IMDb Ratings Prediction

Priyanshu Porwal and Yash Saraiya
ISTM 660: Applied Predictive Analytics
4/28/21

Contents

Executive Summary

The objective of the exercise is to build a predictive model for predicting the IMDb ratings of the movie based on different features. Our goal necessitated the use of data containing IMDb ratings of more than 85000 movies and potentially relevant factors. In modern times, success of movie depends on two important factors, first is its box office collection and second is the perception in the eye of the public i.e., public opinion. We wanted to develop a predictive model which may help different stakeholders to understand different factors which play a vital role in getting a particular IMDb rating.

To achieve this, we used a publicly available dataset from Kaggle called 'IMDb movies extensive dataset' [1]. We primarily worked with two files namely movies.csv and ratings.csv which contained detailed information about movies and analysis of ratings for the movies based on the demography and so on. Based on different features we wanted to predict the IMDb rating of a movie. We started our exercise with the exploratory data analysis involving descriptive statistics and visualizations phase where we wanted to understand the data and get key insights. Based on the insights, we proceeded to the data cleaning phase where several anomalies in the data were removed, also non-relevant columns were removed.

We build our models using 4 predictive algorithms, including linear regression, k-nearest neighbors, decision trees and random forest. RStudio was used to build these models and root mean squared error (RMSE) were used to identify the best prediction model. Based on our study, we find that random forest with bagging is the best model for our use case. K-nearest neighbors performed the worst and had the highest RMSE value.

Past Research

IMDb movie ratings dataset is publicly available on Kaggle, an online community of data scientists and machine learning practitioners. Several attempts were made by different individuals to build a predictive model which will be as accurate as possible. Also, there have been several evidence of exploratory data analyses performed by individuals on the data. All the past research inspired us to use the concepts we learned in the class by Dr. Ketzenberg and try to build a predictive model which can solve the problem statement in hand. We tried to inculcate all the learnings of the class in this real-world use case. We are thankful to all the past research [1] which laid the foundation for our project.

Introduction

With the advancement of technology and digitalization of key sectors of economy there is an enormous amount of data being generated. Companies can leverage this large amount of data to get deeper insights in the areas of financial products, consumer behavior, risk analysis and marketing campaigns to name a few. Predictive modelling is a tool which can help the organizations to turn the raw data into useful information which in turn can help to achieve strategic objectives.

Predictive modeling solutions are a form of data-mining technology that works by analyzing historical and current data and generating a model to help predict future outcomes [2]. It is complex but effective and powerful tool to derive meaningful insights from the data. Predictive modeling involves collecting the data, building statistical models, predicting based on the model built and finally validating the predictions. Predictive models look at previous actions to determine how likely a consumer is to repeat it in the future [2].

Data Overview

Dataset Variables

The data set named ‘IMDb extensive dataset’ was publicly available on an online community of data scientists and machine learning practitioners called Kaggle. Total size of the data was approximately 63 MB. We worked with two files namely movies.csv and ratings.csv. The first file contained records of 85000+ movies which had details like country, year, language, genre, actor, director, average rating, number of votes to name a few. Second file contained detailed breakdown of the ratings received by the movie based on different demographic factors like gender, gender, and country.

In the beginning there were more than 70+ columns in both the files combined. After the exploratory data analysis phase, we kept only relevant columns and hence the number of columns reduced to 35. Some of the features had missing data which was handled by removing the rows.

Data Cleaning

To ensure the dataset is used to its full extent, it is important to transform columns in it that different predictive models can utilize as a predictor for the response variable. Thus, there was a considerable amount of data cleaning efforts put in ensuring all the available data could be used.

Firstly, the dataset was divided into different files and this meant that these different tables had to be merged into a single dataset. This was done by joining the tables on imdb_title_id.

Now, the columns had to be converted from their String equivalents to different formats such as factors, numeric, or Date based on information provided by the data dictionary. In addition to this conversion, new columns were also created using information from older columns such as splitting list values into separate entities such as actor1, actor2, director1, etc. as well as deriving the day of the week from release date. These efforts were applied to ensure Feature Engineering was carried out to derive additional information about movies from the available attributes.

Secondly, now that the format conversions and creation of new columns was concluded, it was time to step into individual records and fix a variety of issues in them. One major issue that was encountered was the presence of accent marks such as umlaut, acute accent, etc. that is present in European words and names. A large part of ensuring the records were consistent was handling of NA values as well as blank terms. In many cases, the empty strings were replaced with keyword of “None”, or just blank cells. These records were to be either converted to 0 or to be converted to NaN so that they were consistent across the dataset. Once, this was done, records with NA values were dropped if there was no information available in terms of its context. This was also done to ensure that the predictors fed into the algorithms did not introduce bias through these missing values.

