UNIVERSITY OF TARTU

Institute of Computer Science

Software Engineering Curriculum

Stanislav Bondarenko

# Prediction of a movie's box office using pre-release data

Master's Thesis (30 ECTS)

Supervisor: Rajesh Sharma PhD

Tartu 2020

**Prediction of a movie's box office using pre-release data**

**Abstract:**

It's difficult to overestimate the impact of the film industry in our lives, it expands our knowledge about the world and culture and entertains. Going to the cinema has become an important leisure activity. Moreover, the total worldwide box office in 2018 hit a significant amount of $41B. This is not surprising as only in 2018 there were released 11,911 feature-length films worldwide. The box office generated from cinema ticket sales is the main source of profit for widely released movies. However, not all movies are successful in terms of profit when the cost of production is compared with the total box office. 78% of movies released worldwide are not profitable and 35% of profitable movies earn 80% of the total profit. Seeing the importance of theatrical screenplays and tough competition for the profit made, we want to be able to predict how successful a movie is going to be and whether it is worth taking the risk of investment. Only pre-release available data is used to be able to make a prediction at the earliest stages. We went through several stages typical for data mining and machine learning to obtain possibly the biggest and feature-rich dataset used in box office gross prediction. We use neural networks and gradient boosting machines to be able to predict the absolute box office gross, predict within which range it is likely to be, and whether a movie will be profitable, and the results obtained are very competitive in the domain.

**Keywords:**

Regression, Classification, Motion pictures, Box office, Neural networks, LightGBM

**CERCS:** P170 Computer science, numerical analysis, systems, control

**Kassahittide ennustamine, toetudes väljaandmiseelsetele andmetele**

**Lühikokkuvõte:**

Filmitööstusel on ühiskonnale märkimisväärne mõju. See avardab inimestele teadmisi maailmast, kultuurist ning on ka meelelahutuseks. Kinos käimine on muutunud oluliseks harjumuseks inimestele. Aastal 2018 linastus maailmas 11 911 mängufilmi, mis moodustas kokku käiveks 41 billion dollarit. Kassahittide piletite müügitulust on võimaldatud pakkuda inimestele laia filmivalikut kinodes. Kuid mitte kõik filmid ei ole kasumi mõttes edukad, võttes arvesse nende produktsioonikulusid. 78% kogu maailma filmidest ei ole tulutoovad. 35% edukatest filmidest, moodustavad kokku 80% filmitööstuse kasumist. Võttes arvesse filmi stsenaariumi tähtsust ning tihedat konkurentsi. Selleks analüüsib käesolev töö, kui edukaks võib film kujuneda ning kas tasub investeerimisriski võtta. Filmi edu ennustamiseks kasutatakse linateose avaldamiseelseid andmeid. Mitmekülgse andmekogumi koostamiseks sai läbi viidud erinevaid tüüpilisi masinõppe etappe. Antud andmekogu kasutatakse filmi edukuse ennustamiseks. Edukuse määramisel kasutatakse tehisnärvivõrke ja gradiendi masinaid. Saadud tulemuste põhjal on võimalik määratleda filmi populaarsus ning selle kasumlikkuse. Saadud tulemused on filmitööstuses vägagi konkuretsivõimelised.

**Võtmesõnad:**

Regression, Klassifikatsioon, Film, Kassahitt, Tehisnärvivõrk, LightGBM

**CERCS:** P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria)

# Table of Contents

# 1 Introduction

The Film industry plays a very important role in our life, it has a crucial impact on society, it expands our knowledge about history and culture, inspires and entertains [1]. As well as having good impact, movies may have influence also on our bad habits such as smoking [2], a more aggressive attitude caused by watching violent movies [3] and even on suicide rate [4].

There are dozens of actors who have more than 10M followers on Facebook, Twitter, or Instagram[1]. Often they are involved in the community and may influence it by sharing their lifestyle, expressing their opinions on various events (including political views), doing charity, or, for example, supporting humanitarian work or doing anti-war activism [5].

One point of view to understand how many people watch movies may be taken from the statistics of Netflix - one of the largest sources of Internet streaming traffic. At the moment (2020) it has 195M paid subscribers worldwide. In 2017 every user has watched approximately 60 films, which equals one movie every 6 days [2].

To understand the scale of the film industry, we will take statistics for 2018. Only in 2018, there were released 11,911 feature-length films worldwide according to IMDb [3]. The total worldwide box office in 2018 hit a significant amount of $41B [4]. To understand the scale better, we use the UNESCO statistics [5]. $3.5M tickets were sold only in Estonia, with on average 3 tickets per capita in 2018. Going to the cinema has become an important practice in our lives. Watching a theatrical release is different compared to watching a movie at home via the internet or television. This action requires a level of commitment and is a cultural practice. Despite the facts given above about the extensive usage of streaming services, they cannot substitute theatrical experience. Steven Spielberg commented on this with "*There's nothing like going to a big dark theater with people you've never met before, and having the experience wash over you.*" [6].

The box office generated from cinema ticket sales is the main source of profit for wide released (more than 600 theaters) movies [6]. However, not all movies are successful in terms of profit by comparing the cost of production with the total box office. Arthur De Vany in his book "The movies" [7] accents on the dominance of extreme events: 78% of worldwide released movies are not profitable, and 35% of movies which make money earn 80% of the total profit. In Hollywood the numbers are more extreme: 80% of Hollywood's profit was earned by 6.3% of movies during 1995-2005.

Seeing the importance of theatrical screenplays and hard competition for the made profit, we want to predict how successful a movie will be in terms of box office gross. Movie success may be interpreted by a number of factors, including its popularity on social media, critics' feedback, and ratings on movie aggregation websites. Fear and Loathing in Las Vegas [7], Donnie Darko [8],

[1] https://fanpagelist.com/category/actors/view/list/sort/fans/

[2] https://techjury.net/stats-about/netflix/

[3] https://www.imdb.com/search/title/?title$_{type} = featureyear = 2018 - 01 - 01, 2018 - 12 - 31sort = num_votes, desc$

[4] https://www.forbes.com/sites/markhughes/2018/12/31/2018-sets-new-box-office-record-with-enormous-41-billion-worldwide

[5] http://data.uis.unesco.org/

[6] https://www.theverge.com/2019/2/18/18229794/steven-spielberg-streaming-theatrical-films-netflix-roma

[7] https://www.imdb.com/title/tt0120669/

[8] https://www.imdb.com/title/tt0246578/

Fight Club [9] (until it would go on to sell six million DVD copies) failed at the box office, but left a significant trace in the history of cinematography and took high places in movie ratings. Even more, consider The Shawshank Redemption [10], a movie that barely paid off became the greatest movie of all time by IMDb rating! By predicting the box office gross, we solve only one aspect of movies' success. However, even with the problem being narrowed to box office prediction, it stays over simplified. It does not include home entertainment (DVD, Blu-ray, etc.), television deals, video on demand, merchandise sales. It also does not include costs on marketing and advertising, physical costs on film distribution, interest, and taxes. Nevertheless, the prediction may show valuable insights and serve as a reference point for movie producers and investors in the early stage of production. Only on the base of accurate box office estimation of a film, can we determine the cinema number to show this film, the propaganda cost, and the period of showing it to get more profit [8].

We will try to predict movies' success in 3 different ways: an absolute value of box office gross, discretized into 9 ranges, box office gross, and yes/no on whether the movie will be profitable. These problems will be solved via regression, multiclass classification, and binary classification, correspondingly. We plan to only use prerelease available data, data which could be obtained before most investments are made. This means we cannot use different ratings, revenue-related features such as opening weekend box office, aggregation features such as total gross of all actor's movies, word-of-mouth data, etc. However, still, we will have a big dataset with a number of various features.

The rest of the paper is organized as follows. Section 2 provides a survey of studies done in the domain area. It will not cover all the papers related to predictions in the film industry, but will focus on the ones on box office gross prediction, with particular focus on pre-release data. Section 3 describes data collection, cleaning, processing, and engineering new features. Section 4 provides information on which models will be developed and how they will be tested. In Section 5 we will show the obtained results and will discuss existing limitations and suggestions for future improvements.

[9] https://www.imdb.com/title/tt0137523/

[10] https://www.imdb.com/title/tt0111161/

# 2 Related Work

We give an overview of the domain by analyzing different types of studies with results and key findings which will help in the current study. Researches on movies' financial success prediction can be classified in a number of ways. The following classification serves only as a basis to understand the differences between studies and does not correspond to the order of the material presented below. In fact, we describe and give examples from the researches only related to our study and gradually narrow the topic without the goal to cover all the domains.

First of all, we will tell about the types of research process, namely exploratory analysis and predictive analysis. Secondly, will discuss research goals, to predict ratings or box office, analyze reviews, or social network service (SNS) data. Then follows a big subsection on used features and datasets. Afterwards, the section finishes with a review of used machine learning algorithms.

The structure of this section can be shown in the following way:

The types of research process:

- Exploratory analysis
- Predictive analysis

Targets to predict/analyze:

- Reviews analysis from critics and from the audience
- Rating prediction from critics and from the audience
- Analysis of mentions in social networks and prediction of the number of mentions
- Prediction of the box office (domestic or worldwide, premiere or all-time)

Data features used for training:

- Pre-production data:
  - Actors, director, genre, production company, producer country, sequels, etc.
- Pre-release data:
  - Metadata: production cost, number of theaters/screens, date of release, competition by other movies during the premiere, MPAA rating, runtime, etc.)
  - Social media data before the release
  - Media (trailers, posters)
- Post-release data:
  - Social media mentions, their number, and sentimentality after release
  - Ratings and reviews on movies' websites
  - Search engine queries and their number, number of translations, etc.
  - Box office each day after release and related numbers.
- Combined data

Machine learning models used

Datasets used

## 2.1 Exploratory Analysis

The goal of the exploratory analysis is to determine in which way movie features and people involved, or the effect of social media can explain key variables such as the box office or rating.

While analyzing pre-release data Tadimari et al. proposed that a movie trailer may influence its box office [9]. They showed that metadata that do not include a trailer (production budget, genre, MPAA rating, release period, the existence of sequels, and number of movies the main actor starred in) can explain up to 61% of the variance in the opening weekend box office. Additionally, different audio and video features extracted from a trailer with configured CNN (convolutional neural network) can explain 11% of the variance and explain 65% of the variance in combination with the metadata. While the mentioned research is not directly related to ours, we would like to mention that the authors emphasize a number of outliers while solving the box office prediction problem. Huge marketing campaigns can increase the box office significantly, while it is not captured directly in any of the metadata features. Examples are "Iron Man 3" and "Hunger games" [9].

Biramane et al. built a graph on 5000 Hollywood and Bollywood movies to establish links between actors and directors, actors, and movie genres they play in, directors and movie genres, production companies, and movie genres [10]. Authors mentioned that sometimes created synergy between movie key people, as for example a movie starring Leonardo Di Caprio and directed by Martin Scorsese most probably will attract many more viewers compared to either of them being absent.

## 2.2 Predictive Analytics

The prediction of target movies can be split into regression tasks, binary classification tasks and multi-class classification tasks. Summary for target variables and metrics used is shown in Table 4 .

Masrury et al. explored different methods to predict movies' profitability in terms of revenue being higher than budget [11]. They gathered a dataset of the top 150 English movies released in the US for each of 2008-2017 years. After removing incomplete ones they left 667 movies to train on and 286 to test.

[11] ANN: Artificial Neural Network

[12] NB: Naive Bayes

[13] SVM: Support Vector Machine

[14] DT: Decision Tree

[15] NN: Neural Network

[16] LR: Linear Regression

[17] CART: Classification And Regression Trees

[18] MR: Multiple Regression

[19] DNN: Deep Neural Network

[20] AB: AdaBoost

[21] RF: Random Forest

[22] SGD: Stochastic Gradient Descent

[23] Logistic Regression

[24] DAN: Dynamic Artificial Neural Network

[25] GNN: Graph Neural Network

[26] DA: Discriminant Analysis

[27] BT: Boosted Trees

Table 1: Results summary for the related studies

| Target variable | Methods | Results | Notes | Source |
|---|---|---|---|---|
| Binary classification | | | | |
| is profitable? | ANN[11], NB[12],<br>SVM[13] | ANN accuracy 0.80<br>ANN F score 0.86 | NN performed the best | [11] |
| is profitable? | DT[14] | accuracy 0.71 | Very small dataset | [12] |
| is half revenue > budget? | NN[15] | accuracy 0.89 | Highly unbalanced<br>3 times more negatives<br>Post-release features | [13] |
| is revenue - budget<br>> 0.25 of one std above mean? | LR[16] | accuracy 0.77<br>F score 0.75<br>AUC-ROC 0.80 | | [14] |
| 1. is RT critics score >60?<br>2. is RT audience score >64?<br>3. is US revenue >$7.49M?<br>4. is US opening weekend gross >$500K?<br>5. is IMDb score >6.5? | SVM | Accuracy:<br>1. 0.68<br>2. 0.69<br>3. 0.88<br>4. 0.87<br>5. 0.71 | Post-release features | [15] |
| is profitable? | NN, CART[17], MR[18] | accuracy 0.93 | Small dataset<br>Post-release features<br>NN performed the best | [16] |
| Multi-class classification | | | | |
| 4 ranges of revenue | NB, SVM | NB accuracy 0.47<br>SVM accuracy 0.41 | | [17] |
| 4 ranges of revenue | DNN[19] | accuracy 0.52<br>1-away 0.88 | | [18] |
| 5 ranges of revenue | NN, SVM | NN accuracy 0.48<br>NN 1-away 0.88<br>SVM accuracy 0.48<br>SVM 1-away 0.84 | Poorly predicts class 3<br>Post-release features | [19] |
| 5 ranges of revenue | AB[20], RF[21], NB,<br>SGD[22], SVM,<br>LR, NN | NN accuracy 0.55<br>NN 1-away 0.85 | NN performed the best<br>Post-release features | [20] |
| 6 ranges of revenue | NN | accuracy 0.68<br>1-away 0.97 | Small dataset.<br>Accuracy is not even, i.e. class 4: 0.35, class 6: 0.65 | [8] |
| 6 ranges of revenue | Log R[23],<br>NB, SVM | accuracy 0.77 | | [21] |
| 9 ranges of revenue | NN, CART, MR | accuracy 0.60 | NN performed the best<br>Post-release features | [16] |
| 9 ranges of revenue | DAN[24] | accuracy 0.74<br>F score 0.71 | | [22] |
| 9 ranges of revenue | GNN[25] + RF | accuracy 0.33<br>AUC-ROC 0.735 | | [23] |
| 9 ranges of revenue | NN, LR,<br>DA[26], RF | NN accuracy 0.37<br>NN 1-away 0.752 | NN performed the best | [24] |
| 9 ranges of revenue | Fusion of NN,<br>SVM, RF,<br>BT[27], CART | accuracy 0.56<br>1-away 0.91 | | [25] |
| Regression | | | | |
| US domestic box office gross for the first weekend | 3 NN models accepting different ranges of data | MAE $4.3M | | [26] |

Lash et al. gave some incites on used features, as that horror genre positively impacts profitability because often they don't require a big budget, and NC-17 MPAA rating impacts profitability very negatively [14]. It's indeed true, as MPAA rating is an indicator of the age demographics and therefore audience size which can potentially view a movie.

Hsu et al. predicted IMDb (Internet Movie Database) movies rating[28] using pre-release metadata (genres, directors, actors, etc.) [27]. The authors developed a large set of categorical nominal features for every attribute for 32968 movies. Insights are that neural networks can forecast users' ratings better than linear combination and multiple linear regression algorithms with 0.69, 0.73, and 0.81 average PAE respectively for the rating with 1-10 scale. They emphasize on the importance of a big dataset and on the fact that user rating significantly depends on actors, director, and writer of a movie. Unfortunately, it is an extremely difficult task to gather a dataset of comparable size which would have both budget and revenue. The exact production budget and advertising budget are known to be industry trade secrets and are not publicly released [8].

