Correlation between Movie Ratings and Gross Revenue

Group 3 – Data Miners | Project 2

Shannon School of Business

Cape Breton University, Sydney, NS, Canada

MGSC-5126-11: Data Mining
Professor: Jamileh Yousefi

Submitted by-

Ajay Pratap Singh Rathore (0242630)

Nitin Rosario Fernandes (0251496)

Ryan Philip Alfabeto (0251738)

Manik Sood (0242207)

Abstract

In the world of entertainment, everyone wants to know how well a movie will perform at the box-office even before its release. With machine learning, it is possible to find a solution for this problem if the gross revenue of a movie is correlated with other variables. In this paper, our goal was to check if there is indeed correlation between movie ratings, reviews, and other variables such as runtime, year of release etc. to the gross revenue that a movie will obtain. The dataset that we considered included information from the top 1000 rated movies on IMDb. Before building our models, we found that critics' ratings don't have much of an impact on the revenue of movies. We have used classification algorithms such as k-NN and Adaboost to check for optimal performance after discretizing our target variable based on median. With Adaboost on R, we were able to correctly predict almost 85% of the time if a movie will bring in revenue below, or equal or above, the median gross revenue in the dataset.

Introduction

Movies have always been a great source of entertainment for people across generations. If the movie is good, you tend to enjoy it; if it is bad, you are likely to dislike it. However, there have been a few movies that have generated higher gross revenues when compared to others. There could be several factors that influence the gross revenue – like the actors and actresses working in the that movie, the director(s), producer(s), with reviews, ratings, runtime etc. also playing a part. As there are so many variables, there is no definite formula that can be used to estimate how much money a given film will generate when it releases.

Having said that, by evaluating the income generated by prior films, a model may be created that can be used to estimate the projected revenue for a certain film. Using that, movie

producers, directors, actors, and other stakeholders can use these predictions to make better decisions.

With gross revenue being a continuous variable, most people might look at building a regression model, but we have used classification models.

To make the prediction (if a movie will have revenue equal or more than the median value of the dataset), we will use the k-NN (k-Nearest Neighbor) algorithm. Once we obtain our results, we will compare it with another algorithm, which is Adaboost. To achieve our goals, we will use R programming language and compare the results with Weka.

Literature Review

In an interesting research on the importance of reviews, Suman Basuroy, S. Abraham Ravid and Subimal Chatterjee examined how film critics affect box-office revenue. It showed that both positive and negative reviews influence weekly for the first two months as negative reviews tend to diminish with time. This research used the hypothesis that critics are judged as influencers, predictors, and both, and to test the hypothesis, multiple regression was performed on a weekly basis. (Basuroy et al., 2003)

In research – Word of Mouth for Movies, Its Dynamics, and Impact on Box Office Revenue – the authors Wenjing Duan, Bin Gu and Andrew B Whinston focus on how the Word of Mouth (WOM) influences the revenue of a film at the box-office. WOM becomes more crucial during the pre-release phase and in the first few weeks. The research used empirical analysis and a framework where WOM is considered before and after the release on a weekly basis. WOM contains ratings and reviews from moviegoers, social media influencers, casts, and various critics. WOM depends on its volume to influence a movie's gross revenue, weekly or aggregated. (Duan et al., 2008)

Another research – Movie Revenue Prediction Based on Purchase Intention Mining Using YouTube Trailer Reviews – by Ibrahim Said Ahmad, Azuraliza Abu Bakar and Mohd Ridzwan Yaakub, focuses on the relationship between the reviews for movie trailers and the movie's revenue. The core of research here is the use of social media data of trailer reviews and its influence on the ticket sales for a movie before its release. The data was gathered from IMDb, Twitter and YouTube and they used purchased mining algorithm and sentimental analysis. This is useful for movie makers to make any last-minute modifications when the trailer has received many negative reviews. (Ahmad et al., 2020)

In a research published just last March by Abhishek Singh, Abhishek Rawat, Shanmukh Rao, Samyak Jain and Uppalpati Yogendra Reddy, a movie recommendation system was made based on k-NN. They felt that customers were having a hard time searching and picking a film from the large number of movies available these days. This was created using collaborative filtering algorithm and k-NN (as both are hybrid-based approaches) at the same time and they feel that such a recommendation technique can help users select a movie to watch. (Singh et al., n.d.)

