genreAdventure', etc. If a movie's genre feature has one of those genres, those features' value becomes '1'.

5.2.3.2 Total Genre Count (TotalGenreCount). This feature shows how many genres the movie has. Like we explained before, a movie would have more than one genre.

5.2.3.3 Production Company List (). Production Company List is like the Genre List, and it does not represent only one feature. We listed all production companies in our dataset and selected the first fifteen of them. Then, we created bool features with their names, such as 'ProductionCompanyUniversalPictures' or 'ProductionCompanyWarnerBros'. If a movie's production company feature has one of those, those features value becomes '1'.

5.2.3.4 Total Production Company Count (TotalProductionCompanyCount). Like the genre, most movies have more than one production company. This feature shows how many production companies the movie has.

5.2.3.5 Production Country List (). Production Country List same as Production Company List. We listed the most common fifteen production countries from our data set and then we created bool features by their names. If a movie's production country feature has one of those, those features' value becomes '1'.

5.2.3.6 Total Production Country Count (TotalProductionCountryCount). Some movies have more than one production country. This feature shows how many production countries the movie has.

5.2.3.7 Spoken Language List (). Spoken Language List same as the above List. We listed the most common fifteen spoken languages from our data set and then we created bool features by their names. If a movie's spoken language feature has one of those, those features' value becomes '1'.

5.2.3.8 Total Spoken Language Count (TotalSpokenLanguageCount). Some movies have more than one spoken language. This feature shows how many spoken languages the movie has.

5.2.3.9 Keyword List (). Keyword List is the same as the above List. We listed the most common twenty keywords from our data set and then we created bool features by their names. If a movie's keywords feature has one of those, those features' value becomes '1'.

5.2.3.10 Total Keyword Count (TotalKeywordCount). This feature shows how many keywords the movie has.

5.2.3.11 Cast List (). Cast List is the same as the above List. It represents the actors or actresses in the movies. We listed the most common thirty cast members from our data set and then we created bool features by their names such as 'CastTomCruise' or 'CastPenelopeCruz'. If a movie's cast feature has one of those, those features' value becomes '1'.

5.2.3.12 Total Cast Count (TotalCastCount). This feature shows the count of cast members.

5.2.3.13 Cast Total Actor Count (CastTotalActorCount). This feature shows how many actors in the movies.

5.2.3.14 Cast Total Actress Count (CastTotalActressCount). This feature shows how many actresses in the movies.

5.2.3.15 Cast Character List (). Cast Character List is the same as the above List. It represents the characters of actors or actresses in the movies. We listed the most common fifteen characters from our data set and then we created bool features by their names such as 'CastCharacterDoctor or 'CastCharacterPilot'. If a movie's cast feature has one of those, those features' value becomes '1'.

5.2.3.16 Total Cast Character Count (TotalCastCharacterCount). This feature shows how many characters in the movies.

5.2.3.17 Crew List (). Crew List is the same as the above List. It represents the crew in the movies. We listed the most common fifteen crew members from our data set and then we created bool features by their names such as 'CrewStevenSpielberg' or 'CrewDeborahAquila'. If a movie's crew feature has one of those, those features' value becomes '1'.

5.2.3.18 Total Crew Count (TotalCrewCount). This feature shows the count of crew members.

5.2.3.19 Crew Total Male Count (CrewTotalMaleCount). This feature shows how many males are in the crew of the movies.

5.2.3.20 Crew Total Female Count (CrewTotalFemaleCount). This feature shows how many females are in the crew of the movies.

5.2.3.21 Crew Position List (). Crew Position List is the same as the above List. It represents the positions in the crew of the movies. We listed the most common fifteen crew positions from our data set and then we created bool features by their names such as 'CrewPositionProducer' or 'CrewPositionDirector'. If a movie's crew

feature has one of those, those features' value becomes '1'.

5.2.3.22 Crew Total Position Count (CrewTotalPositionCount). This feature shows how many positions are in the crew of the movies.

5.2.3.23 Crew Department List (). Crew Department List is the same as the above List. It represents the departments in the crew of the movies. We listed the most common fifteen crew departments from our data set and then we created bool features by their names such as 'CrewDepartmentProduction' or 'CrewDepartmentSound. If a movie's crew feature has one of those, those features' value becomes '1'.

5.2.3.24 Crew Total Department Count (CrewTotalDepartmentCount). This feature shows how many departments are in the crew of the movies.