Sharda et al. may be considered the most noticeable authors in the domain area. Over the 10 years from 2000 to 2010, they published 3 papers on classification by movie's revenue ranges, significantly improving their own results. Their dataset, feature set and target discretization on 9 classes will be reproduced in attempts to improve the result by other authors. We are going to repeat his classes schema as well.

## 2.3 Prediction Features

The summary of used prediction features is shown in Table 2. It should be noted that many features are calculated on the basis of already existing information. If we want to predict based on pre-release data only, we should not take into account any information which appeared after a movie was released. For example, the average actor's box office should be calculated only on movies that were released earlier, not all the movies. It also means we can not use any ratings and review related features as they appear after a movie is released. Most studies don't mention explicitly whether they were using strictly pre-release available information.

The next studies [12, 18, 28, 29, 30, 31, 32] are given to describe the application of different sets of features and provide valuable insights about data processing.

### Social Network Service Data

Taegu Kim et al. provide an excellent example of the box office prediction using social network service data [28]. Taking into consideration how challenging it is to predict the box office as a regression problem, they archived a very good result: approximately 10% MAPE while predicting the first-week premiere box office. They showed that using SNS data such as number of different mentions, their sentimentality, increase in the number of mentions and their emotional increase can reduce MAPE in such prediction up to 40%. Such a good result may be explained by using post-release features such as SNS data and exact data about the numbers of screens and their change during weeks combined with a relatively small and limited dataset of 212 Korean movies. They obtained almost the same result using SVR, GPR and KNN models and accent on the importance and quality of data. The authors give a good insight on how important it is to

[28] https://help.imdb.com/article/imdb/track-movies-tv/ratings-faq/G67Y87TFYYP6TWAVratings

know the number of screens and the number of theatres while predicting the box office. They did not use the data which would show if a movie is released simultaneously with another one (which leads to the lower box office on both of them), but emphasize the importance of such data[28].

In addition to the mentioned above study, Hossein and Miller [29] showed that using Twitter SNS data even before release can help to predict the opening weekend domestic box office as binary classification (hit or flop). Even four days prior to the release twitter data can give 67% F-score which is approximately the same result comparing with 4 days after the release [29]. Authors showed that using binary classification on a limited dataset (86 popular movies) linear kernel SVM gives better results than neural network and Gaussian kernel SVM gives the worst result. This is the only case we found in which neural networks performed worse than other algorithms.

Another word of mouth source used is Wikipedia [33]. It was shown that Wikipedia activity metrics (number of views, number of editors, number of edits) directly reflect movie's popularity and highly correlate with revenue, which makes them useful features in addition to existing ones. The downside of these data is that their obtaining is quite time-consuming.

Ru et al. predicted box office on each day out of 21 days after the movie's release. The study is different from the one we have, but it gives information on how important is word-of-mouth. They used LSTM with micro-blog index and different word-of-mouth metrics. They confirmed that many audiences will choose whether to watch the movie according to the quality of word-of-mouth. Indeed, based on our own experience, a person is much more likely to watch a movie if somebody close advised it. The great value of the micro-blog index before the movie is released shows that the marketing effect of the movie is pretty good. A strong correlation between the micro-blog index and the movie box office was shown [34]. Although the used dataset is quite small (80 movies to train, 34 to test), the box office gross MAPE of 30.1% is a relatively quite high result.

It's no surprise that existing post-release forecasting models give high quality results. However, these forecasts are run too late, when investor's or studio's money already spent. Our goal is to do predictions based only on pre-release data which will give more freedom to those who are in charge of movie financing. "If accurate box-office revenue forecasts can be made before significant investment in development or production, a movie studio could save millions by avoiding a single flop. Due to the scale of investment and expense involved in modern motion pictures, even a marginal increase in the success rate of the “green-lighting” process would bring remarkable financial and reputational benefits to the studios and stakeholders involved" [22].

### Pre-Production and Pre-Release Data

Talking about non-standard pre-release data, we can refer to Zhou and Yen [18, 30]. Additionally to postrelease metadata from IMDb (score, rating, comments, participants, budget, duration, genres) authors used features extracted with a convolutional neural network from film teaser posters. While the data used are not directly related to the current research, the authors show us that multilayered back propagation network (MLBP) or deep neural network (DNN) give significantly better performance compared to support vector machine (SVM) and RF (random forest).

Despite the used approach of the binary box office classification (hit or flop) is using a decision tree, we want to mention the research of Burgos et al. [12]. The authors showed that among the used pre-release metadata features release month, genre and the production cost are the most important ones to predict profitability. The authors used a limited dataset of 100 US movies and emphasize on using a large dataset to produce more representative models.

Regarding the usage of different features and data processing, we want to mention the study of Di et al. [32]. The goal of the study is to predict the absolute box office value on the limited dataset of 104 recent Chinese movies using a multilayered perceptron. The authors proposed a new approach to use actors and directors, not as a nominal binary feature (present or not). The star power of an actor depends on whether the actor is a star or a comprimario and the box office of the 3 most successful movies they starred in. The more time passed between the releases of a such movie and the one to predict, the less contribution it brings to the star power. Also, contribution decays if genres of the movies are not similar. Directors' power depends also on the top 3 movies they directed, which also decays with time passed between the releases of the successful movie and the one to predict. This approach is different from the one taken by Quader et al. [20] to rate actors and directors by their lifetime gross income from previous movies. It is also different from the one taken by Meenakshi et al. [35]. The authors split movies' dataset on clusters depending on the profit (flop, average, success), and calculated star power taking the average cluster of movies they starred in.

Di et al. also propose to take into account movie release date to know if it overlaps with a specific festival or date (as for China these are, for example, the Chinese spring festival, Dragon boat festival) [32]. They also use the power of a distributor based on the number of movies release, the intensity in the release schedule, and a binary variable expressing if the movie is released in 3D and IMAX.

### Combination of Features

An example of combining pre-release metadata with post-release ratings and reviews for binary profitability prediction with a neural network can be found in the study of Rhee and Zulkernine [31]. High performance can be explained by the limited dataset of 375 top IMDb movies used. Similarly to [9] the authors calculate the actor star power by counting how many times an actor appears in the final movie dataset. The average movie actor star-power is used later on while forecasting. The same is done for directors. The authors mentioned the importance to take into account specific dates or festivals, since opening weekend during Valentine's Day, Thanksgiving or Christmas, for example, will gather more audiences in theatres.

### Impact of Post-Release Data on the Prediction Result

Quader et al. [19] compared prediction results using only pre-release data (MPAA rating, actor star power, release month, budget, number of screens, etc.) and those combined with post-release data (IMDb rating, Tomato meter, Tomato rating, Audience Meter, Metascore, IMDb Review Sentiment value etc.). As another study from the same authors [20] they use different machine learning methods to predict the box office class from the range split into 5 classes. Again, the best result is archived using neural networks (compared with SVM). 83.44% accuracy is gained using pre-release data while adding post-release data does not raise the result remarkably - 88.87% for one-away prediction. For the exact match the results are 48.41% and 58.41% respectively. [19]

Table 2: Overview of features used in movie box-office or ranking prediction studies

| Feature | Explanation | Sources |
|---|---|---|
| Age rating | * MPAA rating in binary form (5 features) | [11, 14, 15, 16, 17, 19, 20, 22, 23, 36]<br>[8, 10, 12, 13, 17, 21, 24, 25, 37] |
| | * Movie being restricted in US | [12] |
| | ** Log of average gross for each MPAA rating | [26] |
| Genres | * Genres in binary form | [8, 11, 14, 15, 16, 38],<br>[17, 18, 21, 23, 25, 26] |
| | ** Annual profitability percentage by genre | [14] |
| | ** Annual weighted profit by genre | [14] |
| | ** Average IMDb rating by genre | [27] |
| Runtime | * In minutes | [15, 17, 18, 21, 22, 23, 38] |
| | ** Average of IMDb rating by runtime range | [27] |
| | * Duration range | [12] |
| Budget | * Absolute value | [11, 15, 19, 38],<br>[12, 13, 16, 17, 18, 20, 23] |
| Distribution information | * Number of screens | [8, 19, 20, 22, 24, 25, 26, 38, 39] |
| | * Number of cinemas | [8, 23] |
| | * Number of plays in the initial day of release | [37] |
| | ** Number of movies distributor released | [39] |
| Production company | * Binary form | [15, 38] |
| | ** Number of movies company made | [39] |
| Movie propaganda | * Average number of results in Google | [8] |
| Star power | ** Sum of awards for cast (Academy, Globe, Oscar) | [15, 16, 26] |
| | ** Sum of how many times actor(s) appear in used dataset | [11, 13, 21] |
| | ** Average class (by revenue) of movies where actors were starred | [35] |
| | ** Average gross per actor | [14] |
| | ** Sum of total gross of all movies for all stars | [19, 20] |
| | * Average box office for last N movies | [23, 38] |
| | * Average box office for last N movies with decrease in time or if genres are different | [39] |
| | ** Total and average tenure | [14] |
| | Average number of results in Google | [8] |
| | ** Is present in top 100 box office mojo people index by gross | [16] |
| | ** Present in top 50 top grossing actors from dataset | [26] |
| | Number of Social media likes (Facebook) | [17, 18] |
| | ** Split in classes (low, average, high) | [22, 24, 25] |
| | Average of average of IMDb rating of actors' movies | [27] |
| Director power | ** Sum of how many times director appears in used dataset | [11, 13, 21] |
| | ** Average class (by revenue) of movies director shot | [35] |
| | ** Average director's gross | [14] |
| | ** Sum of total gross of all movies by director | [19, 20] |
| | * Average box office for last N movies | [23, 38] |
| | Average box office for last N movies with decrease in time | [39] |
| | * Average box office to budget ratio for last N movies | [38] |
| | Average number of results in Google | [8] |
| | ** Is present in top 100 box office mojo people index by gross | [16] |
| | ** Sum of awards for director | [15, 26] |
| | Average of IMDb ratings of director's movies | [27] |
| | * Average box office for last N movies | [38] |
| | * Average box office to budget ratio for last N movies | [38] |

Table 2: Overview of features used in movie box-office or ranking prediction studies

| Feature | Explanation | Sources |
|---|---|---|
| Composer | * Average class (by revenue) of movies composer worked on | [35] |
| Writer | ** Sum of how many times writer appears in used dataset | [21] |
| | Average of IMDb ratings of movies by writer | [27] |
| Team features | ** Cohesion, collaboration | [14] |
| Production country | * Home/import weights | [8] |
| | * Made in US (binary) | [21] |
| | Average of IMDb ratings of movies by country | [27] |
| * Languages | Number of languages translated to | [21] |
| Release date | * Month in binary form | [11, 15, 17, 20, 21] |
| | * Month as integer [1,12] | [19] |
| | ** Average total box office for a month | [8] |
| | ** Average total box office for a week | [8] |
| | * Year in binary form | [15, 21] |
| | * Average annual profit | [14] |
| | * Number of consecutive holidays | [38] |
| | * Season (binary) | [14, 16] |
| | * Release on holiday (weekend) binary | [13, 14, 26] |
| | ** Average total box office within a week to a festival | [8] |
| | * Calculated seasonality coefficient | [22] |
| Movie ratings and number of votes | IMDb | [13, 18, 19, 20] |
| | Rotten tomatoes | [13, 16, 19, 20] |
| | Metacritic | [13, 16, 19, 20] |
| Reviews and number of reviews | Sentiment analysis | [19, 20] |
| Text analysis | * Popular words from plot, topic, sentiment analysis of summary | [14, 15, 21] |
| Sequel | * Ordinal number of sequel position | [38] |
| | * Binary yes/no | [13, 16, 24, 25, 37] |
| Competition | * No competition (binary) | [38] |
| | * Number of releases for the particular day | [38] |
| | * Split in classes (low, average, high) | [22, 23, 24, 25] |
| | * 1 / N movies released within 1 week | [13] |
| Songs | * Number of hit songs | [38] |
| Movie base | * True story | [38] |
| | * Book adaptation | [38, 39] |
| Special effects | * Split in classes (low, average, high) | [22, 24, 25] |
| Advertisement expenditure | * Absolute value | [22] |
| Technical | * 3D or IMAX | [39] |

* Features available before release

** Features available before release if calculated only for the previous movies

## 2.4 Datasets

A number of authors mentioned that Box Office Mojo or IMDb data were gathered with help of web-scraper software. We deliberately didn't do so as IMDB's Conditions of Usage state: "You may not use data mining, robots, screen scraping, or similar data gathering and extraction tools on this site, except with our express written consent as noted below" [29]. As it was already mentioned, it is hard to obtain a big dataset of movies which would have budget and box office data. To the best of our knowledge, the biggest dataset used for box office prediction is contained 4260 movies. IMDb is the most popular resource to get movie data. We see a clear pattern that datasets with biggest number of movies are taken from IMDb, but with increasing dataset's quantity, quality rapidly drops. Often the information is so noisy and lacks most needed features, that some authors bought proprietary datasets for the sake of quality. Our goal is to gather a dataset with both budget and revenue with much more movies than existing ones.

The summary of datasets used in related literature is show in Table 3.

Table 3: Brief overview of datasets used in prediction studies

| **N. movies** | **Period** | **Market** | **Source** | **Description** | **Budget/ revenue present?** | **Study** |
|---|---|---|---|---|---|---|
| 86 | 2013-2014 | | Comingsoon.net | | yes | [29] |
| 104 | | | Box office mojo | having the most present features | yes | [12] |
| 104 | 2013-2015 | Chinese | | | yes | **[?** ] |
| 114 | 2015-2016 | China | | Chinese movies with $80M+ box office | yes | [34] |
| 212 | 2011-2013 | Korea | | Movies with original titles and 100K+ audience | yes | [28] |
| 241 | 2005-2006 | China | Wanda Cinema Line Company | Purchased proprietary dataset | yes | [8] |
| 250 | 2014-2017 | India | Wikipedia, RadioMirchi, BoxOfficeIndia | 250 Bollywood movies which have the most filled information | yes | [38] |
| 354 | 1999-2010 | | Kantar | Proprietary dataset, but bought to assess advertising expenditures | yes | [22] |
| 375 | 2012-2014 | Korea | Korean Film Counsil | Bought dataset with lots of additional features | yes | [37] |
| 375 | 2010-2015 | | OpusData | Selected from top 100 for each year | yes | [13] |
| 755 | 2012-2015 | | IMDb, RT, Metacritic, Box Office Mojo | Movies which have the most filled information | yes | [19] |
| 977 | 2006-2011 | | Box office mojo | High grossing 1̃50 movies per year | yes | [23] |
| 1000 | 2008-2017 | US | IMDb | Selected from top grossed 150 English movies released in US for each year | yes | [11] |
| 1353 | 1921-2014 | | Box office mojo | | yes | [14] |
| 1718 | 2005-2009 | | Metacritic and The Numbers | Train movies 2005-2007, validate 2008, test 2009 | yes | [26] |
| 1920 | 2000-2016 | | IMDb and Opus Data | | yes | [16] |
| 2632 | 1998-2006 | USA | | Purchased | yes | [25] |
| 3177 | 2000-2015 | | Opus data quries by IMDb list of titles | All which have revenue | yes | [21] |
| 3807 | | | IMDb | All which have revenue | yes | [18] |
| 4260 | | US | IMDb | Ones which are movies, not adult films, have both budget and revenue, have user rating | yes | [17] |
| 5000 | 2000-2015 | | IMDb | | no | [10] |
| 5043 | 1917-2017 | | IMDb | | no | [36] |
| 6590 | | | IMDb, RT, Wikipedia | | no | [15] |
| 32698 | 2002-2012 | | IMDb | | no | [27] |

[29] https://www.imdb.com/conditions

## 2.5 Machine Leaning Algorithms and Neural Networks in particular

The most comprehensive evaluation of different machine learning algorithms for the movie box office success prediction is the study of Quader et al. [20]. Despite the study is based on using both pre-release and post-release features which is different from our approach, we can refer to it to approve our decision to focus primarily on neural networks. The goal of the study was to predict 5 target classes for the box office using different machine learning algorithms: Multilayer Perceptron, Support Vector Machine (SVM), Stochastic Gradient Descent (SGD), Logistic Regression, Random Forest, AdaBoost, Gaussian Naive Bayes (listed by decreasing performance).