Once this was done, the number of columns that were to be used for predicting the response variable avg_vote was to be determined. By finding correlation between these columns, out of a total of 60 columns, 27 columns were selected that showed a high correlation value with the response variable.

In the end, the dataset size was reduced considerably as the dropping of NA values ensured the input records was reduced to approximately 65,000 records as opposed to the original 80,000+ records available in the dataset.

Data Analysis

Descriptive Statistics

We use various descriptive statistics parameters to get a deeper understanding of the data. We applied descriptive statistics methods on numerical columns. Below is the screenshot which entails mean, median, different percentile values, missing values for different numerical columns like avg_vote, votes, and reviews_from_users.

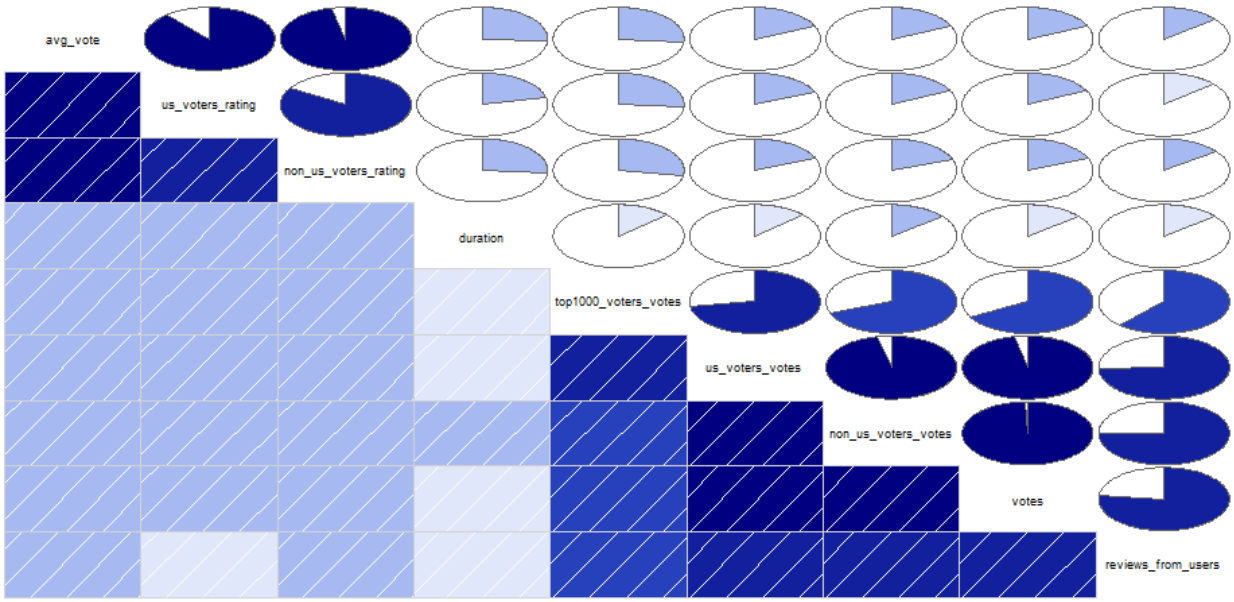
```
---------------------------------------------------------------------------------------------------------------------------
avg_vote
       n  missing distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75      .90      .95
   66052        0       86    0.999    5.966    1.337      3.6      4.3      5.3      6.2      6.8      7.3      7.6

lowest : 1.0 1.1 1.2 1.3 1.4, highest: 9.1 9.2 9.3 9.4 9.7
---------------------------------------------------------------------------------------------------------------------------
votes
       n  missing distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75      .90      .95
   66052        0    14850        1    12223    21978      132      165      294      738     2842    15841    49767

lowest :      99     100     101     102     103, highest: 1780147 1807440 2002816 2241615 2278845
---------------------------------------------------------------------------------------------------------------------------
reviews_from_users
       n  missing distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75      .90      .95
   66052        0     1213    0.998    53.07    83.8      1.0      2.0      4.0     11.0     32.0    104.9    220.0

lowest :     1     2     3     4     5, highest:  7553  7639  8232  8869 10472
```