5.2.3.25 Total Crew Count (TotalCrewCount). This feature shows the count of crew members.

5.2.3.26 Budget to Popularity (BudgetToPopularity). This feature is calculated by dividing the budget into IMDb popularity points.

5.2.3.27 Budget to Popularity TMDb (BudgetToPopularityTMDb). This feature is calculated by dividing the budget into TMDb popularity points.

5.2.3.28 Budget Total Votes Ratio (BudgetTotalVotesRatio). This feature is calculated by dividing the budget by the IMDb total votes

5.2.3.29 Budget to RunTime (BudgetToRuntime). This feature is calculated by dividing the budget into movie run time minutes.

5.2.3.30 Budget to Release Year (BudgetToReleaseYear). This feature is calculated by dividing the budget to the movies release year.

5.2.3.31 Runtime to Release Year (RuntimeToReleaseYear). This feature is calculated by dividing the runtime to the movies release year.

5.2.3.32 Average Vote Popularity Ratio (AverageVotePopularityRatio). This feature shows the count of crew members.

5.2.3.33 Average Vote to Popularity Ratio TMDb (AverageVotePopularityRatioTMDb). This feature is calculated by dividing the TMDb rating value to the TMDb popularity points.

5.2.3.34 Budget Average Vote Ratio (BudgetAverageVoteRatio). This feature is calculated by dividing the budget by the IMDb average votes.

5.2.3.35 Budget to Average Vote Ratio TMDb (BudgetAverageVoteRatioTMDb). This feature is calculated by dividing the budget by the TMDb average votes.

5.2.3.36 Runtime to Average Vote (RuntimeToAverageVote). This feature is calculated by dividing runtime by the IMDb average votes.

5.2.3.37 Popularity to Total Votes (PopularityToTotalVotes). This feature is calculated by dividing IMDb popularity points to the TMDb total votes.

5.2.3.38 Popularity to Total Votes TMDb (PopularityToTotalVotesTMDb). This feature calculated by dividing IMDb popularity points to the TMDb total votes.

5.2.3.39 Budget to US Votes Ratio (BudgetToUSVotesRatio). This feature is calculated by using budget and total US votes.

5.2.3.40 Budget to Male Votes Ratio (BudgetToMaleVotesRatio). This feature is calculated by using budget and male votes.

5.2.3.41 Budget to Female Votes Ratio (BudgetToFemaleVotesRatio). This feature is calculated by using budget and female votes.

5.2.3.42 Is Original Language English (IsOriginalLanguageEnglish). This feature checks the Original Language features. If it's 'English' then the value of this feature becomes '1' if it's not, then the value of this feature becomes '0'.

5.3 Evaluation

In this study, we aimed to make a prediction for how much revenue the movies will make. As it is known, the revenues of these movies are over millions of dollars. Thus, making predictions over such large numbers, we had error rates with large numbers too.

5.4 Model Selection

In our experimentation, we build different sets of features and compare the prediction results of each set. Evaluating all these feature sets on different fore-casting models would take a lot of time. Therefore, we wanted to reach the best-performing forecasting model and continue our experiments with that one.

In model selection, we aimed to find the best fore-casting model and the best set of hyper-parameters. To do this selection we have used a grid search algorithm to find the best hyperparameters for our forecasting models.

For Random Forest, Decision Tree, Linear Regression, and k-Nearest Neighbor algorithms, we have used the implementations of the scikit-learn library (Pedregosa, Varoquaux, Gramfort A, Michel, Thirion, Grisel, Blondel, Prettenhofer, Weiss, Dubourg, Vanderplas, Passos, Cournapeau, Brucher, Perrot and Duchesnay, 2011).

5.4.1 Random Forest.

5.4.1.1 Number of Estimators. Number of trees in the forest.

Possible parameter values: [10, 200, 500, 1000]

5.4.1.2 Max Depth. The maximum depth of the tree.

Possible parameter values: [4, 5, 6, 7]

5.4.1.3 Max Features. Controls the maximum number of features to consider when finding the best split.

Possible parameter values: ['auto', 'sqrt' (square root of the number of features), 'log2' (log of the number of features)]]

5.4.1.4 Criterion. It's the function to measure the quality of a split.

Possible parameter values: ['squared_error', 'absolute_error', 'poisson']

5.4.2 Extreme Gradient Boosting. We have used an open-source Python implementation of the XGB algorithm (XGB, 2021).

5.4.2.1 Min Child Weight. It is the minimum sum of instance weight (hessian) needed in a child. The larger Min Child Weight is, the more conservative the algorithm will be.

