MOVIES: Predicting IMDB Scores and Gross Sales

Project Goals

IMDB.com is one of the most trusted sites for movie ratings both by critics and the users equally. Our team decided we would use text mining, linear regression and random forest regression to predict IMDB score. And because ratings are only one measure of movie success, we decided to also see how effective linear regression and random forest are in predicting gross sales. We compared the accuracy of each algorithm in predicting the IMDB score and the gross sales of movies to see which worked best on these two different measures of success.

Introduction to the IMDB dataset

The dataset being used initially consisted of 5000 rows and 28 fields that has data like movie names, director, actors and their corresponding likes, critic reviews, plot keywords, genres, languages, countries, budget, gross etc.

Data Cleansing

As the first step of our analysis, we realised that data has many different kinds of anomalies and in order to build our model, we have to get rid of those. Following are the steps we took in order to clear up the data.

- **Removed duplicates:** There were some rows that are repetitive in the dataset including all 28 columns in those duplicate rows.
- **Imputing NAs:** Some of the columns had NA values in them, so in order remove those NAs in such a way that the statistics of data remains the same, we have imputed mean, median and mode, whatever was least changing the data structure.
- **Clean Text:** Some of the columns that had character values like title of the movie and keywords associated with each movie had to be cleaned of the undesired characters such

as ‘Â’ and ‘|’. With the help of the ‘stringr’ package and the ‘™’ package we were able to execute that task.

- **Outliers:** Some columns had values that were way too large in comparison with the whole distribution of that column. For example, budget had one single outlier that was skewing the mean of the distribution towards the right. Similarly, we got rid of columns that were using their domestic currency to represent money rather than standard USD, for example India had budget in Rupees and Japan had budget in Yen.

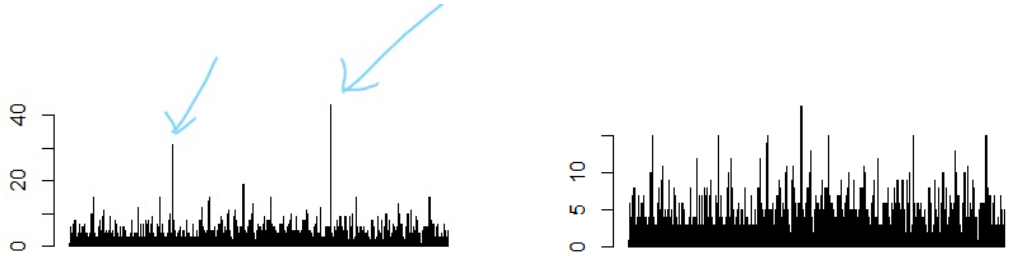

Figure 1 Removing Outliers: Before and After

- We also removed columns that were not consistent with the whole data. In Figure 2, both these charts represents that there is something wrong with director facebook likes. So removing them is was a right option.
- Some columns such as gross sales, budget, Facebook likes spanned many orders of magnitude. We created the natural log of these values as an additional variable.

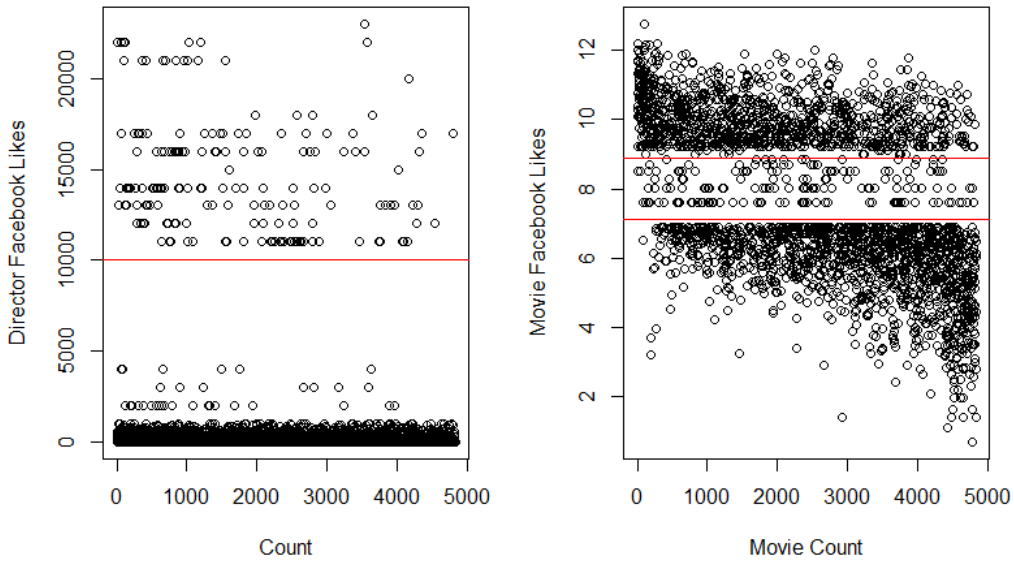

Figure 2 Director Facebook Likes are two-tiered and Movie Facebook likes are missing data

Data Exploration

Be it that a actor is starring in a movie or playing a side role as actor 2 or actor 3, in Figure 3 we see that the facebook likes of each category of actor has median line almost on a same position in distribution. However, actor 3 values more spread out and actor have more concentration values around median.