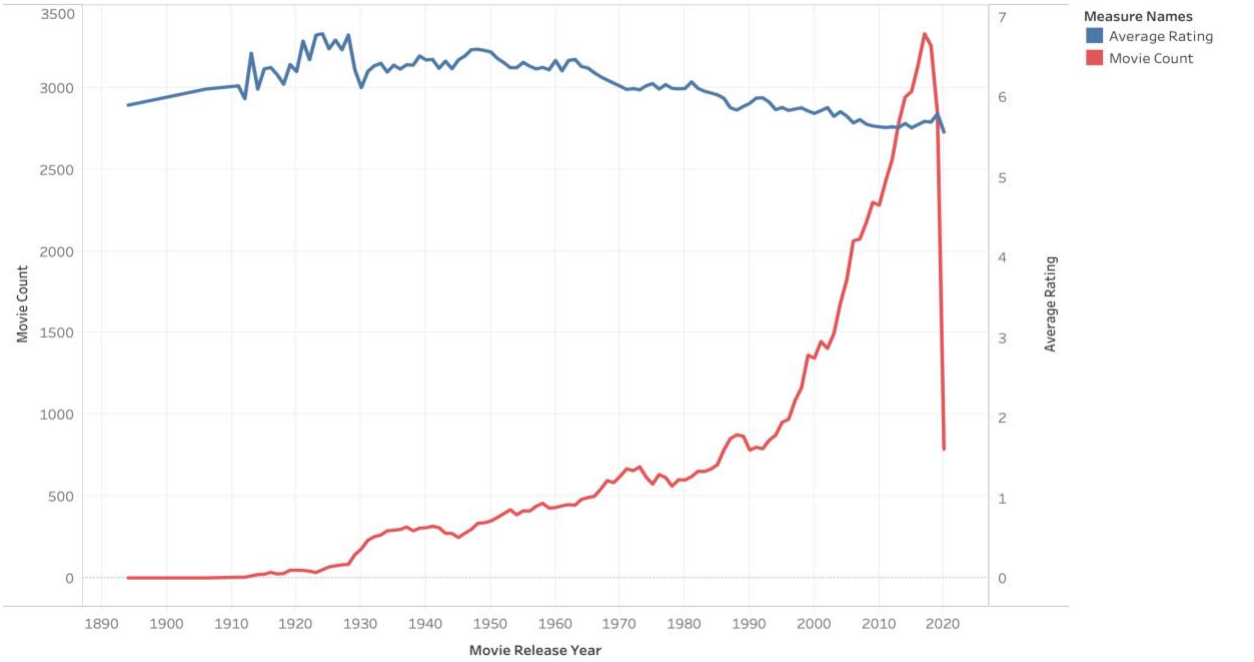
We also tried to find correlation between different features which would help us in selecting the predictor variables in different predictive algorithms which will be used in the later part of the exercise. Below is the screenshot which displays correlation between different variables.



Data Visualizations

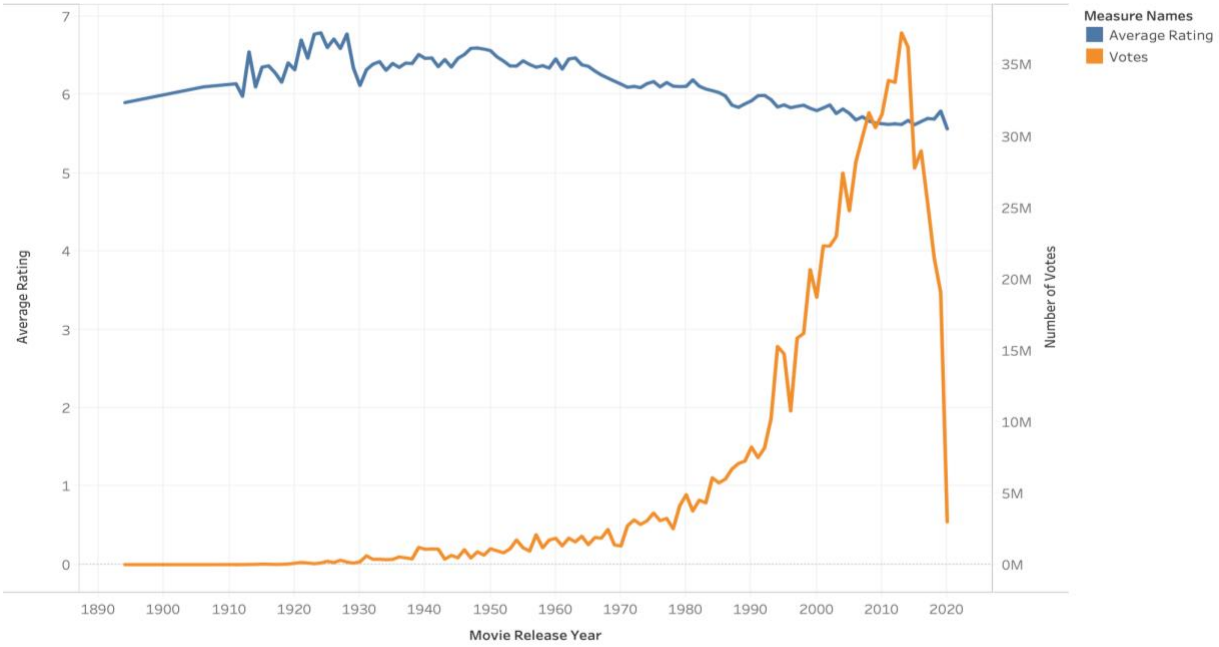
We used Tableau to create different visualizations which will help us to find pattern and get some insights from the data.