In addition, a similar paper published by the team of Rishabh Ahuja, Arun Solanki and Anand Nayyar also used k-NN to recommend movies. They believe that a recommendation system will be like a filtering system used to predict the "rating" and "preference" a user will give to a movie. (Ahuja et al., 2019)

BeiBei Cui designed and implemented a movie recommendation technique based on k-NN and collaborative filtering algorithms. It was to build a system to find how quickly a user would be able to find a movie which they are likely to watch. The use of such a personalized recommendation system can really play an important role, especially if the user has no clear target movie in mind. (Cui, 2017)

Julia Nikulski managed to detect features to predict the revenue of a movie. Using Adaboost algorithm, she found out that these important features namely – popularity, budget, year of release, runtime, and length of tagline – make up as big factors in the resulting revenue of a film. (Nikulski, 2019)

Likewise, in the movie recommendation system published by Prabhakar Eswaran in February 2020 using the Adaboost algorithm, it was noted that ratings determine a viewer's interest in a certain movie, which would eventually lead to an increase in the movie's revenue. (Eswaran, 2020)

Methodology

The dataset – **IMDB Movies Dataset –** was taken from Kaggle, where it is freely available.

The data set consists of 1000 records – the top 1000 movies as rated on the Internet Movie Database (IMDb). It's important to note all these movies are among the best films ever made, if you go by their ratings on IMDb.

The attributes in our dataset included:

- Poster Link – Link of the poster on IMDb
- Series Title – Name of the movie
- Released Year – The year in which the movie was released
- Certificate – The age restriction on the movie
- Runtime – Total runtime of the movie
- Genre – Genre of the movie
- IMDB Rating – Rating by the public on IMDb (out of 10)
- Overview – Summary of the movie

- Meta score – Average reviews by critics (out of 100)
- Director – Name of the Director
- Star1,Star2,Star3,Star4 – Name of lead actors and actresses
- No of votes – Total number of votes on IMDb
- Gross – Revenue of the movie

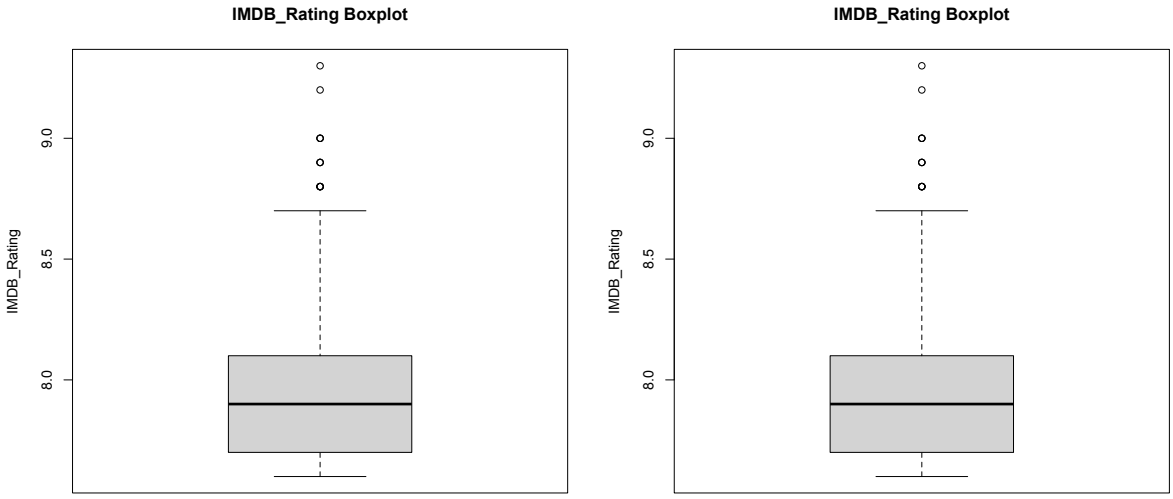
For public reviews, we considered ratings on the IMDb platform, where anyone is free to create an account and give ratings to a movie (IMDb, n.d.). Movies on IMDb are rated out of a maximum rating of 10. On the other hand, for critics' reviews, we considered Metascore – it is a score given by Metacritics after considering a weighted average from a combination of ratings from some of the world's top film critics (Frequently asked questions - metacritic, n.d.). A Metascore can have a maximum value of 100.

