Analysis Of Hollywood Movies

Anjali Patil, Kunal Mehrotra, Nidhi Bodar, Foram Shah

1. Summary

Every year thousands of movies are released and huge revenue is generated contributing to the global economy. However, not all movies make as much money as an investment. Some movies are a super success, while some of them fail to excite the audience. Many factors could contribute to whether a movie can be a success or a complete failure. In this project, we have performed an analysis on various factors like the film genre, the rating of movies, release year, country, main actors, user ratings, production company, number of votes, etc.
The primary data was not in tidy format and hence was cleaned.
We dropped some of the columns as any type of data transformation on them was not possible and thus would not be helpful to us in modeling and EDA. Some columns which were important and had missing values had to be imputed based on some pre-data analysis that we made on the data.
Then we performed exploratory data analysis in order to study more about features and the target variable. The target variable for determining whether a movie is a success or a flop was created by subtracting investment and revenue generated by a movie and by considering the average of user votes. The main aim of this project is to build predictive models- Logistic regression, Decision Tree, K-nearest neighbors, and Random Forest and make comparisons among them and choose the best fit model to determine if a movie is a success or a flop.

2. Data Analysis

2.1 About data:

The dataset used in this project was downloaded from Kaggle. The dataset consists of 6820 movies. Each movie has
1. Budget- the budget of a movie. Some movies don't have this, so it appears as 0
2. Company - the production company
3. Country - country of origin
4. Director - The director of movie
5.Genre - the main genre of the movie.
6.Gross - revenue of the movie
7.Name - the name of the movie,
8.Rating - rating of the movie (R, PG, etc.)
9. Released - release date (YYYY-MM-DD),
10. Runtime - duration of the movie
11.Score - IMDb user rating
12.Votes - number of user votes
13.Star - main actor/actress
14.Writer - writer of the movie
15. Year - the year of release

Methods:

2.2 Data tidying:

After analysis, we have decided to drop the following columns released, director, writer, star, and company as they cannot be encoded and therefore used for predicting. The columns which we will encode with categorical

values are rating, genre, and country. Columns like Rating, budget, and gross have missing values and need to be imputed. For missing values for Rating, we can fill them by Not Rated which is already present as a value, for missing values of Budget we will take the Production company average of the budget and assign accordingly, for Gross we first find a threshold number of votes for which the movies are profitable, not profitable and loss-making, so if the movie is profit-making, we set a value 25% higher than the budget, if not profitable then same as budget and if loss-making then we set 25% below the budget.
The new budget and gross are stored in new columns budget_imputed and gross_imputed.
We now create a Difference column that stores the difference between gross_imputed and budget_imputed.
We create our target column Success_or_not storing 1 and 0 for success or fail, calculated based on the difference and user votes column.

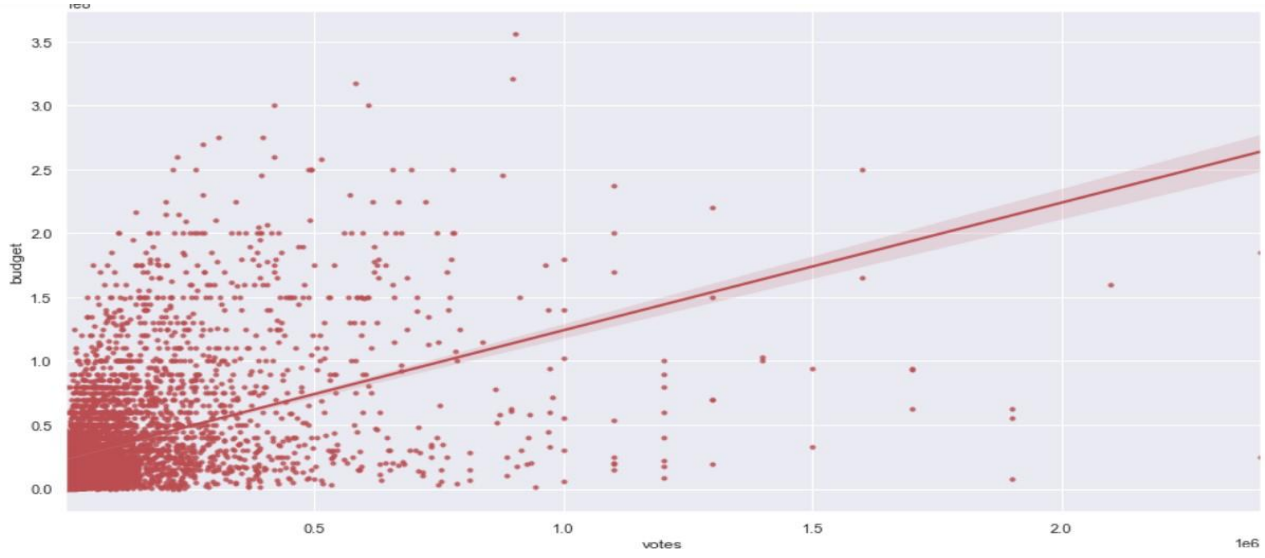
Categorical Variables/Encoding:

Categorical values are encoded for the rating column and stored in rating_encoded for values in the range [0-9].
Categorical values are encoded for the genre column and stored in genre_encoded for values in the range [0-15].
Categorical values are encoded for the rating column and stored in country_encoded for values in the range[0-53].

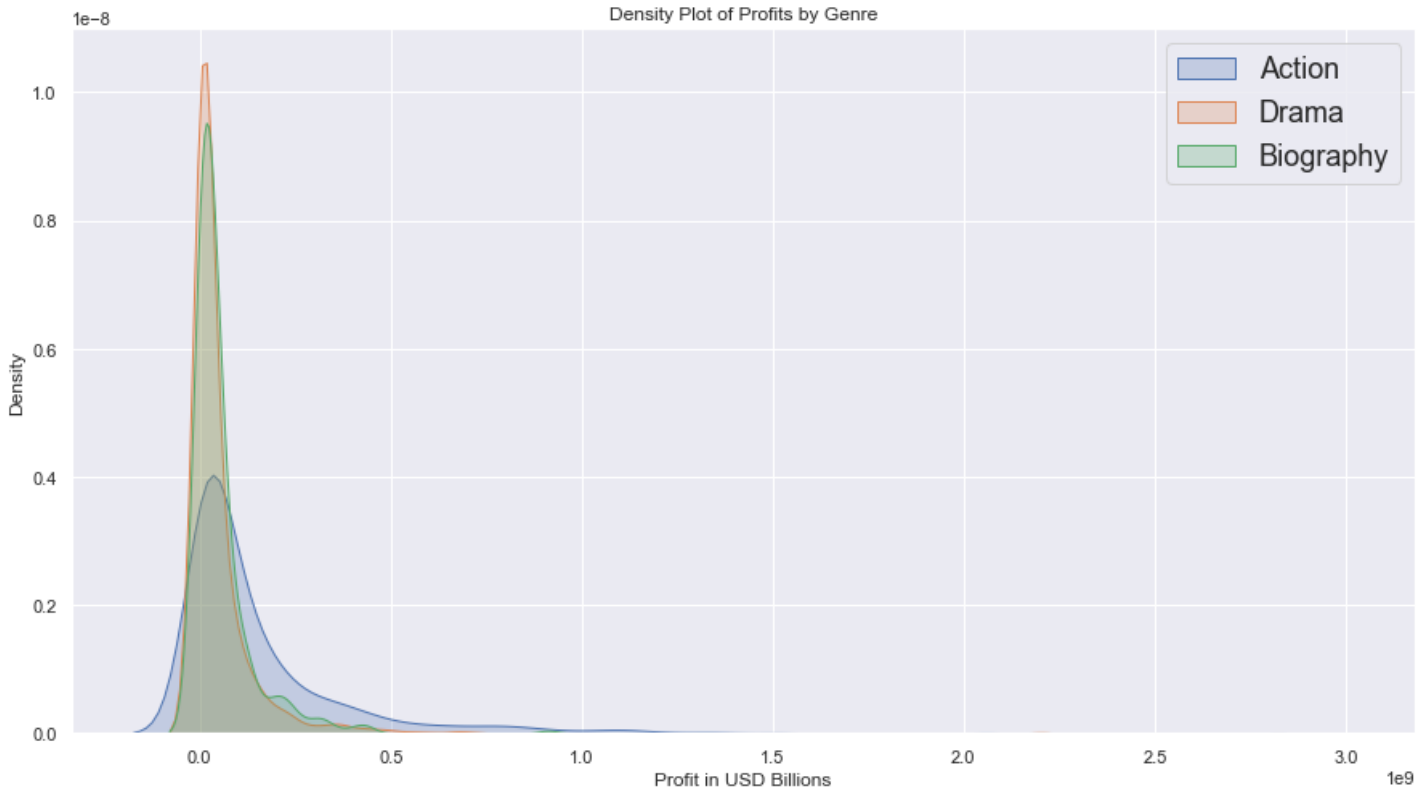
Imputations performed and final state of the dataset after Tidying of data.

