Abstract
Starting from 2020, the film industry has been impacted significantly due to the global pandemic. Theaters and production companies were closed, movies were put on hold, and related events got canceled. People have no choice but to stay at home enjoying streaming platforms services during the quarantine, and it's becoming a trend. With the film industry being challenged, it's crucial for investors to make decisions with minimum risk. In this study, we will analyze the historical data of the film industry with different classification models and provide our recommendations on which models could perform well when predicting if the film could be a success or not. Knowing which movie would more likely turn out successful will help the movie investors and film production companies make appropriate investment decisions.

1 Introduction and background

Filmmaking is a process that requires hard work from a lot of people and a great amount of investment. The evaluation criterion on if the film is a success or not is different from parties involved in the project. Directors and actors place a higher value on receiving a gold in the prestigious awards in the industry, while movie investors and production companies are more concerned about the box office revenue because of the pressure from development cost, production cost, marketing expense and so on. Stephen (2016) presented that a film is likely to be profitable when its box office revenue is at least twice greater than its budget. In our work, we will focus on factors that could affect the revenue of the film which will be used as variables in the classification models to predict the successfulness of the film.

2 Data Sources

We found the dataset in Kaggle with the name of "TMDB 5000 Movie Dataset", which was originally from TMDb, a community built online database with ratings of movies and TV shows. This database includes 4803 movies released from 1916 to 2017. The raw data contains 20 Variables:

Numerical explanatory variables: budget, id, popularity, release date, runtime, vote average, vote count. String explanatory variables: genres, homepage, keywords, original language, original title, overview, production companies, production countries, spoken languages, status, tagline, title.

Response variable: Success (revenue >= 3*budget).

3.1 Data Wrangling and Exploratory Data Analysis

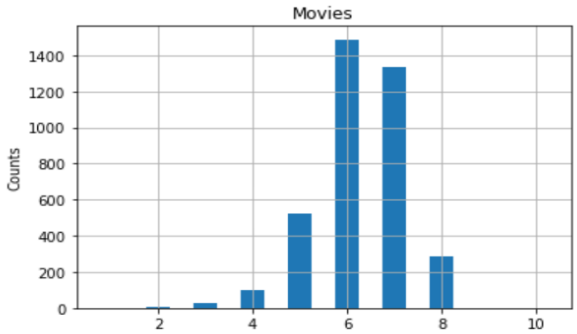
3.1.1 Data Wrangling

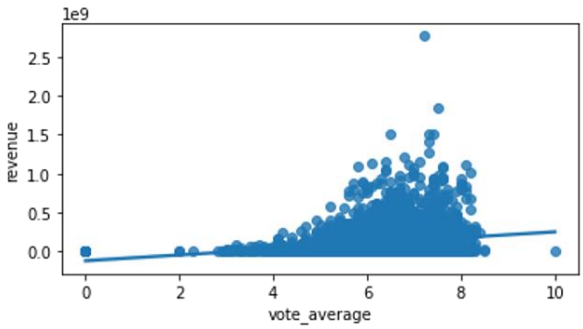
The original data contains missing values, dictionaries, and unnecessary variables, therefore data wrangling is needed before this data can be used for model building. The details of data wrangling is as follows:

1. Drop columns that will not be explored: ["homepage", "id", "original_language", "keywords", "release_date", "overview", "production_countries", "spoken_languages", "status", "tagline"]
2. Drop the entries that has null values
   Drop the entries where “budgets==0”. Having 0 budgets is not possible and may simply because of the lack of information
3. For variable “genres”, it contains a list of dictionaries as is shown below: [{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}, {"id": 14, "name": "Fantasy"}, {"id": 878, "name": "Science Fiction"}]. The name is separated into a list [Action, Adventure, Fantasy, Science Fiction], and dummy variables are created.
4. For variable “production_companies”, similar operations are carried out as in step 3 and dummy variables are created
5. Categorical variable “Success” is created, and this is the response variable for our studies. The successfulness of the movie is determined when the revenue is greater or equals 3*budgets.
6. Drop column “revenue” which is multicollinear with the variable “success”.
7. The final dimension of the data is (3764, 4216).
8. Randomly split the data with 80% for training data and 20% for testing.