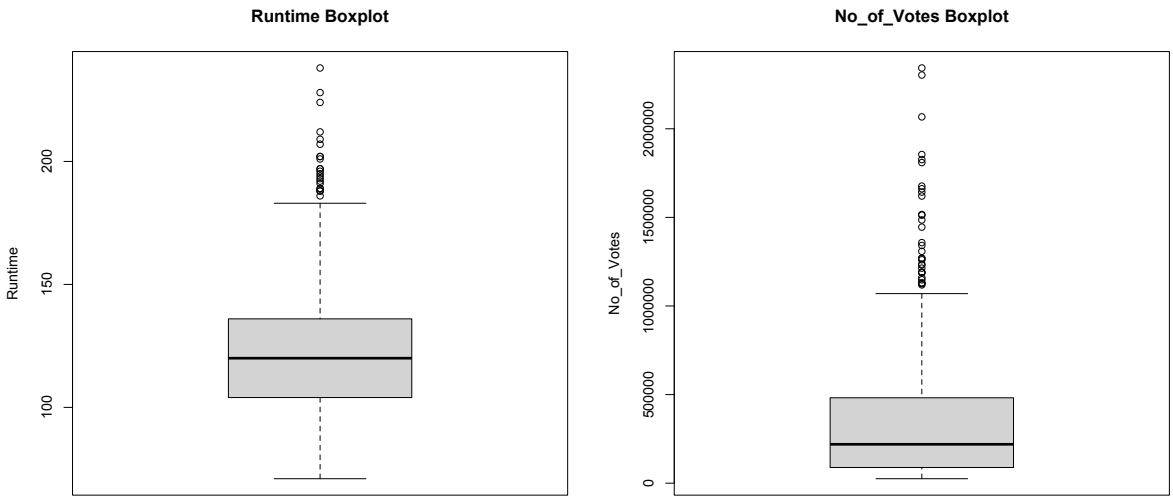
We started with data pre-processing. In this step, we checked the summary of all our variables to check for any issues. We noticed that Runtime, Released_Year and Gross have values in the form of characters, but we needed them to be in numeric form. Hence, we made the change. In addition to that, we also noticed that there were no fake values in the dataset.

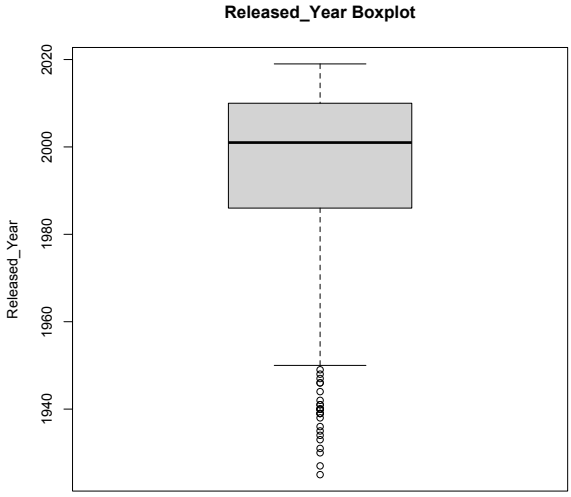
After that, we omitted all the rows which had empty values in our data frame. Here, we lost 250 records.