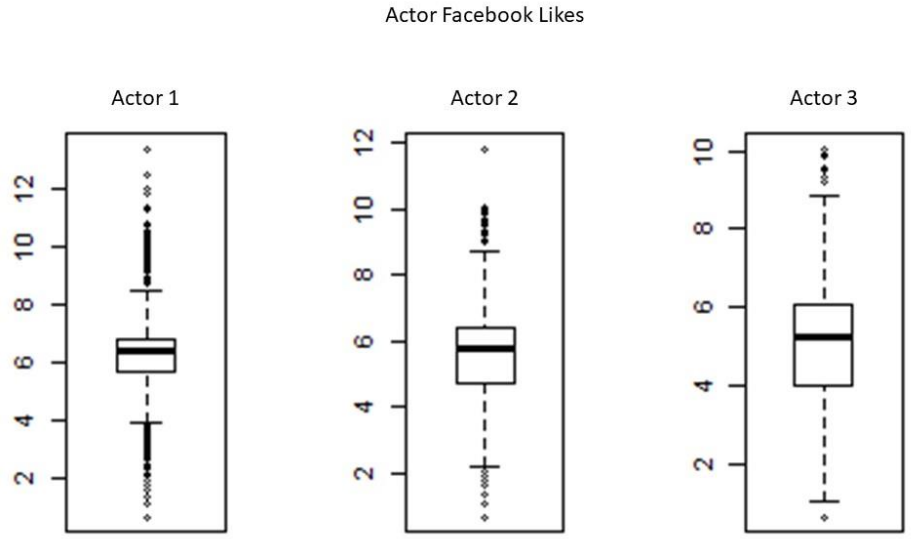

Figure 3 Boxplot statistics of Log(Facebook likes) of Actor 1, Actor 2 and Actor 3 of 5000 movies

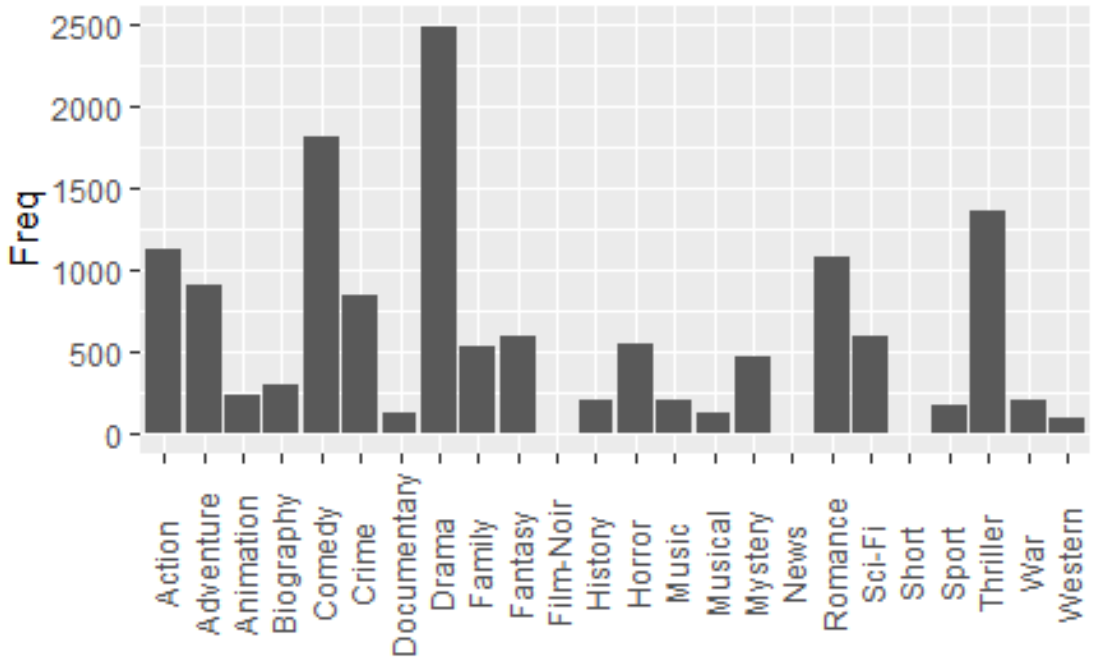

Figure 4 Movie Counts by Genre

Figure 4 shows the counts of genres the movie belongs to. We can clearly see that “Drama” genre have highest number of count followed by comedy and thriller.