1. rating - Missing values of rating are imputed by Not Rated.
2. genre - No imputations.
3. year - Dropped 1 row with an empty value.
4. score - Dropped 3 rows with empty values.
5. votes - Dropped 2 rows with empty values.
6. country - No imputations.
7. budget_imputed - Replaced missing values with the Production company average.
8. gross_imputed - Replaced missing values with a threshold number of votes set and values are based on their particular budget, based on assumption that positive votes are directly dependent on gross in most cases.
9. raring_encoded - Encoded column for ratings.
10. genre_encoded - Encoded column for genre.
11. country_encoded - Encoded columns for country.
12. Success_or_not - Values based on the difference of gross and budget and votes.

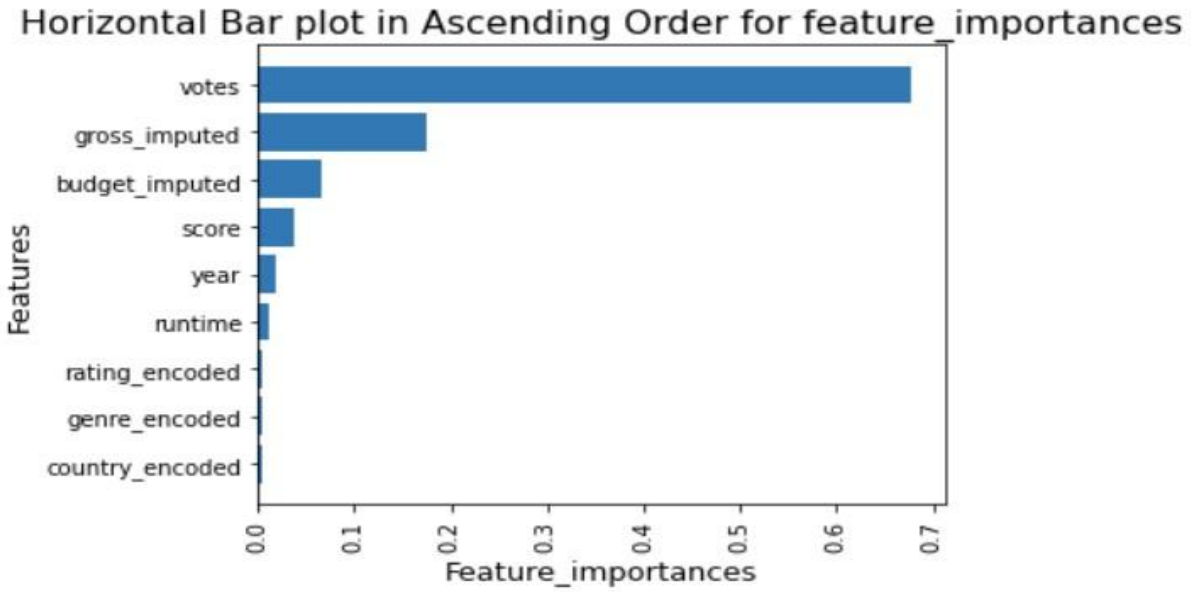
2.3 Exploratory Data Analysis:



The graph depicts budget vs votes which means it shows the relationship between both. It shows whether the increase in the budget results in an increase in votes. To our surprise, it was shown that a smaller number of movies got more votes from people who had a huge budget which means the number of votes did not increase with the increase in the budget. The area near low budget has more density in the scatterplot stating that people liked those movies more or we can rather determine that giving more budget to the movie does not increase the votes or liking of the people towards that movie.



The density plot here shows that the drama genre is most concentrated over 0.1 billion profits. While the concentration of the genre action is less than 0.1 billion, we can see a few movies reaching 1.25+ billion in profit. Also, the biographical movies too have a great density near 0.1 billion, it also has some movies which did produce a good profit.



The above Feature_importance graph shows that votes and gross are the most important features that affect our target variable which is “Success_or_not”. Essentially, the success of movies depends mostly on these two variables of our data.

3. Modeling

- Now that we have cleaned our data, we need to select a predictive model for predicting our target variable. Our target variable is “Success_or_not” which was created by subtracting the budget from gross revenue and by looking at the mean number of votes.
- So if the difference is positive and the number of votes is greater than the mean, the movie is labeled as “a success”, otherwise” Flop”. This was then encoded to “1” and ”0” classes. Classification is the process of predicting the class or label of given data points. For our project "year", "score", "votes", "runtime", "budget_imputed", "gross_imputed", "rating", "genre", "country", input variables and “Success_or_not” is our output variable.
- Splitting data into train and test sets is a crucial part of modeling as it helps in evaluating the performance efficiency of any machine learning algorithm. We have used an 80/20 split for the modeling.