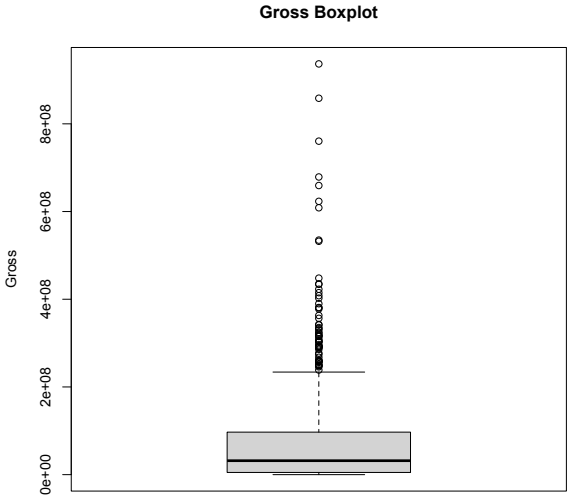
Figure 1 - Checking for null values

We then experimented with boxplots to check for extreme outliers. While we did find outliers in various variables, we decided against deleting or working on them as those values seem realistic and will be important in determining the results of this study.





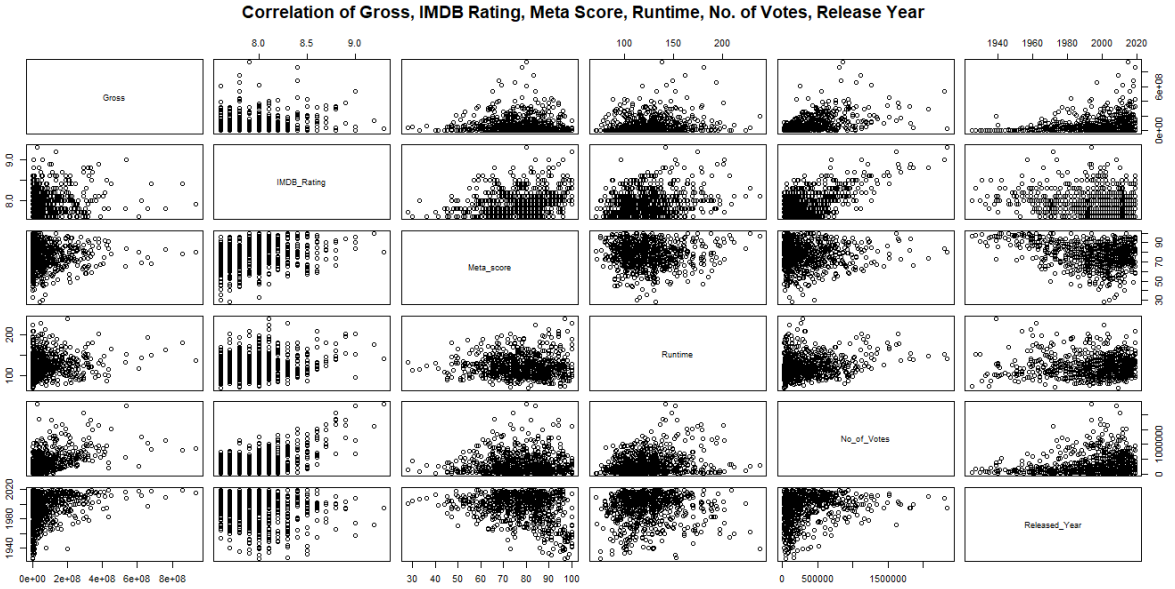




The numeric data in different columns in our data frame had differing ranges and this could cause problems later. To avoid that, we used min-max normalization to bring all the data under the same range. The numeric variables that we normalized were Runtime, IMDB_Ratings, Meta_score, Released_Year, No_of_Votes and Gross.

