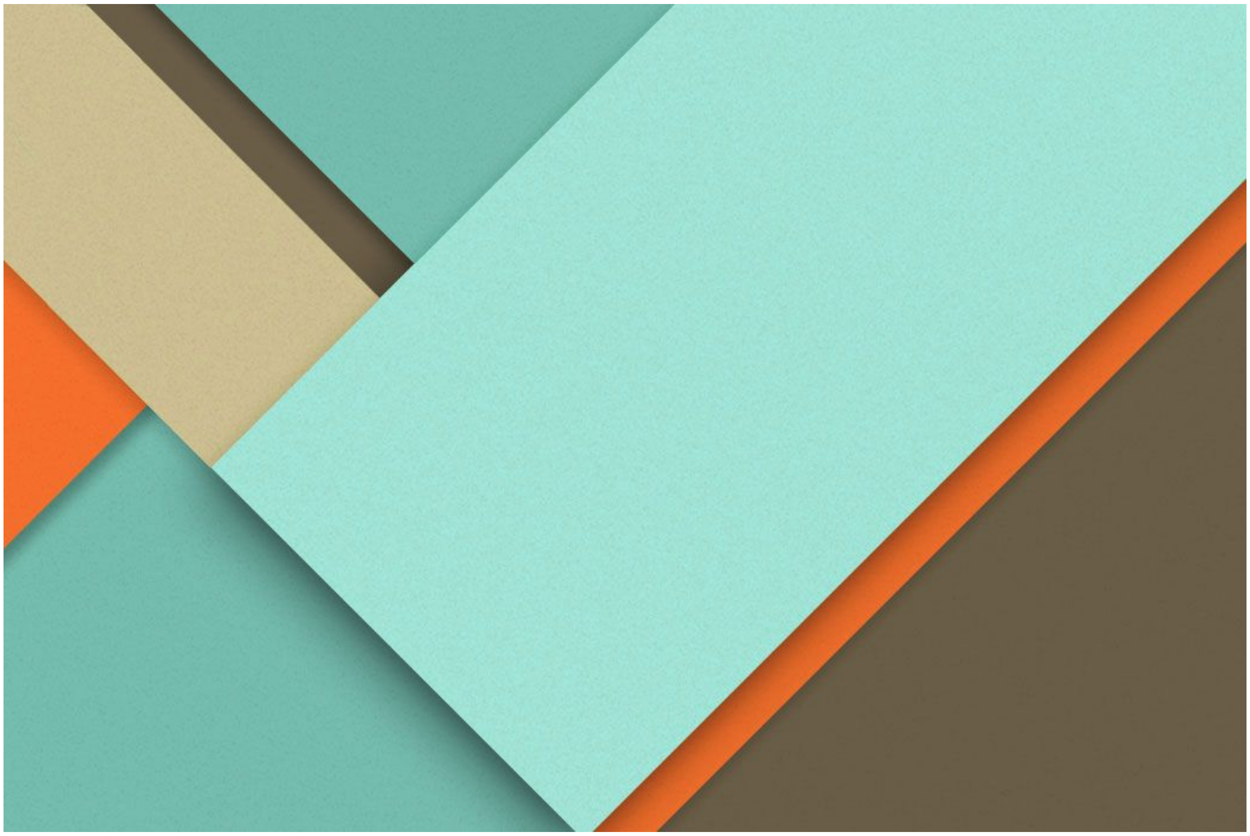
CS 686 Data Mining

Movie Prediction

Project Progress Report



07-November-2016

Mayank Thirani

Ryan Zhu

University of San Francisco

2130 Fulton Street, San Francisco

Contents

Problem Definition

Given that thousands of movies were produced each year, is there a better way for us to tell the greatness of movie without relying on critics or our own instincts?

We are going to predict the movie's greatness before it gets released in cinema by the movie's rating which will be received from IMDB score. Thus, our solution will be more towards Supervised Learning Approach where we will be using **imdb_score** in the dataset as the target variable (or response variable).

Movies will be classified into four different levels as stated below:

- Movies with imdb_score larger than 9.0 will be considered as "**best movies"**
- Movies with imdb_score from 7.0 to 8.0 will be considered as "**good movies**"
- Movies with imdb_score from 6.0 to 7.0 will be considered as "**average movies"**.
- Movies with imdb_score from 1 to 5 will be considered as "**bad movies**".

Above classifications of the imdb_score will be provided a new target name called **imdb_rating** in the dataset.

Data Analysis

Out of the 28 variables provided in dataset, we are especially interested to know how does the IMDB rating score correlate with other variables.

