UNIVERSITY OF TECHNOLOGY SYDNEY

FACULTY OF ENGINEERING AND INFORMATION TECHNOLOGY

SCHOOL OF SYSTEMS, MANAGEMENT AND LEADERSHIP

Assignment2**:** Practical Workplace–Related Data Analytics Project

Contents

1. Business Understanding

Movies have long been a source of entertainment. There is an immense market dedicated for their production. But, as we all know, it is challenging to satisfy viewers in bringing quality movies. Studios dedicate their time and effort in allocating the right budget, directors, cast and many more features which play a pivotal role in making a movie succeed. This measure of success is usually defined in a score or rate and is what most viewers rely on to decide whether a movie is worthy of their time to view. Nowadays, there are numerous resources to view movie scores such as Rotten Tomatoes, IMDB and Guardian.

1.1 Business Objective

In this report, we address the problem of how we can actually perceive how a movie will score in the future. This will provide a metric to relate to when choosing whether a movie is successful or not. We find an appropriate and credible dataset and analyse how its features and dimensions help with our predictions. This will be done by choosing appropriate data mining models used in everyday data analytics.

1.2 Data Source

In order to determine the appropriate dataset that is suitable for our analysis, we had to rely on a reliable source. We looked at options such as “themoviedb”, “bmxdb” and others but found that IMDB was the most comprehensive and commanding source of information on movie ratings. The reason this is so, is because they base their rating on a “weighted average” which means that some votes have different weights then others in their calculations. In other words, if a highly acclaimed movie critic personnel is voting, it will have a higher impact on the movie rating then a regular person voting. Also, IMDB adopts a special formula to calculate how a movie rates based on a “Bayesian estimate”. This formula relies on 4 variables and is described below;

- R = Rating (mean)
- v = # of votes
- m = Minimum votes to list the movie in the top charts
- C = mean vote spread around the entire report

Weighted rating (WR) = $(v \div (v+m)) \times R + (m \div (v+m)) \times C$

This weighted rating and IMDB highly praised information, was enough to convince us to use their database as a resource. However, we still had to find the dataset. We were fortunate enough to find an IMDB movie dataset on Kaggle which had 5000 instances and 29 attributes.

1.3 Measure of success

Predicting if a movie is going to succeed is based on how well it does from 1 to 10 in our prediction results. If a movie gets a 7.5 we rely on how IMDB classifies the scores. (e.g 6 = average, 7= above average, 7.6 = successful) Our concern is not whether a movie succeeds, but predicting the score it is going to get.

1.4 Course of Action

We start to understand and exploit our data by describing the features it presents using statistics, boxplots, bar plots and scatterplots. Now that we have a clear understanding of our dataset, we can go ahead and select, pre-process and transform the attributes we need to proceed with our modelling techniques. We pick the suitable algorithms and evaluate the most reasonable model which is in alignment with our business objective.

2. Data Understanding

2.1 Data Description

Attribute	Format	Description
Directors	0	normal director
	1	Golden Directors (With Oscars)
	2	Silver Directors (Nominated)
	3	Bronze directors (Ranked)
Actors	0	Star actor (Based on IMDB Roster)
	1	Normal Actor (Not ranked in IMDB Roster)
Content_Rating	-1	Not Rated
	0	Approved
	1	G
	2	GP
	3	M
	4	NC-17
	5	Passed
	6	PG
	7	PG-13
	8	R
	9	X
Gross	INTEGER	Movie Total Revenue
Budget	INTEGER	Movie Total Expenditure
imdb_score	INTEGER	Movie Rating from 1 to 10. (E.g 6.5, 7.4)
Num_voted_users	INTEGER	Users votes for a particular movie
Duration	INTEGER	Number of minutes from start to end

2.2 Statistical description of Dimensions

Dimension	Count	Means	Standard Deviation	Minimum Value	Maximum Value	Skewness	Kurtosis
Gross	3891	50,907,893.8159	69,205,645.4235	162	7.61E8	2.9971	13.6188
Budget	3891	45,165,345.1879	2.22E8	218	1.22E10	44.7803	2,347.0972
Imdb_score	3891	6.4631	1.0559	1.6	9.3	0.7258	1.1257
Net Profit	3891	5,742,548.628	2.26E8	1.22E10	5.24E8	43.1679	2,233.4099
ROI	3891	5.2546	129.6391	-1	7,193.4855	47.7373	2,507.6597
Content rating	3891	6.8825	1.8986	-1	9	-2.9936	8.9585

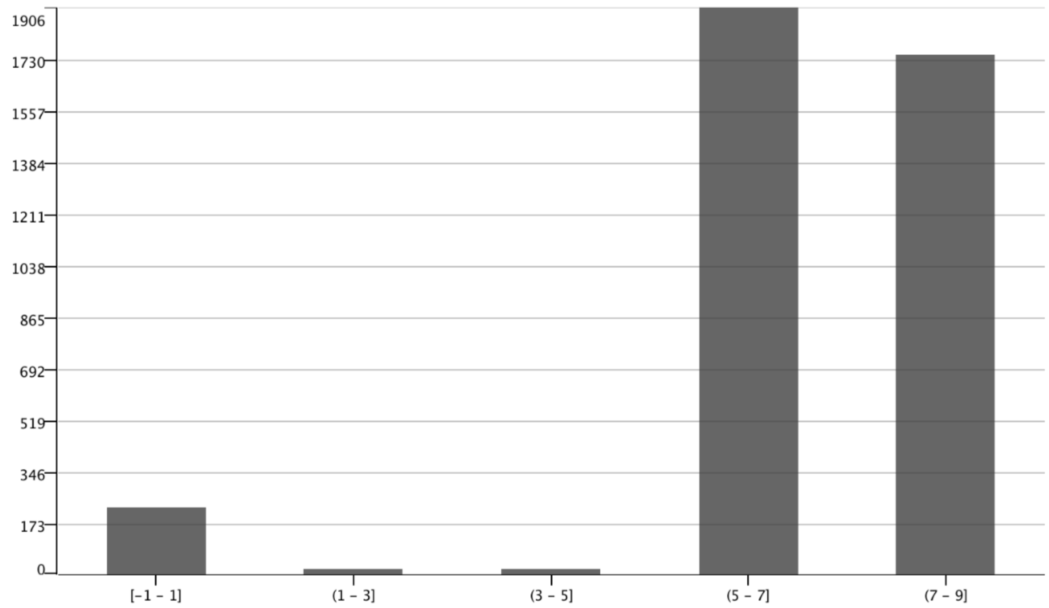


Figure 1: Content rating Histogram

In figure 1, the histogram of the content rating of the given data set has a mean value of 6.88, a standard deviation 1.8986 having minimum value of -1 and maximum 9, skewness -2996 and kurtosis 8.9585 and the graph is not so uniformly distributed. From the histogram we can have the information that there are higher frequency of the movies that is rated passed, PG and PG-13, also have the lower movies, which is rated G, GP, and M.