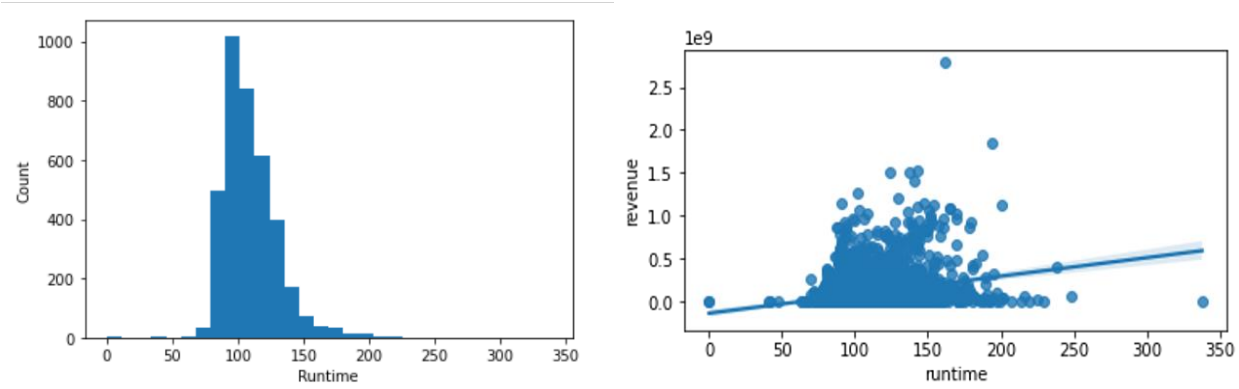
3.1.2 Exploratory Data Analysis

Before diving into our machine learning algorithms, we will first take a look at our data. Below are some interesting facts about our data:
When we took the average vote_average for each movie and put them into a histogram with bins ranging from 0.5 to 10.5 with an interval of 1.0, we found that the majority of our movies have a vote average of 6 or 7 out of 10. There’s a weak correlation between revenue and vote_average

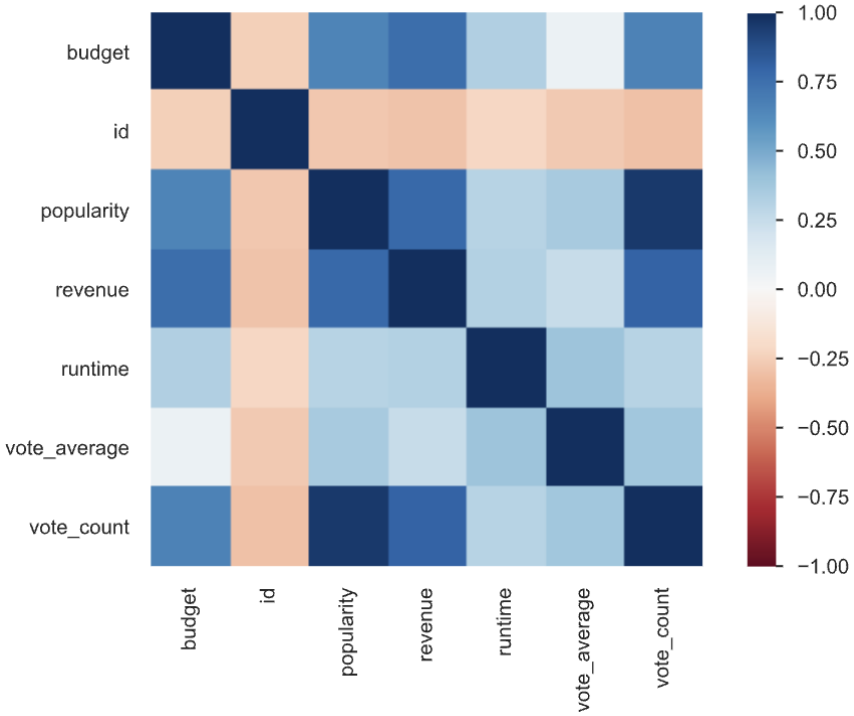




When we did the same for each movie’s runtime, we found that the majority of the movie’s runtime is between 100 minutes to 120 minutes. And there is a weak but positive correlation between revenue and runtime.