While conducting the current research, we want to take as the baseline approach the one used by Ghiassi et al. [22]. They showed in their study that dynamic artificial neural network (DAN) can perform much better compared to SVM while solving the box office forecasting as a multi-class classification problem. The authors took as a baseline the study of Delen and Sharda [25] with the goal to improve their achievement by using refined and improved DAN with extended movies' dataset. Ghiassi obtained a very good result for the 9 classes classification with training dataset of 354 movies: 94.1% F1 test. The authors reported that *"Although larger budgets are correlated with higher revenues, they are not correlated with higher profits; and films with smaller budgets are, on average, more profitable* [22]. Therefore, we want to predict not the box office absolute number or its range, but the ratio of it to the production cost to predict the profitability of a movie.

Another example of better performance of neural network comparing with SVM or Naive Bayes is shown by Masrury et al.[11] while forecasting whether the box office will be larger than the production cost. The authors archieved 86%, 65% and 63% F-score using ANN, NB, and SVM respectively. The authors mentioned that NB gives lower but almost homogeneous results across metrics, compared to NN, and can be used in case of time and power limitation as it is less demanding in terms of computational power. However, the obtained model cannot be considered as a representative as it is trained on data of 150 top grossed released in US movies during 2008-2017. Differently from that, we plan to use a much bigger dataset to archive higher representational power.

A similar but more representational model was developed by Galvao and Henriques [40] using 1920 movies from 2000 to 2016 to predict the worldwide all-time box office as interval value, binary value (is it twice the production cost) and class value (9 distinct classes ranged by profitability). Differently from the described approach, we do not plan to use post-release data as the number of Oscars, awards and different ratings. Neural networks showed significantly better result comparing with multiple regression and decision trees. The authors mention that budget, director and sequels features bring the most of the contribution to prediction success [40].

Ericson et al. solved several tasks in their study with predicting such targets as Rotten Tomatoes score, US box office, IMDb score etc. [15]. They confirmed that while targets are different in the subset of features they need, MPAA rating, runtime, genres and production company were important in each target.

The summary on used models and results is shown in Table 5.2.

Many studies implement features like star power which aggregates revenue over all the movie for given actors. It's okay to do so on small time-wise dataset, as for example movies from 2012 to 2015 years. But once we have a long time span, this estimation won't be correct, as there will be early movies where a given actor is not recognized yet, as well as new movies where old glory of an actor could already vanish. So it will be most precise to aggregate only along previous movies.

We aim to predict movies' box office gross using only pre-production and pre-release data. In the future, such models may help film producers to predict the approximate box office before a movie is released. Knowing the influence of different factors on the box office one may find it useful to change marketing campaigns, change or reallocate the number of theaters, duration of the premiere, etc.

[30] ANN: Artificial Neural Network
[31] NB: Naive Bayes
[32] SVM: Support Vector Machine
[33] DT: Decision Tree
[34] NN: Neural Network
[35] LR: Linear Regression
[36] CART: Classification And Regression Trees
[37] MR: Multiple Regression
[38] DNN: Deep Neural Network
[39] AB: AdaBoost
[40] RF: Random Forest
[41] SGD: Stochastic Gradient Descent
[42] Logistic Regression
[43] DAN: Dynamic Artificial Neural Network
[44] GNN: Graph Neural Network
[45] DA: Discriminant Analysis
[46] BT: Boosted Trees

Table 4: Results summary for the related studies

| Target variable | Methods | Results | Notes | Source |
|---|---|---|---|---|
| Binary classification | | | | |
| is profitable? | ANN[30], NB[31], SVM[32] | ANN accuracy 0.80<br>ANN F score 0.86 | NN performed the best | [11] |
| is profitable? | DT[33] | accuracy 0.71 | Very small dataset | [12] |
| is half revenue > budget? | NN[34] | accuracy 0.89 | Highly unbalanced<br>3 times more negatives<br>Post-release features | [13] |
| is revenue - budget > 0.25 of one std above mean? | LR[35] | accuracy 0.77<br>F score 0.75<br>AUC-ROC 0.80 | | [14] |
| 1. is RT critics score >60?<br>2. is RT audience score >64?<br>3. is US revenue >$7.49M?<br>4. is US opening weekend gross >$500K?<br>5. is IMDb score >6.5? | SVM | Accuracy:<br>1. 0.68<br>2. 0.69<br>3. 0.88<br>4. 0.87<br>5. 0.71 | Post-release features | [15] |
| is profitable? | NN, CART[36], MR[37] | accuracy 0.93 | Small dataset<br>Post-release features<br>NN performed the best | [16] |
| Multi-class classification | | | | |
| 4 ranges of revenue | NB, SVM | NB accuracy 0.47<br>SVM accuracy 0.41 | | [17] |
| 4 ranges of revenue | DNN[38] | accuracy 0.52<br>1-away 0.88 | | [18] |
| 5 ranges of revenue | NN, SVM | NN accuracy 0.48<br>NN 1-away 0.88<br>SVM accuracy 0.48<br>SVM 1-away 0.84 | Poorly predicts class 3<br>Post-release features | [19] |
| 5 ranges of revenue | AB[39], RF[40], NB, SGD[41], SVM, LR, NN | NN accuracy 0.55<br>NN 1-away 0.85 | NN performed the best<br>Post-release features | [20] |
| 6 ranges of revenue | NN | accuracy 0.68<br>1-away 0.97 | Small dataset.<br>Accuracy is not even, i.e. class 4: 0.35, class 6: 0.65 | [8] |
| 6 ranges of revenue | Log R[42], NB, SVM | accuracy 0.77 | | [21] |
| 9 ranges of revenue | NN, CART, MR | accuracy 0.60 | NN performed the best<br>Post-release features | [16] |
| 9 ranges of revenue | DAN[43] | accuracy 0.74<br>F score 0.71 | | [22] |
| 9 ranges of revenue | GNN[44] + RF | accuracy 0.33<br>AUC-ROC 0.735 | | [23] |
| 9 ranges of revenue | NN, LR, DA[45], RF | NN accuracy 0.37<br>NN 1-away 0.752 | NN performed the best | [24] |
| 9 ranges of revenue | Fusion of NN, SVM, RF, BT[46], CART | accuracy 0.56<br>1-away 0.91 | | [25] |
| Regression | | | | |
| US domestic box office gross for the first weekend | 3 NN models accepting different ranges of data | MAE $4.3M | | [26] |

# 3 Data Description

This section describes what data was collected, how it was cleaned, processed, and which features were generated. As a result, a final dataset with 6965 movies with 228 training features and 3 target features has been created.

## 3.1 Data collection

The goal of the research requires gathering the largest dataset possible while meeting the crucial requirement to have both budget and revenue data for each movie. This criterion is based on the fact that we need both of them to know whether a movie is profitable while solving the binary classification task, additionally, budget is one of the most important features while predicting box office. The second important criteria was to obtain the dataset free of charge.

A number of already prepared datasets were considered. For example, *TMDB5000*[47] dataset has 4803 movies with budget and revenue filled in 3229 movies. The OpusData dataset[48] is very comprehensive, but the obtained free of charge extract contains only 1900 movies. MovieLens[49] and OMDb API[50] datasets are focused on ratings and do not contain information on movies' budget and revenue. Free IMDb Datasets[51] are very comprehensive in terms of the number of movies, but are very limited in terms of reflected information about movies. Some studies mentioned that Box Office Mojo or IMDb data were gathered with the help of web-scraper software. We deliberately did not do so as IMDB's Conditions of Usage states: "You may not use data mining, robots, screen scraping, or similar data gathering and extraction tools on this site, except with our express written consent as noted below" [52].

After conducted research, it was decided to use the next resources:

1. The Movie Database (TMDb) API[53]

   Querying such endpoints as *movie*, *movie/credits*, *movie/release_dates*, *movie/keywords* with IDs exported from daily IDs export[54], we obtained each of the 490257 movie records with IDs available on 28.01.2020.

2. The Numbers website[55]

   It was used to obtain the name, release date, budget, and revenue for 5928 movies, 5480 of which had all these columns filled and were matched with movies obtained from TBDb by name and release date. 1066 TMDb movies' budgets and/or revenues were filled using these data.

[47] https://www.kaggle.com/tmdb/tmdb-movie-metadata
[48] https://www.opusdata.com/data.php
[49] https://grouplens.org/datasets/movielens/
[50] http://www.omdbapi.com/
[51] https://www.imdb.com/interfaces/
[52] https://www.imdb.com/conditions
[53] https://developers.themoviedb.org/3
[54] https://developers.themoviedb.org/3/getting-started/daily-file-exports
[55] https://www.the-numbers.com/movie/budgets/all

3. A Corpus of Movie Plot Synopses with Tags (MPST)[41][56]

   It contains 71 fine-grained tags and their associations with 14828 plot synopses of movies released before 2017. It allowed us to add tag features to 4430 movies out of 6965 used in the final dataset. Tag processing will be described in the section 3.2.

4. Internet Movie Database (IMDb) Datasets[57]

   It was used to get missed in TMDB runtime data.

We know that using Social Network Services (SNS) data such as word-of-mouth, micro-blog index, evaluating current actors' popularity with the number of their social media fans can improve the models' performance while still using only pre-release features. However, we wittingly omit these data to keep the scope of current research more condense.

## 3.2 Data cleaning

Initial shape of the TMDB data is 490257 rows with 29 columns listed in Appendix II. NB For simplicity, here and further a movie's worldwide box office gross will be called "revenue".

The next list describes how many movie records were removed.

1. Records which are not possible to use
   - 198 records with API status message "The resource you requested could not be found"
   - 4919 records with corrupted data (API could not return a correct JSON response)
   - 6 records with corrupted crew data

2. Movies which lack budget and/or revenue data or have an improper release date. First of all, it will not be possible to create a target for binary classification (*is_profitable* without knowing a movie's budget. Secondary, the production budget data has been consistently identified as a strong predictor of box-office performance of a movie [22], and the lack of if in a subset of movies can introduce significant noise.
   - 434385 movies which had a release date in neither TMDB nor The Numbers data.
   - 36616 movies which did not have a release date in neither TMDb nor The Numbers data
   - 5468 were not yet released movies
   - 3 movies with a release date in 2020 (at the point of data collection on 28.01.2020, they had no chance to gather most of its revenue)

3. Movies with the lack of other important information that casts doubt on their credibility.
   - 42 movies with no genres set and/or overview.
   - 180 movies which don't have runtime, or it is less than 15 minutes. Data exploration showed that many short movies were actually cartoons and can not be considered feature movies.

[56] https://ritual.uh.edu/mpst-2018/

[57] https://www.imdb.com/interfaces/

- 844 movies which don't have either production country, production company, crew, cast, or languages. While the absence of production country and languages casts doubts on the credibility of this data, production company, crew and cast are important features, lack of which prevents further data engineering. The majority of these movies did not have this information in IMDb datasets as well.

After performed data cleaning, we obtained 7596 movies to work with.

## 3.3 Data Processing

The general rule applied is to make the data usable by machine learning models which require converting them to numerical features. Cocuzzo et al. showed that Naive Bayes can gracefully handle unordered categorical features [17], but this technique was proven to be much less powerful than neural networks or tree ensembles.

Categorical and not ordinal features such as language, country, or genre need to be converted to be binary. Nominal features that are useful but are not feasible to present in a binary way need to be used to engineer new numerical features, it will be described in more detail in section 3.4. Nominal features which are not possible to convert to a useful numerical representation should be dropped. Quasi-constant features (invariant in 99% of samples) are removed.

### Immediately removed features

The next features were dropped as not relevant: *status, api_status_message, api_status_code*. Image or video processing is out of the scope of the defined task, therefore the next features *images_url. poster_url. video_url* were dropped. Another constraint defined by the task is to use only prerelease available data, therefore, *popularity, vote_average, vote_count* features were dropped as well. *adults* feature was removed as only 8 movies had it set to true, but neither of them had the budget and/or revenue data.

### Date processing

Release date feature was split on *day, month, year*. *weekend* was created which equals 1 if a movie was released on a weekend and 0 otherwise. Day and month features are cyclic and discrete, therefore, the proper way to represent them is to use a two-axis coordinate system. *day_sin, day_cos, month_sin, month_cos* features were created, which allows preserving the relation, for example, that December is close to January, and the end of the month is close to the beginning of the month. *year* feature will be used later on in feature engineering.

### Runtime

IMDb dataset was used to replace the runtime in 183 TMDb movies where it was not present or was less than 20 minutes.

### Languages

*original_language* feature was dropped as it strongly correlates with the production country. *spoken_languages* feature was converted from a list of languages to an integer number of different languages.

### Countries

The list of production countries was converted to *ISO 3166-1* country codes. Dataset had 111 unique production countries mentioned. While the maximum number of countries listed in a movie is 12, the average number is 3.08 and the median is 2. Only 17 movies (0.22%) had more than 5 countries listed. 99 countries were met in less than 1% of movies, they were moved to one feature *country__other* as they are quasi-constant (do not pass 0.01 variance threshold), and don't help to increase results. 12 countries were converted to separate binary features: *country__ES, country__JP, country__US, country__CA, country__DE, country__CN, country__IN, country__FR, country__RU, country__IT, country__AU, country__GB*. These features were considered important, since the average USA box office is higher than, for example, Indian, despite the number of tickets sold in India being higher.

### Genres

the 19 different genres mentioned assigned to movies were converted to 19 binary features. The list of genres, can be found in Appendix III. 2 genres *science_fiction* and *tv_movie* were dropped as they had no movies assigned to them. These features were considered important by the fact that action and drama movies gather much higher box office than, for example, musical or documentary.

### Collection

This feature is quite important because a viewer is already familiarized with the movie topic and much more likely to watch a sequel if he likes the previous movie. In addition, the fact of creation sequel itself points to the success of the original movie [22].

1525 movies belong to one of the 852 presented collections. For example, "Star Wars Collection" has 9 movies. Collection names were reviewed and adjusted where necessary to avoid duplication, for example, "batman" and "batman dc universe", "fright night" and "fright night (reboot)". It reduced the number of unique collections to 788. Nominal feature *collection* will be used later on in feature engineering.

### Cast

Dataset has 828 movies (11%) with less than 9 actors. It was decided to keep only the 8 main actors. It leaves 24065 unique actors, 16071 of which have only one movie listed, which means it will not be possible to assess the success of their previous movies while engineering new features. Therefore, keeping more actors in movies will not benefit since most of the data will be empty. These 8 nominal actor features will be used later on in data engineering. Cocuzzo et al. showed in their study on similar data the best number of actors to keep is 10, but they used Baive Bayes algorithm instead or aggregating information over previous movies, so they would have much less missing values [17].

Gender information was not filled in for 6748 actors. To deal with it, NLTK Names corpus [42] was used with NaiveBayesClassifier to predict missing genders based on an actor's name and surname. Thus, 8 binary gender features were created for each of the 8 actors representing "male" as 1 and "female" as 0.

### Crew

849 unique jobs were listed in the movies, 12 unique departments (namely, *costume and make-up, sound, production, directing, art, visual effects, lighting, actors, crew, writing, camera, editing*). People with job names containing one of *assistant, trainee, intern, other, thanks* were removed as not being key crew members. Some misspellings and misplaced departments were fixed manually, information in brackets was removed. It left 739 unique jobs.