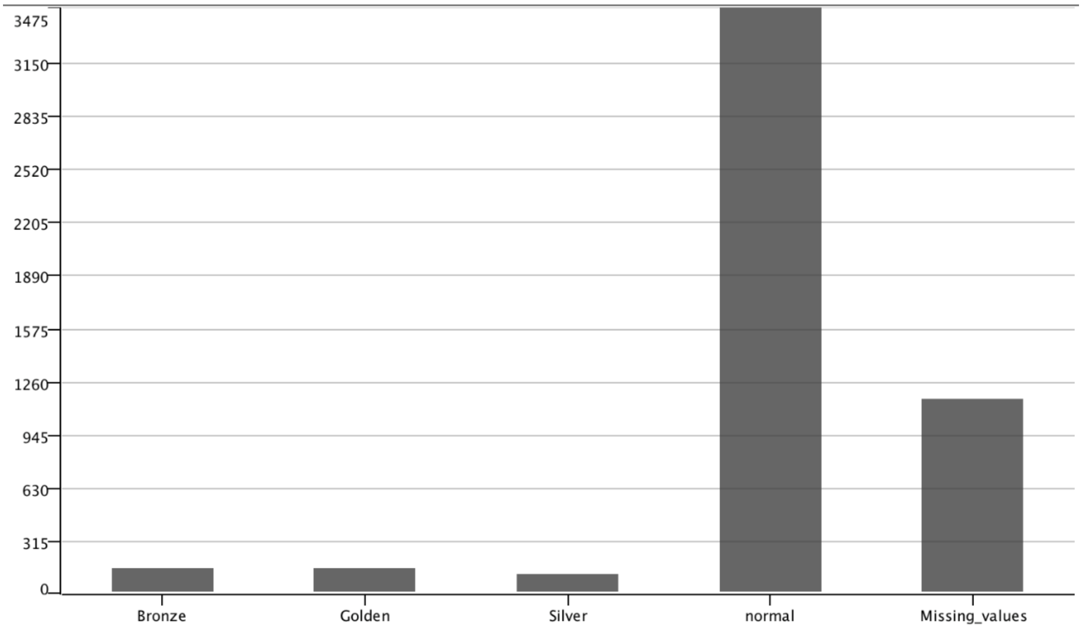


Figure 2: Directors Histogram

The figure 2 shows the histogram of the director of the given data set. From the histogram we can drive the information that most of the movies in the given data set is directed by normal directors, fewer movies which is less than 315 is been directed by directors classified as Bronze Golden and Silver. The data set also consist of missing values as shown by the last bar diagram.

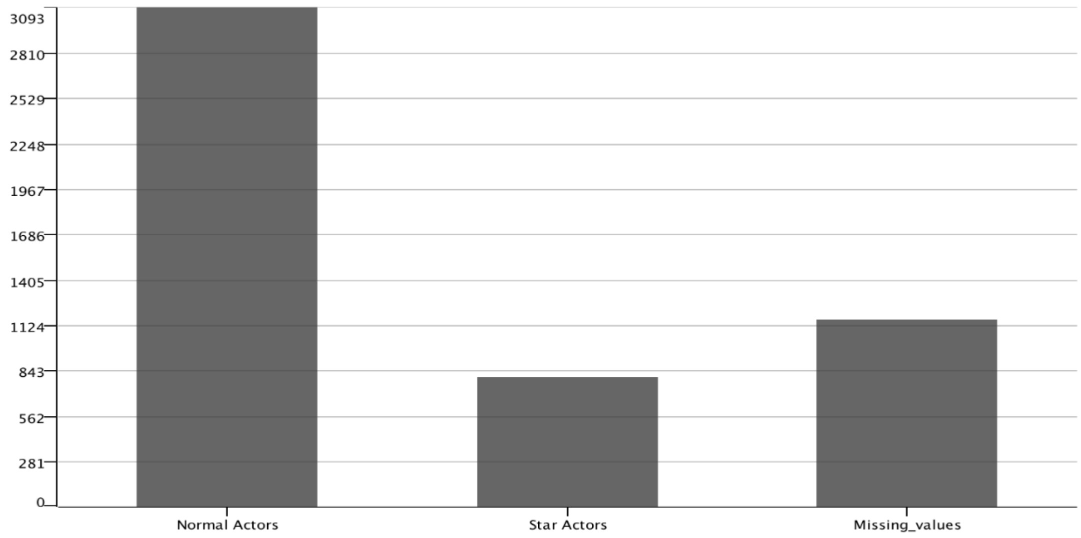


Figure 3: Actors Histogram

The figure 3 shows the histogram of the actors of the given data set. From the histogram we can drive the information that most of the movies in the given data set is played by normal actors, fewer movies which is less than 843 is played by actors classified as Star actors. The data set also consist of missing values as shown by the last bar diagram.

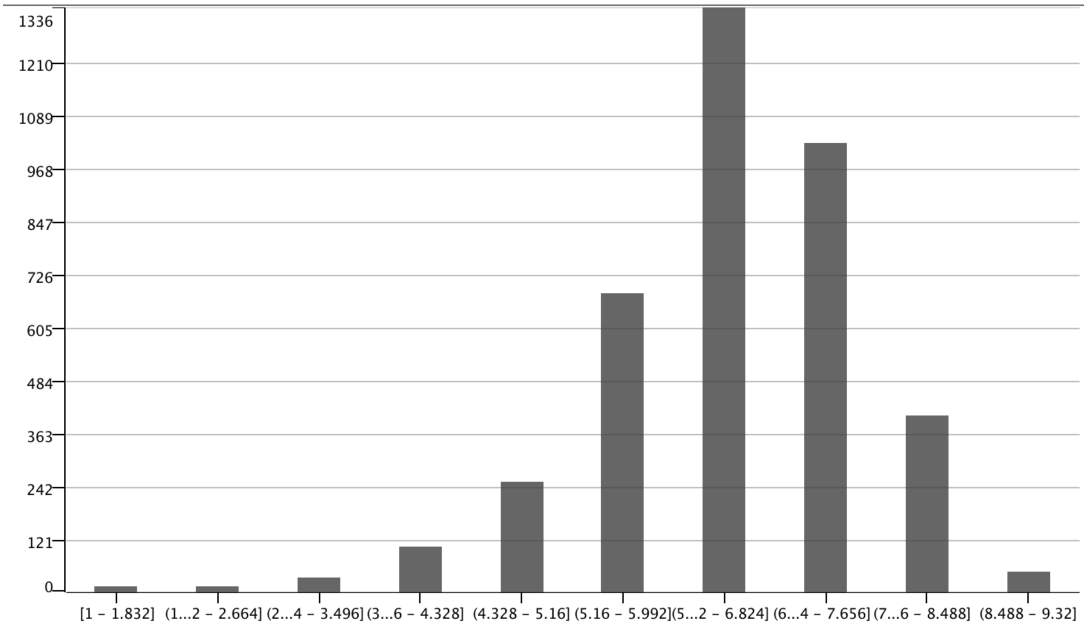


Figure 4:Binning of the IMDB_SCORE

The figure 4 the histogram of the Imbd score rating histogram of the given data set. From the histogram we can have the information that most of the movies in the given dataset is rated 5 – 6 that is average rating for a movie.

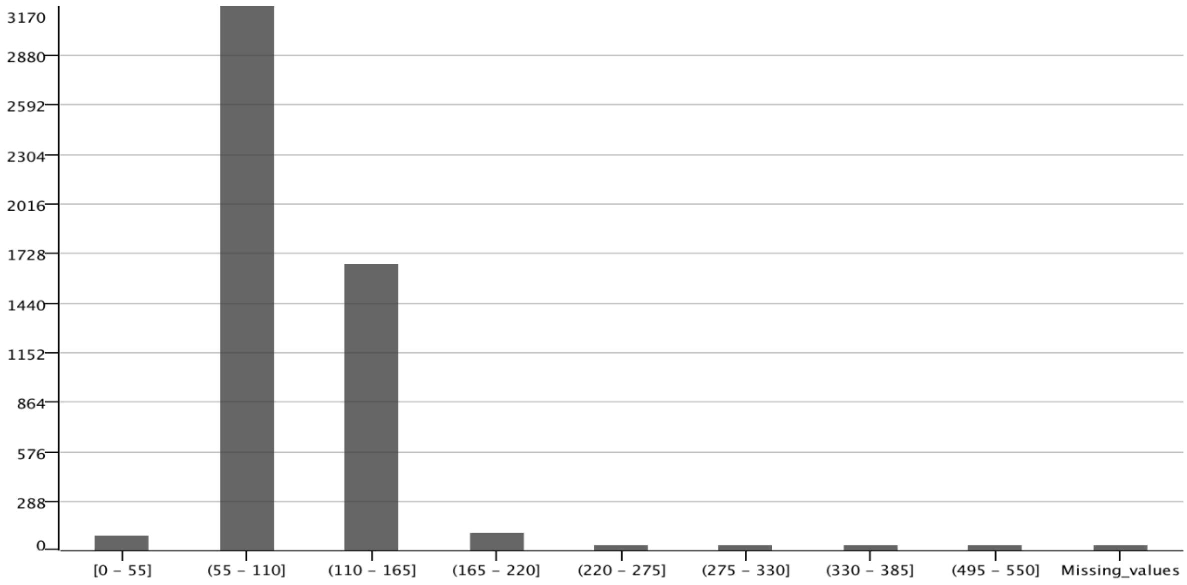


Figure 5: Histogram of duration

The above figure 5 represents the histogram of movie duration time. From the figure we can say that most of the movie duration time is between 55 mins to 110 mins and very few number of movies are of long duration, which is between 495 mins to 550 mins. The data set also contains some missing values under duration entity.