Before the release of a movie, it get lots of criticism based on director, actors (which are in cast), story plot and genre. They have their cool attractive facebook page before the release, provides certain great trailer(s) where the people provides their opinions/ reviews on that. We can get the movie popularity by the number of facebook likes, number of user providing their reviews, number of faces in movie poster and even by popularity of actors/ directors in the cast.

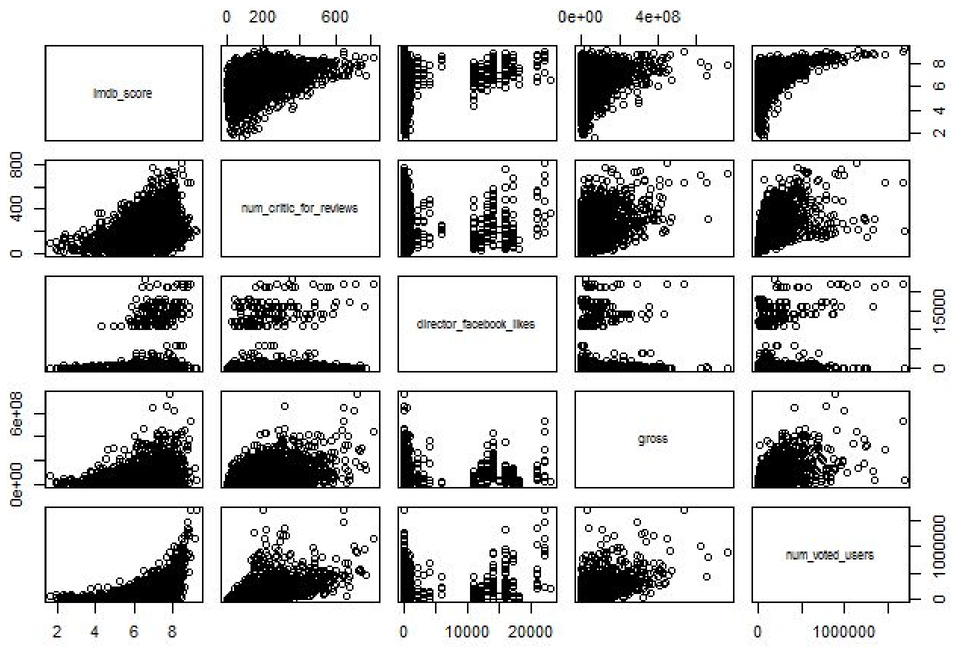
We have filtered the NA values from our data and identified if there is any unique rows or columns during the data pre-processing and make our data set appropriate before processing for analysis.

Below are some of the predictors we have used from our dataset:

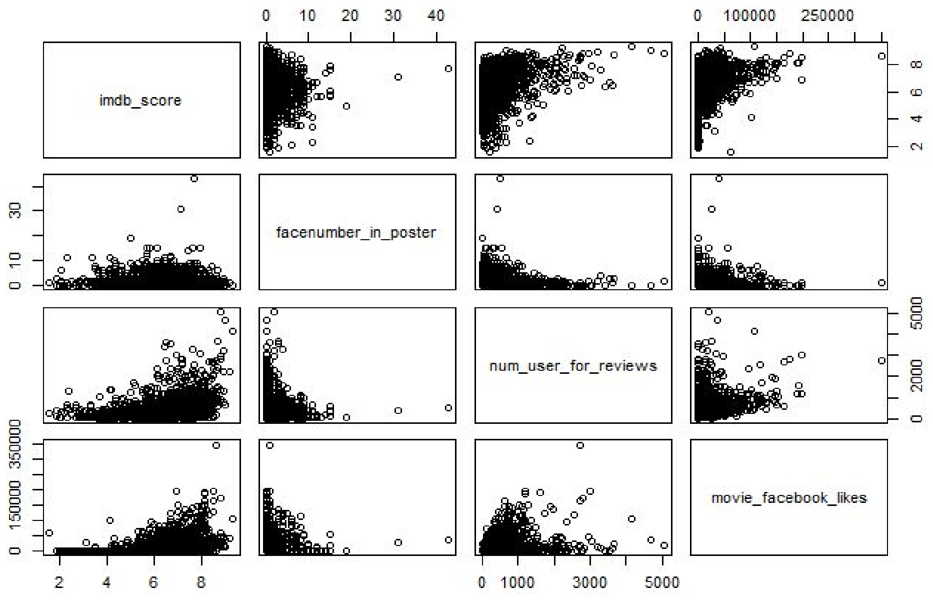
- Num_critic_for_reviews
- Director_facebook_likes
- Num_voted_users
- Gross
- Facenumber_in_poster
- Num_user_for_reviews
- Movie_facebook_likes