First plot shows the trend in the movie count YOY and how average ratings have changes YOY



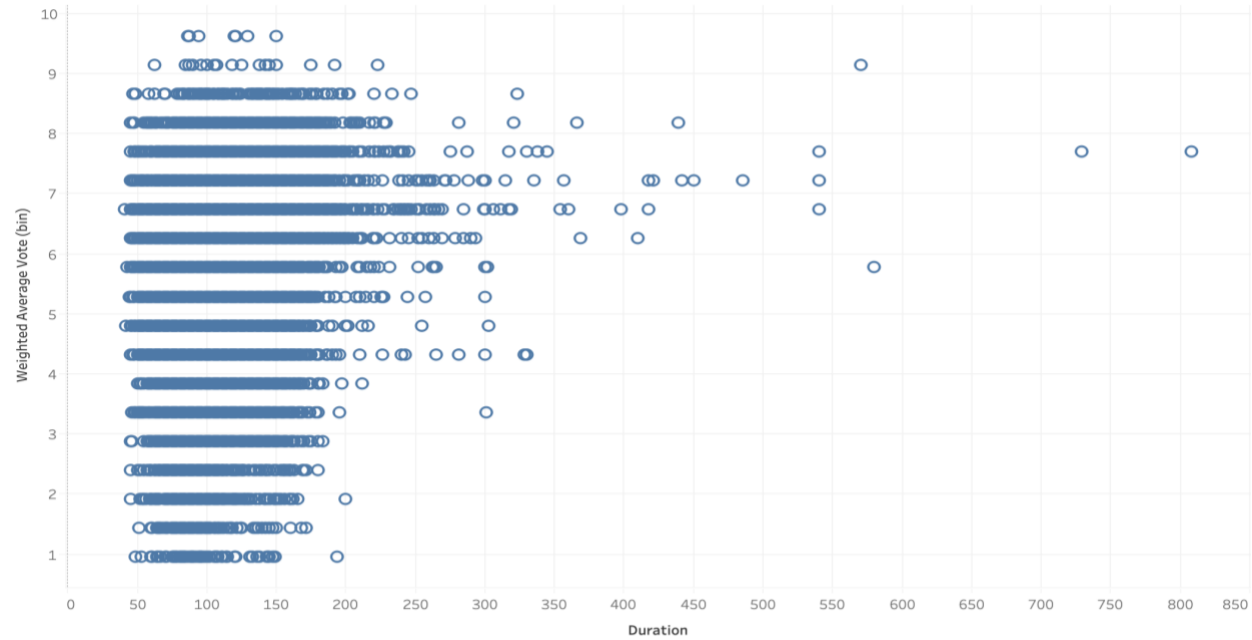
The trends of Movie Count and Average Rating for Year. Color shows details about Movie Count and Average Rating.

First plot shows the trend in the votes count YOY and how average ratings have changes YOY



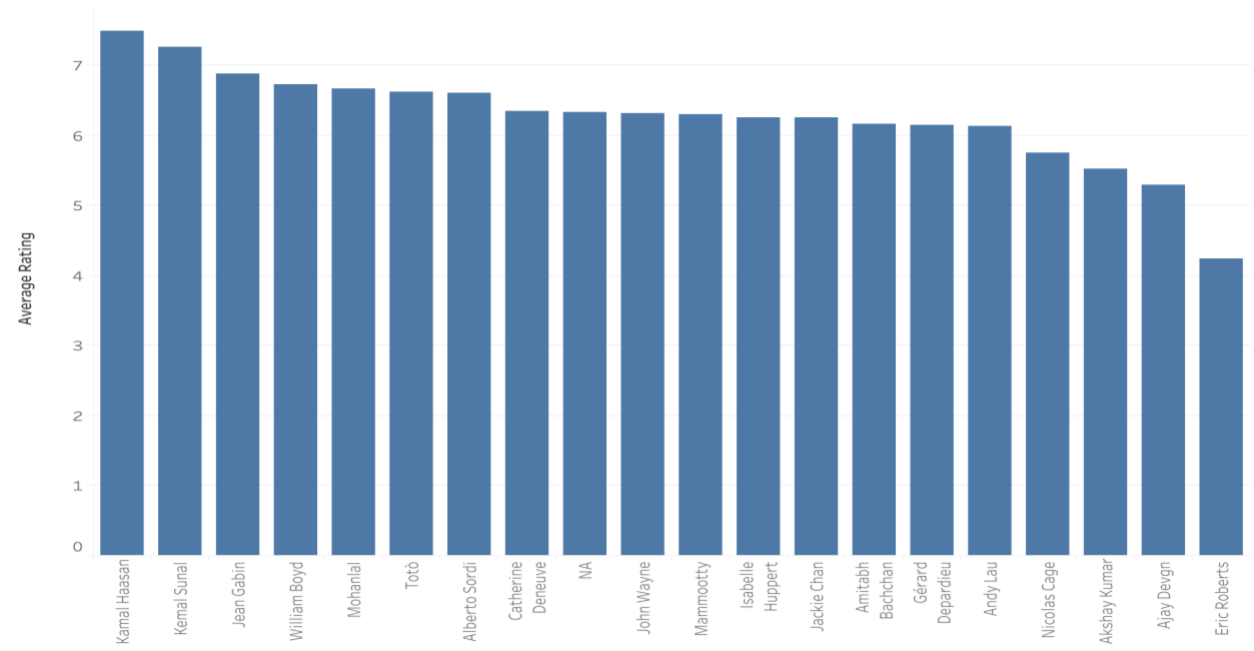
The trends of Average Rating and Votes for Year. Color shows details about Average Rating and Votes.