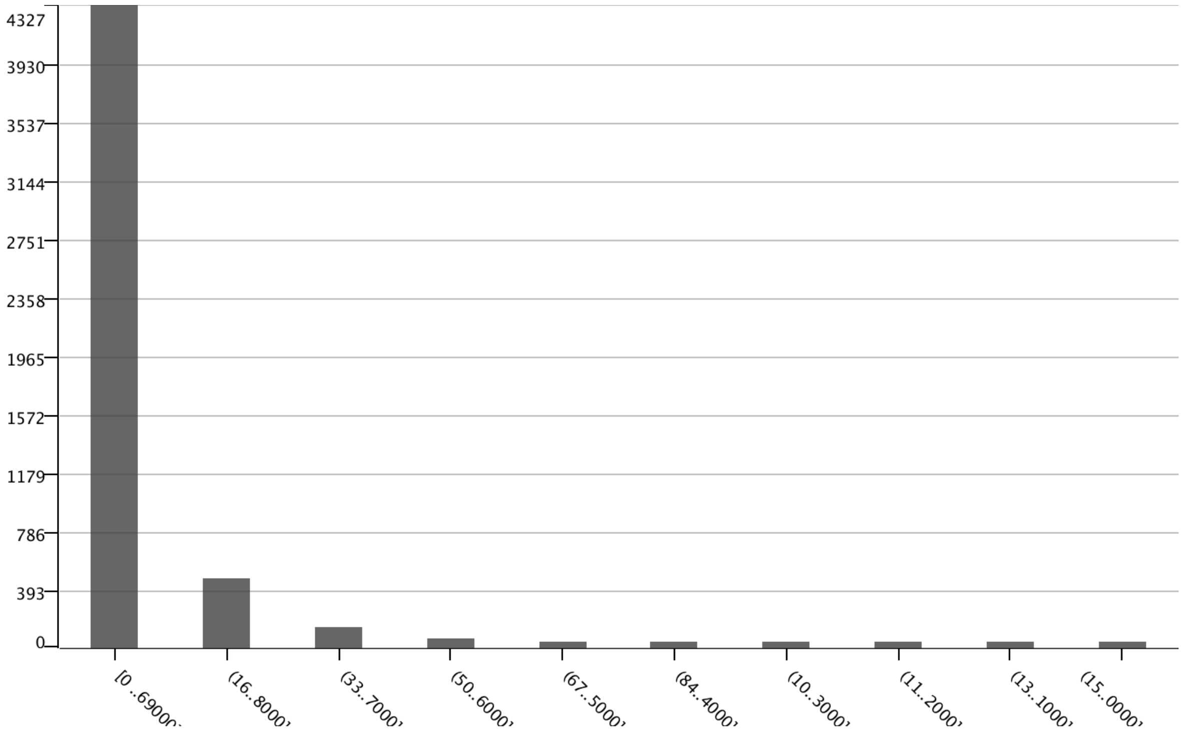


Figure 6: Number of Voted Users Distribution

Figure 6 represents the histogram of number of voted users to the movies. From the figure we can get the information that the vote ranged from 0- 69000 had 4327 number of movies, similarly the vote range from 16-80000 had fewer than 786 numbers of movies and very few movies got the large number of voted user of 150k.

2.3 Boxplots and Outliers

2.3.1 Content rating and imdb score

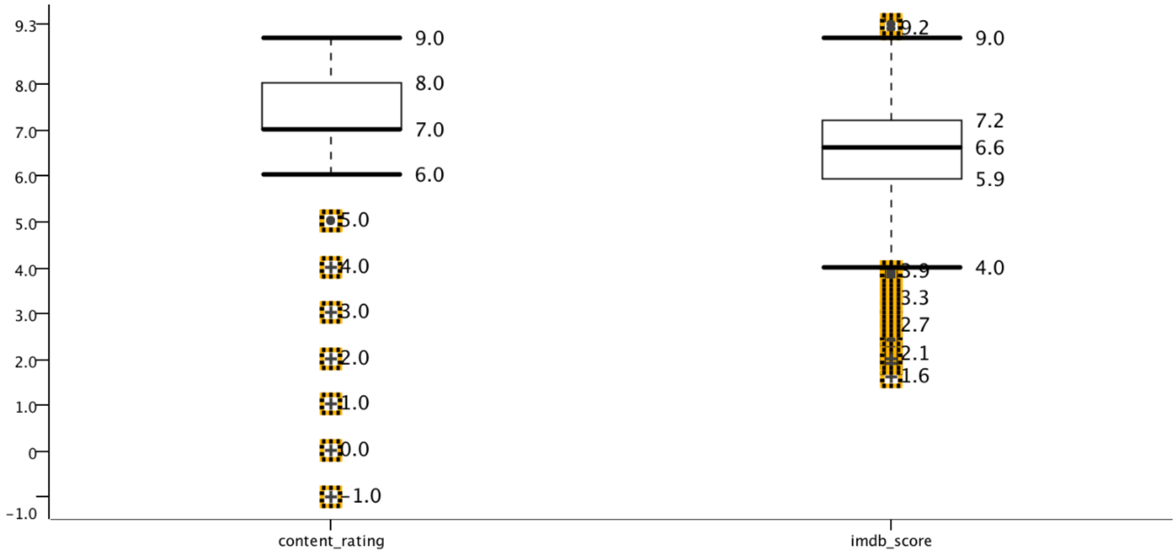


Figure 7: Boxplot of Content rating and imdb_score

Figure 7 shows the boxplot of content rating and imdb score where we can see many outliers in content rating and imdb score moreover 1st quartile of content rating and imdb score is almost same which is 6, median having 7 and 6.6 respectively and third quartile having same value of 9 score.