Possible parameter values: [1, 5, 10]

5.4.2.2 Gamma. It shows the minimum loss reduction required to make a further partition on a leaf node of the tree. The larger the value of gamma means the more conservative the algorithm.

Possible parameter values: [0.5, 1, 1.5, 2, 5]

5.4.2.3 Max Depth. It shows the maximum depth of a tree. Increasing max-depth would make the model more complex and more likely overfitting.

Possible parameter values: [4, 6, 8, 10]

5.4.2.4 Learning Rate. It's for step size shrinkage used in the update to prevent overfitting.

Possible parameter values: [0.01, 0.02, 0.03, 0.04]

5.4.2.5 n-Estimator. It represents the number of gradient boosted trees. Equivalent to the number of boosting rounds

Possible parameter values: [100, 500, 1000, 1500]

5.4.2.6 Sub Sample. It shows the ratio of training instances to be used when building trees. Small values might prevent overfitting.

Possible parameter values: [0.1, 0.2, 0.6, 0.9]

5.4.2.7 Column Sub Sample by Tree. Ratio of columns to be used when building trees.

Possible parameter values: [0.6, 0.8, 1.0]

5.4.3 Decision Trees.

5.4.3.1 Number of Estimators. It controls the maximum depth of a decision tree and therefore deals with the size of the tree to avoid overfitting of trees in the decision tree.

Possible parameter values: [10, 200, 500, 1000]

5.4.3.2 Max Depth. The maximum depth of the decision tree.

Possible parameter values: [4, 5, 6, 7]

5.4.3.3 Max Features. Controls the maximum number of features to consider when finding the best split.

Possible parameter values: ['auto', 'sqrt', 'log2']

5.4.3.4 Criterion. It's the function to measure the quality of a split.

Possible parameter values: ['squared_error', 'absolute_error', 'poisson']

5.4.4 Linear Regression.

5.4.4.1 Fit Intercept. It controls whether or not to calculate the intercept for this model or not.

Possible parameter values: ['True', 'False']

5.4.4.2 Normalize. It controls normalizing the regressor X.

Possible parameter values: ['True', 'False']

5.4.5 k-Nearest Neighbor.

5.4.5.1 Number of Neighbors. It shows the number of neighbors to use in classification decisions.

Possible parameter values: ['1', '2', '3', '5', '9', '10', '20', '30']

5.4.6 Light Gradient Boosting. To build a LGB model, we have used an open-source Python implementation of LGB algorithm [24].

5.4.6.1 Min Child Weight. It is the minimum sum of instance weight needed in a child. The larger Min Child Weight is, the more conservative the algorithm will be.

Possible parameter values: [1, 5, 10]

5.4.6.2 Max Depth. It shows the maximum depth of a tree. Increasing max-depth would make the model more complex and more likely overfitting.

Possible parameter values: [4, 6, 8, 10]

5.4.6.3 Learning Rate. It's for step size shrinkage used in the update to prevent overfitting.

Possible parameter values: [0.01, 0.02, 0.03, 0.04]

5.4.6.4 n-Estimators. It represents the number of gradient boosted trees. Equivalent to the number of boosting rounds.

Possible parameter values: [100, 500, 1000, 1500]

5.4.6.5 Sub Sample. It shows the ratio of training instances to be used when building trees. Small values might prevent overfitting.

Possible parameter values: [0.1, 0.2, 0.6, 0.9]

Table 5 shows the best hype-parameters for our forecasting models.

Table 5 *Forecasting Models' Hyper-parameters*

Model	Parameters
XGBRegressor	[*{*'max depth': 4, 'eval metric'= 'None', 'learning rate': 0.01, 'n estimators': 1500, 'min child weight': 1, 'colsample bytree': 0.8, 'colsample bylevel': 1, 'subsample': 0.9, 'eta': 0.1, 'seed': 42*}*]
LinearRegressor	[*{'fit intercept'*: 'false', 'normalize': 'true'*}*]
RandomForestRegressor	[*{*'max depth': 8, 'max features': 'auto', 'n estimators': 1000, 'random state': 42*}*]
KNeighborsRegressor	[*{*'n neighbors': 9*}*]
DecisionTreeRegressor	[*{*'max depth': 5, 'max features': 'auto', 'max leaf nodes': 20, 'min samples leaf': 8, 'min weight fraction leaf': 0.1, 'splitter': 'random'*}*]
LGBMRegressor	[*{*'num leaves': 30, 'min data in leaf': 20, 'objective': 'regression', 'max depth': 5, 'learning_rate': 0.01, 'boosting': 'gbdt', 'feature fraction': 0.9, 'bagging freq': 1, 'bagging seed': 11, 'metric': 'rmse', 'lambda l1': 0.2, 'subsample': 0.9, 'verbosity': -1*}*]