When we generated a Pearson correlation matrix using all the numerical values in our dataset, we noticed that these four variables: budget, popularity, and vote_count are highly correlated with revenue. Runtime and vote_average have a lower correlation.



There are 20 genres in our dataset and the top 5 movie genres are: drama with 1732 records, comedy with 1311 records, thriller with 1073 records, action with 1015 records, and adventure with 719 records. Some movies might have multiple genres labels thus they would get counted more than once in different genres.

Movie genres with least records: TV with 3 records, foreign with 8 records, documentary with 55 records, western with 66 records and war with 132 records

Below chart shows the top 10 movies with the highest number of votes and their corresponding genres:

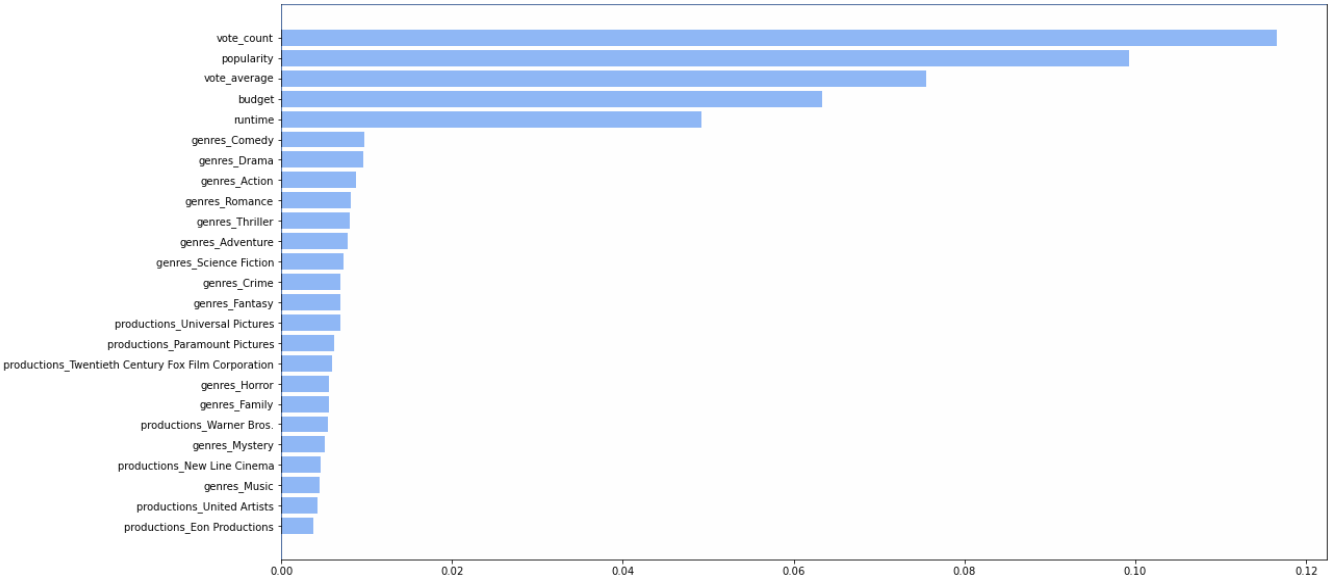
	Votes	Genres
Inception	13752	Action,Thriller,Science Fiction,Mystery,Adventure
The Dark Knight	12002	Drama,Action,Crime,Thriller
Avatar	11800	Action,Adventure,Fantasy,Science Fiction
The Avengers	11776	Science Fiction,Action,Adventure
Deadpool	10995	Action,Adventure,Comedy
Interstellar	10867	Adventure,Drama,Science Fiction
Django Unchained	10099	Drama,Western
Guardians of the Galaxy	9742	Action,Science Fiction,Adventure
The Hunger Games	9455	Action,Adventure,Science Fiction
Mad Max: Fury Road	9427	Action,Adventure,Science Fiction,Thriller

Top 10 movies with the highest average rating - including number of reviews they received: Me You and Five Bucks, The Shawshank Redemption, There Goes My Baby, The Godfather, Fight Club, Schindler's List, Spirited Away, The Godfather:Part II, Pulp Fiction, Whilplast.
Among these top 10 movies with the highest average rating, two of them are outliers with only two votes count for each movie. These two movies are Me You and Five Bucks and There Goes My Baby. If we remove these two outliers, then The Shawshank Redemption would have the highest average rating among all movies in our dataset.