2.3.2 Net profit and imdb score

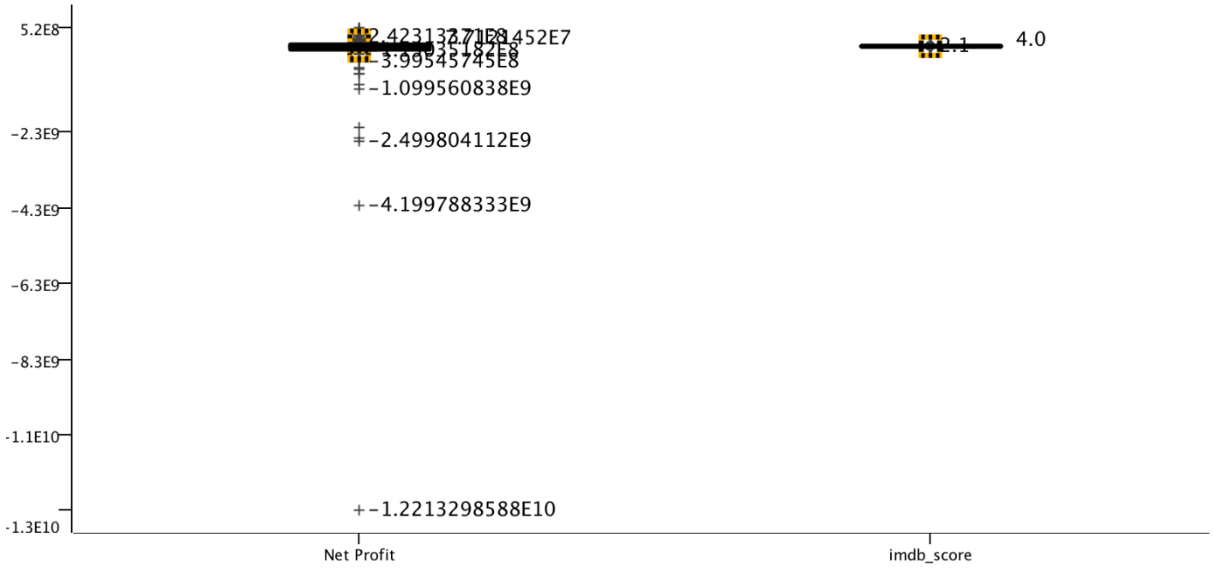


Figure 8: Outliers of net profit and imdb scores

Figure 8 shows the boxplot of net profit and imdb score where we can see many outliers in net profit and imdb score that is highlighted in the given figure.

2.4 Missing and Empty Values

2.4.1 Correlations

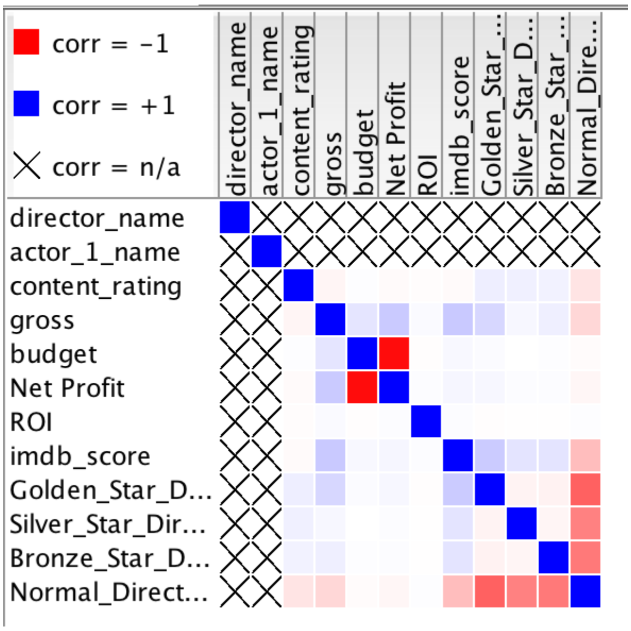
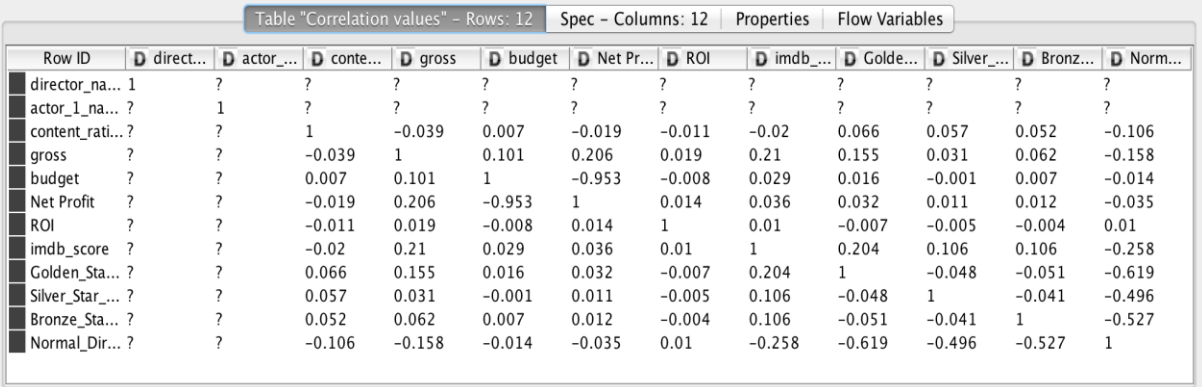


Figure 9: Attibutes Correlations

Figure 7 displays the correlation between all the dimensions. The dark blue color defines the higher correlation between the dimensions and as the intensity of the color decreases the correlation are between the two dimensions also decreases. The field content rating, imdb_score and and net profit have some level of correlation with each other. Actor name and director name have no correlation. Budget and net profit has negative correlation.



Row ID	D direct...	D actor_...	D conte...	D gross	D budget	D Net Pr...	D ROI	D imdb_...	D Golde...	D Silver_...	D Bronz...	D Norm...
director_na...	1	?	?	?	?	?	?	?	?	?	?	?
actor_1_na...	?	1	?	?	?	?	?	?	?	?	?	?
content_rati...	?	?	1	−0.039	0.007	−0.019	−0.011	−0.02	0.066	0.057	0.052	−0.106
gross	?	?	−0.039	1	0.101	0.206	0.019	0.21	0.155	0.031	0.062	−0.158
budget	?	?	0.007	0.101	1	−0.953	−0.008	0.029	0.016	−0.001	0.007	−0.014
Net Profit	?	?	−0.019	0.206	−0.953	1	0.014	0.036	0.032	0.011	0.012	−0.035
ROI	?	?	−0.011	0.019	−0.008	0.014	1	0.01	−0.007	−0.005	−0.004	0.01
imdb_score	?	?	−0.02	0.21	0.029	0.036	0.01	1	0.204	0.106	0.106	−0.258
Golden_Sta...	?	?	0.066	0.155	0.016	0.032	−0.007	0.204	1	−0.048	−0.051	−0.619
Silver_Star_...	?	?	0.057	0.031	−0.001	0.011	−0.005	0.106	−0.048	1	−0.041	−0.496
Bronze_Sta...	?	?	0.052	0.062	0.007	0.012	−0.004	0.106	−0.051	−0.041	1	−0.527
Normal_Dir...	?	?	−0.106	−0.158	−0.014	−0.035	0.01	−0.258	−0.619	−0.496	−0.527	1

Figure 10: Correlation values of the dimensions

Here, figure 10 shows the actual correlation values of the dimensions that is exact same correlation matrix shown in figure 7.