After selecting the hyper-parameters for our forecasting models, we aimed to select the best features for them. Therefore, we checked our features importance scores for each forecasting model and removed the features which had a '0' value. The next two tables show the twenty highest score features for the Extreme Gradient Boosting model and Light Gradient Boosting model.

Table 6

The Highest Importance Score of Twenty Features in Extreme Gradient Boosting Model Without Ranking Features and Generated Features

Feature	**Importance Score**
Budget	1832
BudgetToReleaseYear	1326
BudgetToReleaseDay	920
Runtime	808
BudgetToRuntime	795
TotalKeywordsCount	755
RuntimeToReleaseYear	749
ReleaseYear	678
TotalCastCount	673
ReleaseDay	572
CrewTotalFemaleCount	551
CastTotalActorCount	474
TotalCrewCount	451
ReleaseMonth	441
TotalProductionCompanyCount	427
CastTotalActressCount	411
CrewTotalMaleCount	410
ReleaseDayOfWeek	369
TotalGenreCount	345
GenreDrama	219

Table 7

The Highest Importance Score of Twenty Features in Light Gradient Boosting Model Without Ranking Features And Generated Features

Feature	**Importance Score**
Budget	1559
BudgetToReleaseYear	1529
BudgetToRuntime	1050
ReleaseYear	1005
BudgetToReleaseDay	965
TotalKeywordsCount	949
ReleaseDay	922
RuntimeToReleaseYear	919
TotalCastCount	882
CrewTotalFemaleCount	842
Runtime	777
CastTotalActorCount	665
CrewTotalMaleCount	652
ReleaseMonth	643
TotalCrewCount	515
ReleaseDayOfWeek	509
CastTotalActressCount	491
TotalProductionCompanyCount	429
GenreDrama	285
TotalGenreCount	262

After finding the best hyper-parameters for our forecasting models, we made predictions for each of them. Table 8 shows the results of our forecasting models.

Table 8

The Best Scores of Forecasting Models

Model	R^2 Score	RMSE	MAE	MAPE
XGBRegressor	0.65	113.39M	60.33M	47.20
RandomForestRegressor	0.65	113.87M	61.85M	54.23
LinearRegressor	0.62	118.44M	64.79M	72.60
KNeighborsRegressor	0.54	129.10M	66.87M	65.41
DecisionTreeRegressor	0.48	140.30M	73.34M	47.38
LGBMRegressor	0.64	114.12M	60.30M	48.23

Chapter 6

Experiments And Results

In this chapter, we show our experiment design and explain our classification results. Since the data we created is completely new and there was no open data, we could not directly compare the results with other studies in this field. Our main purpose in this study is to see whether ranking features from IMDb and TMDb provide a significant change in box office movie prediction success. In our tests, we have two different group of models.

The first model group has only the master movie features and the new features that we mentioned in 5.2.2 that we have generated by only using the master movie features. The second model group has the movie master features, ranking features, and all generated features that we mentioned in 5.2.2. We have compared these two models r^2, MSE, MAE, and MAPE scores.

Before we compare the results, we checked our feature importance scores for the Extreme Gradient Boosting model that has ranking features and generated features. Table 9 shows the highest importance score of twenty Features in the Extreme Gradient Boosting model with ranking and generated features.