Third plot displays the relation between movie duration and ratings



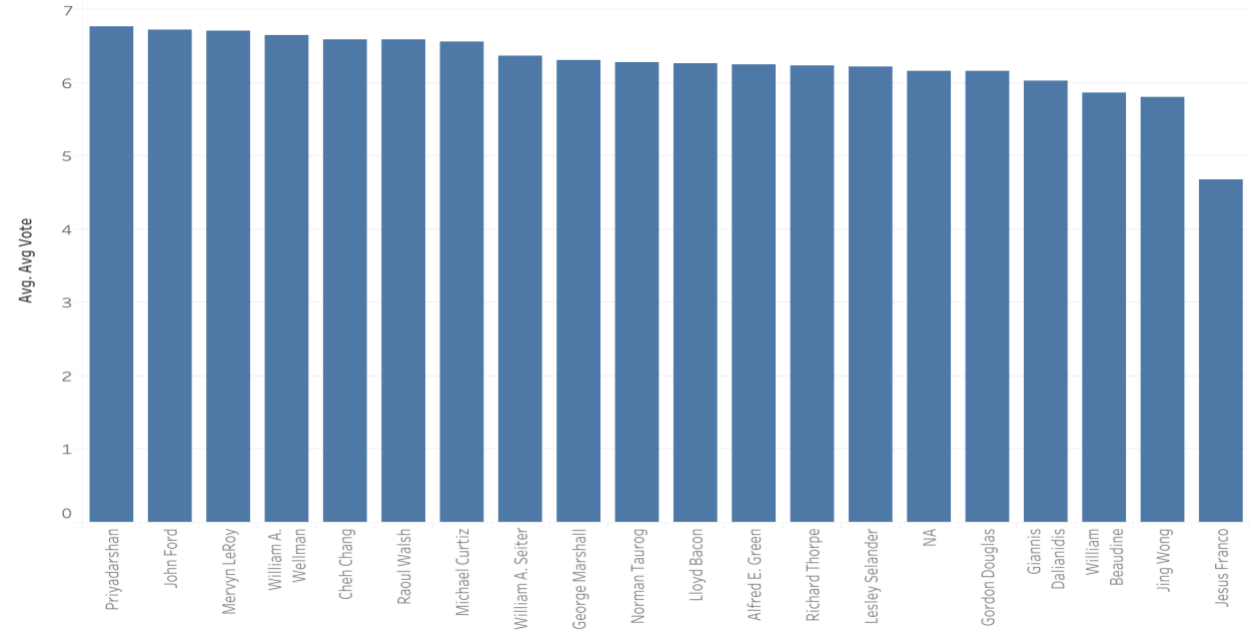
Duration vs. Weighted Average Vote (bin).

Fourth plot shows how top 20 actors in terms of no. of movies have fared on average rating parameter for their movies



Average of Avg Vote for each Actor1. The view is filtered on Actor1, which keeps 20 of 22,484 members.

Fourth plot shows how top 20 directors in terms of no. of movies have fared on average rating parameter for their movies



Average of Avg Vote for each Director. The view is filtered on Director, which keeps 20 of 23,991 members.