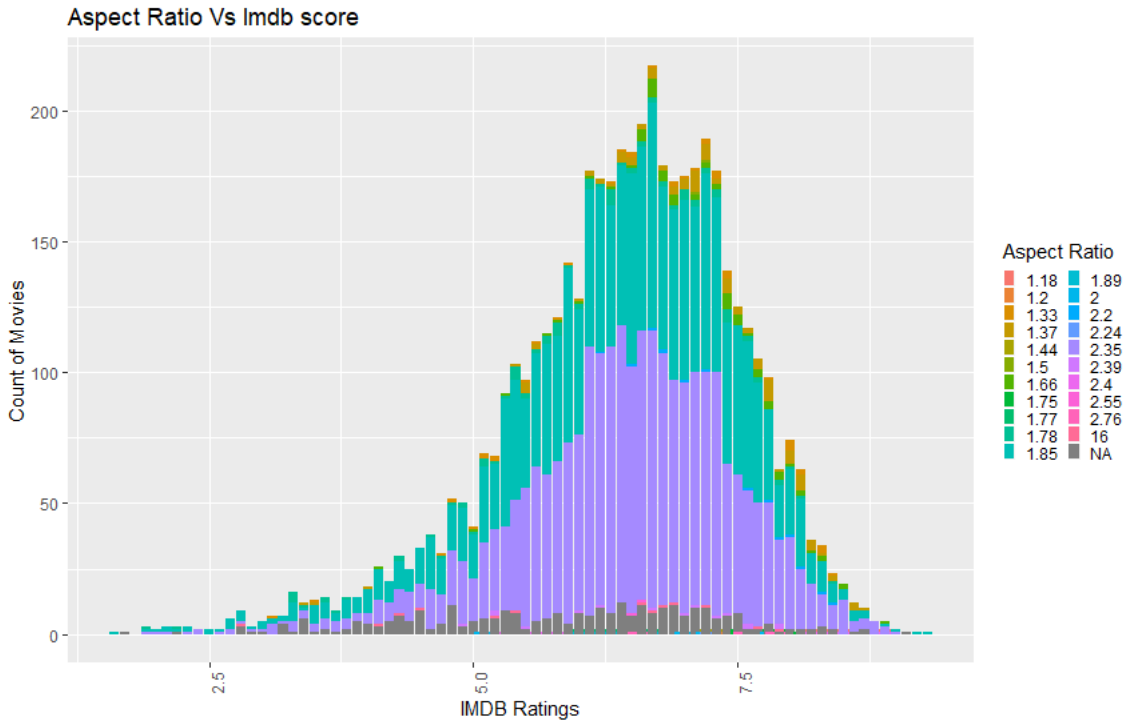

Figure 5 Movie Count by IMDB scores, stacked by Aspect Ratio

In Figure 5, we see that most of the movies have an aspect ratio of 1.85 and 2.35. Because these two categories represent most movies across all the IMDB ratings, we can remove the Aspect Ratio as an input to the models.

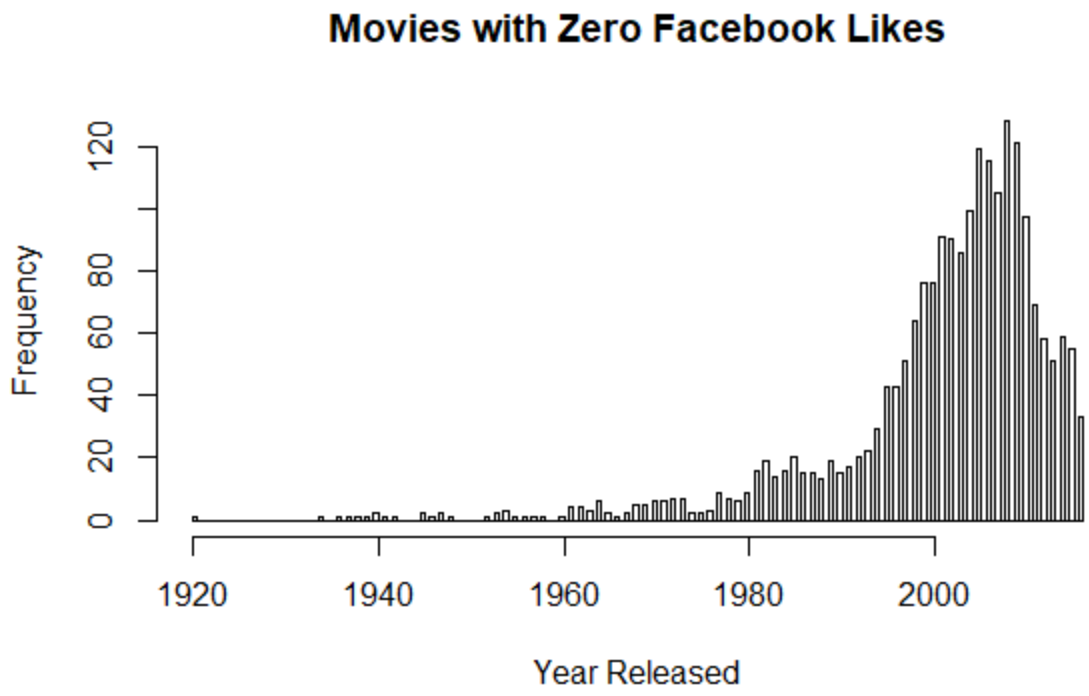

Figure 6 Movies with Zero Facebook likes peak around the year 2000

Figure 6 represents the count of movies that have zero facebook likes. We can clearly see on x-axis that the movie are centered around year 2000. That made us question how come movies in the past have facebook likes even though facebook is invented in modern time, but movie around 2000 are not given any facebook likes. With that suspect in mind, we removed the columns from the data frame.

IMDB vs Gross

In Figure 7, we see that the mean line is getting steeper as the rating of the movies are increasing from top to bottom. The line represent mean value of gross sales for all movies in that category. In simple words, if a movie has higher IMDB rating then there is a better chance that it would have higher gross sale.