Table 9

The Highest Importance Score of Twenty Features in Extreme Gradient Boosting model with Ranking Features and Generated Features

Feature	**Importance Score**
Budget	1181
BudgetToReleaseYear	863
BudgetAverageVoteRatio	585
Popularity	540
ReleaseYear	539
AverageVote	537
CriticsReviewCount	535
Runtime	517
UserReviewCount	485
BudgetTotalVotesRatio	433
PopularityTotalVotesRatio	412
BudgetToReleaseDay	396
TotalVotes	390
RuntimeAverageVoteRatio	377
BudgetUSVotesRatio	365
BudgetFemaleTotalVotesRatio	355
RuntimeToMeanYear	338
BudgetToRuntime	327
TotalKeywordCount	323
AverageVotePopularityRatio	322

Looking at the feature importance scores, the ‘Budget’ feature has the highest score. Of the twenty highest scoring features in the forecasting model, twelve of them are ranking features or ranking generated features. Therefore, we can assume that the ranking features have a crucial place in our prediction models.

After feature importance scores we have checked the result of r^2, RMSE, MAE, and MAPE scores. Table 10 shows the scores of both model groups.

Table 10

The Scores of Two Forecasting Model Groups

Model	**R^2 Score**	**RMSE**	**MAE**	**MAPE**
XGBWithRatingFeatures	0.78	89.98M	42.33M	21.85
XGBWithoutRatingFeatures	0.65	113.39M	60.33M	47.20
RandomForestWithRatingFeatures	0.79	89.77M	44.57M	30.20
RandomForestWithoutRatingFeatures	0.65	113.87M	61.85M	54.23
LinearRegressionWithRatingFeatures	0.76	93.36M	49.78M	35.22
LinearRegressionWithoutRatingFeatures	0.62	118.44M	64.79M	72.60
KNeighborsWithRatingFeatures	0.78	118.02M	57.70M	30.09
KNeighborsWithoutRatingFeatures	0.54	129.10M	66.87M	65.41
DecisionTreeWithRatingFeatures	0.66	119.80M	71.33M	42.20
DecisionTreeWithoutRatingFeatures	0.48	140.30M	73.34M	47.38
LGBMWithRatingFeatures	0.77	90.44M	45.69M	27.15
LGBMWithoutRatingFeatures	0.64	114.12M	60.30M	48.23

As we see in the results, when we add rating features and generated features from our forecasting models, r^2 values increase and the error rates decrease. When we look Extreme Gradient Boosting model with rating features and generated features, the r^2 value increases by 0.13 points, the root mean squared error decreases by 23.41 million, the mean absolute error decreases by 18 million, and the mean absolute percentage error decreases by 25.35 points. After seeing these results, we can say that rating features have improved the prediction accuracy of the base movie features.

Figure 14 and figure 15 shows the plots of Extreme Gradient Boosting models' predictions and real values.

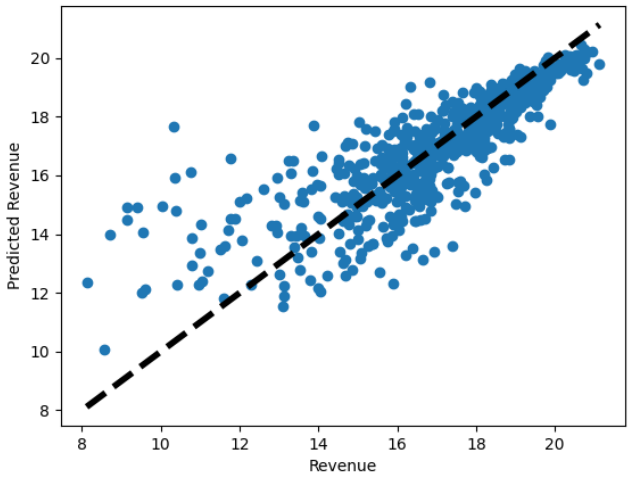

Figure 14. Prediction and real value plots of Extreme Gradient Boosting model with rating features and generated features

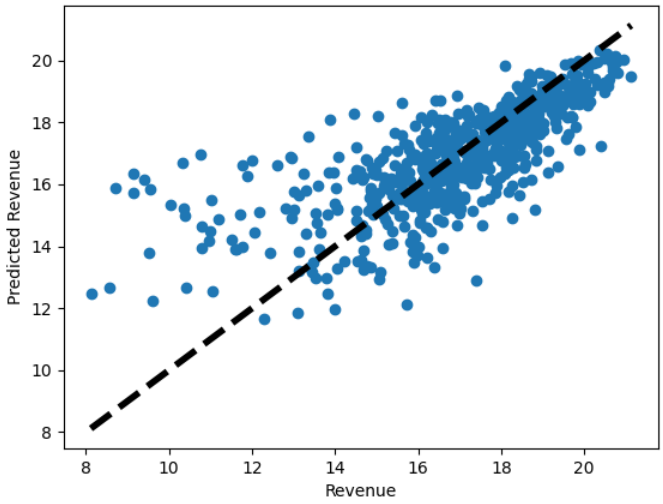

Figure 15. Prediction and real value plots of Extreme Gradient Boosting model without rating features and generated features

Chapter 7

Conclusion

In this study, we have proposed a method to make better predictions on movies' box-office success using the people ratings to the movies from IMDb and TMDb.