Predictive Modeling

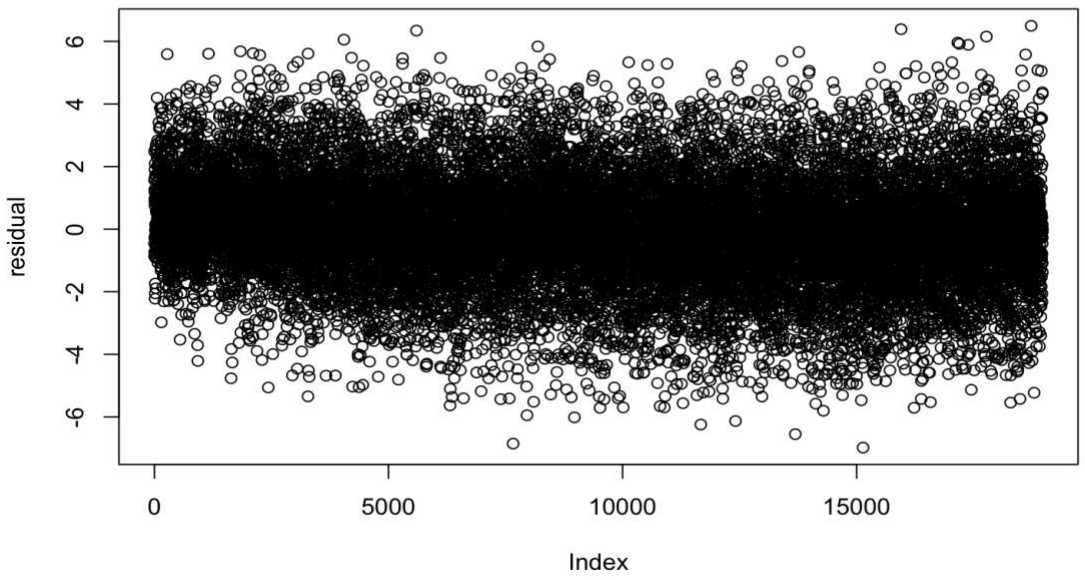
Training-Testing Split

To ensure that the comparison of the algorithms with a common metric of Root Mean Squared Error (RMSE) is applied, the splitting of test data was sampled in the ratio of 75:25. 75% of the dataset will be used to train the models and the remaining 25% will be used to test the models and compute Root Mean Squared Error for this set.

Linear Regression

Linear regression was the starting point of our predictive analysis. Initially, we had 40+ features, vif package in R was used to find the multicollinearity between different data points. Features showing multicollinearity were removed and number of features comes down to 20 which were used to build the model.

Residual plot shows the consistent spread of residual across different data points:



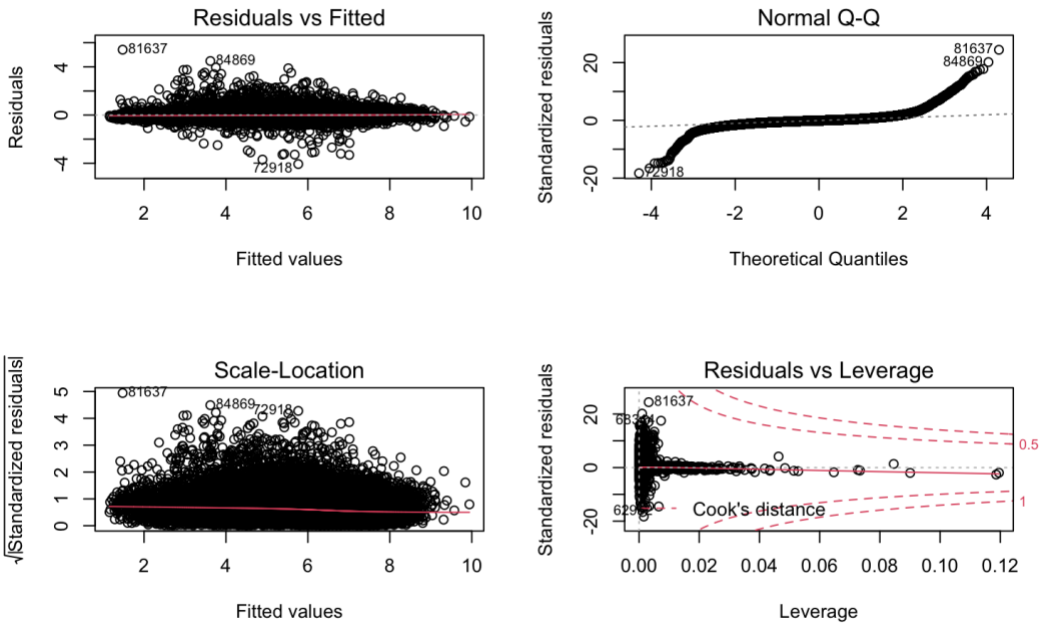
Below is the summary of the model which was built using the linear regression:

```
Call:
lm(formula = avg_vote ~ ., data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-4.0640 -0.0863 -0.0152  0.0622  5.4224

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)           -1.028e-01  8.628e-02  -1.191  0.23348
year                   8.130e-05  4.276e-05   1.902  0.05723 .
duration               7.349e-04  4.549e-05  16.153  < 2e-16 ***
votes                 -6.824e-08  4.139e-07  -0.165  0.86904
reviews_from_users     4.997e-05  9.405e-06   5.313 1.08e-07 ***
reviews_from_critics  -1.845e-04  2.670e-05  -6.910 4.89e-12 ***
males_18age_avg_vote   5.929e-02  1.276e-03  46.453  < 2e-16 ***
males_18age_votes      2.275e-06  1.473e-06   1.545  0.12240
males_30age_avg_vote   3.030e-01  2.702e-03 112.148  < 2e-16 ***
males_30age_votes     -1.159e-07  1.427e-06  -0.081  0.93525
males_45age_avg_vote   3.200e-02  1.496e-03  21.395  < 2e-16 ***
males_45age_votes     -7.489e-06  2.457e-06  -3.048  0.00231 **
females_18age_avg_vote 2.464e-02  7.422e-04  33.198  < 2e-16 ***
females_18age_votes    2.758e-06  2.388e-06   1.155  0.24829
females_30age_avg_vote 6.221e-02  1.155e-03  53.874  < 2e-16 ***
females_30age_votes   -5.271e-06  2.937e-06  -1.794  0.07275 .
females_45age_avg_vote 1.501e-02  8.506e-04  17.647  < 2e-16 ***
females_45age_votes    3.347e-05  8.551e-06   3.914 9.07e-05 ***
us_voters_rating       1.133e-01  1.516e-03  74.735  < 2e-16 ***
us_voters_votes        2.222e-06  1.105e-06   2.011  0.04431 *
non_us_voters_rating   3.794e-01  2.829e-03 134.113  < 2e-16 ***
non_us_voters_votes   -7.768e-07  1.290e-06  -0.602  0.54710
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2223 on 56890 degrees of freedom
Multiple R-squared:  0.9677,    Adjusted R-squared:  0.9677
F-statistic: 8.108e+04 on 21 and 56890 DF,  p-value: < 2.2e-16
```