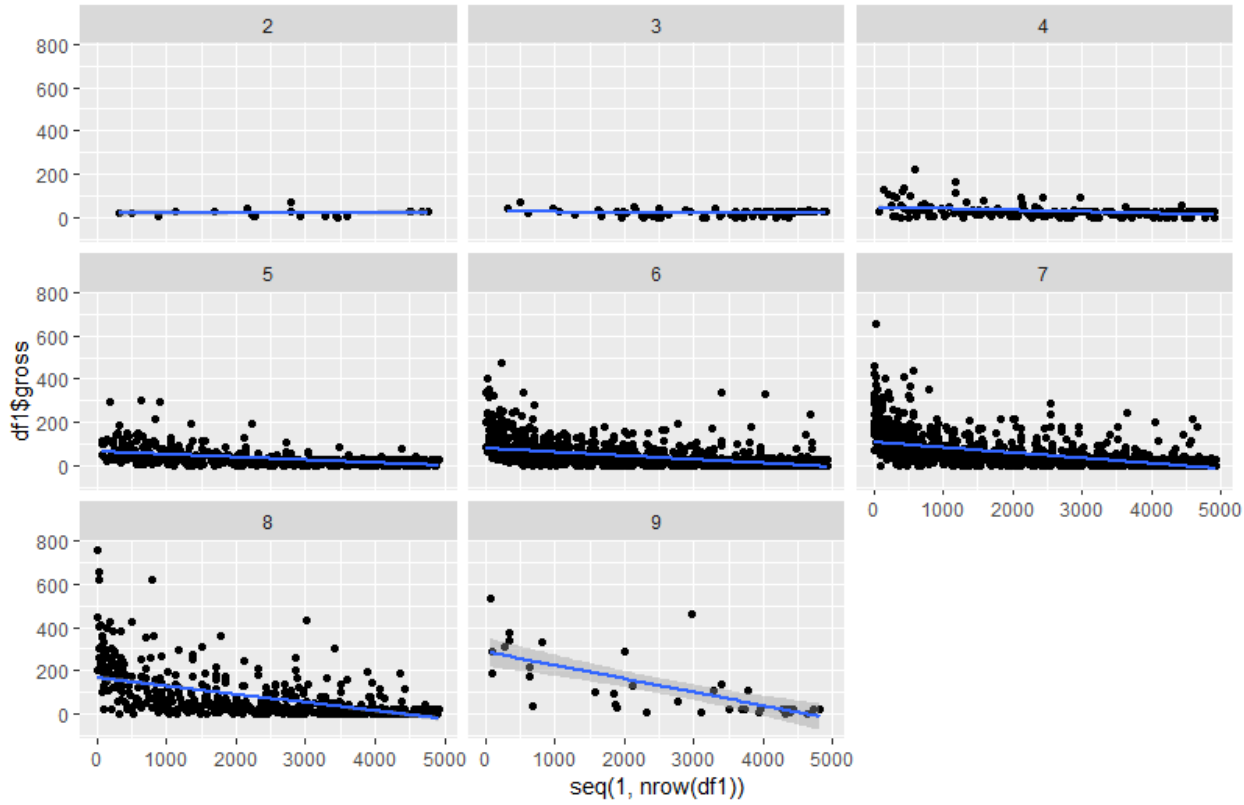

Figure 7 Scatterplots Movie Counts to Gross Sales ($M) by IMDB score category (*e.g* 3 = 2.5 to 3.49)

Correlation Matrix

The correlation matrix in Figure 8 calculates and visually presents the correlation of each variable to all of the others. The figure below substitutes the natural log of six values for their raw numbers. The correlation matrix shows that these six categories of values are highly intercorrelated:

- log(gross sales)
- log(budget)
- log(number of users who voted)
- log(number of users who voted for the movie)
- log(total cast Facebook likes)
- log(movie Facebook likes)

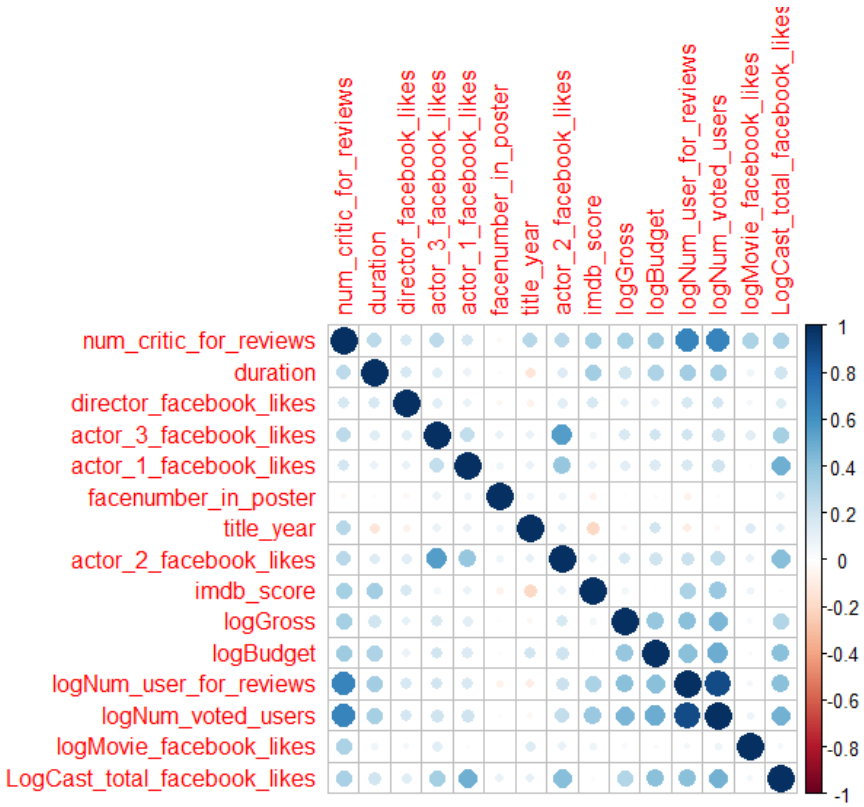

Figure 8 Correlation Matrix of log or raw values of IMDB movie statistics

Values such as director, actor 1, actor 2, actor 3 and number of faces in the movie poster have weak to no correlation to IMDB score or gross sales. Notably, the IMDB score is most, but not very correlated to number of critics for, number of users for and number of users voted. IMDB score is *definitely not* correlated to the budget and the gross sales of the movie.

While correlations help identify inter-relationships between variables, they do not necessarily identify the most important drivers in some of the predictive algorithms we used.

Algorithms

Model Training and Testing

For each predictive algorithm, we randomly divided the 4,836 row data set so that 75% of the rows are used to train the algorithm and 25% are used to test the predictive accuracy of the trained algorithm. We built and ran the model on the train dataset, then made predictions with the test dataset, then compared the predictions to the actual values. Using the confusion matrix we summed the diagonal (true prediction) and divided by the total number of rows in the test dataset to arrive at the accuracy of the model. Using this common measure of accuracy, we compared the ability to predict IMDB score using text mining of plot keywords, linear regression and random forest techniques. We then compared the ability to predict gross sales using linear regression and random forest algorithms.

Linear Regression

The Linear Regression model, the basic and the most commonly used technique, is a modelling approach between the independent variables and the dependent variables.