3.1 Logistic Regression:

Logistic regression is a process of modeling the probability of a discrete outcome given an input variable. The most common logistic regression model a binary outcome; something that can take two values such as true/false, yes/no, and 0/1. Instead of fitting a straight line or hyperplane, the logistic regression model uses the logistic function to squeeze the output of a linear equation between 0 and 1. Since we want to predict whether a movie is a success/flop, logistic regression can be used as predictive modeling. Below are the results on the test data for LR:

Accuracy: 0.759
Precision: 0.520
Recall: 0.818
F1-Score: 0.636

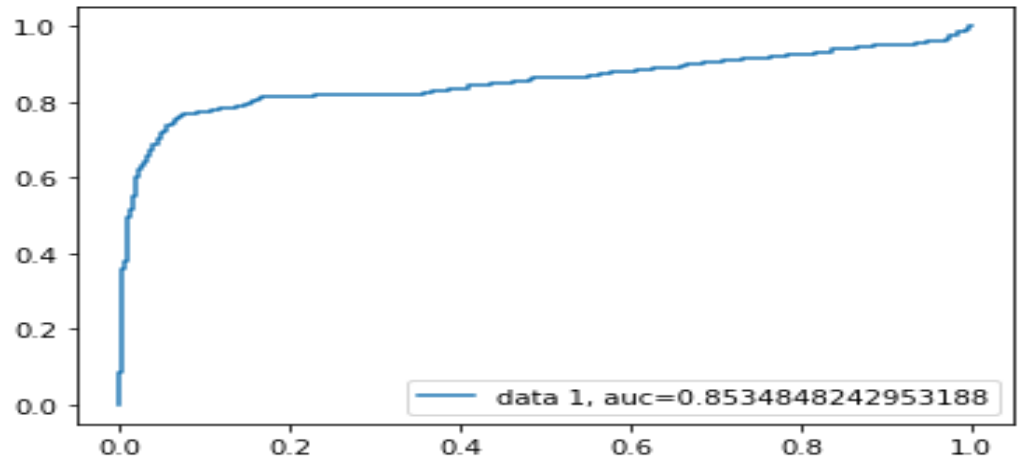
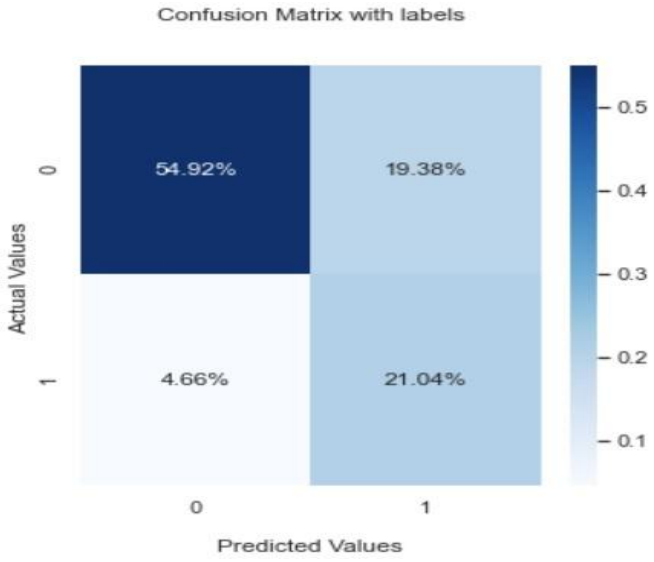


Figure 1: ROC plot.

According to figure 1, AOC is 0.85, which means our model's performance is good.



To evaluate the performance of the classification models, we've represented a confusion matrix as it can be observed that the classifier has accurately predicted 75.96% of the entire data true whereas 24.04% of data predictions are false where it has predicted 'Yes' instead of 'No' and vice versa.

3.2 K nearest neighbors:

K-nearest neighbor is a type of supervised learning algorithm used for both regression and classification. KNN tries to predict the correct class for the test data by calculating the distance between the test data and all the training points. Then select the K number of points that is close to the test data. The KNN algorithm calculates the probability of the test data belonging to the classes of 'K' training data and the class that holds the highest probability will be selected.