Below are the pair plots between imdb_score and the above said predictors:

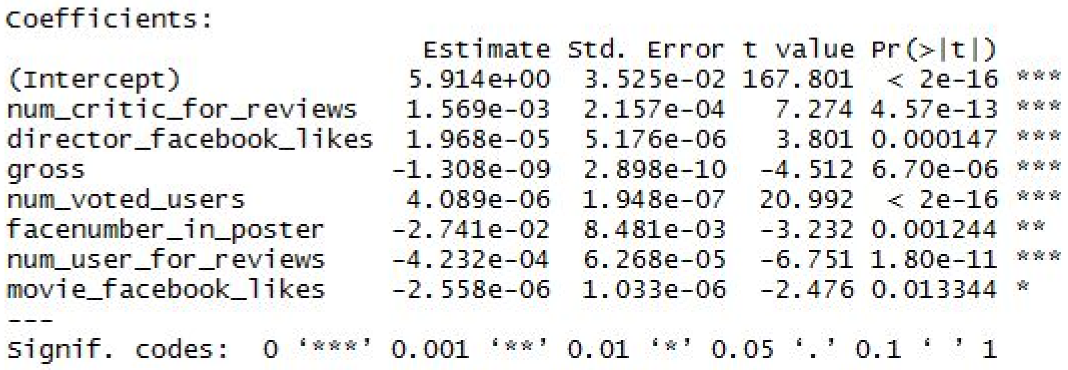
- Plot between imdb_sore and num_critic_for_reviews, director_facebook_likes, gross and num_voted_users



- Plot between imdb_sore and facenumber_in_poster, num_user_for_reviews, and movie_facebook_likes

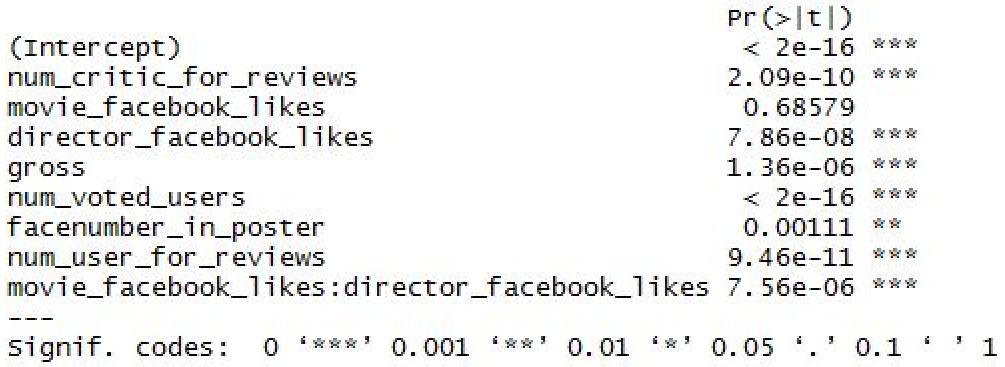


We take those predictors in different linear combinations and interactions to determine the “p” values for them to get their importance of predicting **imdb_rating**. Corresponding “p” values for them as shown below (Table 1):