After the data was prepared, we checked for correlation between the input variables and the target variable which is Gross. The result was as follows:

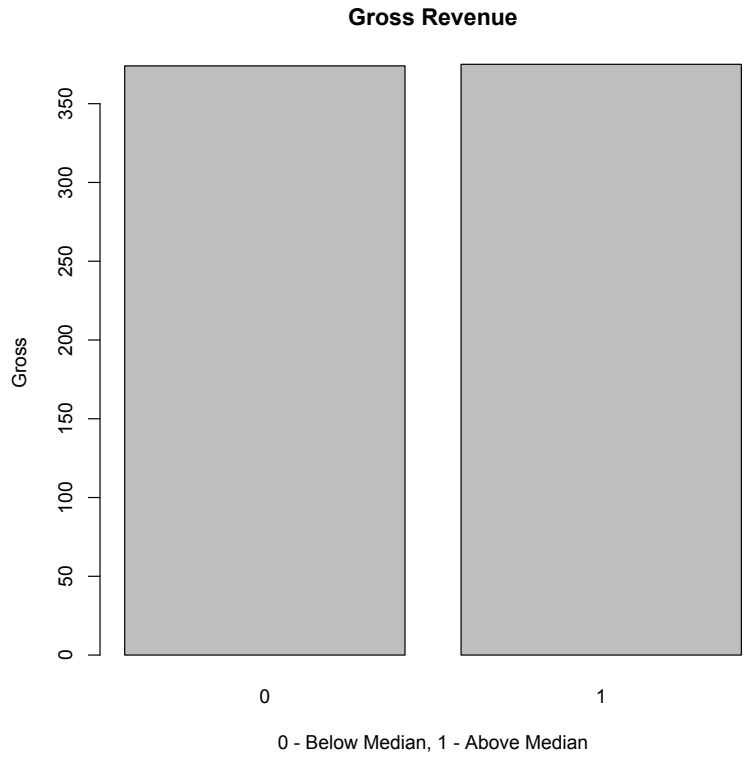
	Gross
Runtime	0.17222646
IMDB_Rating	0.12937777
Meta_score	-0.03055968
Released_Year	0.23595150
No_of_Votes	0.55600264



Here, we can see that No_of_Votes has the highest correlation with the target variable and this makes sense – more votes indicate that more people have seen the film, which in turn contributes to gross revenue. On the other hand, Meta_score has very low correlation with the target variable (interesting to note that critics' reviews haven't affected revenue much when it comes to the top 1000 movies on IMDb). Hence, we have decided to drop the Meta_score variable before we proceed with further analysis.

We then decided to create a new data frame which contained only variables which had numerical data. While we could have converted categorical data to binary using dummy variables, we have too many values under each categorical variable, and this would be a tedious process. Therefore, we decided to drop all these columns.

Since our target variable (Gross) was a continuous variable and as we did not want to use Linear Regression, we proceeded to use discretization. We did so by calculating the median of Gross (the midpoint) and assigning 0 to values below the median and 1 to values equal to or above the median. The median was found to be $31,800,000 (after normalization, it's 0.03394899).



Hence, **the goal of our research was to predict if a movie's gross revenue will be equal to or above the median ($31,800,000) when compared to the top 1000 movies on IMDb**.

Following this, we moved on to creating our models using k-NN and Adaboost algorithms.

Our primary algorithm is k-Nearest Neighbors (k-NN). This is a supervised learning algorithm. In k-NN algorithm, the values are separated into different classes in order to predict

the classification of the target variable (Sutton, 2012). Here, the k is the number of neighbors that you require around a particular data point for prediction.

On the other hand, Adaboost is an algorithm using which we can create a highly accurate prediction rule from a combination of relatively lesser accurate rules (Schapire, 2013). Adaboost principle has attracted researchers because of its good performance in the classification of learning. Its algorithm itself has unique advantages, like the improvement of sample distribution by mistake ratio of classifiers which will focus on hard sample.

Discussion

Before building our model, we need to split the data into training and testing. We have used a 75/25 split – with 75% of the data in training and 25% for testing.

Once we built and ran the model in R, we obtained the following confusion matrix:

	target_test	
knn_model	0	1
0	77	24
1	21	66

From the above results, we can notice that 77 0s (below median of Gross) and 66 1s (equal to or above median) have been predicted correctly. Those numbers are our True Negatives and True Positives respectively.