We have got seventeen features from IMDb and TMDb and generated more than fifty features from those features to create accurate prediction models. Some features were showing the total votes of the members. Some of them were presenting the average of those votes and some of them were dividing by gender. We have used these ranking features with movie features to generate new features. After the experimentation, the best result we got without ranking features and generated features was r^2 *score* = 0.65, *root mean squared error* = 113.39M, *mean absolute error* = 60.33M, and *mean absolute percentage error* = 47.20. Thanks to our ranking features, we improved our models, and got better results (r^2 *score* = 0.78, *root mean squared error* = 89.98M, *mean absolute error* = 42.33M, and *mean absolute percentage error* = 21.85) to predict the movie's revenues.

Chapter 8

Future Work

The world has been dealing with the covid pandemic globally for the last two years. Due to this pandemic, theaters were closed for a long time and people could not go to see movies. Although people find another way to watch movies, they watch the movies and TV series at home using digital platforms (Netflix, Disney+, Amazon Prime, etc.). Besides watching, people also can vote for the movies, give likes, or dislikes and write comments about them.

As an additional direction for future work, getting data (total views, in-platform ranking scores, total comments, likes, and dislikes) from those digital platforms and combining them with IMDb rankings would improve the success of box office movie predictions. study, we have proposed a method to make better predictions on movies' box-office success using the ratings to the movies from IMDb and TMDb.

REFERENCES

Andrei O., Mathias B., Manos T., and Maarten de R. (2012). Predicting IMDB Movie Ratings Using Social Media. *Proceedings of the 34th European conference on Advances in Information Retrieval*, [online]. Available at: https://www.researchgate.net/publication/262401833_Predicting_IMDB_Movie_Ratings_Using_Social_Media [Accessed 19 Apr. 2022].

Gaikar, D. D., Marakarkandy, B. and Dasgupta, C. (2015). Using Twitter data to predict the performance of Bollywood movies. *Industrial Management & Data Systems*. [online], 115(9), pp.1604–1621. Available at: https://www.researchgate.net/publication/282854576_Using_Twitter_Data_to_Predict_the_Performance_of_Bollywood_Movie https://www.researchgate.net/publication/282854576_Using_Twitter_Data_to_Predict_the_Performance_of_Bollywood_Movies [Accessed 25 Mar. 2022].

Ghiassi, M., Lio, D and Moon, B. (2015). Pre-production forecasting of movie revenues with a dynamic artificial neural network. *Expert Systems with Applications*, [online] 42(6), pp.3176–1395. Available at: https://www.researchgate.net/publication/270293976_Pre-production_forecasting_of_movie_revenues_with_a_dynamic_artificial_neural_network [Accessed 1 Apr. 2022].

Golberg, M.A. and Cho, H.A. (2010). *Introduction to Regression Analysis*. Southampton: Wessex Institute of Technology [online]. Available at: https://www.researchgate.net/publication/264700780_Introduction_to_Regression_Analysis [Accessed 19 Apr.2022].

Krushikanth R. A, Merin J., Supreme M., C.-C. Chan, Kathy J. L., and Federico de G. (2013). Prediction of movies box office performance using social media. *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, [online] Available at: https://www.researchgate.net/publication/262249589_Prediction_of_movies_box_office_performance_using_social_media [Accessed 28 Mar. 2022].

LightGBM. (2021). LightGBM Documentation. [online] LightGBM. Available at: https://lightgbm.readthedocs.io/en/latest/index.html [Accessed 26Apr. 2022].

Mitchell, T.M. (1997a). *Machine Learning*. New York: McGraw – Hill Education [online]. Available at:https://www.cin.ufpe.br/~cavmj/Machine%20-%20Learning%20-%20Tom%20Mitchell.pdf [Accessed 19 Apr.2022].