We, tried using linear regression to predict the IMDB scores and the Gross sales for the movies.

IMDB Score:

For the IMDB Score, we use the following formula to fit into the linear model,

```
linearIMDB = lm(imdb_score ~ num_critic_for_reviews+duration+gross+num_voted_users
        +cast_total_facebook_likes+num_user_for_reviews,
        data = train)
```

We obtain the r-squared value of 0.27 and the F-statistic of 213.5 from this model. On the basis of p-values, we conclude that the most favorable variables for IMDB score are Critical Reviews For, Duration, Gross, Users voted for, Number of User For Reviews, Facebook likes of movies. But since, closer the value of R-squared to 1, the better the model. We reject the linear regression model for IMDB score.

Gross Sales:

We used the following formula for the linear model for the Gross sales,

model1 <- lm(gross ~ ., data = numdf1)

We obtained the r-squared value of 0.61 from this model. On the basis of p-values, we conclude that the most favorable variables for Gross sales are are Critical Reviews For, Duration, Users voted for, Number of User For Reviews, total facebook likes of the entire cast, facebook likes of actor1, actor 2 and actor 3. Here, the R-square value is better than that for the IMDB score with slightly higher R-squared value. But, it is still not a very good model and hence, we reject the linear regression model for Gross sales.

Table 1 Summary of Results of Linear Regression models predicting IMDB score

Linear Model	R2	Most Predictive Variables
IMDB Score	0.27 using all numeric, highly correlated	**Critical Reviews For** Duration Gross **Users voted for** **Number of User For Reviews** Facebook likes of movies
Gross Sales	0.60 using all numeric highly correlated	**Users voted** **Users voted for** Cast FB likes

		Actor FB likes **Critical Reviews For**

Table 1 shows that linear regression could not account for more than 27% of the variability in predicting IMDB score, nor more than 60% of the variability in log(Gross Sales). Also we can conclude that there are only a few factors that affect both the IMDB score as well as the gross sales such as Critical Reviews, User Votes and the Users Voted for the reviews.

Text-Mining using K-Nearest Neighbour for IMDB Rating

Knn is a classification algorithm that helps us determine with category does each of the element of our test data belongs to. It utilises the euclidean distance to measure how near or how far the new element is in order to classify it.

In our analysis, we have categories movies having rating 7 or less and movies that have ratings greater than 7.

Euclidean Distance :$\sqrt{(x_2 - x_1)^2 + (y_2 - y_2)^2 + (z_2 - z_1)^2 \ldots\ldots (n_2 - n_1)^2}$

With the help of tm() package we created a corpus of keywords associated with each movie by tokenizing them and converting them into a matrix. This methods creates a matrix significantly greater than the dataset we work with. In our case, for text mining we got the matrix that had more than 4000 rows and more than 5000 columns. Each columns represents a token and each row represents a single document. Since we had around 5000 movies, we say that we have 5000 documents. Each set of words associated with each movie is called a document.

Tokenization

	1 word	2 word	3 word	4 word
Document 1	1	1	0	0
Document 2	0	1	1	0
Document 3	1	0	0	1
. . . .	. . . .	. . . .	. . . .	. . . .
Document n	0	1	1	0

Figure 9 Text mining creates a document term matrix as the basis for prediction

Based on the method shown in Figure 9, we were able to get 83% accurate result in classifying the movies based on imdb score.

Random Forest

The basis of the random forest algorithm is the classification tree. It builds multiple trees and merge them together to get an accurate and stable prediction. In a single classification tree the algorithm splits one (or more) variables sequentially to create a path to a prediction. But that tree or path may be optimal or more likely, it is not a good predictor. Random forest is an ensemble algorithm that runs many trees (we used 400) and essentially takes a vote on the best path overall.

IMDB

We ran the random forest model for the IMDB Scores versus the numerical factors affecting the score. The factors considered here were number of the voted users, number of the reviews by the users, number of reviews by the critics, budget for the movie, its gross,duration of the movie, the total facebook likes for cast of movies. As a part of the process, we plot the Mean Decrease Accuracy and the Mean Decrease Gini.

The variables with a large mean decrease in accuracy are important for the classification of the data. As it is visible from Figure 10, we can conclude that number of the voted users and number of reviews by the users are the most important factors for the classification whereas the total facebook likes for cast of movies is the least important factor for the classification of data.

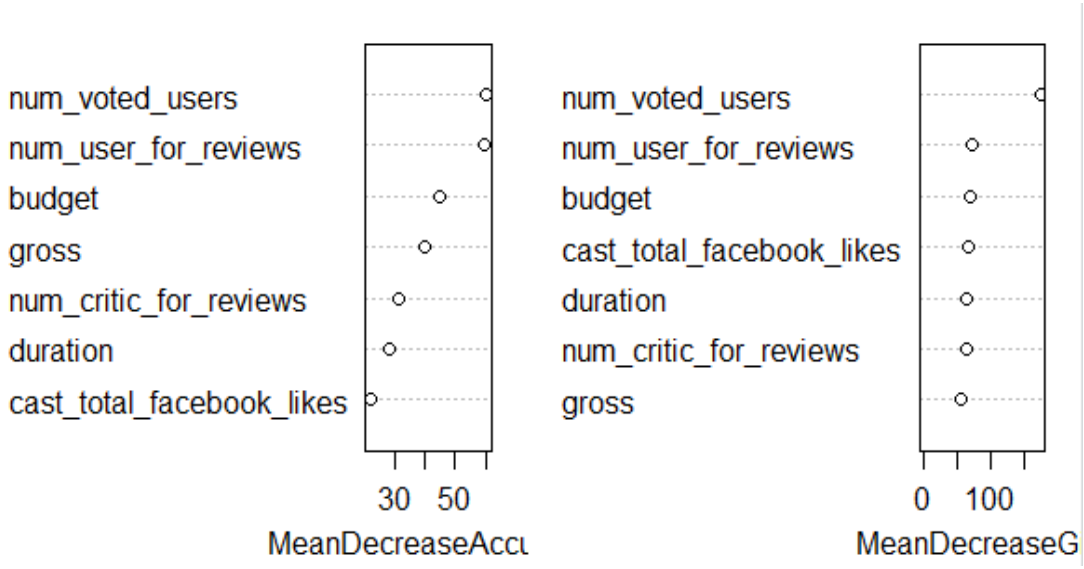

Figure 10. Plot for Mean Decrease Accuracy and Mean Decrease Gini for Random Forest for IMDB.