We can calculate accuracy, precision and recall from the above table. While accuracy gives us the correct predictions divided by the total number of predictions, the formula for recall and precision are as follows:

$$\text{Recall} = \frac{TP}{TP+FN}$$

$$\text{Precision} = \frac{TP}{TP+FP}$$

Figure 2 - (Davis & Goadrich, 2006)

The model has an accuracy of 76.06%, which means it is able to correctly predict if a movie's gross revenue will be equal to or more than $31,800,000 (median) or below it accurately 76.06% of the time. It is able to correctly predict the result for 75.86% of the actual movies whose gross is equal to or above the median (recall), while the precision is 73.33%.

We also ran our dataset on Weka to check if we would get similar results. We followed the same 75/25 split. But there was a slight difference in the number of records in the test data here. While in R, the testing dataset contained 188 records, here it was one less at 187 (this is happening because 25% of 750 is 187.5).

The confusion matrix that we obtained from the k-NN algorithm on Weka was as follows:

```
 a  b   <-- classified as
71 22 |  a = Below
19 75 |  b = Equal or Above
```

From the above results, we can notice that 71 0s (below median of Gross) and 75 1s (equal to or above median) have been predicted correctly. As mentioned earlier in this section, we can calculate accuracy, precision and recall from the above matrix.

The model has an accuracy of 78.07%, which means it is able to correctly predict if a movie's gross revenue will be equal to or more than $31,800,000 (median) or below it

accurately 78.07% of the time. It is able to correctly predict the result for 79.78% of the actual movies whose gross is equal to or above the median (recall), while the precision is 77.32%.

Interestingly, when compared to our model on R, we have got better results on Weka while using k-NN Classifier.

After building our primary model using k-NN, we would like to compare our above results with a second algorithm, and for that, we have selected Adaboost.

Once we built and ran the model on R, we obtained the following confusion matrix:

	Observed Class	
Predicted Class	0	1
0	78	14
1	15	79

From the above results, we can notice that 78 0s (below median of Gross) and 79 1s (equal to or above median) have been predicted correctly. Following this, we calculate the accuracy, recall and precision.

The model has an accuracy of 84.41%, which means it is able to correctly predict if a movie's gross revenue will be equal to or more than $31,800,000 (median) or below it accurately 84.41% of the time. It is able to correctly predict the result for 84.95% of the actual movies whose gross is equal to or above the median (recall), while the precision is 84.04%.

When compared to our primary algorithm, Adaboost definitely has the better performance. An accuracy of close to 85 is a very good score.

Just like we did with the k-NN algorithm, we also ran our dataset using the Adaboost algorithm on Weka to check if we are able to get similar or better results. The confusion matrix that we obtained from the k-NN Classifier on Weka was as follows:

```
 a  b   <-- classified as
72 21 |  a = Below
17 77 |  b = Equal or Above
```

From the above results, we can notice that 72 0s (below median of Gross) and 77 1s (equal to or above median) have been predicted correctly. Following this, we follow the usual process of calculating accuracy, recall and precision to check the performance of the model.

The model has an accuracy of 79.67%, which means it is able to correctly predict if a movie's gross revenue will be equal to or more than $31,800,000 (median) or below it accurately 79.67% of the time. It is able to correctly predict the result for 81.91% of the actual movies whose gross is equal to or above the median (recall), while the precision is 78.57%.

Unlike the k-NN algorithm, we have obtained better results using Adaboost on R rather than on Weka.