Out of these 739 jobs, 192 were filled in less than 10 movies (26% of jobs in 0.13% of movies), 488 jobs were filled in less than 100 movies (66% of jobs in 1.3% of movies). However, only 25 jobs are filled in at least 10% of movies by people who have previous experience in the area. These 25 nominal job features will be used for data engineering.

### Production company

6470 production companies were mentioned. Movies have 3 companies listed on average. Only 802 movies have more than 5 production companies listed. 36% of movies have no company info filled with a company listed in at least one previous movie. This number rapidly grows after the 4th company, so it was decided to keep only the first 3 listed companies in a movie. These features are important as successful production companies are much more likely to release a successful movie than a poorly known company. They will be used later on in feature engineering.

### Age rating

81% of movies have MPAA ratings . 12% of movies have ratings other than MPAA, and 7% have no ratings. After adding the conversion from NO, DE, NL, SE, GB, KR, FR, GR, DK, PT, IN, BR, RU, IT, AU rating systems we got MPAA rating for 90% of movies. The next columns *rating__g, rating__pg, rating__pg-13, rating__r, rating__nc-17* were added as binary features. This is important information since, for example, NC-17 rated movies will be very limited to cinema screenings, and G rated movies will not gather large audiences since they are made for kids.

### Homepage

Homepage is listed in 35% of movies. *homepage_exists* binary feature was created. *homepage_repeats* feature was created with 0 for movies with not filled or unique homepage, and 2+ number for homepages which repeat more than once (91 movies in total). Repeated homepage

Table 5: Revenue ranges used in multi-class classification task in different studies. Ranges are shown in $M

| Researches | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Quader et.al [19] * | <=0.5 | >0.5<br><=1 | >1<br><=40 | >40<br><=150 | >150 | | | | |
| Zhang et al. [8] | <=4 | >4<br><=10 | >10<br><=30 | >30<br><=90 | >90<br><=200 | >200 | | | |
| Flora et al. [21] | <=0.001 | >0.001<br><=0.01 | >0.01<br><=0.1 | >0.1<br><=1 | >1<br><=10 | >10<br><=100 | | | |
| Sharda et al. [24] ** | <=1 | >1<br><=10 | >10<br><=20 | >20<br><=40 | >40<br><=65 | >65<br><=100 | >100<br><=150 | >150<br><=200 | >200 |
| Parimi et al. [23] | <=10 | >10<br><=20 | >20<br><=30 | >30<br><=45 | >45<br><=70 | >70<br><=100 | >100<br><=150 | >150<br><=200 | >200 |

* The same distribution was used in other studies [20].
** The same distribution was used in other studies [16, 22, 25, 38].

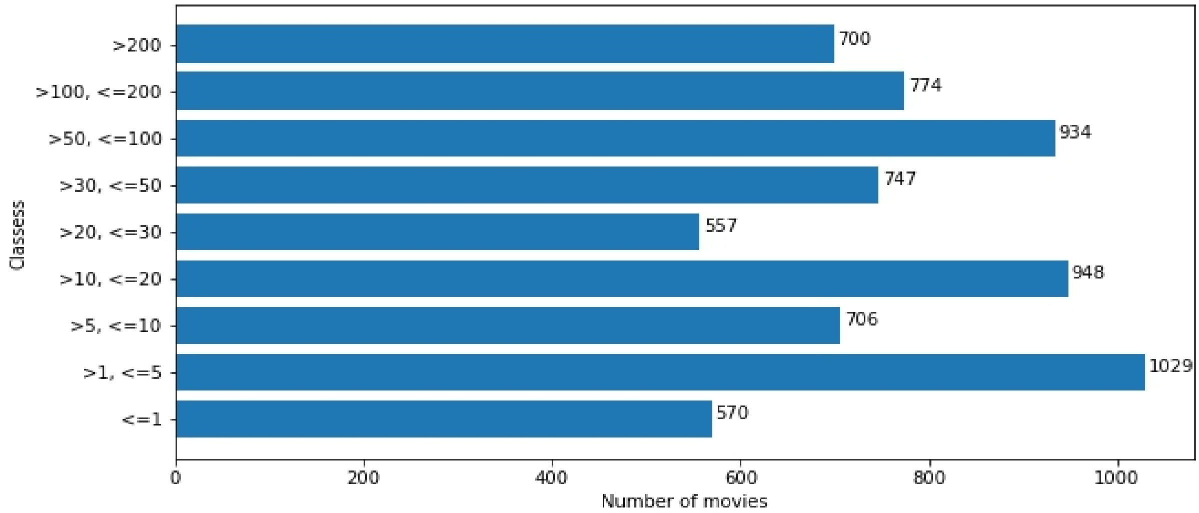


Figure 1: List of 9 class ranges in $M and number of movies per class

indicates some relation between movies, additionally, the existence of a homepage increases the movie's popularity and revenue, which helps to improve prediction.

The target used for the binary classification task is *is_profitable*, which equals 1 when *the revenue* is higher than *the budget* and equals 0 if *the revenue* is less than or equals *budget*.

The target used for the multiclass classification task is *revenue_category*. The revenue range distribution proposed by Sharda et al. [24] turned out to be the most popular. We will follow the given distribution which will allow us to compare results. The revenue ranges used in multiclass classification task in different studies are shown in the Table 5 The list of 9 classes and the number of movies per class shown is shown in the Figure 1.

## 3.4 Feature Engineering

In this study we focus more on creating new features rather than trying to minimize their number. While we did not do granular evaluation on each feature, it's been tested that each set of feature (such as tags, genres, cast average aggregations) improves the overall score.

### Processing text features

Such text features as *tagline, overview, keywords* cannot be used directly and need to be converted to a meaningful numerical representation. The best approach is to create binary features of topics or tags which can characterize a movie. However, TMDb overviews are too short (289 symbols, or 28 words on average), no words which would appear in more than 15% of overviews. After text preparation (lowercase, deleting stop words, deleting words shorter than 3 symbols, lemmatize) and running LDA[43] with Gensim library[58] on a Bag of Words (BoW) corpus and TF-IDF corpus, it still was not possible to obtain a meaningful set of topics on these overviews.

More detailed plot summaries to 42306 movies from Wikipedia up to 02.11.2012 were obtained from CMU Movie Summary Corpus[44]. Overviews on 60% of the own movies' dataset were obtained from this corpus, which significantly increased the overview length (2314 symbols, or 198 words on average). The next data preparation steps were performed: discard single quotes, lowercase, remove punctuation, remove too short or too long words, create bigrams and trigrams, remove stop words, leave only nouns adjectives verbs adverbs, lemmatize. LDA model was run with a big number of passes over BoW for each plot, which allowed to reach a coherence score of 0.529 with the best number of topics equal to 15.

Quite meaningful topics were obtained with words probability higher than 1%, such as

- film, show, music, band, perform, singer, star, performance, play, musical, movie, actor, theater, audience, concert, stage, song, director, tour
- war, soldier, agent, team, order, mission, terrorist, bomb, government, base, force, battle
- team, match, game, win, tournament, race, title, fight, billy, player, sport, play, final, coach, championship, competition, opponent, compete

Unfortunately, it was not enough, and 10-12 out of these 15 features turn out to be quasi-constant (have variance less than 1%). It was decided to abandon the idea to generate movie topics.

Instead, MPST: A Corpus of Movie Plot Synopsis with Tags[41] dataset was used. Out of the present 71 tags, only 47 of them appear in at least 1% of movies. 47 binary features were created based on these 47 tags and a separate binary feature *tags__other* for the rest of tags. A list of used tags is listed in Appendix III.

### Competition

Releases of similar by genre movies on the same date tend to negatively affect box-office gross as it sets movies in direct competition for the audience [45]. Many authors proposed a *competition* feature [8, 13, 22, 46] which would reflect how many movies are released at the same time and compete for viewer's attention. Di et. al. developed this idea and proposed to multiply the competition number with a coefficient of movie similarity by genres[39]. We calculate this feature in the next way:

$$C_i = \sum_{j=0}^{k} \frac{g_i \bigcap g_j}{G} \tag{1}$$

[58] https://pypi.org/project/gensim/1.0.1/

In equation (1), $C_i$ denotes the competition value for movie i, G denotes the total number of genres, g denotes the set of genres of a particular movie, k denotes the number of movies released within a week from movie i.

Mean competition over the dataset is 1.29, the highest is 11.6, 16.7% of movies were released without competition (which means there are no movies released within a week, or they have different genres).

### Engineered new numeric features

As it was mentioned in section 3.3, cast, crew collection, and production company are important nominal features that are not possible to represent in a binary form, therefore they should be converted to a meaningful numerical representation.

The most objective way to assess how successful an actor, a crew member, the company, or the collection is in a particular movie is to calculate the average revenue and average profit (the difference between revenue and budget) from the previous movies they had. Another feature which allows assessing the experience of the crew or cast members is the number of movies they worked on before. Additionally, cast experience in terms of the number of years passed since their first movie to some extent reflects the experience as well.

Knowing a year's average revenue helps to access what mean box office can be gathered in a particular year. Especially it helps while predicting old movies since the budget and revenue amounts are not adjusted for inflation.

These calculations need to be done carefully as we cannot afford target leakage, and we should calculate only using the information available before a movie's release.

Therefore, the next new features were calculated:

- Average profit, average revenue, and number of movies before for each of the 25 crew members if present
- Average profit, average revenue, number of movies before, and experience for each of the 8 cast members if present
- Average profit and average revenue for the collection if present
- Average profit and average revenue for each of the 3 production companies if present
- Average revenue for every year
- Average profit, average revenue, average experience, and the average number of movies before as aggregated features over cast members.

## 3.5 Outliers Removal

Outliers are observations or measures that are suspicious because they are much smaller or much larger than the vast majority of the observations [47]. Some of them may exist because of data entry errors. Data used in this research are mostly obtained from TMDb website, which is managed by volunteer contributors and is not prone to human errors while filling or editing a record. We assume that the great majority of data is correct, and outliers occurred due to data entry error will not affect the results significantly.

The real treat of current research is legitimate outliers. Those are the movies that exist indeed but are so rear, that it is not feasible to teach a model to handle them properly. For example, the movie Paranormal Activity[59] has \$193M box office with an estimated budget of \$15,000. The Adventures of Pluto Nash[60] gathered only \$7M box office with a budget of \$100M. These movies are legitimate outliers, but it would be impossible to predict their box office before release, without knowing the ratings and reviews. Such movies are quite rare, but their presence can impact the actual results significantly. Another example is The Irishman[61], a movie with top cast, crew, production companies and a budget of \$159M, but with only \$1M box office since it was released on Netflix and had an extremely limited release in movie theaters, but we do not have this information in the dataset. Unfortunately, there is no unanimously accepted approach to handle outliers.

### Removing on the basis of revenue and/or budget value

The main challenge while performing the regression task is the target's range. Dataset of 7 thousand samples is extremely insufficient to predict the target with an order of magnitude 9. We are not trying to predict the target for each movie equally well, but rather concentrate on the ability to better predict the target of the majority of movies. With this being said, the resulting models will not be able to deal efficiently with movies which revenue lays around the revenue bounds.

While manually checking movies, the budget of which exceeds the revenue hundreds of times, many misleading and false data was found. We ensured the first hundred of these extreme movies to have a valid revenue value. Additionally, 23 movies with budget or revenue equal to \$1 were found, which is false information.

9 movies with a budget exceeding revenue in more than 1000 times or revenue exceeding budget in more than 1000 times were removed as obvious outliers.

To reduce the target's order of magnitude, it was decided to remove 568 movies with budget and/or revenue being less than \$100000. It included 25 movies with it being less than \$100 and 132 movies with it being less than \$1000.

### Outliers Removal Techniques

The goal was to remove as few outliers as possible while maximally decreasing the mean average percentage error (MAPE) and mean absolute error (MAE). More information on the used metrics can be found in section 4.1. It was set to have a contamination level not higher than 0.05 (to delete not more than 5% of movies).

The next outlier detection techniques were explored:

- Isolation Forest [48] gives the best result with max features 0.8, max samples 6500, 5000 estimators, and contamination 0.016 (removing 126 movies)
- One-Class Support Vector Method [49] gives the best result with auto gamma, *RBF* kernel, and contamination 0.034 (removing 275 movies)
- Local Outlier Factor [50] method gives the best result with 9 neighbors, *ball tree* algo-

[59] https://www.imdb.com/title/tt1179904/

[60] https://www.imdb.com/title/tt0180052/

[61] https://www.imdb.com/title/tt1302006/

rithm, leaf size 30, *minkowski* metric, and contamination 0.021 (removing 161 movies)

- Custom outliers detection technique was created. The next 44 lists were created with movies having:
  - revenue less than $250000 as it's a minority of movies error in predicting which significantly increases MAPE
  - budget less than $250000
  - year less than 1970 as it is a minority of movies which are quite different from the main movies cluster, additional error may be added by inflation
  - profitability lower than the 1st decile or higher than the 9th decile
  - production company's average profit lower than 1st decile or higher than 9th decile
  - 8 lists for each cast member if their average profit is lower than 1st decile or higher than the 9th decile
  - cast average profit lower than the 1st decile or higher than 9th decile
  - 26 lists for each crew member if their average profit is lower than 1st decile or higher than the 9th decile
  - collection average profit lower than the 1st decile or higher than 9th decile
  - year average profit lower than 1st decile or higher than 9th decile

  Where profitability denotes for revenue to budget ratio and profit denotes for revenue to budget difference. The best result was obtained by removing movies that appear in at least 15 of these lists (removing 63 movies).

Out of these 4 outliers detection techniques and all their combinations, the best result was obtained by using only the custom detection technique. It can be explained by the fact that we put attention specifically to the features which directly impact the ability to predict the target, while other techniques try to search outliers through the whole feature space.

It resulted in a dataset of 6965 movies with 228 features.

# 4 Methods

This section describes which methods were used to solve regression, binary classification, and multi-class classification tasks. It also describes how data was transformed to obtain better result and metrics which were used to evaluate the results.

## 4.1 Metrics for result evaluation

Model performance evaluation is a crucial component of any machine learning task. The goal of each metric used in this research is to illustrate how close the predictions are to the actual values. It means that the metrics used in binary classification, multiclass classification, and regression will be different due to the nature of the target feature.

Next metrics will be discussed:

Regression metrics

- Mean absolute error (MAE)

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - x_i| \tag{2}$$

- Root mean squared error (RMSE)

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2} \tag{3}$$

- Mean absolute percentage error (MAPE)

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}|\frac{y_i - x_i}{y_i}| \tag{4}$$

- Weighted absolute percentage error (WAPE)

$$WAPE = 100\%\frac{\sum_{i=1}^{n}|y_i - x_i|}{\sum_{i=1}^{n}y_i} \tag{5}$$

- Symmetric mean absolute percentage error (SMAPE)

$$SMAPE = \frac{100\%}{n}\sum_{i=1}^{n}\frac{|y_i - x_i|}{(|y_i| + |x_i|)/2} \tag{6}$$

- Adjusted R squared $\bar{\mathrm{R}}^2$

$$\bar{R}^2 = 1 - (1 - R^2)\frac{n-1}{n-p-1} = 1 - \frac{\sum_{i=1}^{n}(y_i - x_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}\frac{n-1}{n-p-1}, \tag{7}$$

where n denotes the number of samples and p denotes the number of features

Common classification metrics

- AUC-ROC score
- Number of true positive samples (TP)
- Number of true negative samples (TN)
- Number of false positive samples (FP)
- Number of false negative samples (FN)
- Cohen's kappa (k)

$$k = \frac{TP + TN - ((TP + FN)(TP + FP) + (FP + TN)(FN + TN)}{FP + FN} \tag{8}$$

- Matthews correlation coefficient (MCC)

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{9}$$

- Precision (PPV)

$$PPV = \frac{TP}{TP + FP} \tag{10}$$

- Recall (TPR)

$$TPR = \frac{TP}{TP + FN} \tag{11}$$

- F1 score (F$_1$)

$$F_1 = 2 \cdot \frac{PPV \cdot TPR}{PPV + TPR} \tag{12}$$

- Accuracy (ACC)

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{13}$$

Metrics used specifically in the binary classification task

- Error samples (E)

$$E = FP + FN \tag{14}$$

Metrics used specifically in the multiclass classification task

- One away accuracy (ACC1)

$$\text{ACC1} = \frac{\sum_{i=1}^{n} 1(x_i \in \{y_1 - 1, y_i, y_i + 1\})}{n}; y \in \{0, 1, ..., c\}, x \in \{0, 1, ..., c\} \tag{15}$$

where c denotes the ordinal number of the class

**Regression metrics**

RMSE, MAE, and MAPE are the most used metrics in regression tasks[51]. The popularity of them to some extent is caused by their easy interpretation.