The Mean Decrease in Gini is the mean of various factors' decrease in node impurity. As it can be seen from the plot, the number of voted users have the highest node impurity. The other factors have almost same amount of mean decrease in gini.

The IMDB ratings are divided into bins with values 0-4, 4-8 and 8-10 inclusive. The random forest function was run for IMDB versus the number of critic reviews, number of users voted, number of users reviewed the movie, total facebook likes for the cast, gross and budget for movie for the training data set. 500 trees with 5 splits at each node. With this model we obtain the out-of-box estimate error rate of 6.33%, which means this model has the prediction accuracy of almost 94%. The command used to run the random forest model is

rf_IMDB = randomForest(binned_score ~ num_critic_for_reviews + duration + gross + num_voted_users + cast_total_facebook_likes + num_user_for_reviews + budget, data = train, ntree = 500, mtry = 5, importance = TRUE, na.action = na.roughfix, proximity = TRUE)

Table 2 shows the confusion matrix generated from the random forest for IMDB for the predicted as well as actual values, where the diagonal values are the values predicted correctly. As it is seen from the table, the bucket for the 0-4 IMDB score has the highest number of the error rate, as it predicted, 124 of 129 incorrect with an error rate 96%.

	(0,4]	(4,8]	(8,10]	class.error
(0,4]	5	123	1	0.961240310077519
(4,8]	4	3348	19	0.00682290121625628
(8,10]	0	85	82	0.508982035928144

Table 2. The confusion matrix for random forest for IMDB Score bins.

The bucket of 4-8 bucket predicted with the highest accuracy with 0.68% error, whereas the bucket with 8-10 bucket has error rate of 50% with 82 out of 167 correct predictions. Below Figure 11 is the graph for the confusion matrix.

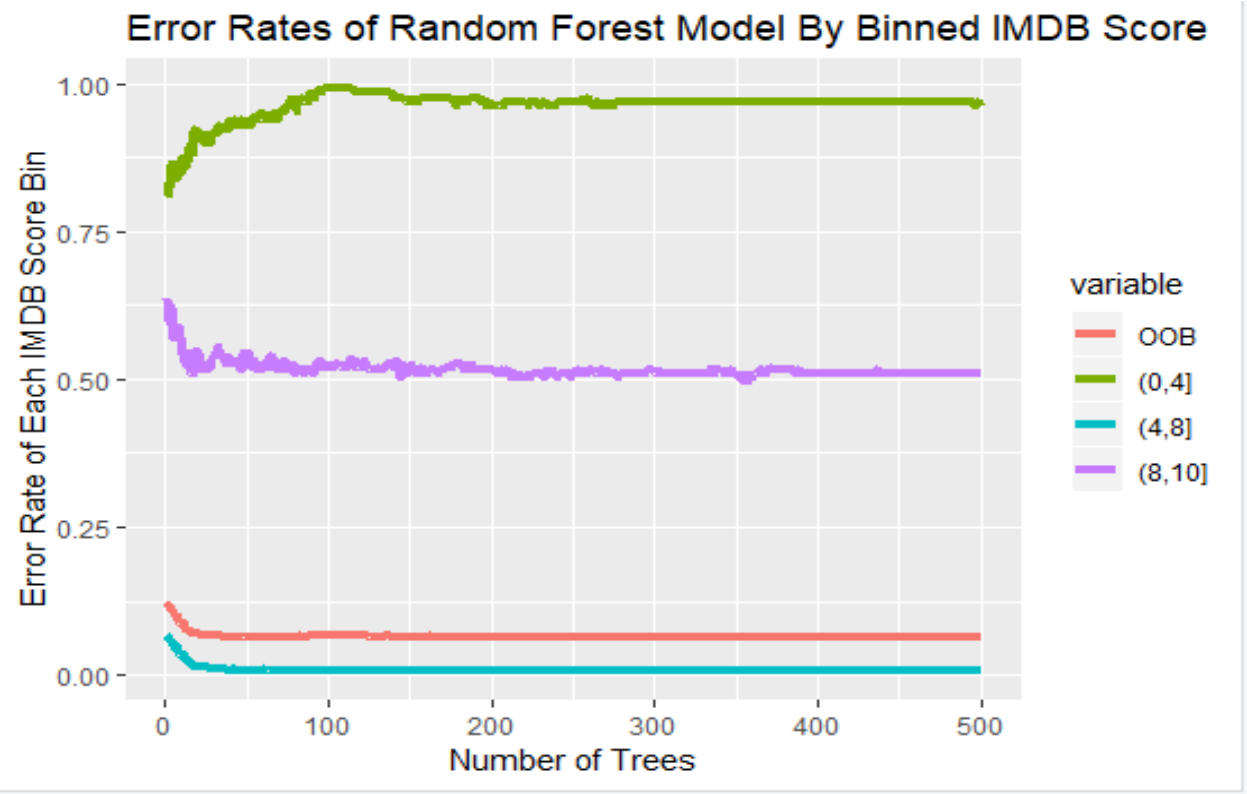

Figure 11. Error Rate for Random Forest by Binned IMDB Scores

In Figure 11, the red line shows, the over all Out of box (OOB) Estimate Error rate which gets constant as the number of trees increases. Also as seen in the confusion matrix, the bin with 4-8 IMDB score has the lowest error rate as shown with the blue line in the matrix, whereas the 0-4 bin has the highest error rate, as shown by the green plot.