3.2 Implementation and Methodology

3.2.1 Random Forest Classifier

First Random Forest Classifier is used for feature selection followed by prediction. Using the default random forest parameters of n_estimators, max_features, min_samples_leaf, and min_samples_split, the training error is 0 and the testing error was 0.2084. As mentioned previously after data wrangling we had 4216 features. Random Forest Classifier showed the importance of all the features and the top 25 are shown in the plot below. To reduce the computation cost, we will only be targeting the top 25 features, as the importance does not decrease drastically.



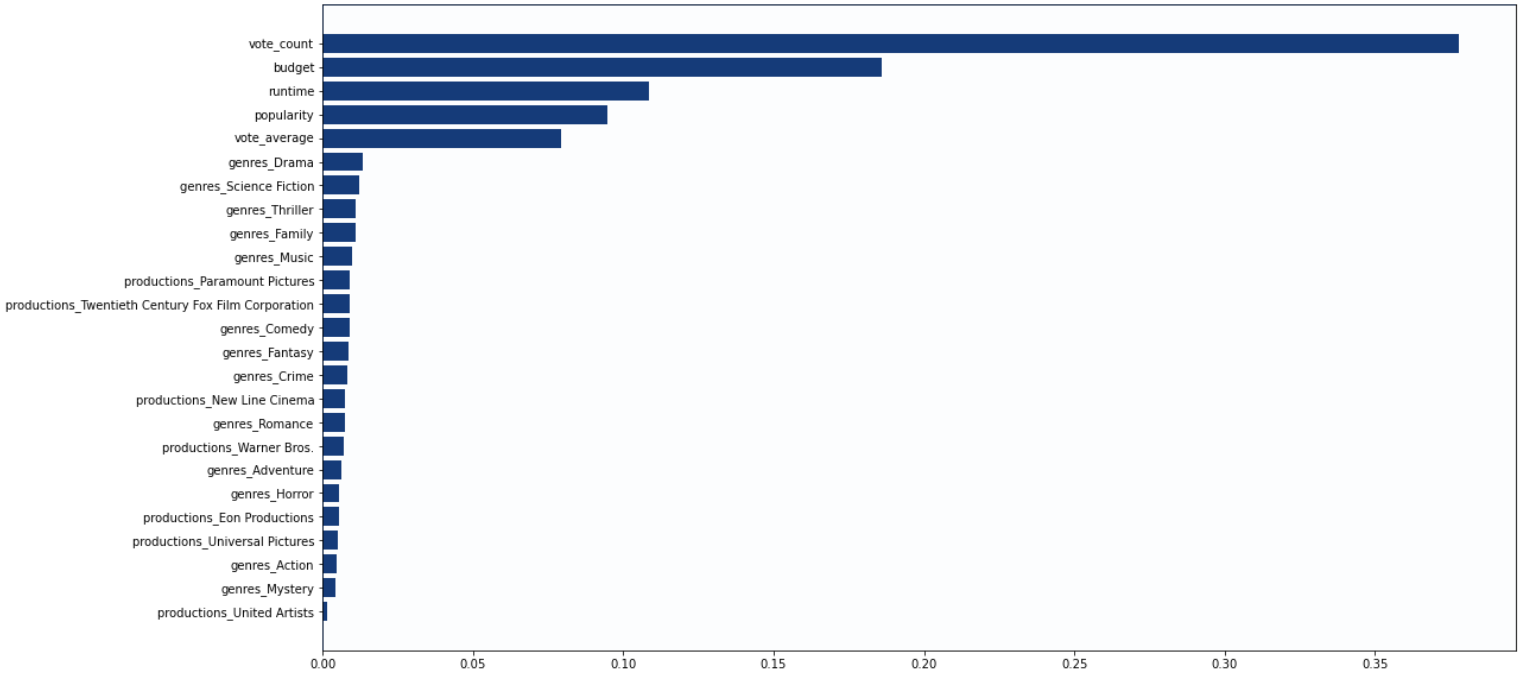
(magnify to see the details)