In this research, RMSE and MAE can be directly interpreted as the difference in $ between actual and predicted revenue. However, these metrics, RMSE especially, suffer from outliers and differences in target magnitude. Relatively small absolute error of $10M for a movie with hundreds of millions of revenue will greatly shadow a relatively big error of \$1M for a movie with hundreds of thousands of revenue. While providing a good reference, they do not show the full information on our dataset.

To abstract from absolute numbers, MAPE is used. At a first glance, this metric is easy to interpret, but it has an issue of asymmetric penalizing: negative errors are penalized higher than positive ones, and this imbalance increases with the decay of the true value [52]. For example, a movie with an actual revenue of \$0.5M and a predicted revenue of \$5M will result in 900% MAPE, while swapped revenues (actual \$5M and \$0.5M predicted) will result in 90% MAPE.

WAPE is used to overcome this issue, but it introduces an opposite one: WAPE under-penalizes negative errors because, for example, 1M error predicted for the movie with an actual revenue of \$3M is much more important than the same error for the movie with \$30M actual revenue.

SMAPE overcomes the mentioned issues with MAPE and WAPE, but it is hard to intuitively interpret since it has an upper bound of 200%.

Adjusted R squared metric overcomes the issue of R squared, namely, not penalizing useless features [53] and was mostly used to check whether an added feature was helpful.

To conclude, MAE and MAPE were used as the main metrics. WAPE, SMAPE, and Adjusted R squared were used as general references if the model improves. RMSE was not used directly, but it was used in the early stop condition while the model was being trained in conjunction with Mean squared error (MSE) in the loss function.

### Classification metrics

Classification metrics are more straightforward and less biased due to the limited nature of the target. However, it should be noted that the class distribution is unbalanced and classes should be weighted to be able to predict classes around the threshold 0.5.

At first, attention should be drawn to FP and FN samples. In the binary classification task, 69% of the samples are positive, which means the model will reach its maximum accuracy while predicting more false positive samples than false negative ones as this prediction pattern is easier. When the model is tuned correctly and the classes are weighted, the maximum accuracy and F-score will be reached around a threshold of 0.35-0.4, which corresponds to the target's ratio. The maximum value of AUC-ROC score, MCC, and Cohen's kappa will be correctly reached around a threshold of 0.5, keeping the ratio of FP to FN around 0.7.

The cost of type I error (predicting a flop movie being profitable) is much higher than the cost of type II error (rejecting a profitable movie). The outcome predictions can be made more strict by applying higher weight to the negative class, or simply by increasing the prediction threshold.

Thus being said, the main metric used for the binary classification task is AUC-ROC score. Cohen's kappa and MCC are supporting to check the model's correct behavior. Precision and recall curves along with the threshold's *[0,1]* space also serve to check model's correct behavior meaning that they should not have sudden jumps (except for border values). FP and FN metrics are checked to ensure the prediction ratio's sanity, such that a model doesn't try to simply predict all samples as positive ones. Accuracy and F score metrics should be increasing, but without sacrificing AUC-ROC score.

### Multi-class classification metrics

While it is important to predict the exact revenue range category, a movie producer might be glad to predict within one category on either side [24]. The authors calculated 1-away correct classification rate, which was replicated later on by Zhang et al. [8]. The 1-away prediction approach is used in this research as well to be able to compare the results with published papers.

The rest of the metrics follow the same rules as described above with the difference that not adjustable threshold exists (the prediction classes are the maximums of the softmax output to the model's output).

## 4.2 Preprocessing data

Even after the data are cleaned and features are prepared, the data should be transformed in a certain way to ensure a machine learning model will be able to learn from them in a desired way. It includes filling missing data, changing data scale, and distribution.

### Handling missing data

The used dataset is quite rich in missing data. Many cases of this issue were solved during data preparation. Still, for example, if a movie does not have a specific genre, we can't be sure whether it is indeed a different genre movie or the information simply wasn't filled. Or if an actor has 0 previous movies, we cannot be sure whether it's indeed their first movie, or simply we do not have previous movies in our dataset. This uncertainty increased with the fact that we limited the number of actors we account for in a movie by 8. The reason to take into account only the roles where an actor has a main or at least a secondary role is to avoid noise in the data from cameos, short episodes, and off-camera work [22]. Authors of the mentioned research proposed to make star power calculation even more strict by taking into account only recent movies to avoid the influence of forgotten roles. They could do it because of the decision to calculate star power just in 3 categorical values *[high, medium, low]*, while we cannot do so as it would make our dataset very limited for us to be able to calculate a person's average revenue or average profit.

As was said above, the main source of missing values are features that aggregate the average revenue and average profit from previous movies for a person or a production company. Strictly saying, we cannot be sure whether the values are missing because they were missed in the initial data obtained from TMDB API, or because there indeed were no previous movies to count.

The percentage of missing values varies significantly, from 21% for *cast_1_avg_revenue* to 81% for *crew__sound__sound_designer_avg_revenue*. Another source of missing values is the average profit and average revenue of a movie's collection (91%), which is explained by the fact that most of the movies don't belong to a collection.

There are different ways to handle missing data. We deliberately do not remove records with missing data as we want to keep our dataset as big as possible.

While neural networks cannot work with missed data at all, decision trees in particular cases can work with it. Twala et al. showed that using the approach "missingness incorporated in attributes" (MIA) can have a relatively good performance [54], but in practice, it is close to

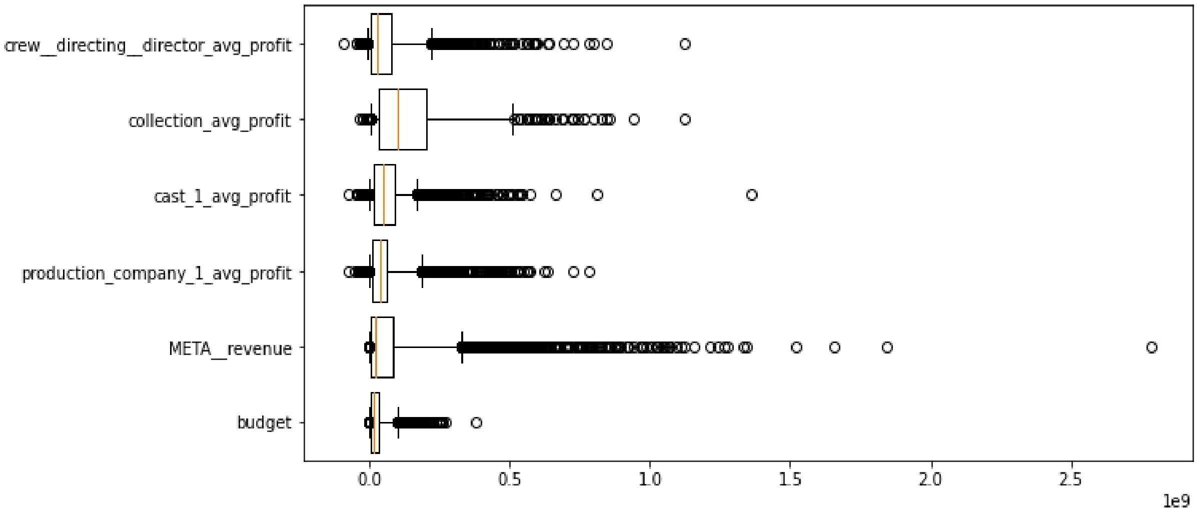


Figure 2: Box plot examples of features connected with budget and revenue
Orange line denotes the mean value, box borders denote the 0.25 and 0.75 percentiles, and ticks denote the 0.05 and 0.95 percentiles.

assigning a well-matched constant value for missing values in a particular feature.

Using single imputation techniques, such as mean substitution, median substitution, or standard deviation is easy and computationally cheap. However, taking into account the high number of features with missing data, high percentages of data being missed, and big range or target feature, using these techniques will make the model highly biased towards the mean values and will result in a poor performance for movies which revenue is far from the mean one.

It was shown that k-Nearest Neighbors (KNN) imputer performs better in filling missing data than Singular Value Decomposition (SVD) [55]. We decided to use this method as it is available in the sklearn library and can be easily included in a data processing pipeline. After rounds of tuning, it was found that the best result is archived with hyperparameters of 30 neighbors and *distance* weights, which means that closer neighbors of a query point will have a greater influence than the neighbors which are further away. Euclidean distance is used with discard of missing values.

Although the needs of our research can be fulfilled with KNN imputer and improving in handling missing data is not a direct goal of this research, we noticed that there are better ways to solve the missing data issue. It was shown that Random Forest (while being a type of nearest neighbor method) is better than KNN imputer for imputing missing values with low to medium data correlation [56]. Neural network methods such as Denoising autoencoder with partial loss (DAPL) perform comparably or better than KNN imputer while having less computational burden [57]. However, we leave this topic for further improvements.

### Data scale and distribution

Data scaling (normalization) is an essential step in data preprocessing as models trained on scaled data usually have significantly higher performance compared to the models trained on unscaled data [58]. However, it depends on the used machine learning models. Decision trees make decisions based on the learned set of rules, which makes them invariant to the monotonic transformation of features [59]. However, normalization is particularly useful for algorithms involving neural networks or distance-based measurements such as nearest-neighbor classification [60], it also helps neural networks converge faster [61].

Data skewness describes the amount of asymmetry compared to the normal distribution. While in normal distribution mean, median, and mode are equal, the mean, median, and mode values for skewed data will be different. It was shown that the higher the absolute value of skewness, the lower the accuracy of a neural network model [62]. Decreasing data skewness can benefit decision trees as well because squashing the input in a more uniform way over the provided space gives more freedom in the choice of split points. Due to data skewness, some models may treat the tail region samples as outliers and result in very poor performance in that region. There are a number of techniques to decrease data skewness.

Budget, revenue, and profit are extremely skewed in our dataset. This creates skewness in all derived features. An example of distributions is shown in Figure 2. Other features derived from the number of movies of a particular person or company, as well as the number of experience years for the cast members are highly skewed as well.

It was found that in our particular case, Yeo-Johnson transformation [63] brings better results than Box-Cox transformation (which is limited to strictly positive inputs) or simpler techniques such as cube root, square root, or logarithm transformation. However, it should be used carefully since it distorts the initial distribution of samples and in some cases may worsen the result. It works best in combination with standardization (over min max scaling and robust scaling). Applying min-max scaling, robust scaling, standardization alone does not help that much as without reducing the data skewness, the range of the most common values still remains comparatively small.

### Data pipelines

After evaluation of different approaches and their combinations, the next data preprocessing pipelines for specific tasks were created:

- Regression
  1. Filling missing input data with KNN Imputer (30 neighbors, weighted distance)
  2. Yeo-Johnson power transformation of the input data with standardisation.
  3. Yeo-Johnson power transformation of the target data
- Classification with tree-based algorithms
  1. Filling missing input data with KNN Imputer (30 neighbors, weighted distance)
- Classification with neural networks
  1. Filling missing input data with KNN Imputer (30 neighbors, weighted distance)
  2. Yeo-Johnson power transformation of the input data with standardisation.

Input data preprocessors, namely, KNNImputer [62] and PowerTransformer [63], were attached to a model by means of the sklearn Pipeline [64]. Target variable preprocessor was attached to a model by means of TransformedTargetRegressor [65].

[62] https://scikit-learn.org/stable/modules/generated/sklearn.impute.KNNImputer

[63] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.PowerTransformer

[64] https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline

[65] https://scikit-learn.org/stable/modules/generated/sklearn.compose.TransformedTargetRegressor

## 4.3 Model selection

Model selection is an important decision to make. Using a model with a limited capacity to learn may prevent proper prediction no matter how well the model is tuned. A number of previous studies showed the superiority of neural networks in comparison with other algorithms, which influenced our decision to use neural networks in our research:

- Binary classification task whether the movie's box office is higher than budget [11]
- Multi-class classification task for predicting movie box office in one out of
  - 5 revenue ranges [19, 20]
  - 9 revenue ranges [16, 24, 38]
- Regression task on predicting IMDb movie rating [27]

We decided to use Tensorflow v2.3.0 implementation of Keras API [66].

Apart from using neural network, we decided to explore one more algorithm to be able to compare performance. We compared MAPE and MAE of 10 fold cross-validation regression task for 25 Sklearn regression algorithms [67], as well as gradient boosting frameworks XGBoost (Extreme Gradient Boosting) [68] and LightGBM (Light Gradient Boosting Machine) [69] running gradient boosted decision trees (GBDT). Default hyperparameters of decision tree based regressors lead the model to extreme overfitting, while the default hyperparameters of support vector based regressors and elastic net lead the model to extreme underfitting (predicting a constant). Therefore, some hyperparameters were changed to prevent obvious overfitting or underfitting, but the models were not tuned extensively, so we cannot state we have seen the best results. However, it was enough to see that among Sklearn models, the best result in terms of MAPE was obtained with ExtraTreesRegressor [70], and the best result in terms of MAE with HistGradientBoostingRegressor [71]. But the boosting models performed better. We decided to keep using LightGBM as it gives almost the same result as XGBoost while taking considerably less time to train. "Random forest and gradient boost are the two algorithms which are giving the best accuracy (compared to SVM, KNN and AdaBoost Classifier)" [36].

LightGBM implements Sklearn interfaces with LGBMRegressor [72] and LGBMClassifier, [73] which allows us to use these models right away in Sklearn's pipelines, cross-validation, k-fold split etc. To be able to use a Keras model with Sklearn interfaces, we use the corresponding wrappers KerasClassifier [74] and KerasRegressor [75].

## 4.4 Model tuning

Choosing a set of optimal hyperparameters for a machine learning model is crucial. We simultaneously want not to limit the model's learning capacity while preventing overfitting.

[66] https://www.tensorflow.org/api_docs/python/tf/keras

[67] https://scikit-learn.org/stable/supervised$_learning$

[68] https://xgboost.readthedocs.io/en/latest/

[69] https://lightgbm.readthedocs.io/en/latest/

[70] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesRegressor

[71] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.HistGradientBoostingRegressor

[72] https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.LGBMRegressor

[73] https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.LGBMClassifier

[74] https://www.tensorflow.org/api_docs/python/tf/keras/wrappers/scikit_learn/KerasClassifier

[75] https://www.tensorflow.org/api_docs/python/tf/keras/wrappers/scikit_learn/KerasRegressor

Together with manual tuning and exploring different hyperparameter combinations, we considered tuning algorithms. The chosen models (neural network and LightGBM) have dozens of hyperparameters, and it would not be feasible to run an exhaustive grid search over them to find the best combination, or to do it manually. It was shown that randomized grid search can achieve a better result with fewer iterations than exhaustive grid search [64].

We tried to use randomized grid search via RandomizedSearchCV [76], but it still was not sufficient, since with 10 folds cross-validation we would need 10 models would need to be trained for each search iteration, and the number of iterations needed to arrive to a potential sweet spot rapidly grows with the increase of the number of hyperparameters.

