Project Report

DATS6103 - Intro to Data Mining

Predicting Movie Success using Machine Learning Classification Algorithms

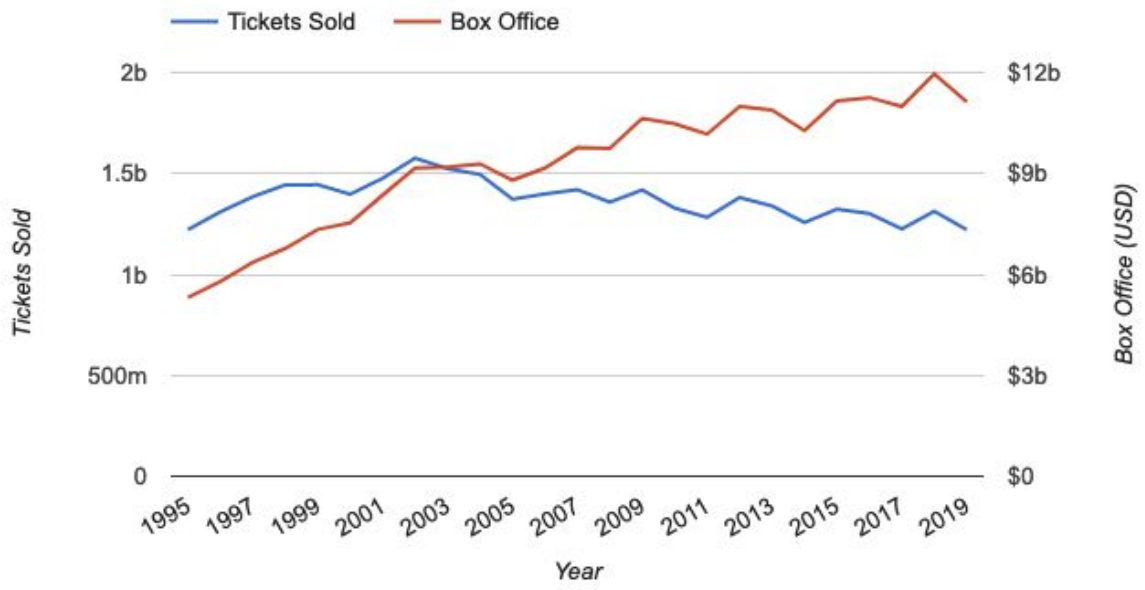
Group 8

Amna Gul, Madhuri Yadav, and Hemanth Koganti

Table of Contents:

Introduction:

Film industry is one of the top grossing industries in the world. Millions of dollars are invested in the making of each movie expecting high margin of profit. From the graph below, as per the data available for movies released in United States 1.2 Billion cinema tickets were sold generating a revenue of approximately $11 Billion in 2019 alone.



So the primary goal by undertaking this project is to investigate the influential factors affecting the success or failure of a movie which could be game changing not only for the producers but also the audience. With all this in our mind we set out on a quest to answer this question: Whether is it possible to use machine learning algorithm to predict if movie will be a success (generate at least as much revenue as its budget) or a flop?

The primary source of our data set is Kaggle but some columns were added by parsing data made publicly available on IMDb's website. Data Dictionary pertaining to our data set is appended at end of this document for reference.

Following are the modifications that we applied to our raw data set.

Data Cleaning:

The first operation performed was to remove all those columns from our data set which were irrelevant e.g. "homepage" and "poster_path". Then we searched for corrupt values in the columns e.g. "budget" column contained alpha-numerical values so we removed all such rows.

Two variables "Genre" and "Production_Companies" were in JSON format in raw data set so one of the challenges we faced during cleansing data was extracting and converting them to

string data type. “release_date” column was also converted to its proper datetime format using pandas. Based on the months in this column, we created a new column because we believed that movies released in vacation months e.g. summers or at Christmas/New Year time have a higher chance of getting good profits.

Since the dataset from Kaggle was two years old and “average_rating” and “vote_count” are values which keep updating on very regular basis as more people watch and rate the same movies so we got those as well as the “Director” names from the largest and most authentic movie source i.e. IMDb.