To find the best optimal model, the parameters need to be optimized. For this purpose, I turned to the "GridSearchCV" package in sklearn.model_selection for hyperparameter tuning and code can be found in the APPENDIX. Using this method, the optimal value for the rf model is max_features = 4, min_samples_leaf = 1, min_samples_split = 4, n_estimators = 150. Cross-validation of the model using the best selected parameters with random the split of training (80%) and testing data (20%) resulted in a training error of 0 and testing error of **0.2058.**

3.2.2 Gradient Boosting Classifier

Next we used the Gradient Boosting Classifier to build the model. As was done previously with the random forest model, the hyperparameters of this model need to be tuned. This was done using the "GradientBoostingClassifier" package as is shown in the APPENDIX. The optimal parameters were selected using a cross-validation approach with 3 splits. The results given are learning_rate = 0.5, max_depth = 7, n_estimators = 200. Using these parameters, the training error was 0 and the testing error was **0.2138**.

The plot below shows the relative importance of the features.



(magnify to see the details)

3.2.3 Simple Baseline Methods

We also performed five simple baseline models to the dataset with top 25 features selected from above so we have a better understanding about the basis for modeling the prediction and the accuracy improvement of random forest and gradient boosting methods.

Linear discriminant analysis (LDA): the discriminant functions are linear functions and it is the Bayes rule under the assumption that the densities are multivariate normal with common covariance. It is expected to perform better than logistic regression if the normality assumption is met.

Quadratic discriminant analysis (QDA): the discriminant functions are based on posterior distributions. And the mean and covariance matrix are estimated for each class, which is the main difference from LDA.

K-nearest neighbors(KNN): a non-parametric and non-interpretable method that is practical for large training datasets. In our work, we randomly chose five k values, which are 31,59,101,201,301, to predict the training error and testing error.

Naive Bayes: a generative model assumes that all predictors are independent. The performance could be not good as expected when some of the features are correlated.

Logistic Regression: a very efficient supervised model for classification problems with a binary outcome. It doesn't require the linear assumption and normality assumption, which could be the reason that it is safer and more robust than LDA and QDA.

4 Results and Findings

4.1 Summarized results of all methods

One Split:

	Train Error	Testing Error
LDA	0.2405	0.2457
QDA	0.2710	0.2709
Naive Bayes	0.2933	0.2815
Logistic Regression	0.2288	0.2390
KNN (K = 31)	0.3819	0.3174
KNN (K = 59)	0.3613	0.3267
KNN (K = 101)	0.3427	0.3280
KNN (K = 201)	0.3414	0.3267
KNN (K = 301)	0.3414	0.3267

Cross validation with 100 iterations:

	Avg Train Error	Avg Testing Error
Random Forest	0	0.2058
Gradient Boosting	0	0.2138
LDA	0.2429	0.2455

QDA	0.2643	0.2765
Naive Bayes	0.2897	0.2939
Logistic Regression	0.2299	0.2330
KNN (K = 31)	0.3710	0.3121

4.2 Findings

Drama, comedy, thriller, action, and adventure are the most popular genres in the period from 1916 to 2017, and the top 10 movies of the dataset include at least one of these genres. Also, the movies with the runtime between 100 minutes to 120 minutes are the most acceptable and comfortable for the audience. These factors could be taken into consideration when the investors and production companies make choices among plenty of proposals.

Furthermore, according to the Random Forest and Gradient Boosting Classifier, the most important features are "vote_count", "popularity", "budget", "runtime", and "vote_average", although the rankings are different. The most important genres for RF model are "comedy", "drama" and "action" while for GSC are "drama", "fiction" and "thriller".

The Random Forest method performed the best in terms of training and testing error. Not only does it give the lowest testing error, but also it's advantageous for several reasons. First, the tree-based methods do not require normality assumptions like LDA/QDA does. Nor does it require the observations to be independent of each other. Additionally, tree-based methods generally perform well with many features.

However, the result from the Random Forest is not interpretable because Random Forest ran hundreds of models in the background and then used the calculation from those models to come up with the single result. In this regard, Logistic regression is more advantageous as it sacrifices a little accuracy in prediction, but is much easier to interpret.