2.5 Scatter Plots

2.5.1 Gross vs IMDB score

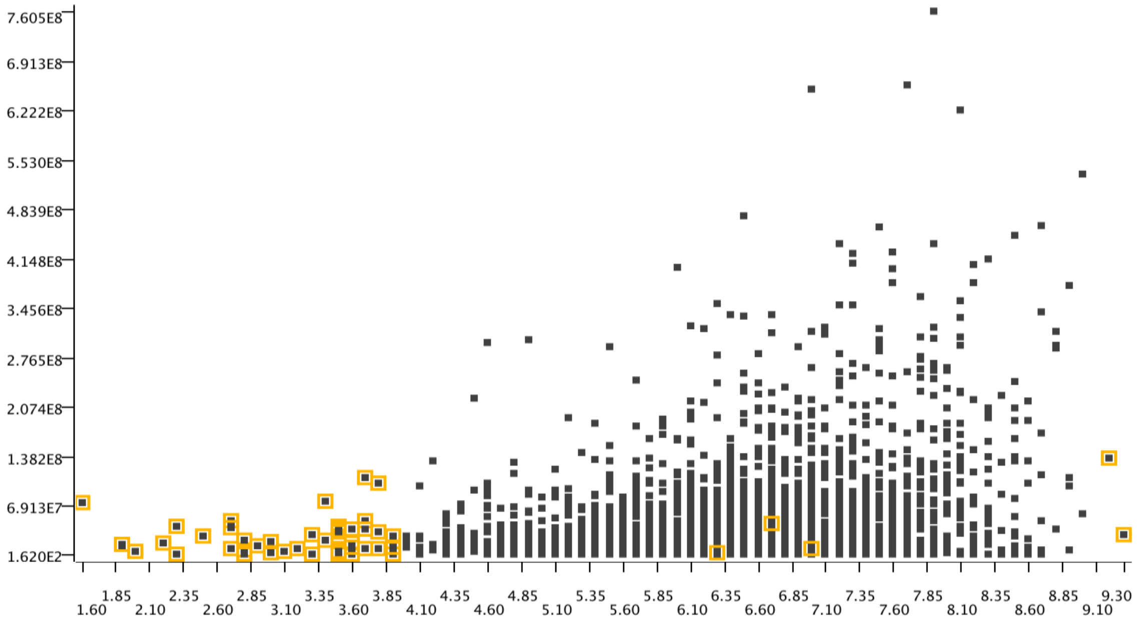


Figure 11: Gross vs IMDB score

The scatter plot of gross and imdb score is shown by the above figure 11. From the figure we can see that the data on the x-axis is mostly concentrated in the upper middle where the imdb score ranges between 6.1 to 8.6. Any imdb score lower then 6.1 shows a low gross in the y-axis.

2.5.2 Content rating vs IMDB score

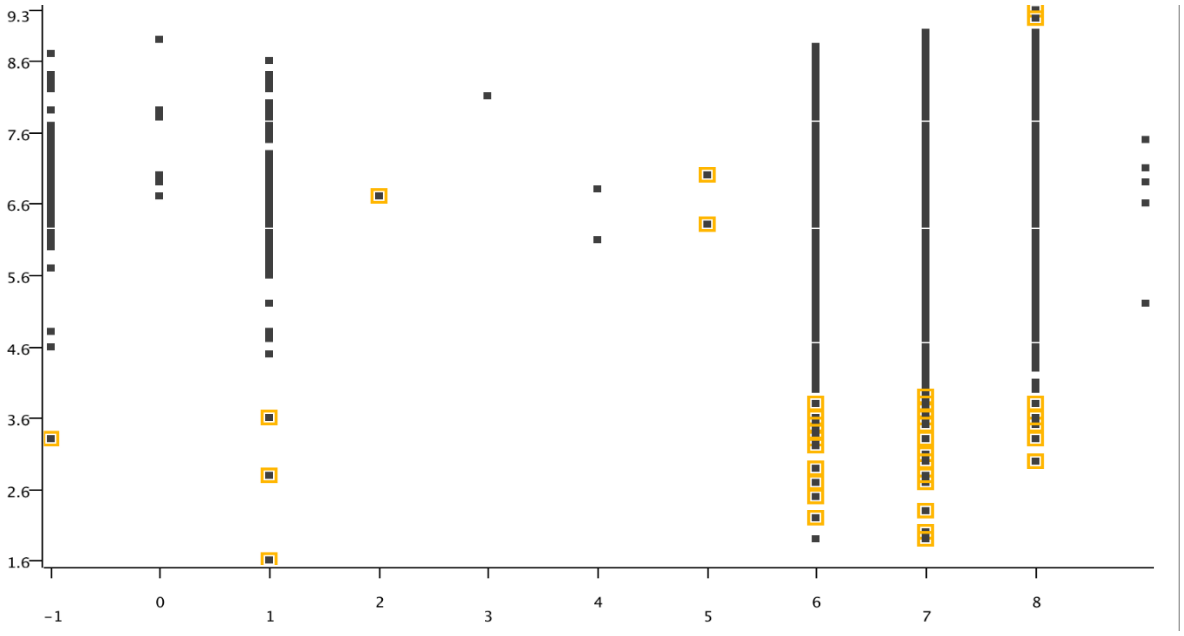


Figure 12: Content Rating vs IMDB score

The scatter plot of content rating and imdb score is shown by the above figure 9. From the figure we can see that the data in x-axis is mostly concentrated upper far area where the content rating ranges between 6 and 8. This means that most of the movies of content rating PG PG-13 and R are mostly rated with imdb score 5.6 and above.

2.5.3 Number of voted users vs IMDB score

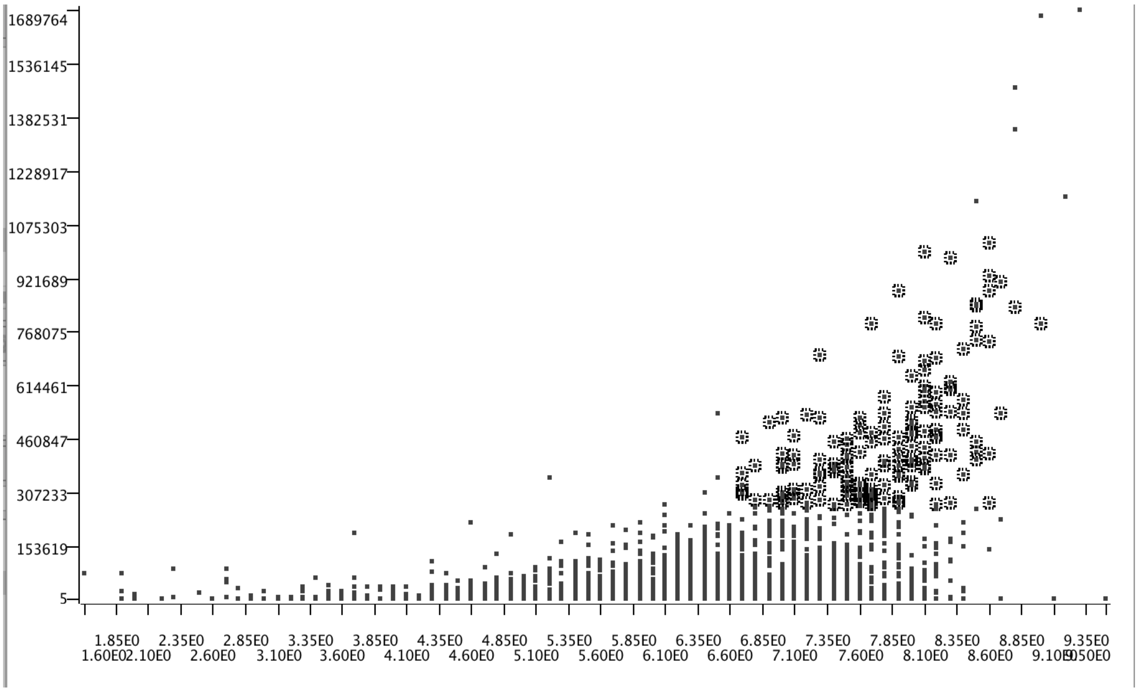


Figure 13: IMDB score vs number of voted users

The above figure is the representation of scatter plot between imdb score and number of voted user. From the figure what can be drawn is that the imdb score is greater than 6.8 where there is high number of voter user having number of votes greater than 307233. It is represented by highlighted dense instances at the higher x-axis area and lower y-axis area.