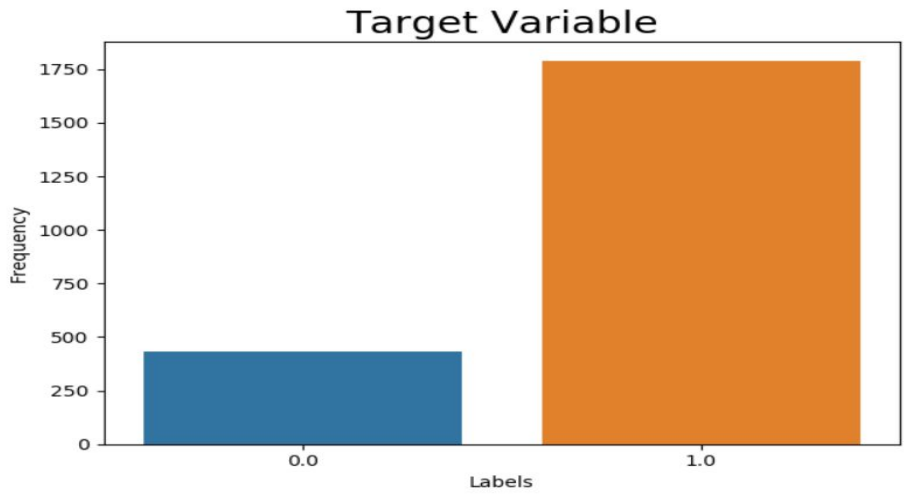
To keep the outliers in our data set to a minimum we excluded all those rows for which budget value was less than $100,000 and revenue value less than $1000. Afterwards we created our target column by dividing revenue by budget. If the value obtained was greater than 1, we categorized that movie as “success” else “flop”.

Last step performed was to remove all duplicate values. We started with a dataset containing 45,000+ rows but after cleaning ended up with having only 2,222.

Exploratory Data Analysis:

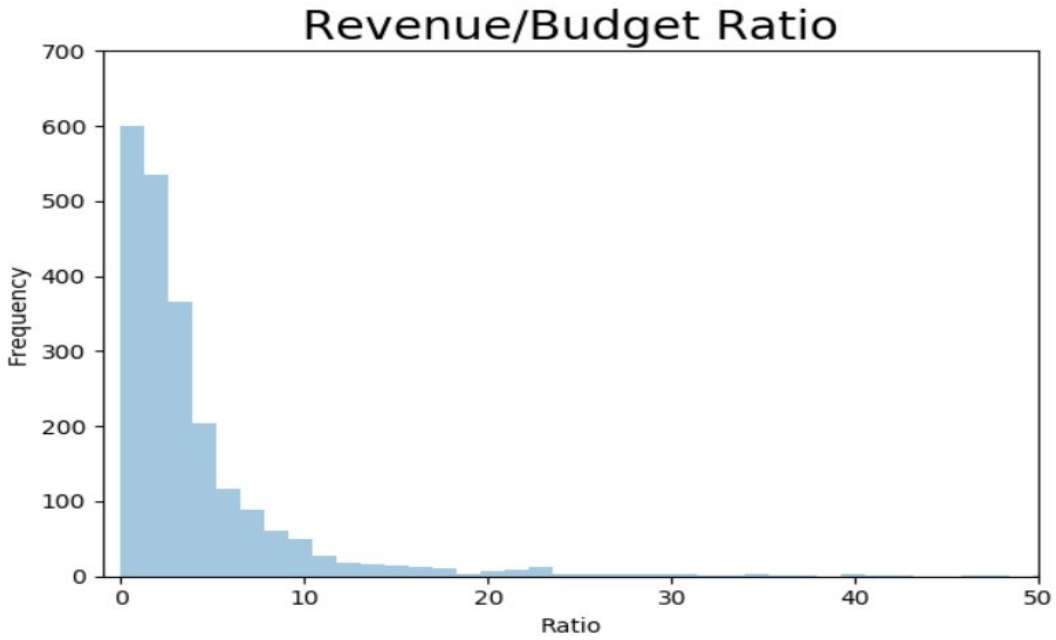
Dependent Variable:

First lets take a look at our target variable. From the histogram below we see that there are approximately 1750 successful & 450 flop movies in our cleaned data frame.

