Individual Final Report
Madhuri Yadav

For the final project, I worked mainly on Data Modeling and little of feature engineering.

Feature Engineering:

Release_Month column: Since release month is correlation with no of movies released. For example, we observe lot of movies released during December and least in January and February. This might be due to Holidays and also because movies released in December can be nominated for Oscar and the crew can use this factor for Promotion and many other such reason contribute for reason number of movies released varies with months. To extract Release_Month from Release_Date I converted string Release_Date values to date time format, then I extracted the Month to column called Release_Month and converted it to category type for later use in Modeling.

Modeling:

After preprocessing and EDA was done I worked on Modeling and Prediction. Before applying the model on our dataset 1 performed following steps:

1. Data Splitting: I 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 I stratify the sampling. Here I have used RandomOverSampler method.
3. Label Encoding: To deal with the categorical values I use one-hot encoding using LabelEncoder() in our code.
4. Scaling: To deal with the numerical values in the dataset I used MinMaxScaler(). The formula for MinMaxScalar is as follows.
   Xsc = (X-Xmin)/(Xmax - Xmin)

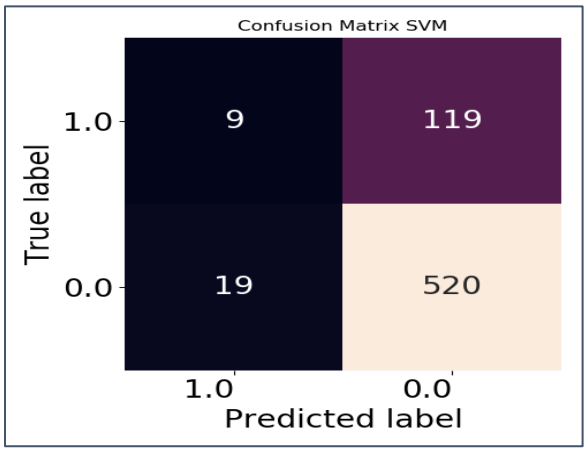
Initially I started with 4 features 'runtime', 'averageRating', 'Genre', 'Production_Company' and I applied DT(Gini), DT(Entropy), SVM, KNN, NB classification models on our data. Following are the results.

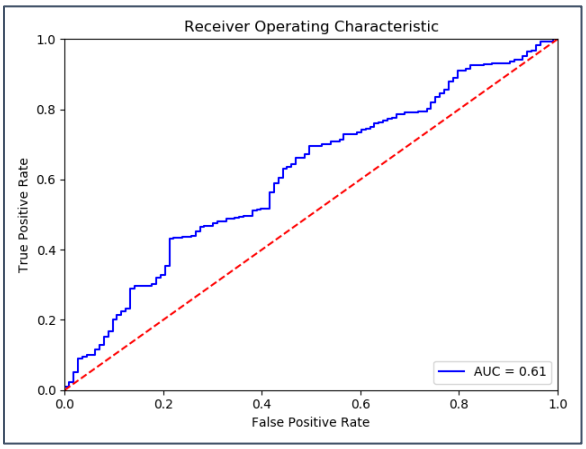
Model	Accuracy
Accuracy DT Entropy	69.26%

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)

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





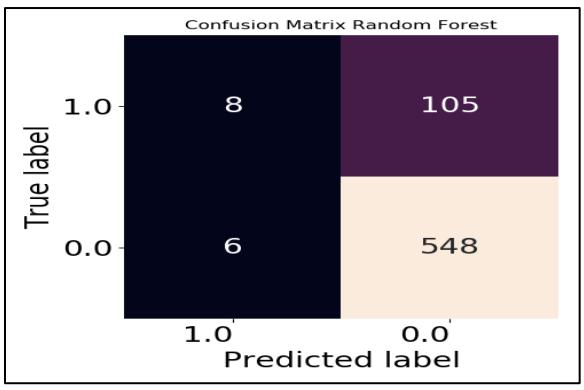
From the confusion matrix it is observed that 119 values are still incorrectly labeled as 0 This might be due to Biased Label.

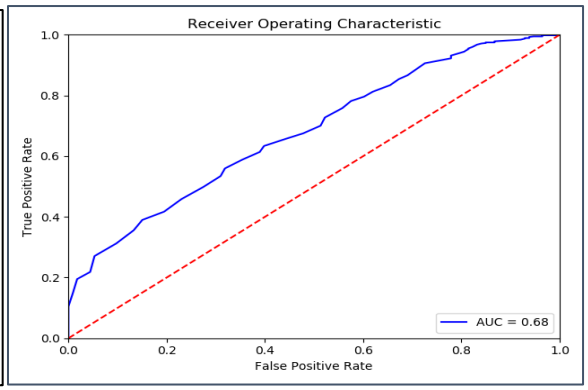
To further improve the model, I used all 7 features which were relevant to be used. And this was the exhaustive list with our dataset.

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%

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





From AUC I saw that value no was increased from 61% to 68% But I 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 I got better accuracy in DT, SVM and RF we use Hard Voting on the three samples, but I observe that it did not contribute much to the improvement.

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

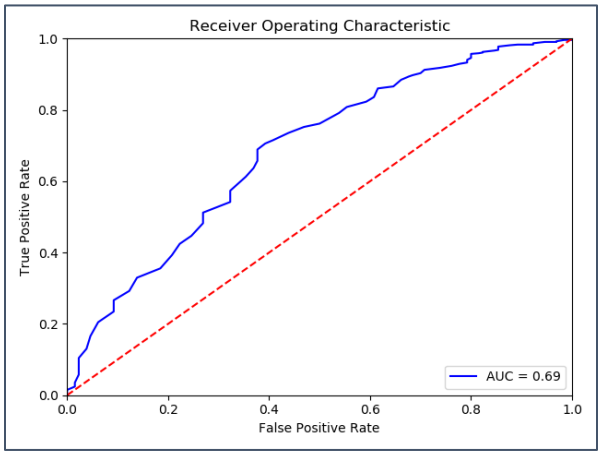
Then I performed Adaptive Boosting on random forest.

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

And following are the results.

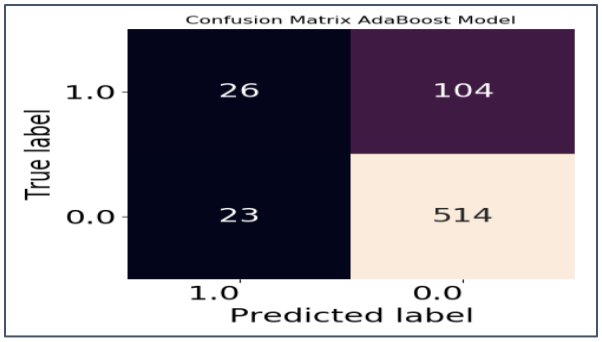
Model	Accuracy
Decision Tree (Entropy)	72.56%
Support Vector Machine	69.12%
Random Forest	79.91%
Bagging (Hard Voting)	78.41%
Adaptive Bootstrap	80.96%

I 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 analysis Adaptive Boosting on random forest fits better than other models I trained. However, might be because of biased label results are not satisfactory. But the model is still better than probability (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

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