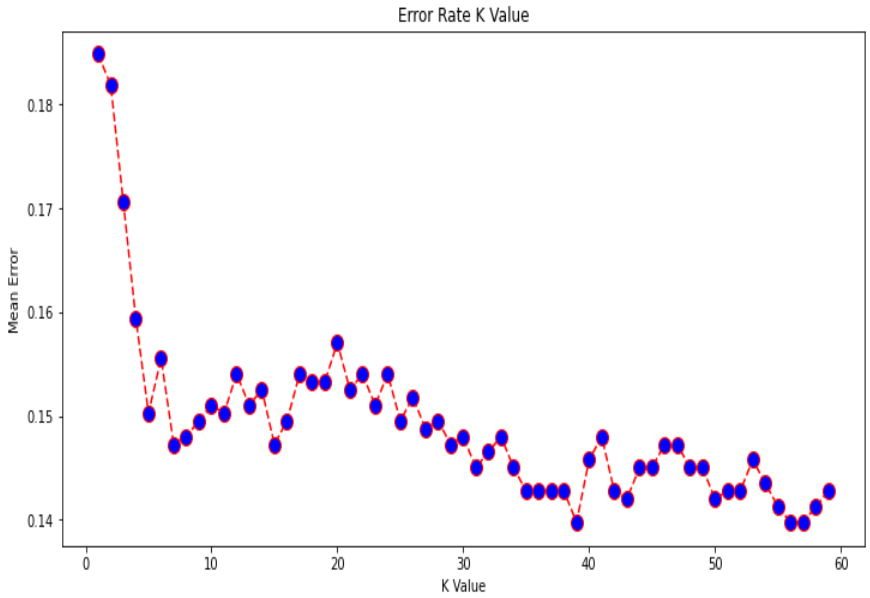
There is no ideal value for K and it is selected after testing and evaluation, however, to start out, we picked K=6.

Below are results when K=6:

Accuracy: 0.844

Precision: 0.752
Recall: 0.587
F1-Score: 0.660

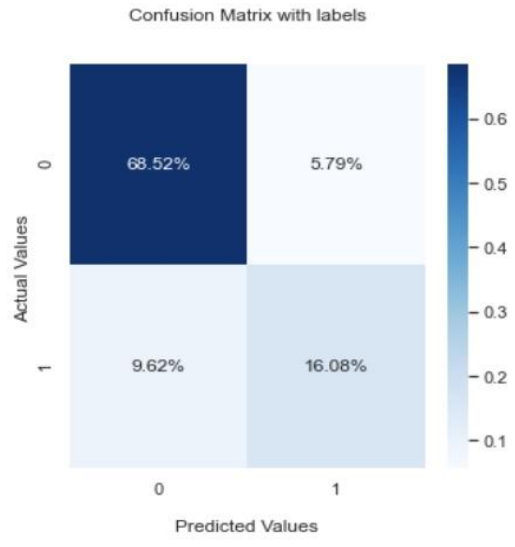
Though we are getting good results for k=6, one way to help you find the best value of K is to plot the graph of K value and the corresponding error rate for the dataset.



As we can see from the above plot, there is no significant change in error for different values of k. However, for k=39, the mean error is minimum. So, results when k=39 are:

Accuracy: 0.860
Precision: 0.774
Recall: 0.643
F1-Score: 0.7028753993610224

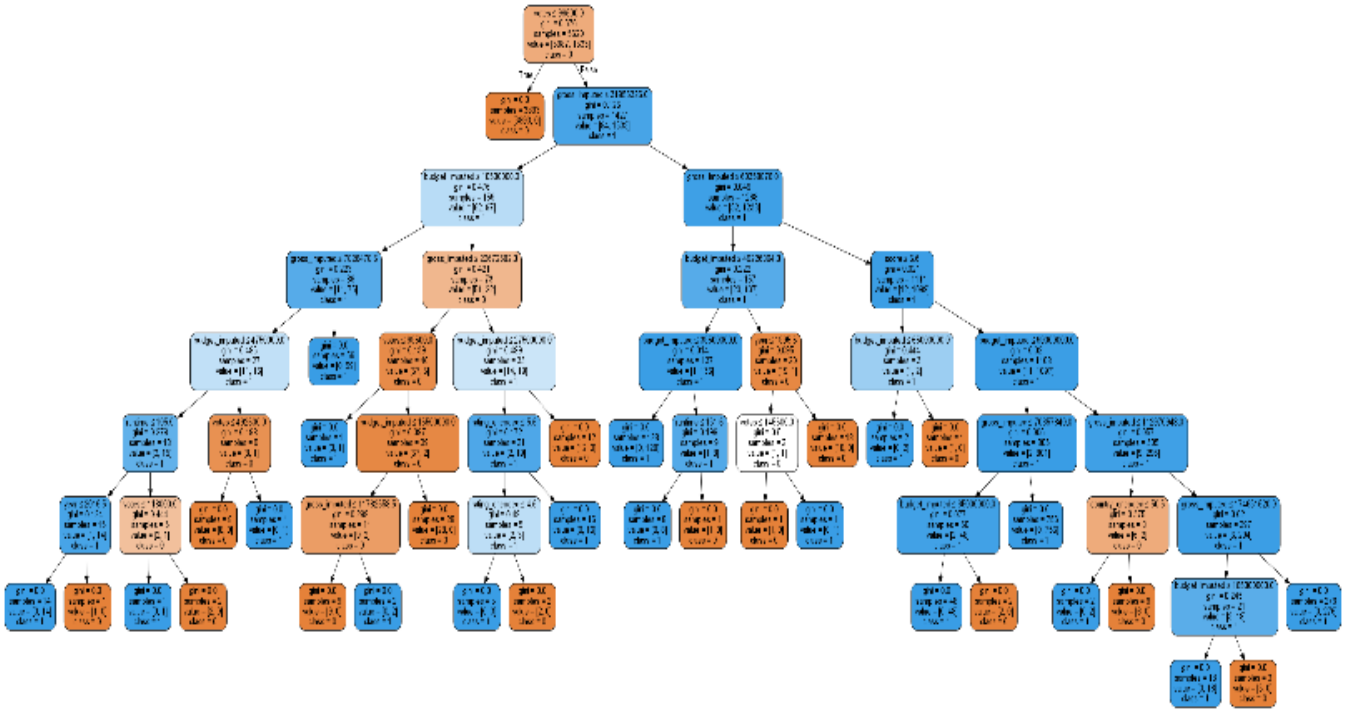
We can see that though accuracy, precision, and recall improved a bit, the values do not show much difference. This is not surprising as we can see in the above plot that the error ranges from 0.18 to 0.14.



It can be observed from the above confusion matrix for the K-nearest neighbors classifier that it has accurately predicted 84.6% of the entire data true whereas 15.4% of data predictions are false where it has predicted ‘Yes’ instead of ‘No’ and vice versa which is slightly better than the Logistic regression model.

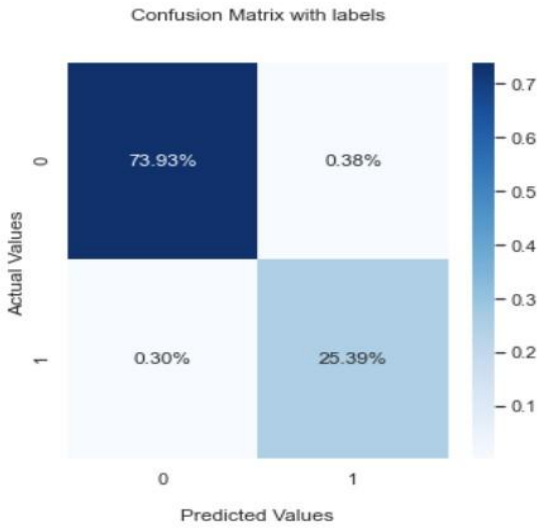
3.3 Decision tree:

Decision trees are a non-parametric supervised learning method used for classification and regression The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.



Above is the pictorial representation of our decision tree model.

Below are the results for our decision tree model:
Accuracy: 0.993
Precision: 0.985
Recall: 0.988
F1-Score: 0.986



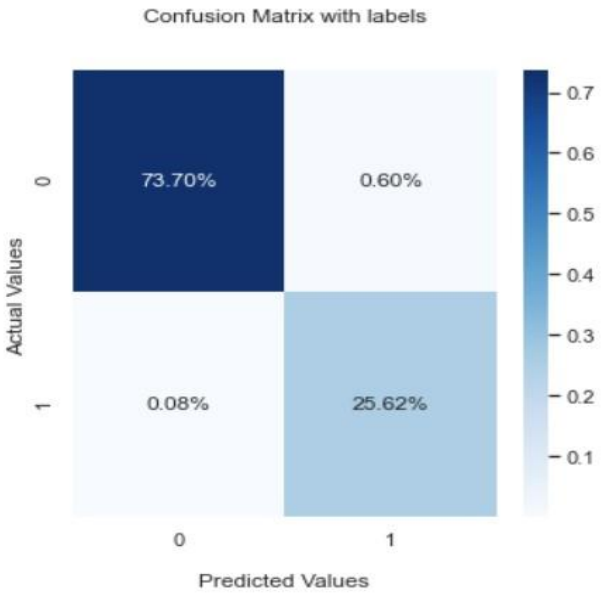
From the above confusion matrix of Decision Tree we can say that we've represented confusion matrix as it can be observed that the classifier has accurately predicted 99.32% of the entire data true whereas 0.68% of data predictions are false which is far better than the Logistic regression and KNN model's prediction.