Therefore, we used 2 more sophisticated hyperparameter optimization algorithms which would perform the selection based on the expected improvement criterion. This deliberate selection helps to find a hyperparameter sweet spot in fewer iterations than randomized search, and even more exhaustive grid search. We used exactly two of them to be able to compare their performance and ensure they result in a similar set of hyperparameters.

Bayesian optimization was proven to be useful in a wide range of machine learning models [65]. However, the performance of a straight-forward implementation rapidly decreases with the increase of the number of parameters needed to be optimized.

This limitation can be overcame with a sequential model-based optimization Tree of Parzen Estimators (TPE) [66] which proved its effectiveness in optimizing even hundreds of parameters [67]. We used it via Hyperopt [77] implementation.

Another technique which allows to overcome the limitation is Sequential Domain Reduction [68]. We used it via Bayesian Optimization Python library [78] implementation.

Both used algorithms attempt to find the maximum (BayesianOptimization) or minimum (Hyperopt) value of an unknown function in as few iterations as possible. The next tasks have corresponding values to be optimized:

- Regression: average SMAPE from 10 folds of cross-validation.
- Binary classification: average AUC-ROC from 10 folds of cross-validation.
- Multi-class classification: average Cross-entropy loss from 10 folds of cross-validation.

During the optimization of neural networks, the next notes were concluded:

- Among tried loss functions (MSE, MAE, MAPE, MSLE, Cosine similarity, Huber loss, Log Cosh) the best result was obtained with MSE. MAE and Huber showed a comparable but still lower results. For classification, the best loss function is Cross-entropy loss (binary or categorical correspondingly to task).
- Batch size is a crucial hyperparameter. Model with big batch size tends to be highly biased towards mean values. Reducing the batch size to 8-32 helps to at least partially grasp the target's tails. Small batch size makes it harder for the model to converge, it has time to learn more before starting to overfit. Downside of a small batch size is a significant increase in training time.
- Among tried optimizers (SGD, RMSProp, Adam, Adamax, Nadam), the best result was obtained with Adamax. Its parameters, namely, learning rate, beta 1, beta 2, will be tuned.

[76] https://scikit-learn.org/stable/modules/generated/sklearn.model$_{s}election.RandomizedSearchCV$

[77] https://github.com/hyperopt/hyperopt

[78] https://github.com/fmfn/BayesianOptimization

Learning rate is an important hyperparameter. Big learning rate led to fast overfitting, while a small learning rate led to the model's inability to predict high values. Adamax' hyperparameters were tuned differently for classification tasks.

- Early stopping of regression is used by monitoring SMAPE. Due to the small batch size, model trains in a few long epochs, and early stopping with patience of 5 epochs shows the best performance while keeping the time to train short. Early stopping of multiclass classification should be done even faster, otherwise the model may quickly overfit by overloading some classes.
- We tried different model architectures: 1-4 hidden layers with 16-2048 neurons in each layer. The best number of hidden layers is 2. Practice shows that having more than 2 layers makes the model to overfit fast and train longer. 1 hidden layer can not explain the dependencies between our big number of features. Found through experiments, the best number of neurons is around the number of features used. Tuning of the binary classifier showed that it is harder for the model to perform well with such a big decrease in neurons (from 256 to 1 of output), and using 3 layers with decreasing number of neurons is more beneficial.
- There is no activation on the output layer (linear activation) for regression. Classification model use sigmoid and softmax. The best activation function found for the regressor's hidden layers is sigmoid. Tuning for classification showed the best result with using different hidden layer activations.
- Dropout layer is essential, especially on the last layer before the output. We tried dropout in the range [0.1-0.9] as well as its absence on each layer.
- We tried to tune *l1* and *l2* regularization, but did not find it useful or failed to tune it properly.
- We did not see a noticeable impact of initialization functions (for both kernel and bias).
- Class weights for binary classification are simply the inverse number of class instances. For multiclass classification it is more complicated, since even with the inverse number of class instances it may concentrate around particular classes (marginal classes mostly). The most balanced model we found has an inverse number of class instances with number of instances multiplied by 1.2, 1.05, 0.93, 0.97 for 1, 3, 4, and 5 classes correspondingly.

During optimization of LightGBM the next notes were concluded:

- There is no magic formula for regularization since a model with low restrictions of maximum depth, number of leaves, etc. will require high *l1* and *l2* regularization, and vice versa.
- The number of estimators should definitely be above 1000. However, too high number of estimators make the model less flexible.
- Number of leaves should be tuned together with the max depth, and usually with the increase of the number of leaves the max depth should be increased as well.
- Learning rate for all tasks is approximately the same with the magnitude of 0.01.
- Bagging or subsampling helps to increase the model's generalization, it acts similarly to batch size in neural networks.
- Early stopping after around 30 rounds on the used loss metric is good.
- Class weights of the number of negative samples / the number of positive samples works well enough.

The best hyperparameters found are shown in Table 6.

Table 6: Tuned hyperparameters used in final evaluation

| | | | **Regression** | **Binary classification** | **Multi-class classification** |
|---|---|---|---|---|---|
| **Neural network** | **loss function** | | MSE | binary crossentropy | categorical crossentropy |
| | **early stopping** | **patience** | 10 | 10 | 3 |
| | | **metric** | validation MSE | validation AUC-ROC | validation categorical accuracy |
| | **batch size** | | 16 | 16 | 4 |
| | **1st hidden layer** | **neurons** | 256 | 1024 | 1024 |
| | | **activation** | sigmoid | tanh | sigmoid |
| | | **kernel initialization** | glorot uniform | glorot uniform | lecun uniform |
| | | **dropout** | 0.1 | 0.5 | 0.5 |
| | **2nd hidden layer** | **neurons** | 256 | 512 | 256 |
| | | **activation** | sigmoid | relu | elu |
| | | **kernel initialization** | glorot uniform | glorot uniform | lecun uniform |
| | | **dropout** | 0.5 | 0.5 | 0.75 |
| | **3nd hidden layer** | **neurons** | NA | 192 | NA |
| | | **activation** | | relu | |
| | | **kernel initialization** | | glorot uniform | |
| | | **dropout** | | 0.5 | |
| | **output layer** | **neurons** | 1 | 1 | 9 |
| | | **activation** | linear | sigmoid | softmax |
| | | **kernel initialization** | glorot uniform | glorot uniform | lecun uniform |
| | **Adamax optimizer** | **learning rate** | 0.001 | 0.001 | 0.007 |
| | | **beta 1** | 0.958 | 0.9 | 0.88 |
| | | **beta 2** | 0.987 | 0.999 | 1.0 |
| **LightGBM** | **tree learner** | | data | serial | serial |
| | **early stopping** | **patience** | 30 | 30 | 30 |
| | | **metric** | validation MSE | validation AUC-ROC | validation categorical crossentropy |
| | **loss function** | | MSE | AUC-ROC | categorical crossentropy |
| | **regularization** | **l1** | 0.617 | NA | 0.0647 |
| | | **l2** | 0.435 | NA | 11.26 |
| | | **min data in leaf** | 20 | 70 | 13 |
| | | **min sum hessian in leaf** | 47 | 25.78 | 9.965 |
| | **learning rate** | | 0.018 | 0.0367 | 0.02 |
| | **num leaves** | | 98 | 88 | 178 |
| | **max depth** | | 52 | 79 | 74 |
| | **bagging freq** | | 3 | 3 | 1 |
| | **bagging fraction** | | 0.8 | 0.932 | 0.9 |
| | **n estimators** | | 7149 | 8100 | 8000 |
| | **boos from average** | | NA | NA | true |

# 5 Results

## 5.1 Models results

The results obtained from models using 10 folds cross-validation with keeping 5% of the data for validation on early stopping condition are shown in the Table 7.

Table 7: Models prediction results

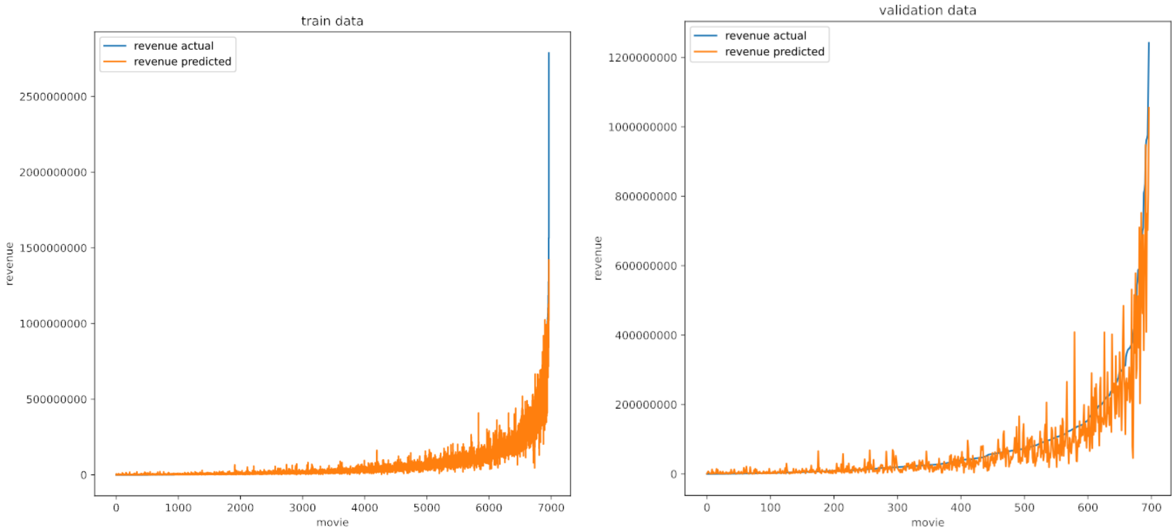
| | | **Neural network** | **LightGBM** |
|---|---|---|---|
| **Regression** | SMAPE | 65.868 | 54.821 |
| | WAPE | 42.625 | 32.411 |
| | MAPE | 192.5 | 149.725 |
| | MAE | 32162029 | 24457248 |
| | RMSE | 73068313 | 59561111 |
| | Adjusted R^2 | 0.601 | 0.818 |
| **Binary classification** | accuracy | 0.761 | 0.815 |
| | F1 | 0.820 | 0.865 |
| | AUC-ROC | 0.750 | 0.791 |
| | Cohen's kappa | 0.470 | 0.571 |
| | MCC | 0.477 | 0.572 |
| | precision | 0.865 | 0.878 |
| | recall | 0.779 | 0.853 |
| | TN | 153.9 | 154.4 |
| | FP | 59.1 | 57.6 |
| | FN | 107.1 | 71 |
| | TP | 377.4 | 413.5 |
| | error samples | 166.2 | 128.6 |
| **Multi-class classification** | 1-away accuracy | 0.757 | 0.804 |
| | accuracy | 0.363 | 0.433 |
| | Cohen's kappa | 0.283 | 0.359 |
| | MCC | 0.291 | 0.359 |
| | F1 | 0.333 | 0.429 |
| | precision | 0.378 | 0.427 |
| | recall | 0.362 | 0.433 |
| | AUC-ROC | 0.833 | 0.861 |

All tasks were solved better by LightGBM with significant superiority. As it is partially visible from Figure 3 and quite visible from Figure 4, our LightGBMRegressor model fails to predict both tails of the revenue range, while providing a relatively good predictions in between $10M to $100M.

Prediction, for example, a target within one order of magnitude gives more interpretable metrics. Training the model and predicting revenue from a subset of movies from $10M to $100M of revenue gives 41% of MAPE and $13.4M of MAE. While the model undoubtedly learns features and moves in the correct direction, the result is still far from using it in the real world. This can be solved by adding new features which proved their help in other studies, such as word-of-mouth metrics, activity on social media, and other things which can be tracked before release.

Figure 3: Example of a CV fold results, absolute value

Predicted revenue (orange) over sorted actual one (blue)

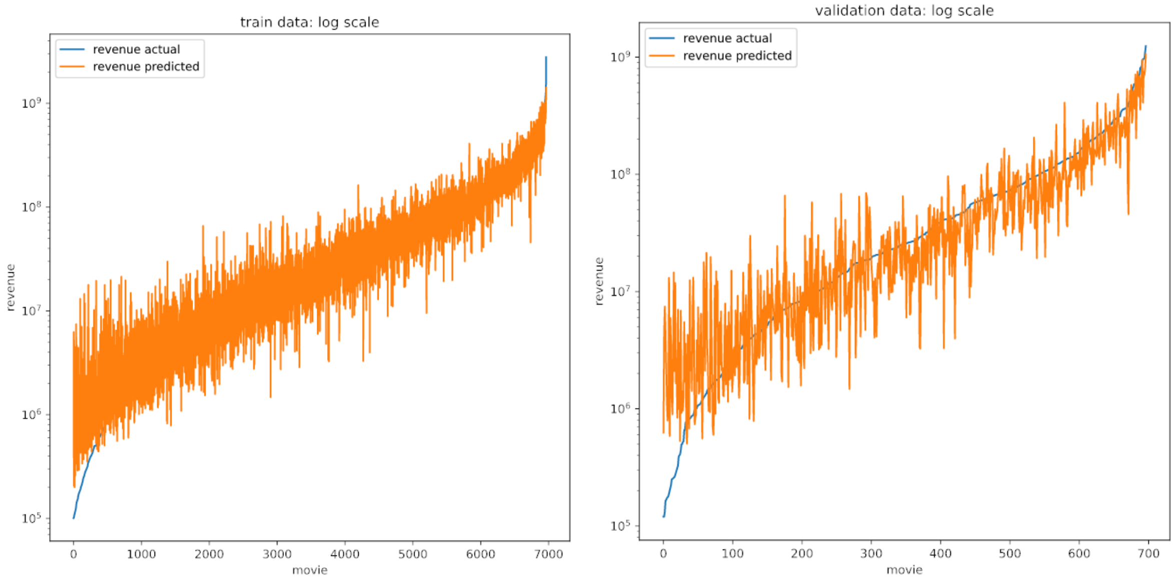


(a) Predicting train data

(b) Predicting test data

Figure 4: Example of a CV fold results, log scaled value

Predicted revenue (orange) over sorted actual one (blue)