Log (gross sales)

Before we start predicting values with the random forest algorithm, we need to create buckets with the range of gross revenues. We took the natural log of gross revenues and bucketized them into: 0-10, 10-12, 12-14, 14-16, 16-18, 18-20.

To predict the log(Gross), we specified taking 5 variables at a time (mtry = 5), chosen randomly. We also specified the algorithm to quantify the importance of each value being input to the random forest model. Therefore the command to run random forest was:

rfGross=randomForest(binned_LogGross ~ ., data = numdf4, mtry=mtry,ntree=400, importance = TRUE)

After running random forest several times, we step-wise eliminated the lowest importance variable until 10 remained as the final input to the random forest algorithm.

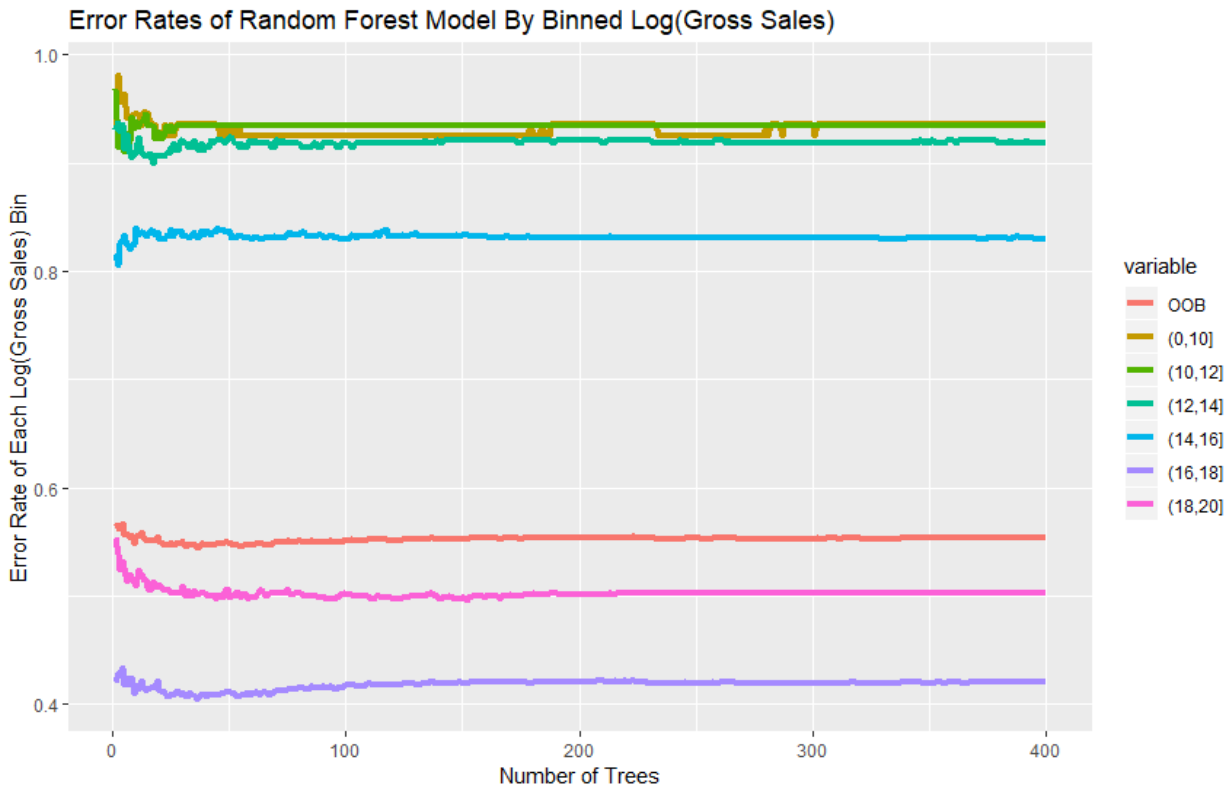

Figure 12 Error Rates of Random Forest Model by Binned Log(Gross Sales)

In Figure 12, we can see the red line is the error of the overall model (out of bag error, or OOB) The random forest model error rate is very high (80-90%) for low valued revenues (logGross = 0-10, 10-12, 12-14 and 14-16) or gross revenues up to $8.9M. The model does a great job predicting revenues from $8.9M to $760M.

The confusion matrix for this model compares the predicted log(Gross) revenue to the actual value (column names) in the Test dataset to the predicted values (row names). The confusion matrix in Table 3 shows an accuracy of 98%. We calculate accuracy by adding the values in the diagonal (or the correctly predicted values) and dividing that sum by the total of values in the table.

Table 3 Confusion Matrix of Random Forest Prediction of Log(Gross Sales).