2.5.4 Duration vs IMDB score

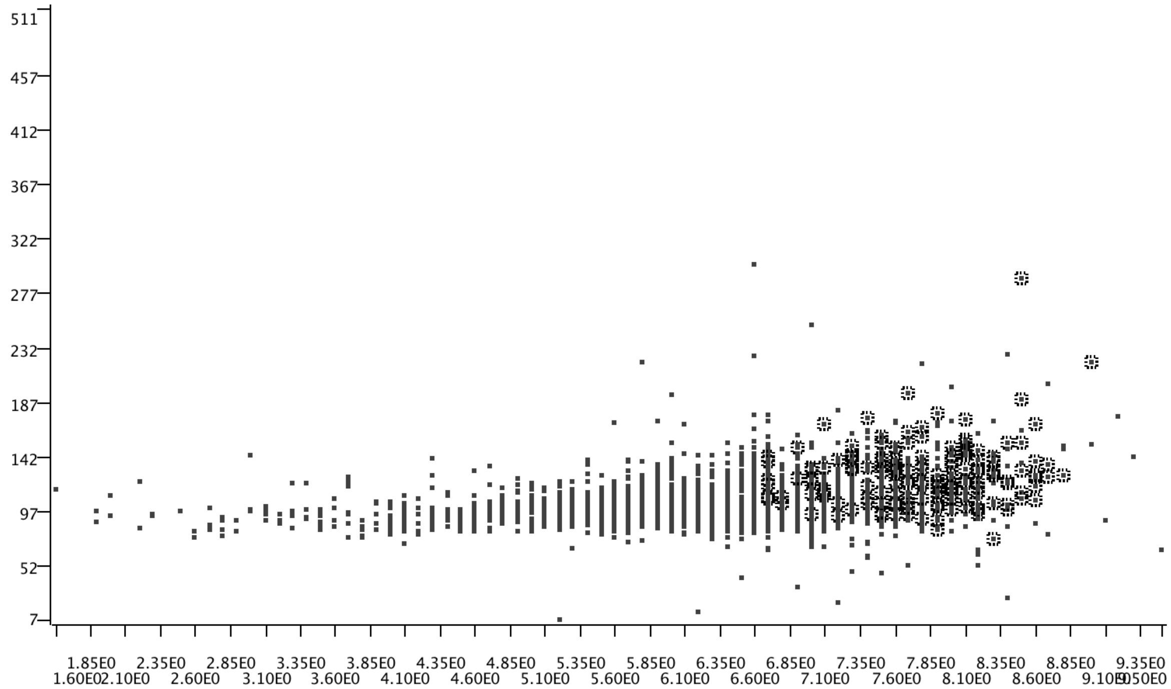


Figure 14: IMDB score vs Duration

The Figure 14 represents the scatter plot between imdb score and duration of the movie. From the figure we can see that most of the movies have a similar duration between 52 mins to 187 mins and the rating of the movies are observed through out movies of these duration.

3. Data Preparation

Data Cleaning and Observations

Data Cleaning

Our initial dataset constituted of 28 attributes. Only 8 were chosen since they were significantly relevant in analysing the business problem. The remaining attributes were deleted from the dataset.

Deleted the occurrences of the character "¬" from the movie_title attribute which had no relevance to the movie title.

Located and corrected Words which had the HTML Unicode such as "é" and "ç" instead of the acute letters È, Ì, ÿ, Â, Á were corrected. Example: √mile Gaudreault is now émile Gaudreault.

Data Observations:

Located and interpreted missing values in instances that belonged to attributes such as director_name, director_facebook_likes, budget and content_rating.

These were not deleted since other attributes related to those rows had significant data which contributed to our overall analysis.

For instance, for the director_name and director_facebook_likes, there were only 103 out of 5044 rows with missing data in them. This would not dramatically skew the data and are considered as data outliers.

The maximum amount of data missing was found in the budget attribute which had 492 values missing which is around 20% of the instances. This would skew the data in our result set and is an important observation to note in our analysis.

It is relevant to note the 0 values which were observed in many of the movie_facebook_likes instances. These could be either considered as missing values, 0 facebook likes or experimental errors (data outliers).

The genres attribute instances include a vertical bar. E.g " Comedy|Drama " It acts as a separator between the different movie genres. Knime could be used to Bin the different genres of movies instead of separating them as so.

The content_rating attribute also could be binned in Knime.

Data Pre-processing and transformations

The most important part here was to find a solution on how to classify our actors and directors into categories. We had to rely on a credible roster to build our dictionary so we can categorise our data. We did a thorough investigation and found a list of directors which had won Oscars, another list of directors which did not win any Oscars but were highly distinguished and directors which were ranked but did not win any rewards. Similarly, we applied the same technique to the actors and used excel to transform strings into categorised integers.

We relied on more then 350 stars actors and more then a 100 directors and searched whether each row pertains to any of the dictionary list. If the row contained the last name or first name of a director or actor, it was assigned an appropriate ranking. (Ses section 2.1 Data description) Below the Formula snippet from excel.

```
IF(OR(AND(ISNUMBER(SEARCH("Christopher",A2)),ISNUMBER(SEARCH("Nolan",A2))),
AND(ISNUMBER(SEARCH("David ",A2)),ISNUMBER(SEARCH("Fincher",A2))),
AND(ISNUMBER(SEARCH("Peter",A2)),ISNUMBER(SEARCH("Jackson",A2))),
AND(ISNUMBER(SEARCH("Steven",A2)),ISNUMBER(SEARCH("Spielberg",A2)))),1,0)
```

4. Modelling

In order to understand which methods to use to predict our IMDB scores we first have to check how the response variable is distributed.

We do this in R by loading the ggplot2 library

The “Supervised Learning” branch of algorithms is used to train the machine in order to get the desired outputs from the chosen movie dataset. In this research three models are chosen to achieve this purpose; the Linear Model, the Regression trees Model and the Random Forests Models. The algorithms are developed using the R platform.

4.1 The Linear Regression Model

To develop our linear model for our response variable we need to visualize how other variables relate to the response variable.

We do this using the correlations function cor() in R. Below the code.

```
relationships <- x()
for (k in 1:dim(tmp)[2]) {
    relationships[k] <- cor(tmp[,k],tmp[,’imdb_score’])
}
relationships
```

We get the Result set below:

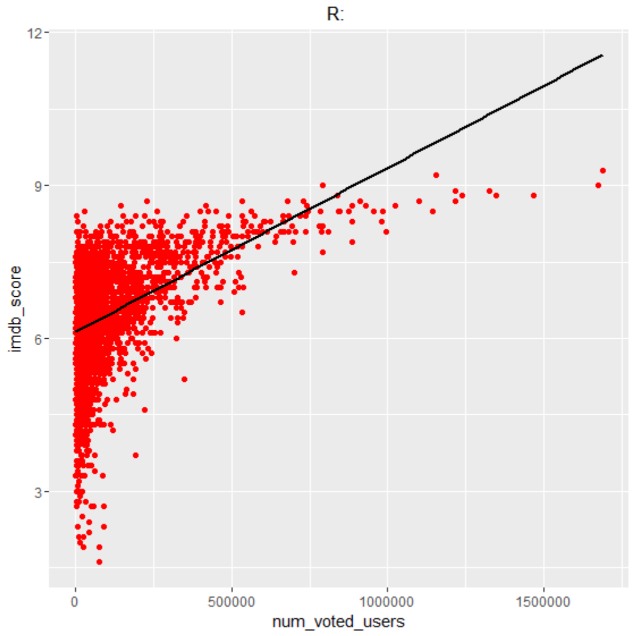
```
[1] 0.37321535 [2] 0.21994704 [3] 0.47532192 [4] 0.03519452
[5] 1.00000000 [6] 0.28625569 [7] 0.20836975 [8] 0.17732731
```

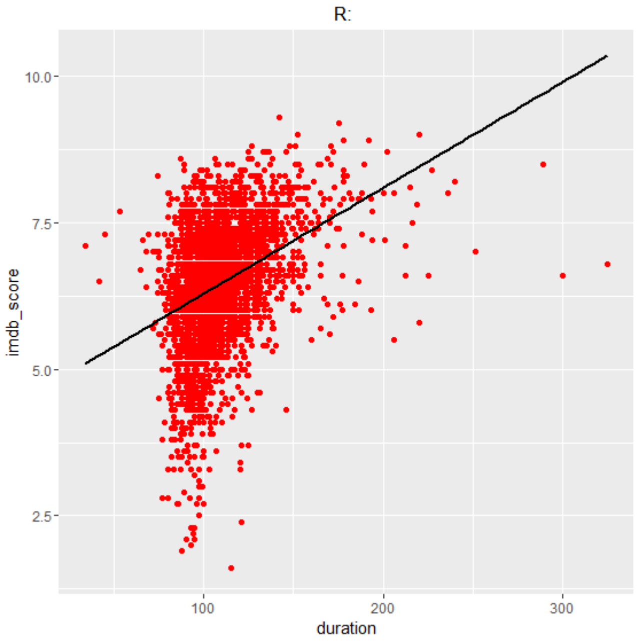
Using all our vectors for our linear model would increase the margin of errors. For this reason we decided to pick the top two variables with respect to how close they relate to the responsive variable. The first and third attributes are picked since their value is the highest in this case. These are the “duration” and “num_voted_users” attributes.