Strengths

Linear regression was simple to implement and interpret. It worked as a good starting point for our project and provided us with test RMSE of 0.23 which was good.

Weaknesses

One issue we encountered with linear regression was overfitting of the data. We used vif package to overcome the problem by removing features which were showcasing multicollinearity. Our data set contained features having non numerical vales, linear regression didn't had any means to handle it the way random forest does.

KNN Regression

K-Nearest Neighbors (KNN) model was built with a 75/25 train test split on the data and a seed set at 25. Standardization of the variables were done to cutoff the possibility of higher numerical value of a variable affecting the prediction results. It led to variables having value between -1 and 1. KNN model for prediction uses 'feature similarity' to predict the values of the data points based on the given data. Selecting the proper value of K is very critical in the performance of the model, hence, model performance was measured for different values of K starting from 1 to 100. KNN model with K=14 gave the best model performance in terms of RMSE. Mean value of the data in the testing set was around 6.07 and the RMSE for K=14 was around 0.90.

Strengths

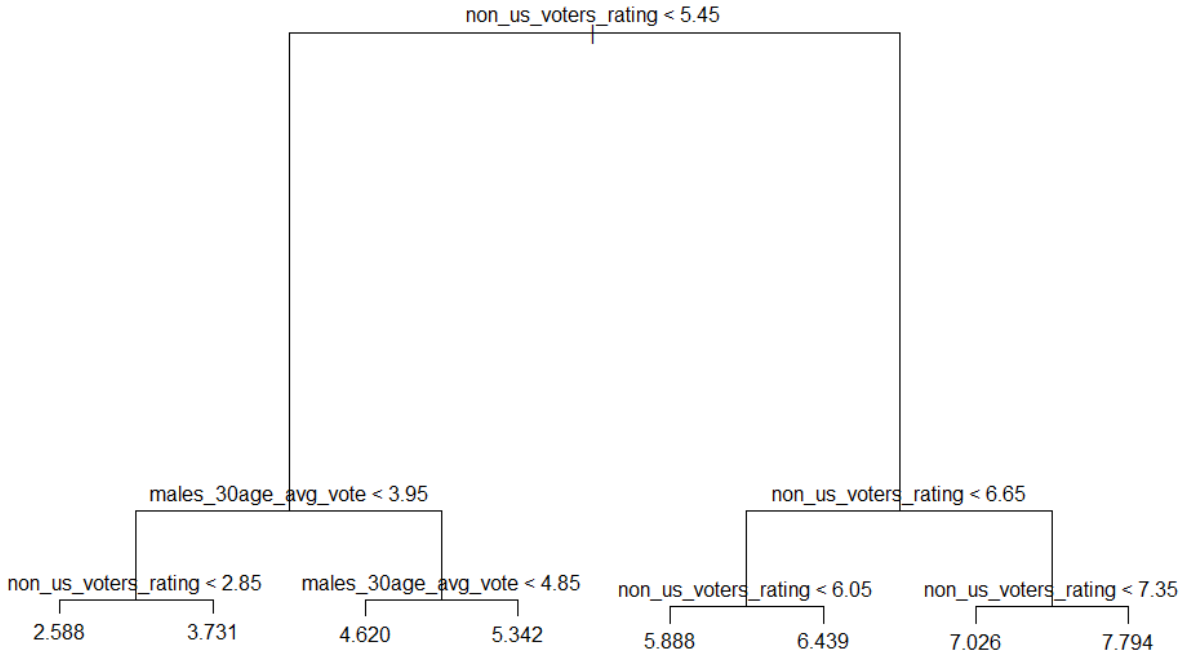
In general, KNN model is simple and intuitive. Also, KNN model is easy to implement for the small size of data. In our case, KNN served as a good starting point from which we can build on models with better performance which can overcome the drawbacks of KNN.