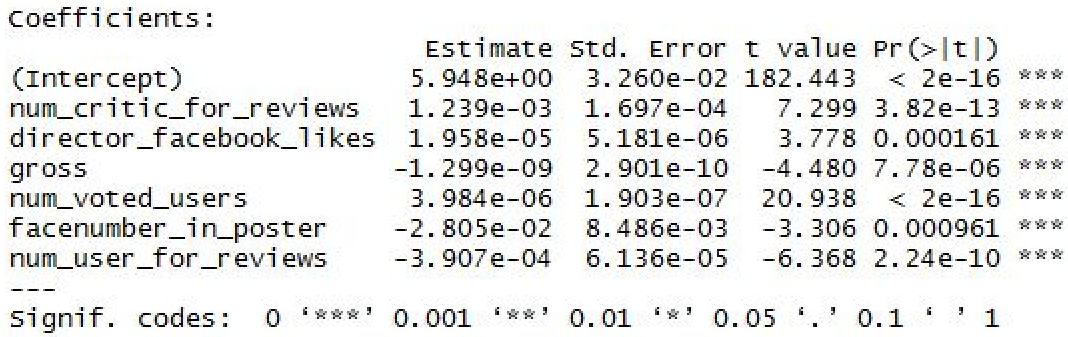
```
Coefficients:
                           Estimate Std. Error t value Pr(>|t|)    
(Intercept)               5.914e+00  3.525e-02 167.801  < 2e-16 ***
num_critic_for_reviews    1.569e-03  2.157e-04   7.274 4.57e-13 ***
director_facebook_likes   1.968e-05  5.176e-06   3.801 0.000147 ***
gross                    -1.308e-09  2.898e-10  -4.512 6.70e-06 ***
num_voted_users           4.089e-06  1.948e-07  20.992  < 2e-16 ***
facenumber_in_poster     -2.741e-02  8.481e-03  -3.232 0.001244 ** 
num_user_for_reviews     -4.232e-04  6.268e-05  -6.751 1.80e-11 ***
movie_facebook_likes     -2.558e-06  1.033e-06  -2.476 0.013344 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Since movie_facebook_likes seems to be an important predictor based on the plot diagrams so we tried to make an interaction between movie_facebook_likes and director_facebook_likes. Corresponding “p” value for the interaction becomes significant as shown below (Table 2) but significance for movie_facebook_likes decreases.



```
                                             Pr(>|t|)    
(Intercept)                                   < 2e-16 ***
num_critic_for_reviews                       2.09e-10 ***
movie_facebook_likes                          0.68579    
director_facebook_likes                      7.86e-08 ***
gross                                        1.36e-06 ***
num_voted_users                               < 2e-16 ***
facenumber_in_poster                          0.00111 ** 
num_user_for_reviews                         9.46e-11 ***
movie_facebook_likes:director_facebook_likes 7.56e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Based on the above statistics, below set of predictors (Table 3) provides a higher F statistics:



```
Coefficients:
                           Estimate Std. Error t value Pr(>|t|)    
(Intercept)               5.948e+00  3.260e-02 182.443  < 2e-16 ***
num_critic_for_reviews    1.239e-03  1.697e-04   7.299 3.82e-13 ***
director_facebook_likes   1.958e-05  5.181e-06   3.778 0.000161 ***
gross                    -1.299e-09  2.901e-10  -4.480 7.78e-06 ***
num_voted_users           3.984e-06  1.903e-07  20.938  < 2e-16 ***
facenumber_in_poster     -2.805e-02  8.486e-03  -3.306 0.000961 ***
num_user_for_reviews     -3.907e-04  6.136e-05  -6.368 2.24e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Models/ Prediction Algorithm

We have used Multiple Linear Regression Models, LDA, KNN and Single decision trees for predicting **imdb_score** / **imdb_rating**.

As we have classified the **imdb_rating** into four different levels so we have used LDA/ KNN for our approach and Multiple Linear Regression model is used basically for predicting the **imdb_score** as we have multiple predictors in data set.

Single decision trees models is used to provide more accurate results for predicting **imdb_rating.**

Prediction Accuracy

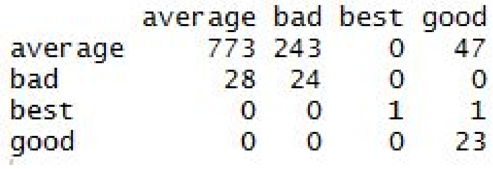
We have used the best predictors based on their “p” values as stated above on the models described earlier. So we considered taking the predictors from the table shown in 3 in Data Analysis as they seems to provide more better prediction rather than other set for our first round.

Below is the model accuracy based on the Multiple Linear Regression:

Testing data set which does not matches with their corresponding imdb_score is 0.387. So we get the accuracy of around (1-0.387)*100 = ~62%

Below is the model accuracy based on the LDA:

Confusion matrix obtained from the LDA model:

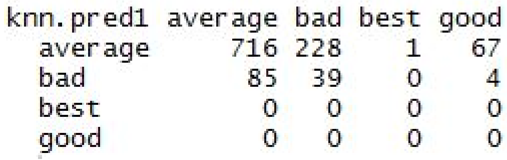