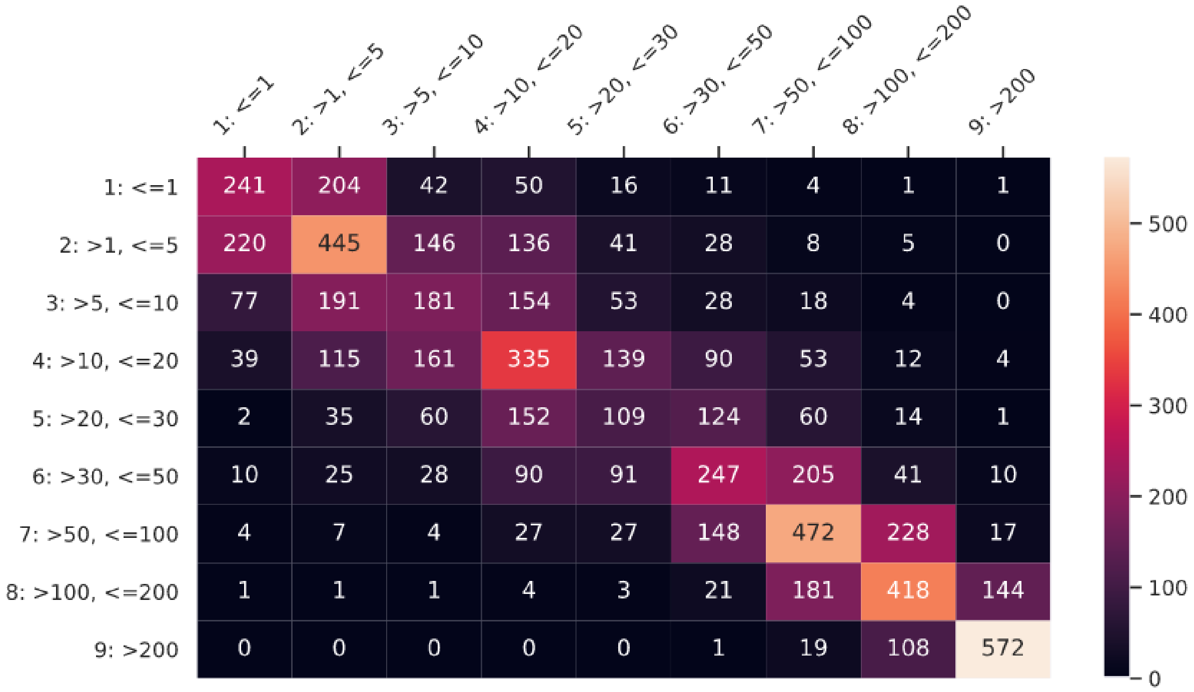


(a) Predicting train data

(b) Predicting test data

Figure 5: Confusion matrix of LightGBM Classifier

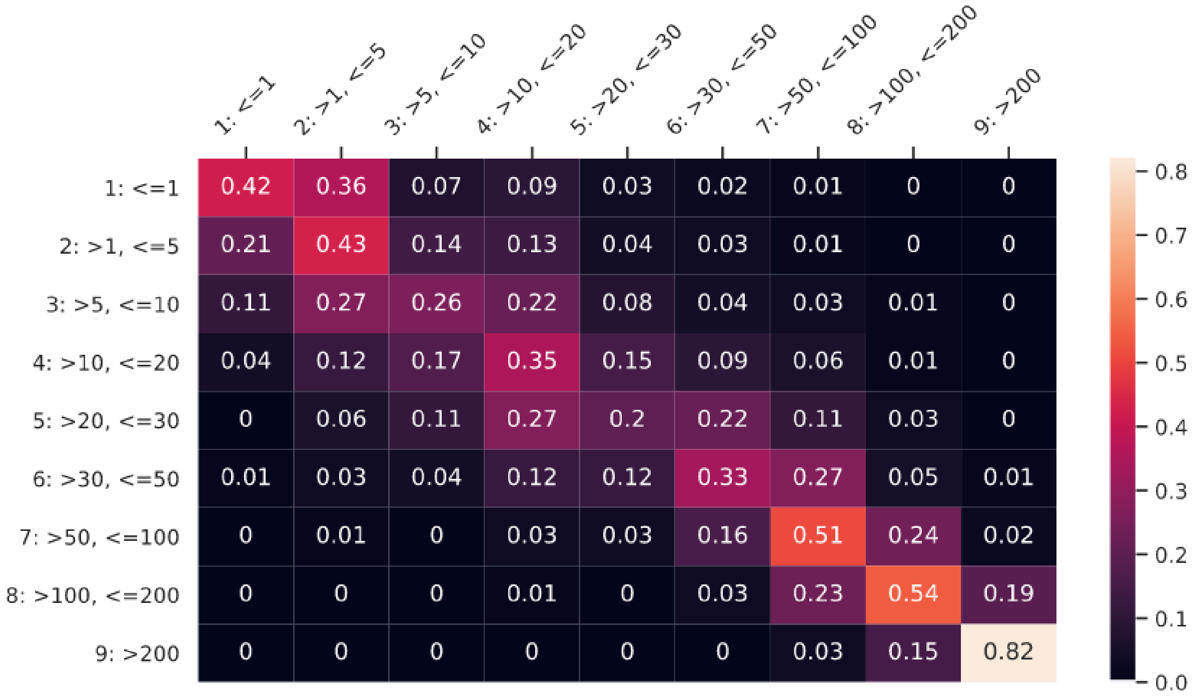
The results of 10 folds cross-validation



|  | 1: <=1 | 2: >1, <=5 | 3: >5, <=10 | 4: >10, <=20 | 5: >20, <=30 | 6: >30, <=50 | 7: >50, <=100 | 8: >100, <=200 | 9: >200 |
|---|---|---|---|---|---|---|---|---|---|
| 1: <=1 | 241 | 204 | 42 | 50 | 16 | 11 | 4 | 1 | 1 |
| 2: >1, <=5 | 220 | 445 | 146 | 136 | 41 | 28 | 8 | 5 | 0 |
| 3: >5, <=10 | 77 | 191 | 181 | 154 | 53 | 28 | 18 | 4 | 0 |
| 4: >10, <=20 | 39 | 115 | 161 | 335 | 139 | 90 | 53 | 12 | 4 |
| 5: >20, <=30 | 2 | 35 | 60 | 152 | 109 | 124 | 60 | 14 | 1 |
| 6: >30, <=50 | 10 | 25 | 28 | 90 | 91 | 247 | 205 | 41 | 10 |
| 7: >50, <=100 | 4 | 7 | 4 | 27 | 27 | 148 | 472 | 228 | 17 |
| 8: >100, <=200 | 1 | 1 | 1 | 4 | 3 | 21 | 181 | 418 | 144 |
| 9: >200 | 0 | 0 | 0 | 0 | 0 | 1 | 19 | 108 | 572 |

Figure 6: Normalized confusion matrix of LightGBM Classifier

The results of 10 folds cross-validation



|  | 1: <=1 | 2: >1, <=5 | 3: >5, <=10 | 4: >10, <=20 | 5: >20, <=30 | 6: >30, <=50 | 7: >50, <=100 | 8: >100, <=200 | 9: >200 |
|---|---|---|---|---|---|---|---|---|---|
| 1: <=1 | 0.42 | 0.36 | 0.07 | 0.09 | 0.03 | 0.02 | 0.01 | 0 | 0 |
| 2: >1, <=5 | 0.21 | 0.43 | 0.14 | 0.13 | 0.04 | 0.03 | 0.01 | 0 | 0 |
| 3: >5, <=10 | 0.11 | 0.27 | 0.26 | 0.22 | 0.08 | 0.04 | 0.03 | 0.01 | 0 |
| 4: >10, <=20 | 0.04 | 0.12 | 0.17 | 0.35 | 0.15 | 0.09 | 0.06 | 0.01 | 0 |
| 5: >20, <=30 | 0 | 0.06 | 0.11 | 0.27 | 0.2 | 0.22 | 0.11 | 0.03 | 0 |
| 6: >30, <=50 | 0.01 | 0.03 | 0.04 | 0.12 | 0.12 | 0.33 | 0.27 | 0.05 | 0.01 |
| 7: >50, <=100 | 0 | 0.01 | 0 | 0.03 | 0.03 | 0.16 | 0.51 | 0.24 | 0.02 |
| 8: >100, <=200 | 0 | 0 | 0 | 0.01 | 0 | 0.03 | 0.23 | 0.54 | 0.19 |
| 9: >200 | 0 | 0 | 0 | 0 | 0 | 0 | 0.03 | 0.15 | 0.82 |

Figure 5 and Figure 6 show how good is the classification's result. However, a minor imbalance still exists. We clearly see that the best accuracy is reached on class 9, which is not surprising since it is the largest by absolute value range. And the least accuracy was gained by the smallest class 5. Overall, the diagonal line of the confusion matrix is clearly distinguishable.

If we compare the conducted study with Masrury et al. who reached for binary classification 80% accuracy and 86% F score[11], it is noticeable that their dataset is much smaller (1000 selected movies out of 150 top grossed US movies during 2008-2017 years) and is highly unbalanced.

Our multiclass classification results are quite high, they outperform the ones obtained by Parimi et al. (2013),[21] and Sharda et al. (2006)[24]. However, they are still worse than the ones shown by Sharda et al. (2010),[25] and Ghiassi et al. (2015)[22].

Binary classification outperforms the mentioned studies of Masrury et al., [11], Burgos et al., [12], Lash et al., [14]. At the same time, it performed worse than the models of Rhee et al., [13] and Galvao et al. [16]. This can be explained by the fact that both superior models use post-release features and are tested on much smaller datasets.

He et al. proposed to use an ensemble of models trained on different data [26]. They split the data into 3 groups (split by number of screens) and trained 3 corresponding models. Number of screens is a train feature and is known ahead, so they predict a movie from the model which corresponds to movie's number of screens.

Delen et al. showed the benefit of using models ensembles [25]. They used average prediction from NN, SVM, RF, BT and CART models. Lee et al. used voting for AdaBoost, GTB, Linear discriminant, LR, NN, RF, SVC models with a privilege to the best performing model GTB [37].

Apart from trying these approaches, we want to propose future researches to make an ensemble of models based on the target feature. Models would be trained on the corresponding subsets of data, then test data would be predicted with each model, and the best result would be taken.

## 5.2 Data Preparation

To the best of our knowledge, the dataset of 6965 movies created in this study is the biggest ever used in box office gross prediction. However, at the same time, the high magnitude range of the box office feature limits the model's performance and distorts the metrics.

A big set of new features was develop, which resulted in 228 input feature spaces.

Next steps are proposed for future improvements:

- Granular feature selection
- More sophisticated missing data imputation
- Acquire a bigger dataset, even a proprietary
- Acquire number of theaters or screens
- Acquire marketing expenditure
- More substantial plot analysis
- Involve star ratings from IMDb and Rotten Tomatoes as one of the criteria to assess a cast member

# 6 Conclusion

We successfully went through several stages typical for data mining and machine learning to obtain possibly the biggest and feature-rich dataset used in box office gross prediction. Engineered feature have proven they improvement to model's score.

We used neural networks and gradient boosting in the following tasks:

- Regression
  The regression result is satisfying enough from the machine learning point of view, but it is very far from being used in the real world. Nevertheless, considering the absence of similar regression studies in the domain, the current research may potentially become a benchmark and give a fresh start for new studies. LightGBM model showed better results than neural network.
- Classification
  Our obtained results in both binary classification and multiclass classification are very competitive. LightGBM performed better than neural network here as well.

Data driven models such as decision tree ensembles or neural networks require acquiring a large amount of data to show their true power. We tried to fulfill this need and gather the biggest to our knowledge dataset which was used to predict box office gross. The limitation of the budget and revenue data which we acquired is that it does not give us information on how exactly this budget was spent, what marketing expenditures were, for example, and does not tell what other income streams apart of the box office were involved. The weak side of the used data-driven models is that they act as a black-box method which does not give us the opportunity to determine causal relationships as, for example, Linear Regressors or Decision Trees would do. We may follow a feature's contribution to the common result, but we still will be far from understanding how it impacts the prediction from the model's point of view.

# References

[1] A. Tesser, K. Millar, and C.-H. Wu, “On the perceived functions of movies,” *The Journal of Psychology*, vol. 122, no. 5, pp. 441–449, 1988.

[2] A. Charlesworth and S. A. Glantz, “Smoking in the movies increases adolescent smoking: A review,” *Pediatrics*, vol. 116, no. 6, pp. 1516–1528, 2005.

[3] M. S. Hassan, B. Hassan, M. N. Osman, and Z. Sabaghpour Azarian, “Effects of watching violence movies on the attitudes concerning aggression among middle schoolboys (13-17 years old) at international schools in kuala . . . ,” *European Journal of Scientific Research ISSN*, vol. 38, pp. 1450–216, 12 2009.

[4] M. Gould and D. Shaffer, “The impact of suicide in television movies,” in *Suicide and Its Prevention*, p. 331, EJ Brill, Leiden, 1989.

[5] V. R. O’Regan, “The celebrity influence: do people really care what they think?,” *Celebrity Studies*, vol. 5, no. 4, pp. 469–483, 2014.

[6] G. J. J. Young, S. Mark and W. Van der Stede, “The business of making money with movies. strategic finance,” *Strategic Finance*, pp. 35–40, 2010.

[7] A. D. Vany, “Chapter 19 the movies,” vol. 1 of *Handbook of the Economics of Art and Culture*, pp. 615–665, Elsevier, 2006.

[8] L. Zhang, J. Luo, and S. Yang, “Forecasting box office revenue of movies with bp neural network,” *Expert Systems with Applications*, vol. 36, no. 3, Part 2, pp. 6580 – 6587, 2009.

[9] A. Tadimari, N. Kumar, T. Guha, and S. S. Narayanan, “Opening big in box office? trailer content can help,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2777–2781, Mar. 2016.

[10] V. Biramane, H. Kulkarni, A. Bhave, and P. Kosamkar, “Relationships between classical factors, social factors and box office collections,” pp. 35–39, 01 2016.

[11] R. Masrury, M. Saputra, A. Alamsyah, and M. Primantari, “A comparative study of hollywood movie successfulness prediction model,” 2019.

[12] M. Burgos, M. Campanario, J. Lara, and D. Lizcano, “Using decision trees to characterize and predict movie profitability on the us market,” vol. 1, pp. 274–279, 2015.

[13] T. Rhee and F. Zulkernine, “Predicting movie box office profitability: A neural network approach,” pp. 665–670, 12 2016.

[14] M. Lash and K. Zhao, “Early predictions of movie success: The who, what, and when of profitability,” *Journal of Management Information Systems*, vol. 33, 06 2015.

[15] J. Ericson and J. Grodman, “A predictor for movie success,”

[16] M. Galvao and R. Henriques, “Forecasting model of a movie’s profitability,” in *2018 13th Iberian Conference on Information Systems and Technologies (CISTI)*, pp. 1–6, 2018.

[17] D. C. dcocuzzo and S. Wu, “Hit or flop : Box o ffi ce prediction for feature films december,” 2013.

[18] Y. Zhou, L. Zhang, and Z. Yi, “Predicting movie box-office revenues using deep neural networks,” *Neural Computing and Applications*, vol. 31, no. 6, pp. 1855–1865, 2019.

[19] N. Quader, M. Gani, D. Chaki, and M. Ali, “A machine learning approach to predict movie box-office success,” vol. 2018-January, pp. 1–7, 2018.

[20] N. Quader, M. Gani, and D. Chaki, “Performance evaluation of seven machine learning classification techniques for movie box office success prediction,” vol. 2018-January, pp. 1–6, 2018.

[21] R. Parimi and D. Caragea, “Pre-release box-office success prediction for motion pictures,” pp. 571–585, 07 2013.

[22] M. Ghiassi, D. Lio, and B. Moon, “Pre-production forecasting of movie revenues with a dynamic artificial neural network,” *Expert Systems with Applications*, vol. 42, no. 6, pp. 3176–3193, 2015.

[23] R. Parimi and D. Caragea, “Pre-release box-office success prediction for motion pictures,” pp. 571–585, 07 2013.

[24] R. Sharda and D. Delen, “Predicting box-office success of motion pictures with neural networks,” *Expert Systems with Applications*, vol. 30, no. 2, pp. 243 – 254, 2006.

[25] D. Delen and R. Sharda, “Predicting the financial success of hollywood movies using an information fusion approach,” *Industrial Engineering Journal (a Turkish/English Journal for Industrial Engineers)*, vol. 21, pp. 30–37, 01 2010.

[26] G. He and S. Lee, “Multi-model or single model? a study of movie box-office revenue prediction,” in *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, pp. 321–325, 2015.

[27] P.-Y. Hsu, Y.-H. Shen, and X.-A. Xie, “Predicting movies user ratings with imdb attributes,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8818, pp. 444–453, 2014.

[28] T. Kim, J. Hong, and P. Kang, “Box office forecasting using machine learning algorithms based on sns data,” *International Journal of Forecasting*, vol. 31, no. 2, pp. 364–390, 2015.

[29] N. Hossein and D. Miller, “Predicting motion picture box office performance using temporal tweet patterns,” *International Journal of Intelligent Computing and Cybernetics*, vol. 11, no. 1, pp. 64–80, 2018.

[30] Y. Zhou and G. G. Yen, “Evolving deep neural networks for movie box-office revenues prediction,” in *2018 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–8, July 2018.