As we can see, all the metrics are exceptionally good and hence this can be a good classifier for our project.

3.4 Random Forest:

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression. Below are the results for our random forest model:

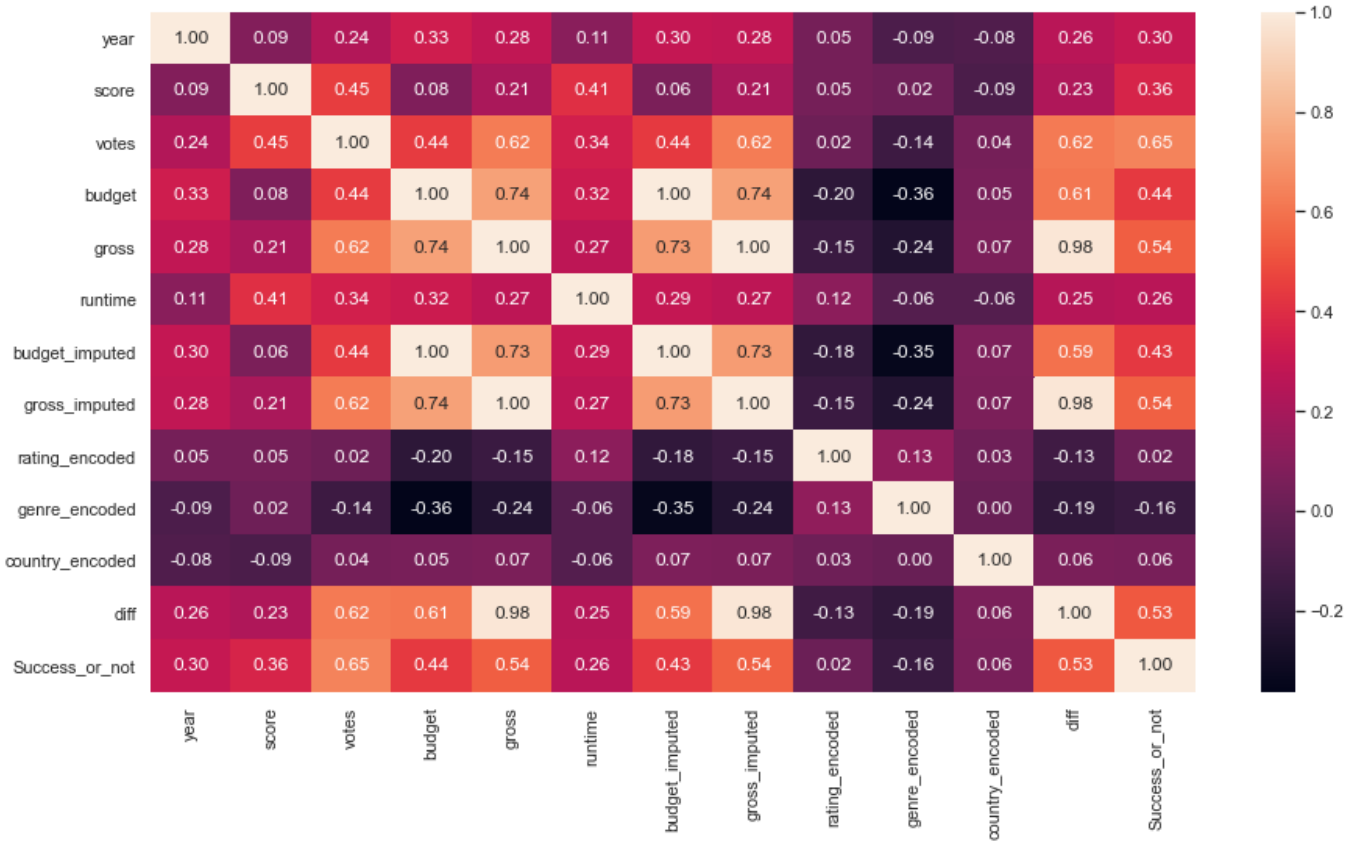
Accuracy: 0.9924
Precision: 0.9742
Recall: 0.997
F1-Score: 0.985



As far as Random Forest gives the truest values which are nearly 99.32% similar to Decision Tree's prediction and only .68% of the data, it predicts the false values i.e. False-positive and False Negative.