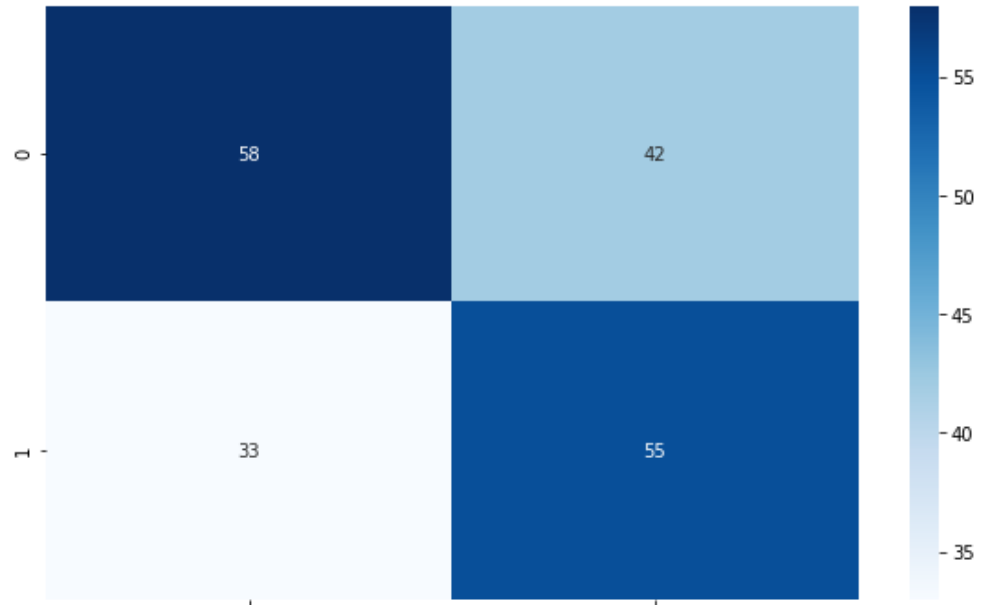
Comparing the performance of the two algorithms on R and Weka.

Algorithm	**k-NN (on R)**	**k-NN (on Weka)**	**Adaboost (on R)**	**Adaboost (on Weka)**
Accuracy	0.7606383	0.780749	0.844086	0.796791
Recall	0.7586207	0.797872	0.8494624	0.819149
Precision	0.7333333	0.773196	0.8404255	0.785714

The Adaboost algorithm on R programming language gave us the best results across accuracy, recall and precision.

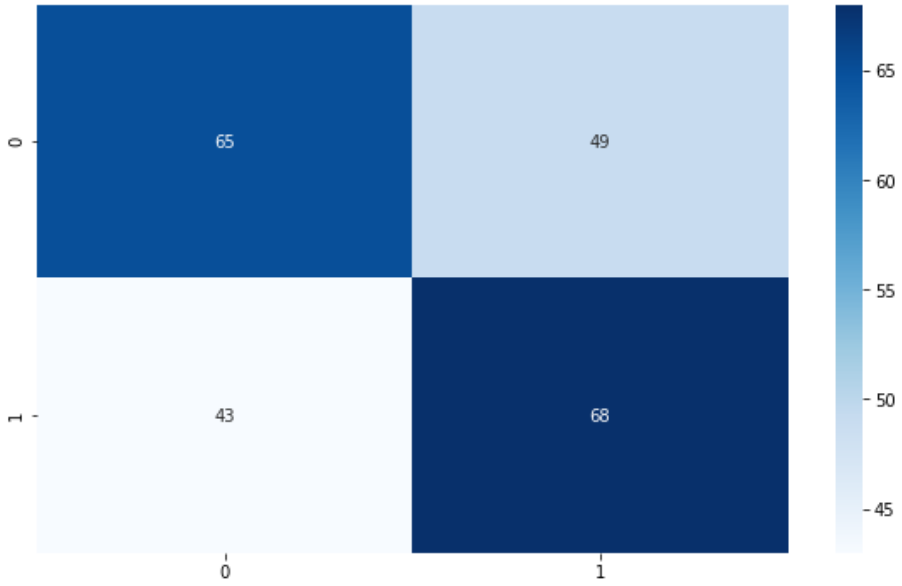
In addition to R and Weka, we also tried to run the above-mentioned algorithms on Python. The results there, though, weren't as good.

On Python, with k-NN algorithm, the confusion matrix was as follows:



As we can notice above, there are a lot of False Positives and False Negatives which is not ideal. The accuracy that we were able to get here was 0.601064, while the recall was 0.625000 and the precision was 0.567010. These results pale in comparison to our k-NN algorithm models on R and Weka.