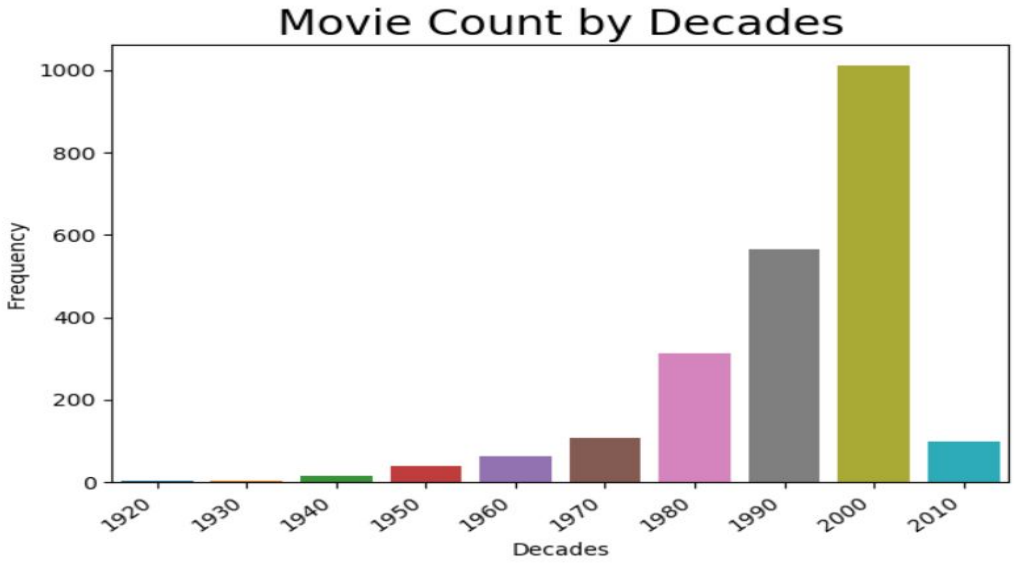


Independent Variables:

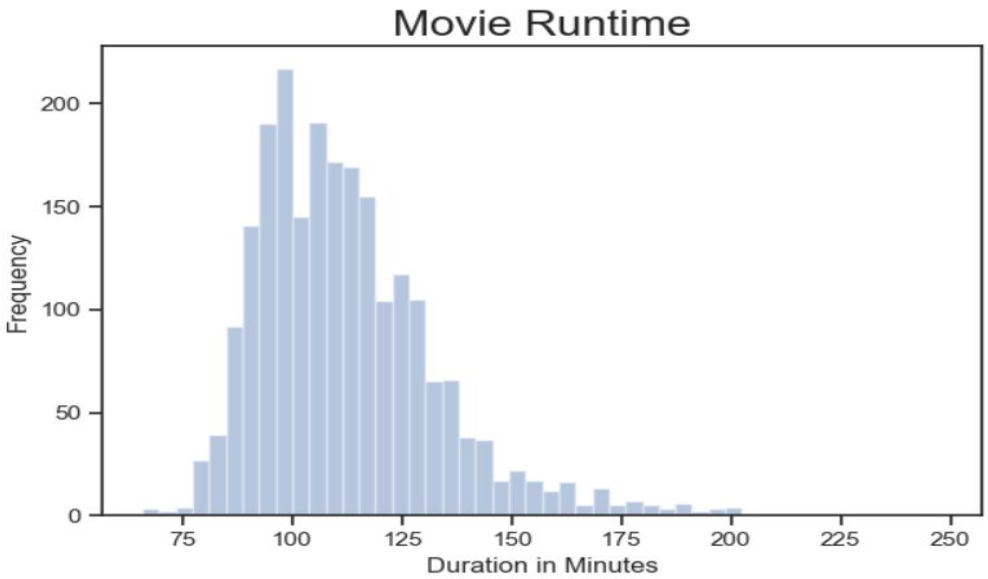
Out of 2,222 movies the ratio of Revenue/Budget ("status" column) that they generate lies between 0 & 643. Most of the data is concentrated around 0 & 10.