[31] T. G. Rhee and F. Zulkernine, “Predicting movie box office profitability: A neural network approach,” in *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 665–670, Dec. 2016.

[32] Z. Di, J. Xiu, J. Lin, and Y. Qian, “Research on movie-box prediction model and algorithm based on neural network,” pp. 224–228, 2016.

[33] M. Mestyán, T. Yasseri, and J. Kertész, “Early prediction of movie box office success based on wikipedia activity big data,” *PloS one*, vol. 8, p. e71226, 08 2013.

[34] Y. Ru, B. Li, J. Liu, and J. Chai, “An effective daily box office prediction model based on deep neural networks,” *Cognitive Systems Research*, vol. 52, pp. 182 – 191, 2018.

[35] K. Meenakshi, G. Maragatham, N. Agarwal, and I. Ghosh, “A data mining technique for analyzing and predicting the success of movie,” vol. 1000, 2018.

[36] R. Dhir and A. Raj, “Movie success prediction using machine learning algorithms and their comparison,” in *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*, pp. 385–390, 2018.

[37] K. Lee, J. Park, I. Kim, and Y. Choi, “Predicting movie success with machine learning techniques: ways to improve accuracy,” *Information Systems Frontiers*, vol. 20, 08 2016.

[38] A. Kanitkar, “Bollywood movie success prediction using machine learning algorithms,” in *2018 3rd International Conference on Circuits, Control, Communication and Computing (I4C)*, pp. 1–4, 2018.

[39] Z. Di, J. Xiu, J. Lin, and Y. Qian, “Research on movie-box prediction model and algorithm based on neural network,” in *2016 4th International Conference on Cloud Computing and Intelligence Systems (CCIS)*, pp. 224–228, 2016.

[40] M. Galvao and R. Henriques, “Forecasting model of a movie’s profitability,” vol. 2018-June, pp. 1–6, 2018.

[41] S. Kar, S. Maharjan, A. P. López-Monroy, and T. Solorio, “MPST: A corpus of movie plot synopses with tags,” in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)* (N. C. C. chair), K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, and T. Tokunaga, eds.), (Paris, France), European Language Resources Association (ELRA), May 2018.

[42] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python*, pp. 274–279. O’reilly, 2009.

[43] M. Omar, B.-W. On, I. Lee, and G. S. Choi, “Lda topics: Representation and evaluation,” *Journal of Information Science*, vol. 41, no. 5, pp. 662–675, 2015.

[44] D. Bamman, B. T. O’Connor, and N. A. Smith, “Learning latent personas of film characters,” in *ACL*, 2013.

[45] F. Gutierrez-Navratil, V. Blanco, L. Orea, and J. Prieto-Rodriguez, “How do your rivals’ releasing dates affect your box office?,” *J Cult Econ*, vol. 38, pp. 1–14, 02 2012.

[46] R. Parimi and D. Caragea, “Pre-release box-office success prediction for motion pictures,” pp. 571–585, 07 2013.

[47] D. Cousineau and S. Chartier, “Outliers detection and treatment: A review,” *International Journal of Psychological Research*, vol. 3, 01 2010.

[48] F. T. Liu, K. Ting, and Z.-H. Zhou, “Isolation forest,” pp. 413 – 422, 01 2009.

[49] A. Bounsiar and M. G. Madden, "One-class support vector machines revisited," in *2014 International Conference on Information Science Applications (ICISA)*, pp. 1–4, 2014.

[50] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: Identifying density-based local outliers," vol. 29, p. 93–104, May 2000.

[51] A. Botchkarev, "Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology," *ArXiv*, vol. abs/1809.03006, 2018.

[52] S. Makridakis, "Accuracy measures: theoretical and practical concerns," *International Journal of Forecasting*, vol. 9, no. 4, pp. 527 – 529, 1993.

[53] J. Karch, "Improving on adjusted r-squared," Sep 2019.

[54] B. Twala, M. Jones, and D. Hand, "Good methods for coping with missing data in decision trees," *Pattern Recognition Letters*, vol. 29, pp. 950–956, 05 2008.

[55] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman, "Missing value estimation methods for DNA microarrays ," *Bioinformatics*, vol. 17, pp. 520–525, 06 2001.

[56] F. Tang and H. Ishwaran, "Random forest missing data algorithms," 2017.

[57] Y. L. Qiu, H. Zheng, and O. Gevaert, "A deep learning framework for imputing missing values in genomic data," *bioRxiv*, 2018.

[58] X. H. Cao, I. Stojkovic, and Z. Obradovic, "A robust data scaling algorithm to improve classification accuracies in biomedical data," *BMC bioinformatics*, vol. 17, p. 359, 09 2016.

[59] W.-Y. Loh, "Classification and regression trees," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, pp. 14 – 23, 01 2011.

[60] K. Muralidharan, "A note on transformation, standardization and normalization," 02 2010.

[61] J. Sola and J. Sevilla, "Importance of input data normalization for the application of neural networks to complex industrial problems," *Nuclear Science, IEEE Transactions on*, vol. 44, pp. 1464 – 1468, 07 1997.

[62] R.-S. Guh, "Effects of non-normality on artificial neural network based control chart pattern recognizer," *Journal of the Chinese Institute of Industrial Engineers*, vol. 19, no. 6, pp. 13–22, 2002.

[63] I.-K. Yeo and R. A. Johnson, "A new family of power transformations to improve normality or symmetry," *Biometrika*, vol. 87, no. 4, pp. 954–959, 2000.

[64] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *The Journal of Machine Learning Research*, vol. 13, pp. 281–305, 03 2012.

[65] J. Wu, X.-Y. Chen, H. Zhang, L.-D. Xiong, H. Lei, and S.-H. Deng, "Hyperparameter optimization for machine learning models based on bayesian optimizationb," *Journal of Electronic Science and Technology*, vol. 17, no. 1, pp. 26 – 40, 2019.

[66] J. Bergstra, R. Bardenet, B. Kégl, and Y. Bengio, "Algorithms for hyper-parameter optimization," 12 2011.

[67] J. Bergstra, D. Yamins, and D. D. Cox, “Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures,” in *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML’13, p. I–115–I–123, JMLR.org, 2013.

[68] N. Stander and K. Craig, “On the robustness of a simple domain reduction scheme for simulation-based optimization,” *International Journal for Computer-Aided Engineering and Software (Eng. Comput.)*, vol. 19, 06 2002.

# Appendix

## I License

### Non-exclusive licence to reproduce thesis and make thesis public

I, **Stanislav Bondarenko**,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to

   reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   **Prediction of a movie's box office using pre-release data**,

   supervised by Rajesh Sharma.

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Stanislav Bondarenko
***12/11/2020***

## II TMDb API Columns

1. adult
2. backdrop_path
3. belongs_to_collection
4. budget
5. cast
6. crew
7. genres
8. homepage
9. id
10. imdb_id
11. keywords
12. original_language
13. original_title
14. overview
15. popularity
16. poster_path
17. production_companies
18. production_countries
19. release_date
20. results (information on releases in different countries were used to access age ratings)
21. revenue
22. runtime
23. spoken_languages
24. status
25. status_code
26. status_message
27. tagline
28. title
29. video
30. vote_average
31. vote_count

## III Final dataset features list

Total of 228 features: 70 features are independent, 126 features generated for each dataset separately basing on movies before (values change depending on movies present in dataset).

**Features which values don’t change depending on which movies are present in dataset**

- General features (19 features)
    1. budget
    2. runtime
    3. spoken_languages
    4. weekend
    5. day_sin
    6. day_cos
    7. month_sin
    8. month_cos
    9. competition
    10. cast_1_gender
    11. cast_2_gender
    12. cast_3_gender
    13. cast_4_gender
    14. cast_5_gender
    15. cast_6_gender
    16. cast_7_gender
    17. cast_8_gender
    18. homepage_exists
    19. homepage_repeats
- Genres (17 genres)
    1. genre__war
    2. genre__western
    3. genre__mystery
    4. genre__music
    5. genre__crime
    6. genre__romance
    7. genre__action
    8. genre__adventure
    9. genre__thriller
    10. genre__animation
    11. genre__family
    12. genre__drama
    13. genre__comedy
    14. genre__documentary
    15. genre__history
    16. genre__fantasy
    17. genre__horror
- Countries (12 countries + 1 other)
    1. country__es
    2. country__jp

    3. country__us
    4. country__ca
    5. country__de
    6. country__cn
    7. country__in
    8. country__fr
    9. country__ru
    10. country__it
    11. country__au
    12. country__gb
    13. country__other
- Rating (5 ratings)
    1. rating__g
    2. rating__pg
    3. rating__pg-13
    4. rating__r
    5. rating__nc-17
- Tags (48 tags)
    1. tag__murder
    2. tag__violence
    3. tag__flashback
    4. tag__romantic
    5. tag__cult
    6. tag__revenge
    7. tag__psychedelic
    8. tag__comedy
    9. tag__suspenseful
    10. tag__good_versus_evil
    11. tag__humor
    12. tag__satire
    13. tag__entertaining
    14. tag__neo_noir
    15. tag__action
    16. tag__sadist
    17. tag__insanity
    18. tag__tragedy
    19. tag__fantasy
    20. tag__paranormal
    21. tag__boring
    22. tag__mystery
    23. tag__horror
    24. tag__melodrama
    25. tag__cruelty
    26. tag__gothic
    27. tag__dramatic
    28. tag__dark
    29. tag__atmospheric
    30. tag__storytelling
    31. tag__sci_fi

32. tag__psychological
33. tag__historical
34. tag__absurd
35. tag__prank
36. tag__sentimental
37. tag__philosophical
38. tag__bleak
39. tag__alternate_reality
40. tag__depressing
41. tag__plot_twist
42. tag__realism
43. tag__cute
44. tag__stupid
45. tag__home_movie
46. tag__thought_provoking
47. tag__inspiring
48. tag__other

**Features which values change depending on which movies are present in dataset**

- Crew (3 features for each of 26 crew members)
    1. crew__production__producer_1_avg_profit
    2. crew__production__producer_1_avg_revenue
    3. crew__production__producer_1_movies_before
    4. crew__production__producer_2_avg_profit
    5. crew__production__producer_2_avg_revenue
    6. crew__production__producer_2_movies_before
    7. crew__sound__music_editor_avg_profit
    8. crew__sound__music_editor_avg_revenue
    9. crew__sound__music_editor_movies_before
    10. crew__sound__original_music_composer_avg_profit
    11. crew__sound__original_music_composer_avg_revenue
    12. crew__sound__original_music_composer_movies_before
    13. crew__sound__sound_designer_avg_profit
    14. crew__sound__sound_designer_avg_revenue
    15. crew__sound__sound_designer_movies_before
    16. crew__sound__sound_effects_editor_avg_profit
    17. crew__sound__sound_effects_editor_avg_revenue
    18. crew__sound__sound_effects_editor_movies_before
    19. crew__sound__sound_re_recording_mixer_avg_profit
    20. crew__sound__sound_re_recording_mixer_avg_revenue
    21. crew__sound__sound_re_recording_mixer_movies_before
    22. crew__sound__supervising_sound_editor_avg_profit
    23. crew__sound__supervising_sound_editor_avg_revenue
    24. crew__sound__supervising_sound_editor_movies_before
    25. crew__directing__director_avg_profit
    26. crew__directing__director_avg_revenue
    27. crew__directing__director_movies_before

28. crew__directing__script_supervisor_avg_profit
29. crew__directing__script_supervisor_avg_revenue
30. crew__directing__script_supervisor_movies_before
31. crew__production__casting_avg_profit
32. crew__production__casting_avg_revenue
33. crew__production__casting_movies_before
34. crew__production__executive_producer_avg_profit
35. crew__production__executive_producer_avg_revenue
36. crew__production__executive_producer_movies_before
37. crew__editing__editor_avg_profit
38. crew__editing__editor_avg_revenue
39. crew__editing__editor_movies_before
40. crew__costume__costume_designer_avg_profit
41. crew__costume__costume_designer_avg_revenue
42. crew__costume__costume_designer_movies_before
43. crew__costume__costume_supervisor_avg_profit
44. crew__costume__costume_supervisor_avg_revenue
45. crew__costume__costume_supervisor_movies_before
46. crew__costume__makeup_artist_avg_profit
47. crew__costume__makeup_artist_avg_revenue
48. crew__costume__makeup_artist_movies_before
49. crew__crew__stunt_coordinator_avg_profit
50. crew__crew__stunt_coordinator_avg_revenue
51. crew__crew__stunt_coordinator_movies_before
52. crew__writing__screenplay_avg_profit
53. crew__writing__screenplay_avg_revenue
54. crew__writing__screenplay_movies_before
55. crew__art__art_direction_avg_profit
56. crew__art__art_direction_avg_revenue
57. crew__art__art_direction_movies_before
58. crew__art__production_design_avg_profit
59. crew__art__production_design_avg_revenue
60. crew__art__production_design_movies_before
61. crew__art__property_master_avg_profit
62. crew__art__property_master_avg_revenue
63. crew__art__property_master_movies_before
64. crew__art__set_decoration_avg_profit
65. crew__art__set_decoration_avg_revenue
66. crew__art__set_decoration_movies_before
67. crew__visualeffects__visual_effects_supervisor_avg_profit
68. crew__visualeffects__visual_effects_supervisor_avg_revenue
69. crew__visualeffects__visual_effects_supervisor_movies_before
70. crew__camera__director_of_photography_avg_profit
71. crew__camera__director_of_photography_avg_revenue
72. crew__camera__director_of_photography_movies_before
73. crew__camera__steadicam_operator_avg_profit
74. crew__camera__steadicam_operator_avg_revenue
75. crew__camera__steadicam_operator_movies_before
76. crew__camera__still_photographer_avg_profit

77. crew__camera__still_photographer_avg_revenue
78. crew__camera__still_photographer_movies_before

- Production company (3 features for each of 2 companies)
    1. production_company_1_avg_profit
    2. production_company_1_avg_revenue
    3. production_company_1_movies_before
    4. production_company_2_avg_profit
    5. production_company_2_avg_revenue
    6. production_company_2_movies_before
    7. production_company_3_avg_profit
    8. production_company_3_avg_revenue
    9. production_company_3_movies_before
- Cast (4 features for each of 8 cast members)
    1. cast_1_avg_profit
    2. cast_1_avg_revenue
    3. cast_1_experience
    4. cast_1_movies_before
    5. cast_2_avg_profit
    6. cast_2_avg_revenue
    7. cast_2_experience
    8. cast_2_movies_before
    9. cast_3_avg_profit
    10. cast_3_avg_revenue
    11. cast_3_experience
    12. cast_3_movies_before
    13. cast_4_avg_profit
    14. cast_4_avg_revenue
    15. cast_4_experience
    16. cast_4_movies_before
    17. cast_5_avg_profit
    18. cast_5_avg_revenue
    19. cast_5_experience
    20. cast_5_movies_before
    21. cast_6_avg_profit
    22. cast_6_avg_revenue
    23. cast_6_experience
    24. cast_6_movies_before
    25. cast_7_avg_profit
    26. cast_7_avg_revenue
    27. cast_7_experience
    28. cast_7_movies_before
    29. cast_8_avg_profit
    30. cast_8_avg_revenue
    31. cast_8_experience
    32. cast_8_movies_before
- Aggregated cast features (3 features)
    1. cast_avg_avg_revenue
    2. cast_avg_avg_profit
    3. cast_avg_experience

- Aggregated year features (2 features)
  1. year_avg_profit
  2. year_avg_revenue
- Aggregated collection features (2 features)
  1. collection_avg_profit
  2. collection_avg_revenue

**Target features**

The dataset contains 3 target features for 3 different tasks:

- Binary feature whether revenue is higher than budget (for binary classification)
- Categorical ordinal feature containing one of 9 classes of revenue split (for multi-class classification)
- Continuous revenue feature (for regression)