Then, we proceeded to attempt Adaboost algorithm on Python as well. For this, the resultant confusion matrix was as follows:



Here, the accuracy that we got was 0.591111, while the recall was 0.612613 and the precision was 0.581196. Once again, these results are not as good when compared to our Adaboost algorithm models on R and Weka.

We also tried experimenting with Naïve Bayes Classification on Python, but again, the results were poor, so we didn't consider it.

Conclusion

The goal of our project was to find if there is a correlation between movie ratings and other such variables, and the gross revenue of a movie. After working with multiple algorithms, we can safely say that, considering the top 1000 movies on IMDb, there is a good amount of correlation. Due to this, we were able to predict the class of the gross revenue of a movie (below median/equal to or above median) based on a few input variables – in this case, it was the IMDb rating, runtime of the movie, the year the movie was released and the number of votes the movie has got on IMDb.

While we experimented with more than one algorithm and on multiple software, the best results that we obtained were through the Adaboost algorithm on R. With this method, we got an accuracy of 84.41%, with a high recall value of 84.95% and precision at 84.04%. Using this algorithm, we can be confident of predicting the gross revenue of a movie when we consider median gross revenue.

Therefore, if we were to choose an algorithm for this analysis, we would reject our original choice – k-NN algorithm – in favour of the Adaboost algorithm.

References

Ahmad, I. S., Bakar, A. A., & Yaakub, M. R. (2020). Movie revenue prediction based on purchase intention mining using YouTube trailer reviews. *Information Processing & Management*, *57*(5), 102278. https://doi.org/10.1016/j.ipm.2020.102278

Ahuja, R., Solanki, A., & Nayyar, A. (2019). Movie recommender system using K-means clustering AND K-nearest neighbor. *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*.

Basuroy, S., Chatterjee, S., & Ravid, S. A. (2003). How critical are critical reviews? The box office effects of film critics, star power, and budgets. *Journal of Marketing*, *67*(4), 103–117. https://doi.org/10.1509/jmkg.67.4.103.18692

Cui, B.-B. (2017). Design and implementation of movie recommendation system based on knn collaborative filtering algorithm. *ITM Web of Conferences*, *12*, 04008. https://doi.org/10.1051/itmconf/20171204008

Davis, J., & Goadrich, M. (2006). The relationship between Precision-Recall and ROC curves. *Proceedings of the 23rd International Conference on Machine Learning - ICML '06*.

Duan, W., Gu, B., & Whinston, A. (2008). The dynamics of online word-of-mouth and product sales—An empirical investigation of the movie industry. *Journal of Retailing*, *84*(2), 233–242. https://doi.org/10.1016/j.jretai.2008.04.005

Eswaran, P. (2020). Movie recommendation system using data mining. *AICTE Sponsored International Conference on Data Science and Big Data Analytics for Sustainability*.

Frequently asked questions - metacritic. (n.d.). Metacritic.Com. Retrieved April 11, 2022, from https://www.metacritic.com/faq

IMDb. (n.d.). Imdb.Com. Retrieved April 11, 2022, from https://help.imdb.com/article/imdb/track-movies-tv/ratings-faq/G67Y87TFYYP6TWAV

Nikulski, J. (2019, August 19). *Predicting movie revenue with AdaBoost, XGBoost and LightGBM*. Towards Data Science. https://towardsdatascience.com/predicting-movie-revenue-with-adaboost-xgboost-and-lightgbm-262eadee6daa

Schapire, R. E. (2013). Explaining AdaBoost. In *Empirical Inference* (pp. 37–52). Springer Berlin Heidelberg.

Singh, A., Rawat, A., Rao, S., Jain, S., & Uppalpati, Y. (n.d.). *A research paper on machine learning based movie recommendation system*. Irjet.Net. Retrieved April 11, 2022, from https://www.irjet.net/archives/V8/i3/IRJET-V8I3205.pdf

Sutton, O. (2012). *Introduction to k nearest neighbour classification and condensed nearest neighbour data reduction*. Le.Ac.Uk. http://www.math.le.ac.uk/people/ag153/homepage/KNN/OliverKNN_Talk.pdf