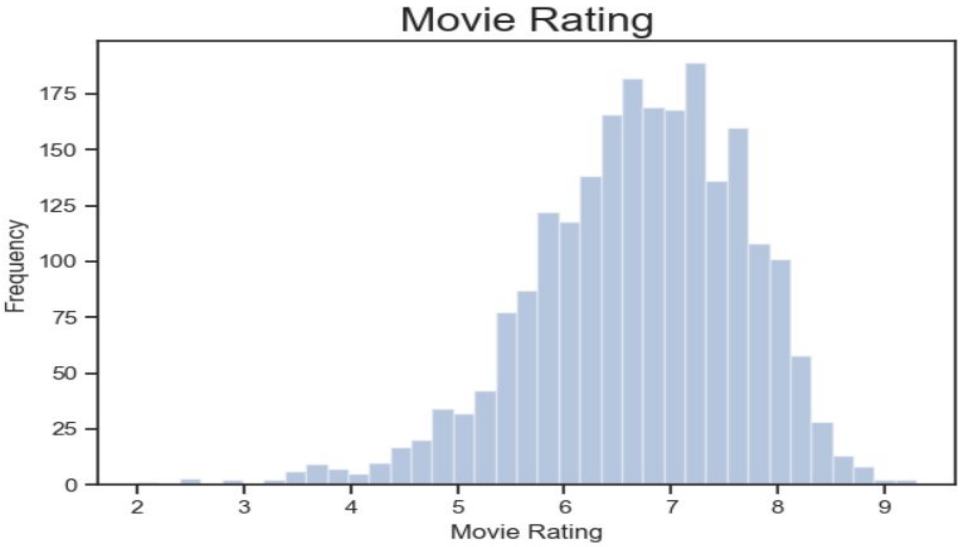
Coming towards the release year data, we found out that the oldest movie in our dataframe was released in 1921 whereas the most recent one was released in 2017 (*Kaggle dataset we are using is 2 years old).*



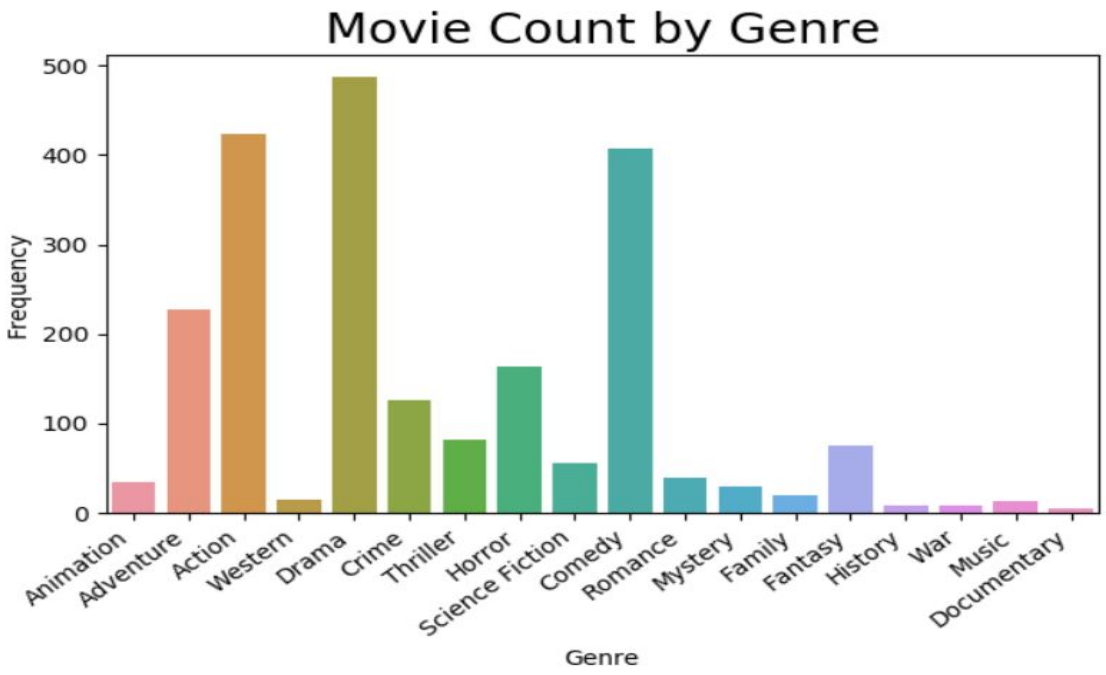
The distribution of “runtime” depicts that most movies are concentrated around 80 to 120 minutes.