```
        average bad best good
average     773 243    0   47
bad          28  24    0    0
best          0   0    1    1
good          0   0    0   23
```

Testing data set which does not matches with their corresponding imdb_rating is 0.279. So we get the accuracy of around (1-0.279)*100 = ~72%

Below is the model accuracy based on the KNN where value of k has been chosen to 20:

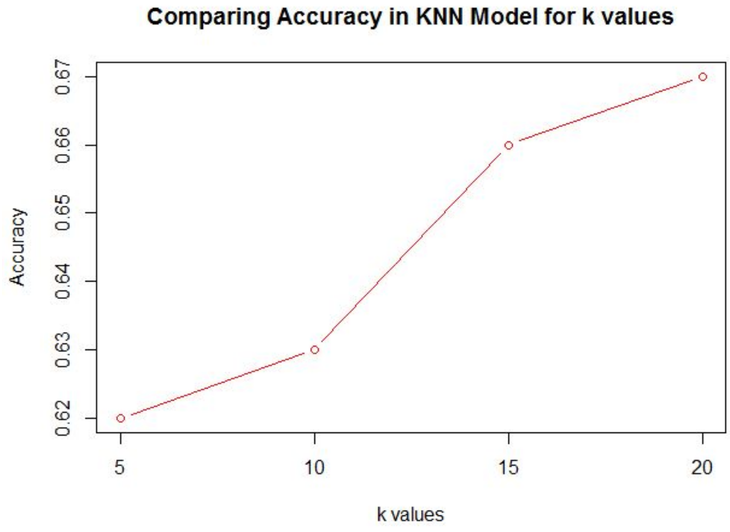
Confusion matrix obtained from the KNN model:



```
knn.pred1 average bad best good
  average     716 228    1   67
  bad          85  39    0    4
  best          0   0    0    0
  good          0   0    0    0
```

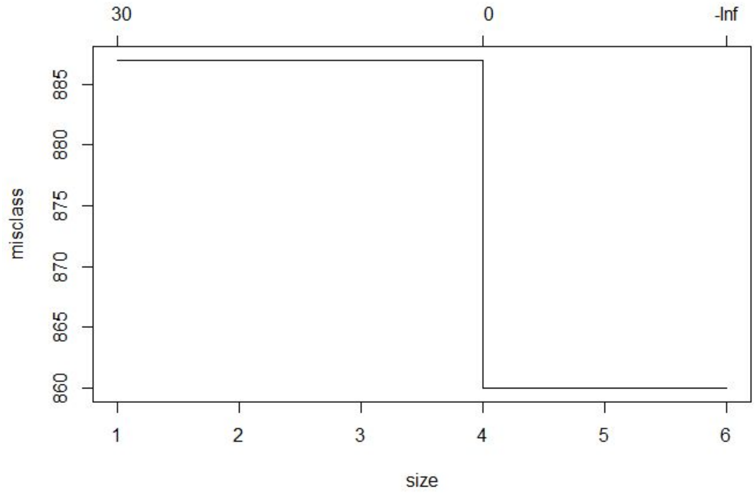
Testing data set which does not matches with their corresponding imdb_rating is 0.337. So we get the accuracy of around (1-0.337)*100 = ~67%

Why k=20 was chosen? Below chart provides a general idea for the same:

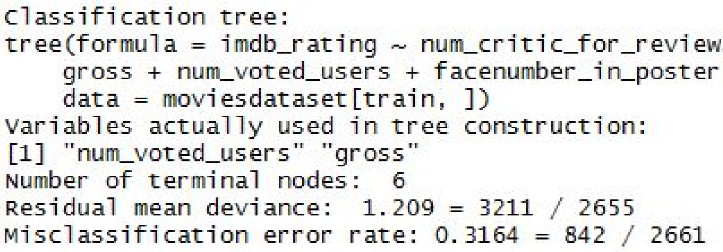


Below is the model accuracy based on the Single decision trees where we have done cross validation and then predicted the test result on the best pruned subtree:

Plot for misclassification versus size of terminal nodes as a result of cross validation:

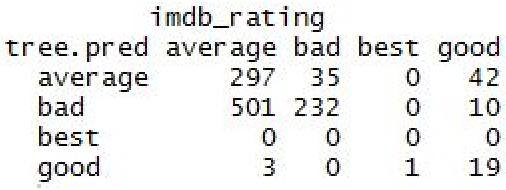


Below is the summary of the tree model obtained when size (=4) of terminal nodes are optimal:



```
Classification tree:
tree(formula = imdb_rating ~ num_critic_for_reviews + director_facebook_likes +
    gross + num_voted_users + facenumber_in_poster + num_user_for_reviews,
    data = moviesdataset[train, ])
Variables actually used in tree construction:
[1] "num_voted_users" "gross"
Number of terminal nodes:  6
Residual mean deviance:  1.209 = 3211 / 2655
Misclassification error rate: 0.3164 = 842 / 2661
```

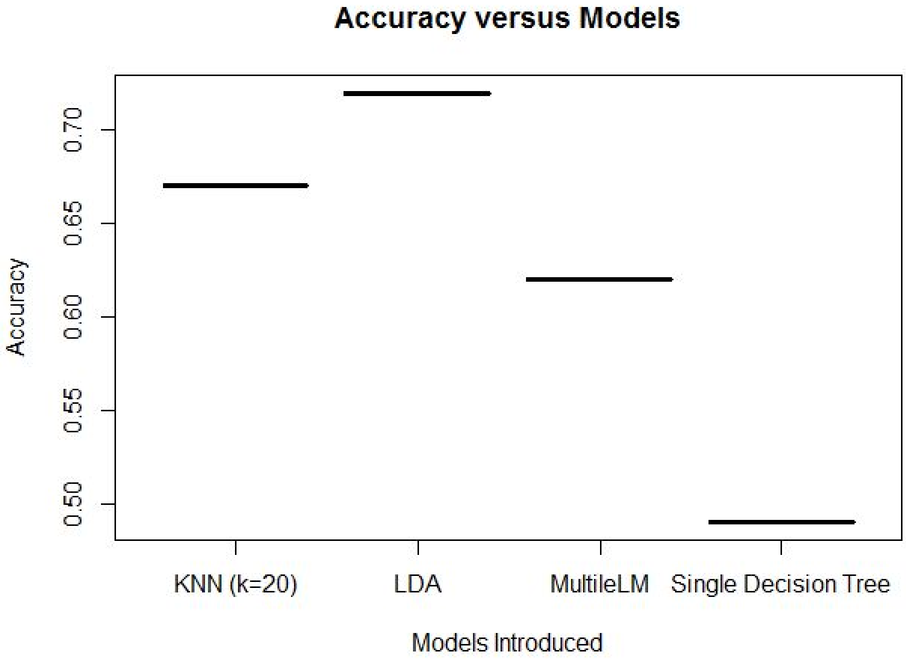
Confusion matrix found obtained from the tree model:



```
          imdb_rating
tree.pred average bad best good
  average     297  35    0   42
  bad         501 232    0   10
  best          0   0    0    0
  good          3   0    1   19
```

Testing data set which does not matches with their corresponding imdb_rating is 0.519. So we get the accuracy of around (1-0.519)*100 = ~49%

Below is the summary of accuracy [0,1] for different models used so far:



NOTE:

1. LDA was the best model determined so far with the accuracy of ~72%.
2. Even performing certain rounds/ iterations, we could not improve the accuracy of the model compared to the current LDA model.

Rounds/ Iterations

To improve the model accuracy, we have used more predictors over to top of already used predictors and have also implemented cross validation approach. We also introduced interactions between certain predictors on Multiple Linear Regression, LDA, KNN models to get the better result.

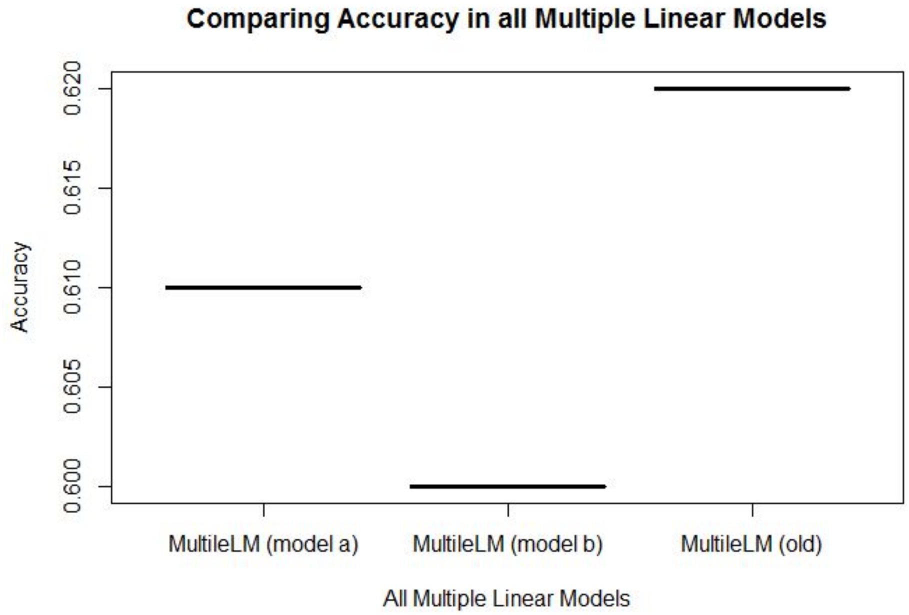
New models based on new additional set of predictors in Multiple LM/ LDA and KNN are:

1. Introducing interaction between movie_facebook_likes and director_facebook_likes: **Model a** (as seen from Table 2 of Data Analysis)
2. Introducing movie_facebook_likes as a new predictor with no interaction: **Model b** (as seen from Table 1 of Data Analysis)

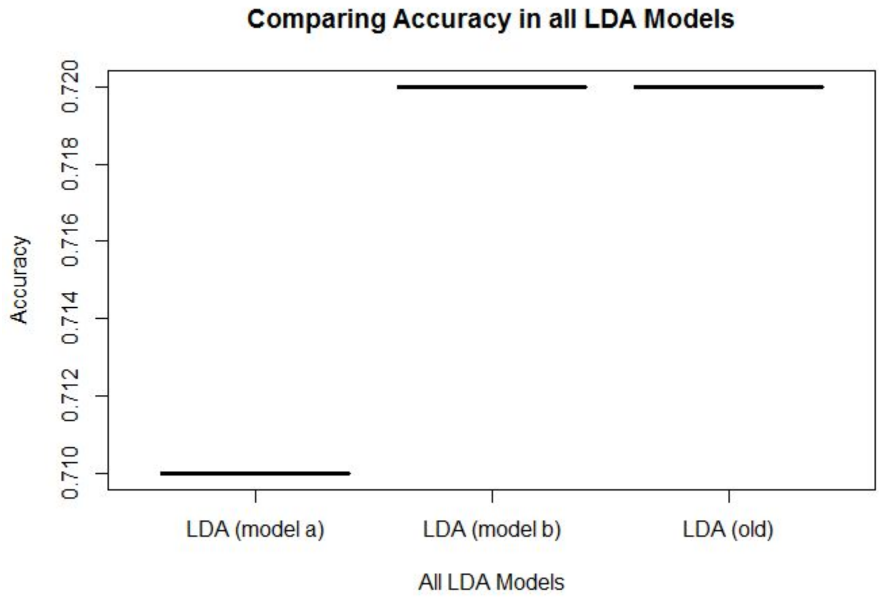
Using cross validation on Model a, b and old model for Multiple Linear Regression, we found the result to be more or less same. Value for **CV: 0.74** (k=5)**,** it was a k fold approach where k was taken to be 5, 7 and 10. Results for all these k values were more or less same.

Adding new set of predictors or CV result indicates that our model does not get better even we introduced some more features into our model.

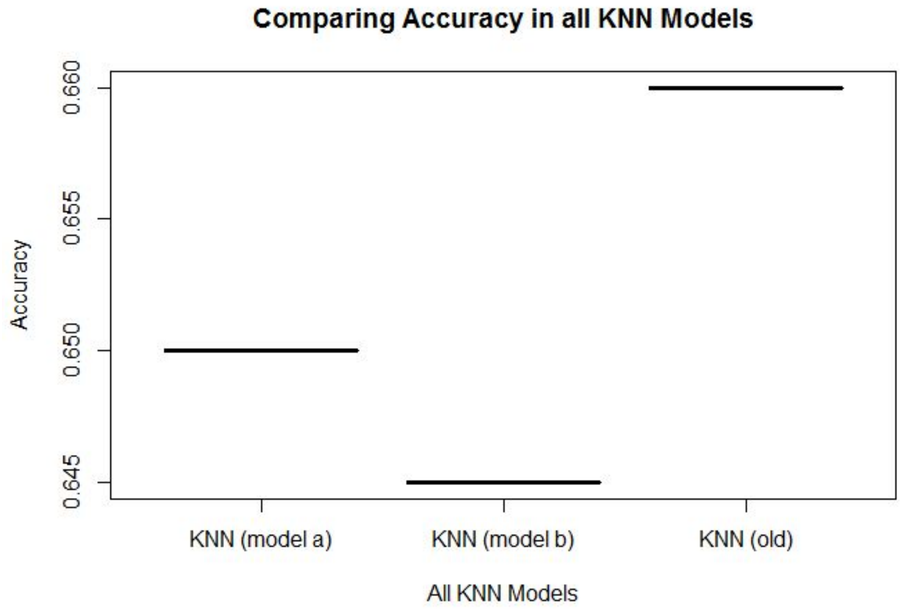
Below is the accuracy plot [0,1] for **Model a and b** based on the Multiple Linear Regression which is being compared with old model which we were using earlier for predicting the test data set. Accuracy numbers for all the three model seems to be more or less same even we introduced certain complexity.



Below is the accuracy plot [0,1] for **Model a and b** based on the LDA which is being compared with old model which we were using earlier for predicting the test data set.