Visualization the correlations using a scatter plot

In order to visualise this better we use the R ggplot2 library. We produce 2 graphs the num_voted_users vs imdb_score and duration vs imdb_score. One snippet of the code is provided below. The ggplot function uses the aesthetics (aes) function to plot the axis.

```
     ggplot(tmp, aes(x=num_voted_users, y=imdb_score))
     ggplot(tmp, aes(x=duration, y=imdb_score))
```





Now that we have a clear view of how the attributes are distributed against our response variable, we start to build our model by using both the number of users’ votes and the duration variables. We split our data set into two: the test and training set.

Here is the coded data split:

```
     set.seed(10)
     training <- sample(dim(tmp)[1],dim(tmp)[1]*0.7)
     train_tmp <- tmp[training,]
     test_tmp <- tmp[-training,]
```

```
lm_fit = lm(imdb_score ~ num_voted_users + duration, data=train_tmp)
summary(lm_fit)
```

And the results:

```
Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)     5.006e+00  8.942e-02   55.98   <2e-16 ***
num_voted_users 2.731e-06  1.228e-07   22.23   <2e-16 ***
duration        1.076e-02  8.346e-04   12.89   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8927 on 2739 degrees of freedom
Multiple R-squared:  0.2666,    Adjusted R-squared:  0.266
F-statistic: 497.7 on 2 and 2739 DF,  p-value: < 2.2e-16
```

The value to observe in this summary set is the “R squared” metric which helps in assessing how our linear model interprets the variability in our dataset.(highlighted in green) Our value here is 0.2666 which is not close to R square of 1 and is considered a poor value. We expected this result since both explanatory vectors picked when we ran the correlation function were < 0.5.

Let us calculate the MSE (mean squared error) that corresponds to our model and see the margin of error it presents. This is coded below.

```
prediction <- predict(lm_fit, test_tmp)
mean((test_tmp$imdb_score-prediction)^2)
```

The result follows.

```
[1] 0.8227051
```

4.2 Regression Trees

Let’s try an alternative method to get the imdb scores predictions. We will give the regression trees a go since this specific model do not take into consideration the linearity in our dataset. Also the decision tree tend to be based on logic so it is easy to visualize the logical conditions to interpret.

We divide our data into a test and training data sets and apply the train set to rpart().

```
library(rpart)

set.seed(4)
t.rpart <- rpart(imdb_score~.,data=train_tmp)
t.rpart
```

We get the following Results:

```
1) root 2742 2975.94200 6.459227
   2) num_voted_users< 119867.5 2064 2048.06800 6.207364
     4) duration< 105.5 1184 1293.07000 5.952872
       8) budget>=4950000 891  935.48370 5.786756
        16) num_voted_users< 44136.5 557  613.07800 5.574506
          32) gross>=3637876 426  422.63820 5.416197 *
          33) gross< 3637876 131  145.04500 6.089313 *
        17) num_voted_users>=44136.5 334  255.46620 6.140719 *
       9) budget< 4950000 293  258.23370 6.458020 *
```

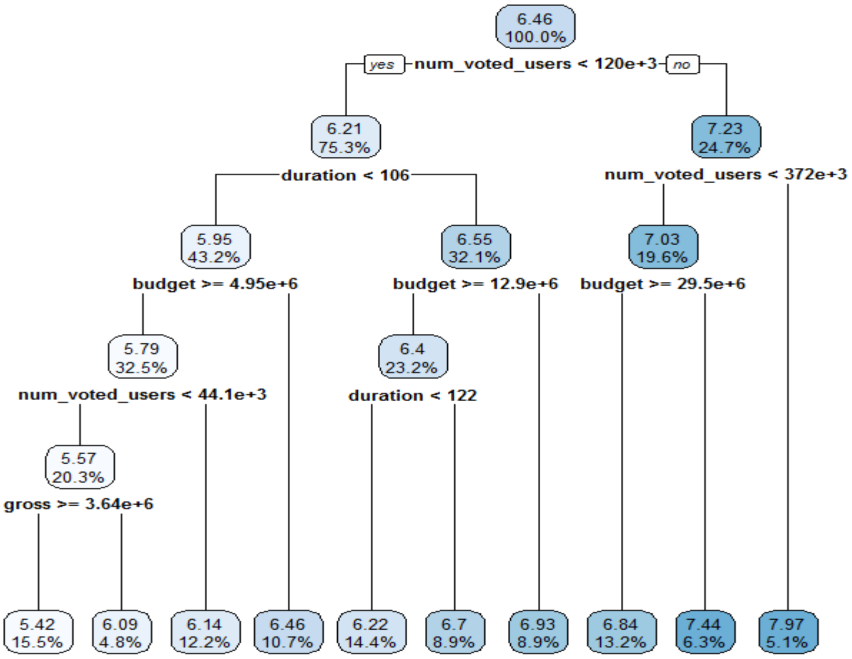
```
  5) duration>=105.5 880  575.14000 6.549773
   10) budget>=1.29e+07 637  369.90430 6.402983
     20) duration< 121.5 394  223.19440 6.222843 *
     21) duration>=121.5 243  113.19410 6.695062 *
   11) budget< 1.29e+07 243  155.52960 6.934568 *
3) num_voted_users>=119867.5 678  398.36310 7.225959
  6) num_voted_users< 371847 537  263.53070 7.031099
   12) budget>=2.95e+07 363  163.32790 6.837190 *
   13) budget< 2.95e+07 174   58.07908 7.435632 *
  7) num_voted_users>=371847 141   36.78638 7.968085 *
```

It seems that from the above results, the hierarchy has an emphasis on the num_voted_users and duration. (rows #2, #3 and #4) We will consider again these variables as pivotal to construct our tree.