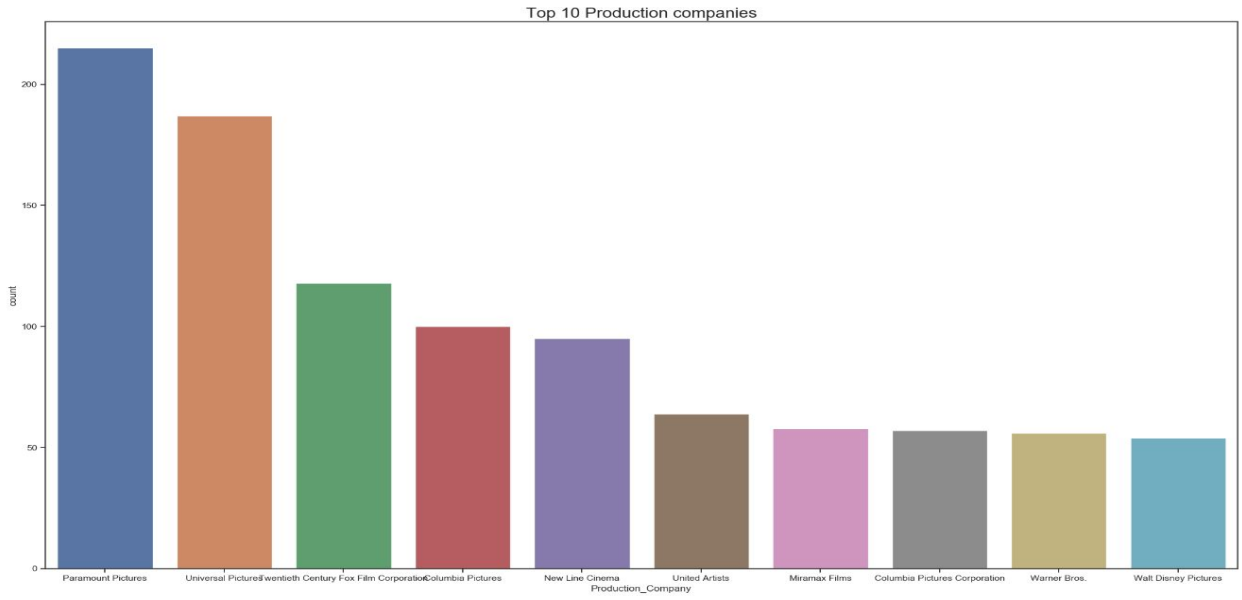
Then we plotted histograms for “VoteAverage”. Movie that got the maximum rating was “The Shawshank Redemption”.



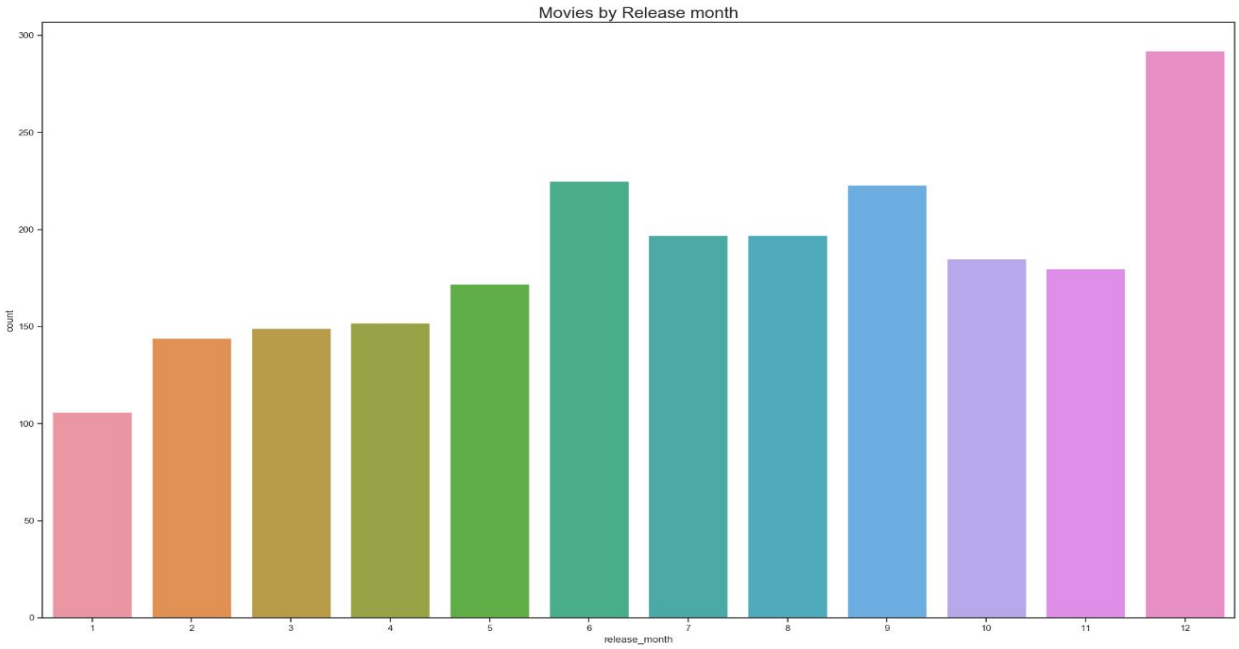
Although there were 18 genres in our data set but most of the movies fell under the category of Action, Drama and Comedy.