As we can see, all the metrics are exceptionally good and hence this can be a good classifier for our project.

Choosing Model Predictors:



The main concept to represent heatmap is the collinearity of the multiple variables in the dataset. According to the palette that we've used, the lighter the color the relationship The above heatmap represents that votes and gross (0.62), as well as budget and gross(0.74), are more correlated with each other than any other variables i.e. These variable states a positive linear relation between them. Also, our target variable which is Success_or_not is highly linearly correlated with votes (0.65) and gross (0.54).

Additionally, it can also be seen that runtime and genre have too little to no relation between them. Also, budget and rating follow a slightly negative linear relationship.

4.Results

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.759	0.520	0.818	0.636
KNN	0.844	0.752	0.587	0.660
Decision tree	0.993	0.985	0.988	0.986
Random Forest	0.9924	0.9742	0.997	0.985

From the above table we can see that Decision Tree and Random Forest performed better than Logistic regression and KNN.

A basic model such as Logistic regression and CNN performed better with our dataset when compared to complex models like decision trees. Logistic performs worst amongst the four. This might be due to class imbalance in data or the target label has no linear correlation with the features. Hence decision trees and random forests seem to perform better. As a result, in the future, given a movie and its features, if we test them against a Decision tree and Random forest, there is a good chance that we would be able to predict whether it will be a hit or a flop movie.

5. Discussion

The main idea behind the project was to predict whether a movie is successful or not based on certain features affecting its success rate. It can be derived that whether a movie is successful or not is highly correlated to the number of votes that movie got from the viewers. It was very obvious that people who liked the movie would definitely vote more (eg. giving 5 stars to a movie that we like or getting 9 + points on IMDP for a particular movie.) But as we know that votes can only be calculated after the movie's release. To find the second-best correlated feature, we use the heatmap and check the correlations for "success_or_not". Here we can see that the budget plays an important role too in the success of a movie. However, it gives only partially correct results which means that there can be a possibility that people might like certain actors more or they like stories from some particular production companies, etc.

Apart from this, we can notice that the quality of movies or the creativity might have increased over a time period as in the early 80s till 2000 the number of unsuccessful movies outnumbered the successful ones and it was the opposite after around 2001. One noticeable thing we could see was the number of movies in the states was more than in other countries which might be the reason why The United States has more successful movies. Certain relations can also be seen between the runtime of the movie and the score it gets which means the longer the movie, the more it is liked by the audience. One relation that we can derive here through one of our graphs(scatterplot) is that the higher the budget we have for the movie, the more votes it will get. It is already known that the movies with more votes tend to be successful. So here we can say that the money any production company puts on a movie has a lot to do with how successful it will get and the gross it brings. The number of action and comedy movies is much higher than animation and adventure but we can see that the budget of Animation seems to be highest and then follows Action and adventure so there are more chances of animation and action movies getting successful than the others.

Models such as decision trees and random forests have great accuracies i.e 99.3% and 99.2% respectively. However, KNN and logistic regressions do not perform well here. In the future, we can try making our data better so that simplest models like logistic regressions can perform well. We can also look forward to getting derivations for relations between the stars and the production company and the success rate. This way the productions can try working on particular genres with some particular actors and release in the countries that bring the max gross and make the maximum profit.

6.Statements of Contributions

Kunal Mehrotra - Date Preprocessing, Pipeline, and Documentation.

Anjali Patil - Data Modeling and Analysis.

Nidhi Bodar - Data Visualization and Documentation.

Foram Shah - Data Visualization and Documentation.

7. References

- Grijalva, Daniel. “Movie Industry.” *Kaggle*, 23 July 2021, https://www.kaggle.com/datasets/danielgrijalvas/movies.
- https://www.canr.msu.edu/news/what_do_movie_ratings_mean
- https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
- https://scikit-learn.org/0.16/modules/generated/sklearn.linear_model.LogisticRegression.html
- https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html
- https://scikit-learn.org/stable/modules/neighbors.html
- https://www.sciencedirect.com/topics/computer-science/logistic-regression

8. Appendix

The code used for the project lies in this GitHub repository. - Link