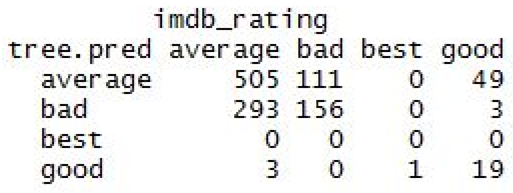


Below is the accuracy plot [0,1] for **Model a and b** based on the KNN which is being compared with old model which we were using earlier for predicting the test data set.



We have even introduced a new model for single decision tree where we chose around 14-15 predictors of data set and compared with the old single tree model (of only 6 predictors).

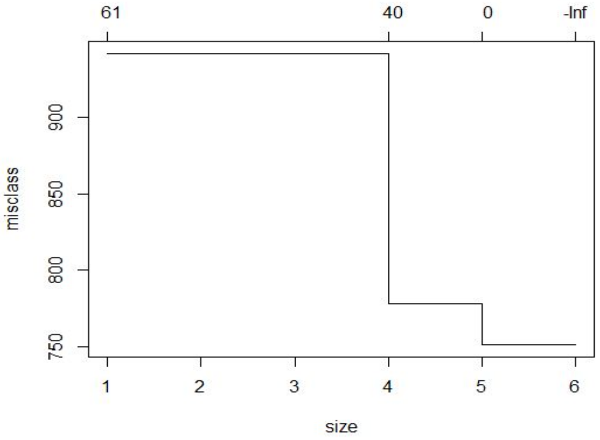
Confusion matrix found obtained from the new single tree model:



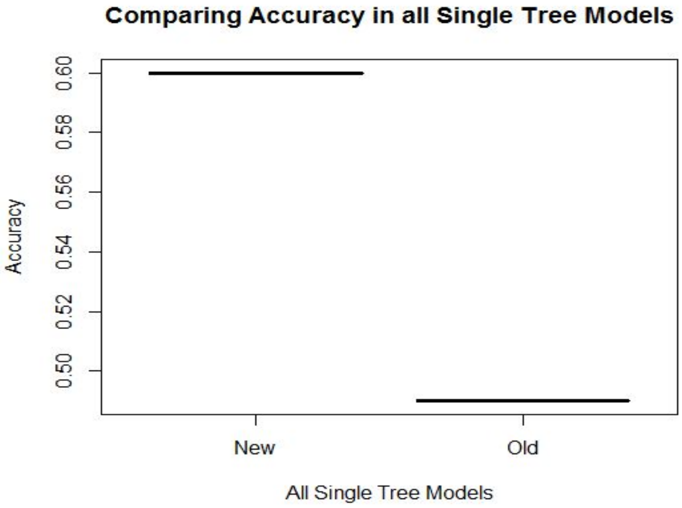
```
             imdb_rating
tree.pred average bad best good
  average     505 111    0   49
  bad         293 156    0    3
  best          0   0    0    0
  good          3   0    1   19
```

Testing data set which does not matches with their corresponding imdb_rating is 0.400. So we get the accuracy of around (1-0.400)*100 = ~60%

Plot for misclassification versus size of terminal nodes for new tree model as a result of cross validation:



Below is the accuracy plot [0,1] for **Tree Models** (Old and New Single Tree Model):



Challenges

- We have tried to use subset selection model to get a prediction of predictors to determine if the selected predictor matches with it but due to long running time of systems, we have agreed to move without taking the subset selection function into account.

- We have faced certain issue in KNN approach earlier due to the bad syntax used in KNN function. Thus, KNN took a longer time for experimenting it..

- Cross Validation approach cannot be used as a one-fold approach as the system becomes unresponsive so we used k fold cross validation approach.

Timeline

No	Item description	To do date	Status
1	Meet with team, decide on topic, features and target variable(s), split tasks	09/26	Done
2	Download data, setup git repo with team, install R packages	09/30	Done
3	Data exploration / preprocessing Project Proposal Due	10/03	Done
4	Data preprocessing	10/10	Done
5	Feature selection / creation	10/17	Done
6	Run models, evaluate accuracy, plot accuracy results (Round 1)	10/24	Done
7	Update models, run cross validation, evaluate accuracy, plot accuracy results (Round 2)	10/31	Done
8	Round 3 (with significance of results always tested)	11/03	Done
9	Project Progress Report Due	11/07	Done