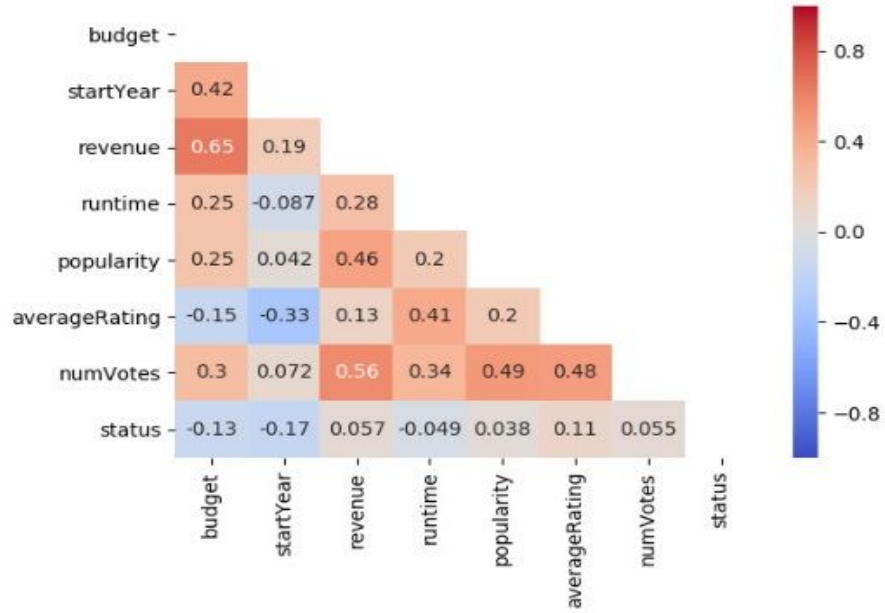
Then, we plotted the top 10 Production companies in our cleaned data set. We can see in the below graph that Paramount pictures produced more number of movies.



Here in the below picture, we can observe the number of movies released in each month of the year. More movies were released in December whereas January has less number of releases.



Finally we plotted the correlation heatmap between numerical variables. The only figure that is worth mentioning is 65% of positive correlation between budget & revenue. But overall we see no significant relationship between any of our numerical features.



Modeling:

After preprocessing and EDA we move to Modeling and Prediction. Before applying the model on our data set we performed following steps:

1. Data Splitting: We split our data into train(70%) and test(30%). We have 2222 observations out of which 1555 are test and 667 in test.
2. Stratified Sampling: As our data is highly biased out of 1555 observations in train set only 304 values belong to class 0(Flop) Hence we stratify the sampling. Here we have used RandomOverSampler method.
3. Label Encoding: To deal with categorical values we use one-hot encoding using LabelEncoder() in our code.
4. Scaling: To deal with the numerical values in the data set we use MinMaxScaler(). The formula for MinMaxScalar is as follows.

   Xsc = (X-Xmin)/(Xmax - Xmin)

5. Feature selection: In feature selection we eliminate unrelated columns like Poster_Path contains only the path/URL to the image. The Adult column which had False value throughout except for 19 observations. Then We eliminated Cast feature since it had around 1000 unique categories and did not add any value to the model.