Mitchell, T.M. (1997b). *Machine Learning*. New York: McGraw – Hill Education [online]. Available at:https://www.cin.ufpe.br/~cavmj/Machine%20-%20Learning%20-%20Tom%20Mitchell.pdf [Accessed 19 Apr.2022].

Mitchell, J.B.O. (2014a). Machine learning methods in chemoinformatics. *Wiley interdisciplinary reviews: Computational Molecular Science.* [online] 4(5). Available at: https://www.researchgate.net/publication/260436143_Machine_learning_methods_in_chemoinformatics [Accessed 13 Apr.2022].

Mitchell, J.B.O. (2014b). Machine learning methods in chemoinformatics. *Wiley interdisciplinary reviews: Computational Molecular Science.* [online] 4(5). Available at: https://www.researchgate.net/publication/260436143_Machine_learning_methods_in_chemoinformatics [Accessed 13 Apr.2022].

Mesty´an, M., Yasseri, T and Kertesz, J. (2013). Early prediction of movie box office success based on Wikipedia activity big data. *PloS one*, [online] 8(8). Available at: https://www.researchgate.net/publication/256291608_Early_Prediction_of_Movie_Box_Office_Success_Based_on_Wikipedia_Activity_Big_Data [Accessed 15 Apr.2022].

Mojo. (2019). *Domestic Box Office For 2019*. [online] Mojo. Available at: https://www.boxofficemojo.com/year/2019/?grossesOption=calendarGrosses [Accessed 8 Apr. 2022].

Mojo. (2019). *2019 Worldwide Box Office*. [online] Box Office Mojo.Available at: https://www.boxofficemojo.com/year/world/2019/ [Accessed 5Apr. 2022].

Oh, C., Roumani, Y. and Nwankpa, J.K., Hu H.F. (2017). Beyond likes and tweets: Consumer engagement behavior and movie box office in social media. *Information & Management*. [online], 54(1), pp.25–37. Available at: https://www.researchgate.net/publication/299020569_Beyond_Likes_and_Tweets_Consumer_Engagement_Behavior_and_Movie_Box_Office_in_Social_Media [Accessed 27 Mar. 2022].

Pedregosa, F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V., Vanderplas J., Passos A., Cournapeau D., Brucher M., Perrot M. and Duchesnay E. (2011), Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, [online] 1 2 , pp.2825–2830. Available at: https://www.researchgate.net/publication/51969319_Scikit-learn_Machine_Learning_in_Python [Accessed 29 Apr. 2022].

Reddy, A. S. S., Kasat, P. (2012). Box-Office Opening Prediction of Movies based on Hype Analysis through Data Mining. *International Journal of Computer Applications*. [online], 56(1), pp.1–5. Available at: https://www.researchgate.net/publication/258652717_Box-Office_Opening_Prediction_of_Movies_based_on_Hype_Analysis_through_Data_Mining [Accessed 26 Mar. 2022].

Rhee, T. G., and Zulkernine, F. (2016). Predicting movie box office profitability: a neural network approach. *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, [online] pp.665-670. Available at: https://www.researchgate.net/publication/313455341_Predicting_Movie_Box_Office_Profitability_A_Neural_Network_Approach [Accessed 25 Mar. 2022].

The-Numbers. (2021). *Movie Production Companies*. [online] The-Numbers. Available at: https://www.the-numbers.com/movies/production-companies/ [Accessed 5Apr. 2022].

The-Numbers. (2021). *Movie Budget and Financial Performance Records*. [online] The-Numbers.Available at: https://www.the-numbers.com/movie/budgets.htm [Accessed 10Apr. 2022].

Wasu J. (2013). Prediction of Movie Success using Sentiment Analysis of Tweets. *The Proceeding of International Conference on Soft Computing and Software Engineering 2013*, [online], 3(3), 3(3). Available at: http://www.jscse.com/papers/vol3.no3/vol3.no3.46.pdf [Accessed 28 Mar. 2022].

XGB. (2021). *XGB Documentation*. [online] XGB.Available at: https://xgboost.readthedocs.io/en/latest/index.html, [Accessed 26Apr. 2022].

Zhou, Y., Zhang, L., and Yi, Z. (207). Predicting movie box-office revenues using deep neural networks. *Neural Computing and Applications*, [online] pp.1-11. Available at: https://www.researchgate.net/publication/318831837_Predicting_movie_box-office_revenues_using_deep_neural_networks [Accessed 23 Mar. 2022].