Five k values were used in KNN method, and it turns out that the model performed the best when k equals to 31. As k increases, the training error decreases and the testing error first increases and then decreases. The performance of the model didn't improve with k over 200.

Compared with logistic regression, random forest and boosting methods, the LDA, QDA and Naive Bayes show moderate performance in training and testing error. The normality assumptions for LDA, QDA and Naive Bayes may be one of the reasons why they don't perform as well.

5 Conclusions

To conclude, we have taken a rather complicated dataset with thousands of observations and multiple variables containing dictionaries and missing values and cleaned it. We have created new

variables to facilitate our studies. A total of 7 models were evaluated and compared. Cross validation was performed to reduce overfitting and gave a better approximation for model accuracy. We have found that Random Forest, Gradient Boosting and Logistic Regression show good predictive power for this particular dataset. The accuracy could reach as high as 79%. LDA, QDA, Naive Bayes, KNN models are not recommended for our purpose.

6 Lessons we have learned

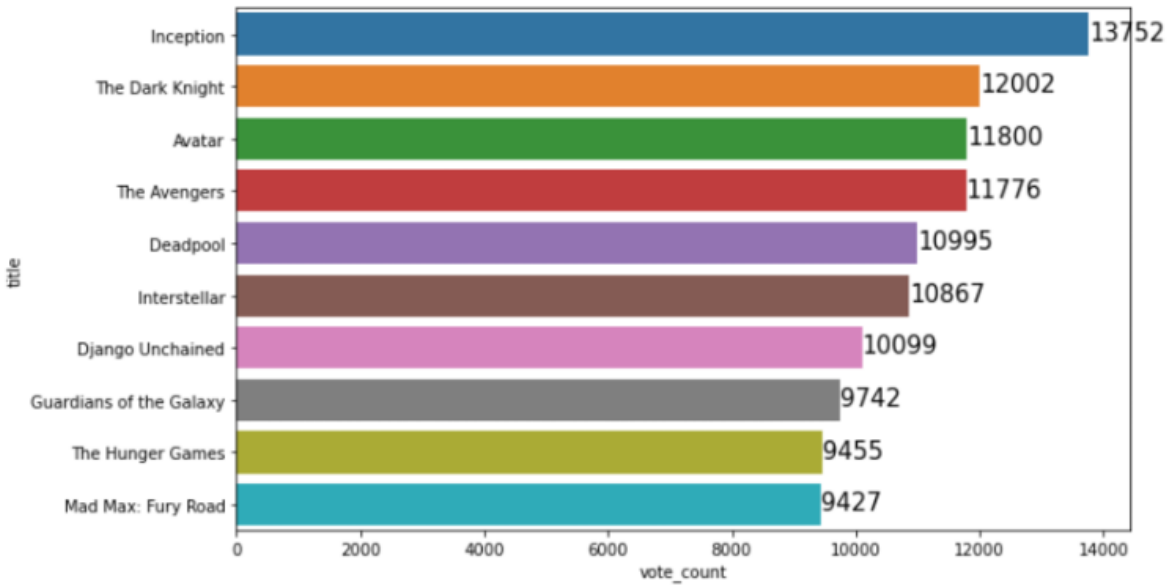
The real-world data comes in various forms such as paragraphs, dictionaries, lists, missing values and invalid values. Having them cleaned up so that we can fit the data into our mathematical models takes time and effort. Also, deciding which explanatory variables to keep has a significant impact on the final outcome of our models because just like the old sayings says, garbage in and garbage out. If we chose inappropriate variables as our explanatory variables and used them to predict our response variables, then the final outcome would be unreliable.

For a dataset with 4216 variables at the beginning, random forest is very handy in feature selection, as the Lasso and Ridge regression in R failed to handle too many features due to "too many weights".

The selection of models could also have a big impact on the accuracy of prediction. We need simple baseline models to set the basis of modeling accuracy and compare it with the accuracy of other machine learning models to ensure the results are reasonable. For example the random forest model has an overall testing error of 0.2058 and the KNN model has an overall testing error of 0.3121. It's hard to predict the relative performance of all models for classification, and therefore, trying out different models is essential.

Appendix

Top 10 movies with the highest number of votes



Top 10 movies with the highest average rating - including number of reviews they received