Prediction Accuracy = 98%

Predicted Values	Actual Values					
	(0,10]	(10,12]	(12,14]	(14,16]	(16,18]	(18,20]
(0,10]	20	0	0	0	1	0
(10,12]	0	43	0	2	1	0
(12,14]	0	0	80	1	1	0
(14,16]	0	0	1	154	0	0
(16,18]	3	3	1	1	655	1
(18,20]	0	0	0	0	0	240

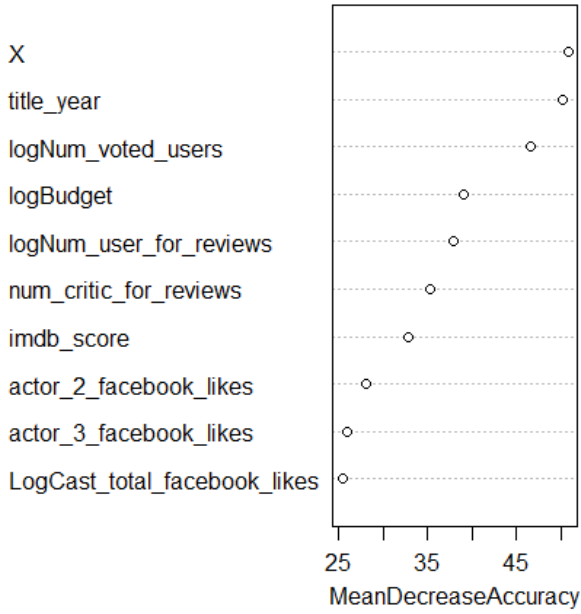

Figure 13 Importance of Variables Input into Random Forest to Predict log(Gross)

Assessing both the MeanDecreaseAccuracy and MeanDecreaseGini, the random forest model gives more importance to these variables. The order may be a little different because they assess importance in different manners. MeanDecreaseAccuracy ranks importance by the impact of removing one variable and scrambling the observation while keeping the proportion correct. Does that change the outcomes a lot or a little? MeanDecreaseGini assesses how pure the split for that variable is. So if number of users who voted splits 90% one way in the tree and 10% the other way, that's pretty pure. If the split is closer to 50/50, then that has a higher Gini weight.

Figure 13 shows the two ways to assess importance of the variables in the random forest model predicting log(Gross sales). Important variables are once again:

- Number of users who voted
- Year the movie came out
- Number of user FOR reviews
- Number of critic FOR reviews
- Cast total FB likes

Actor 1, 2 and 3 Facebook likes, all rank lower, but still high enough to remain in the model. It's interesting that the title year (year the movie came out) is actually a strong predictor.

To summarize, the random forest algorithm is incredibly powerful and accurate, yielding predictions of the log(Gross Sales) that are 98% accurate. Another surprising aspect is that the factors that are important in the random forest prediction can be very different from those we saw highly correlated with log(Gross sales). An example is Title Year, which is not very correlated to log(Gross sales), but is important in the random forest algorithm.

Conclusion and Future Scope

Table 4 Model Accuracies of 3 Algorithms on 2 Measures of Movie Success

Prediction Algorithm	IMDB Score Accuracy	log(Gross sales) Accuracy
Linear Regression	0.27	0.60
Text Mining	0.84	NA
Random Forest	0.94	0.98

Table 4 compares the accuracies of predicting IMDB scores and log(Gross Sales) using 3 different prediction algorithms:

- Linear regression
- Text mining
- Random forest

Linear regression did a poor job of predicting IMDB scores or log(Gross sales) even when including all the numerical fields in the models. The IMDB score linear model accounted for only 27% of the variability in IMDB scores. The log(Gross) linear model accounted for only 60% of the variability in log(Gross sales).

Random forest produced outstanding accuracy of 94% in predicting IMDB scores and 98% in predicting log(Gross sales).

But most surprisingly, text mining plot keywords was able to predict IMDB scores with 84% accuracy. The plot keywords field contains between 5 and 25 words, as very small number of words upon which to base a prediction. Yet text mining produced more accurate predictions than could be accomplished with linear regression

We realised that factors determining the success of IMDB rating and factors determining the success of gross sales had some common elements using the random forest. However, the linear model does not had a good accuracy on this kind of data set because the predictors were not able to explain the variation in the dependent variable and hence the R-sq value was pretty low.

On the other hand, our team represented many different aspect of story along the way. Also, we concluded that this analysis can be used to boost the gross sales of a movie up to some extent.

But, for making this model more robust and useful we may build a recommendation model based on clustering algorithms utilizing packages such as CARET in order to target appropriate audience to maximize the gross sales or the IMDB rating

References

The original dataset has been replaced on the Kaggle website with a larger one. The Dataworld site has the 5000 movie data set that was used in this project: https://data.world/data-society/imdb-5000-movie-dataset

Appendix

Data Dictionary

Variable Name	Description
movie_title	Title of the Movie
duration	Duration in minutes
director_name	Name of the Director of the Movie
director_facebook_likes	Number of likes of the Director on his Facebook Page
actor_1_name	Primary actor starring in the movie
actor_1_facebook_likes	Number of likes of the Actor_1 on his/her Facebook Page
actor_2_name	Other actor starring in the movie
actor_2_facebook_likes	Number of likes of the Actor_2 on his/her Facebook Page
actor_3_name	Other actor starring in the movie
actor_3_facebook_likes	Number of likes of the Actor_3 on his/her Facebook Page
num_user_for_reviews	Number of users who gave a review
num_critic_for_reviews	Number of critical reviews on imdb
num_voted_users	Number of people who voted for the movie
cast_total_facebook_likes	Total number of facebook likes of the entire cast of the movie
movie_facebook_likes	Number of Facebook likes in the movie page

plot_keywords	Keywords describing the movie plot
facenumber_in_poster	Number of the actor who featured in the movie poster
color	Film colorization. ‘Black and White’ or ‘Color’
genres	Film categorization like ‘Animation’, ‘Comedy’, ‘Romance’
title_year	The year in which the movie is released (1916:2016)
language	English, Arabic, Chinese, French, German, Danish, Italian etc.
country	Country where the movie is produced
content_rating	Content rating of the movie
aspect_ratio	Aspect ratio the movie was made in
movie_imdb_link	IMDB link of the movie
gross	Gross earnings of the movie in Dollars
budget	Budget of the movie in Dollars
imdb_score	IMDB Score of the movie on IMDB