Initially we started with 4 features 'runtime', 'averageRating', 'Genre', 'Production_Company' and We applied DT(Gini), DT(Entropy), SVM, KNN, NB classification models on our data. Following are the results.

Model	Accuracy
Accuracy DT Entropy	69.26%
Accuracy SVM	81.40%
Accuracy RF	80.35%
Accuracy KNN	71.81%
Accuracy NB	22.93%

accuracy score = (Obs Accuracy – Exp Accuracy)/(1 - Exp Accuracy)

We got Accuracy Score of 81.4% for SVM. However, AUC was just 61%.

From the confusion matrix we see that 119 values are still incorrectly labeled as 0 This might be due to Biased Label.

To further improve the model we used all 7 features which were relevant to be used. And this was the exhaustive list with our dataset.

Following are the results with 7 features
[runtime','averageRating','budget','Genre','Production_Company','release_month', 'popularity']

Model	Accuracy
Accuracy DT Gini	73.91%
Accuracy SVM	81.41%
Accuracy RF	83.35%

We ignored DT(Entropy), KNN, NB here since our results weren't significantly different than before. Since we have 83.35% accuracy which is good we look further into our results.

From AUC we see that value no was increased from 61% to 68% But we still have 105 of 1.0 labels are predicted as 0.0

Hence to further improve the model Bagging, Oversampling, and Boosting is performed.

Since we get better accuracy in DT, SVM and RF we use Hard Voting on the three samples but we observe that it did not contribute much to the improvement.

Since our data is highly biased, we oversample our train data. After over sampling now instead of 1555 we have 2502 observations. Here Random over sampling is used.

Then we performed Adaptive Boosting on random forest.

```
AdaBoostClassifier(RandomForestClassifier(n_estimators=100,random_state=seed),n_estimators=100,random_state=seed)
```

And following are the results.

Model	Accuracy
Decision Tree (Entropy)	72.56%
Support Vector Machine	69.12%
Random Forest	79.91%
Bagging (Hard Voting)	78.41%
Adaptive Bootstrap	80.96%

We do not see any improvement as such accuracy is still 80.96%. But when we look at AUC it is increased to 69%.
Also, the Cohen Kappa score is 20% which has been increased from 5 when we started initially. However confusion matrix almost has similar no of miss classified values.

Conclusion

As per our analysis Adaptive Boosting on random forest fits better than other models we trained. However, might be because of biased label results are not satisfactory. But the model is still better than probability which 50%.

Limitations

- Biased Label: If values were unbiased we might have been able to build better model
- Missing Values & Invalid Data : We lost 43,000 observation due to missing data values, We could have built a better model.
- Additional Features like main actors/directors rating, number of Oscars they have won etc. would have helped our model perform better.

References:

https://www.the-numbers.com/market/
http://www.diva-portal.org/smash/get/diva2:1106715/FULLTEXT01.pdf
https://io9.gizmodo.com/how-much-money-does-a-movie-need-to-make-to-be-profitab-5747305
Maklin, C. (2019). *AdaBoost Classifier Example In Python*. [online] Medium. Available at: https://towardsdatascience.com/machine-learning-part-17-boosting-algorithms-adaboost-in-python-d00faac6c464

Data Dictionary:

adult - whether the movie is adult rated or not(True or False).
belongs_to_collection - the collection to which the movie belongs to.
budget - The budget in which the movie was made.
genre - The genre of the movie, Action, Comedy ,Thriller etc.
homepage - A link to the homepage of the movie.
id - A unique identifier for each movie.
imdb_id - IMDB id of the movie.
keywords - The keywords or tags related to the movie.
original_language - The language in which the movie was made.
original_title - The title of the movie before translation or adaptation.
overview - A brief description of the movie.
popularity - A numeric quantity specifying the movie popularity.
poster_path - A link to the poster of the movie.
production_companies - The production house of the movie.
production_countries - The country in which it was produced.
release_date - The date on which it was released.
revenue - The worldwide revenue generated by the movie.
runtime - The running time of the movie in minutes.
spoken_languages - The languages that are spoken in the movie.
status - "Released" or "Rumored".
tagline - Movie's tagline.
title - Title of the movie.
vote_average - average ratings the movie received.
vote_count - the count of votes received.