Now we visualize the tree so we can examine the conditions and rules in order to predict the IMDB scores. We plot the tree using the rpart.plot() function.

```
rpart.plot(t.rpart,digits = 4)
```

The tree looks similar to a flowchart. Here is the visualization below:

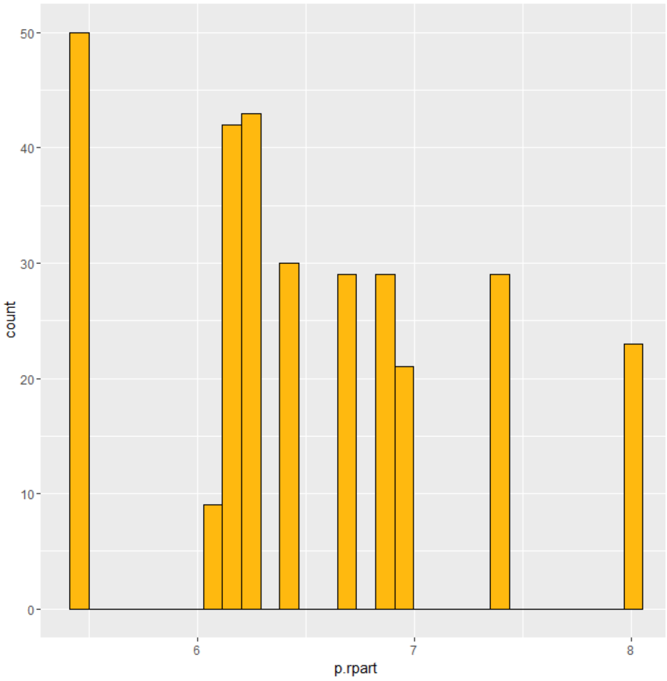


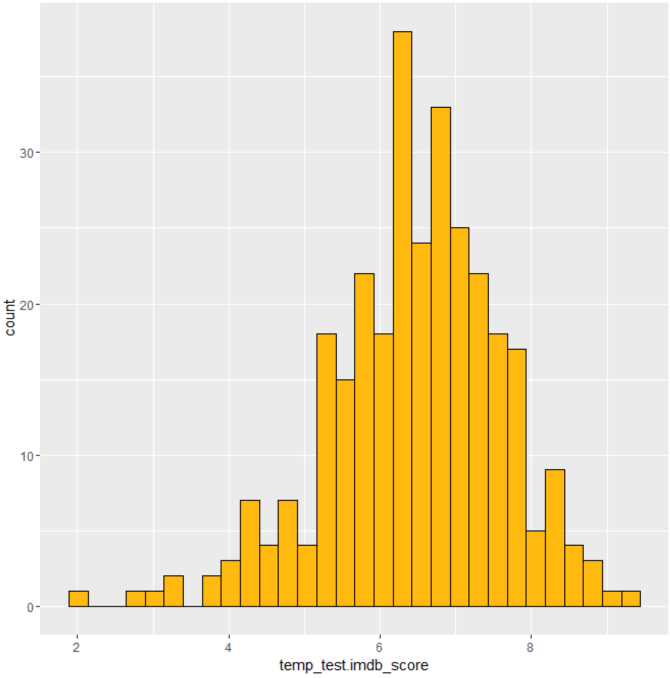
Now we apply this model to the test dataset.

```
s.rpart <- predict(t.rpart,test_tmp)
```

Next we examine how both the predicted and actual models are different from each other. We do this using histograms.

```
flowchart_tree <- data.frame(s.rpart,test_tmp$imdb_score)
ggplot(flowchart_tree, aes(x=s.rpart)) + geom_histogram(fill="darkgoldenrod1",
colour="black")
```





There is a significant difference in the distributions of data between the two histograms which tell us that the predictions and real values are totally different. We can still examine if there are correlations between the actual and predicted values. Again we use the cor() function.

```
cor(s.rpart,test_tmp$imdb_score)
```

We get the following result:

```
[1] 0.5438364
```

This is an acceptable correlation score. But it doesn't give us a certain measurement to how the actual values diverge from the predicted values. We calculate the MSE and we get the following result:

```
[1] 0.7532995
```

A fair improvement from out previous linear regression.

4.3 Random Forests

The training algorithm for our random forest model relies on the known technique of bagging or bootstrap aggregations to tree learners. We also use random selections of features to improve variety to the decision tree. Here is the code snippet below. We pick 600 ntree to train.

```
rf_algo <-randomForest(imdb_score~.,data=tmp[training,], +
ntree=600,mtry=floor(dim(tmp)[2]/3))
```

```
pred_rf_algo <- predict(rf_algo,temp[-training,]) +
mean((pred_rf_algo-tmp[-training,]$imdb_score)^2)
```

The result:

```
[1] 0.5166582
```

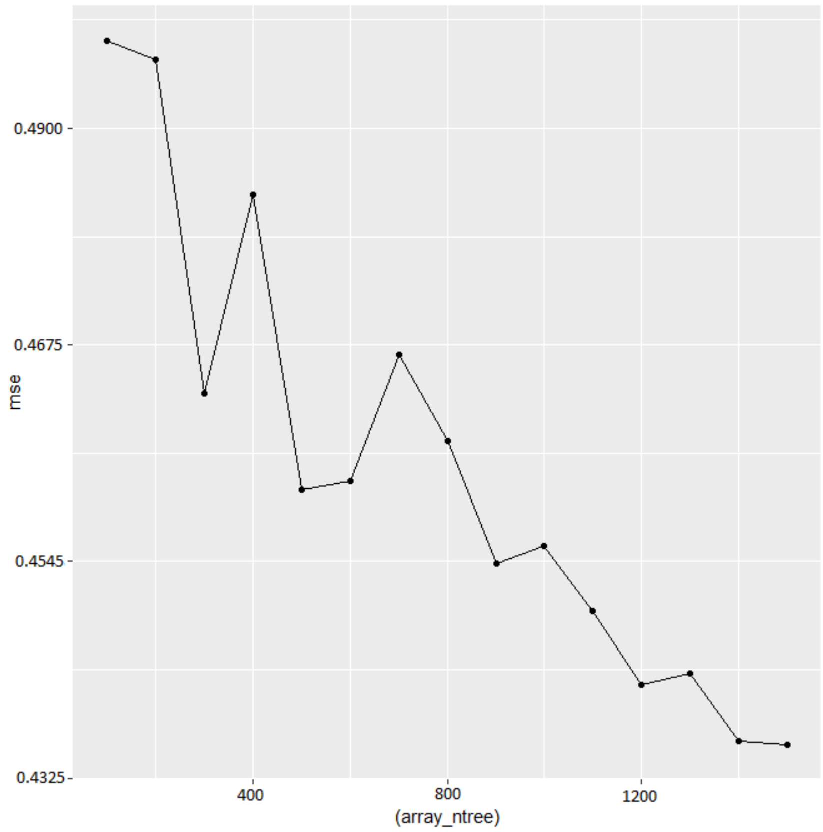
An acceptable result compared to our previous model. We change the number of ntrees parameter to see which one gives us a lower MSE.

```
ntree_pick<- c(100,200,300,400,500,600,700,800,900,1000,1100,1200)
error_measure <- c()
k <-1
for(i in ntree_pick)
{
        set.seed(15)
        rf_algo <- randomForest(imdb_score~.,data=tmp[training,],ntree=i, +
        mtry=floor(dim(tmp)[2]/3))
        pred_rf_algo <- predict(rf_algo,tmp[-training,])
        error_measure[j]<-mean((pred_rf_algo-tmp[-training,]$imdb_score)^2)
        k=k+1
}
```

Let us plot the MSE vs the ntrees we picked. We will only pick the value of ntree with the minimum MSE.

```
mse_value <- data.frame(ntree_pick,error_measure)
ggplot(mse_value, aes(x=(ntree_pick), y=error_measure)) + geom_line() +
geom_point()
```

We can tell in the graph below that the higher the number of ntrees the lower the MSE.



We extract this measure and we get the result below:

```
[1] 0.4365980
```

A major improvement when we rely on the most optimistic number of trees.

5. Evaluation

We picked the Random Forest algorithm to be our suitable model in order to predict our movie scores because of the high predictability measure. Trying different ntrees truly boosted our results and convinced us that this was the appropriate mode. We think the only thing which would have improved our results (we didn’t for the sake of time) would have been to include AdaBoost. We would have wished that the directors and actors classifications gave a better correlation to our response variable since in real life people tend to ask which director or actor is staring in the movie. A sentiment analysis could have also benefited this theory but this is an entire branch which can be tackled and is not related to our models we chose. All in all, movie scores can be predicted with our model, but it is always advantageous to rely on peer reviews and personal preference. For instance, a horror movie might score high in IMDB but that does not mean a person who hates horror movies is going to watch it based on the score in IMDB.