Weaknesses

When it comes to weakness of using KNN model in our problem statement, we encountered quite a few to them. Firstly, processing time for KNN models is a main concern for large dataset. Also, KNN is particularly not a good algorithm for noisy data as it can lead to models with bad performance.

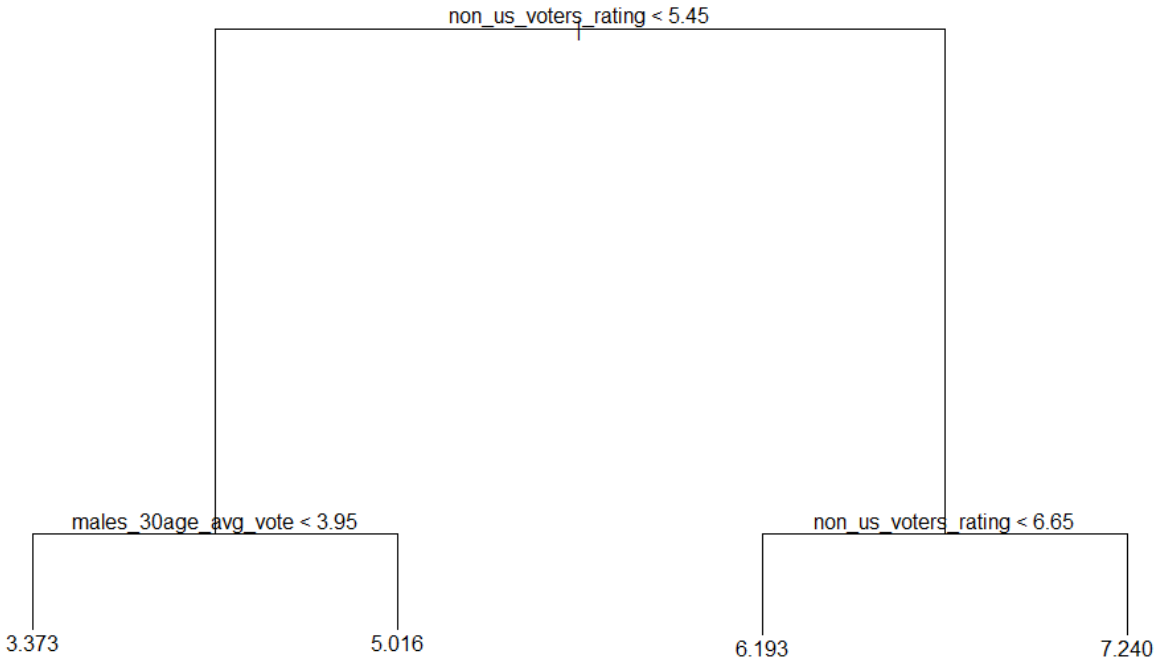
Decision Tree

Decision Tree was one of the first tree-based models used to predict the response variable of avg_vote. Using this method, the below Decision Tree was generated by using the predictors that were generated after the data cleaning procedures.



The Root Mean Squared Error (RMSE), the common metric used for determining the performance of models across the project, turned out to be 0.3422. This was lower than the RMSE of KNN regression and Linear regression and thus, Decision Tree was one of the best predictors so far in the project.

Next, to reduce the problem of overfitting, cost complexity pruning was carried out on the above decision tree. Using this, the tree was reduced to:



Here, the main parameters to affect a movie's IMDb rating were shown to be the effect of Non-US voters followed by the effect of Males in the age bracket of 30-45. This increased the Root Mean Squared Error to 0.48 but tackled the problem of overfitting.

Strengths

Decision Tree based models do offer higher interpretability by displaying how data points are classified, even in a regression problem.

Weaknesses

Like any other highly interpretable models, Decision Trees are not known for their predictability. This can be seen with a more complex model of Random Forest in the next section where the interpretability is traded-off for better predictive performance.

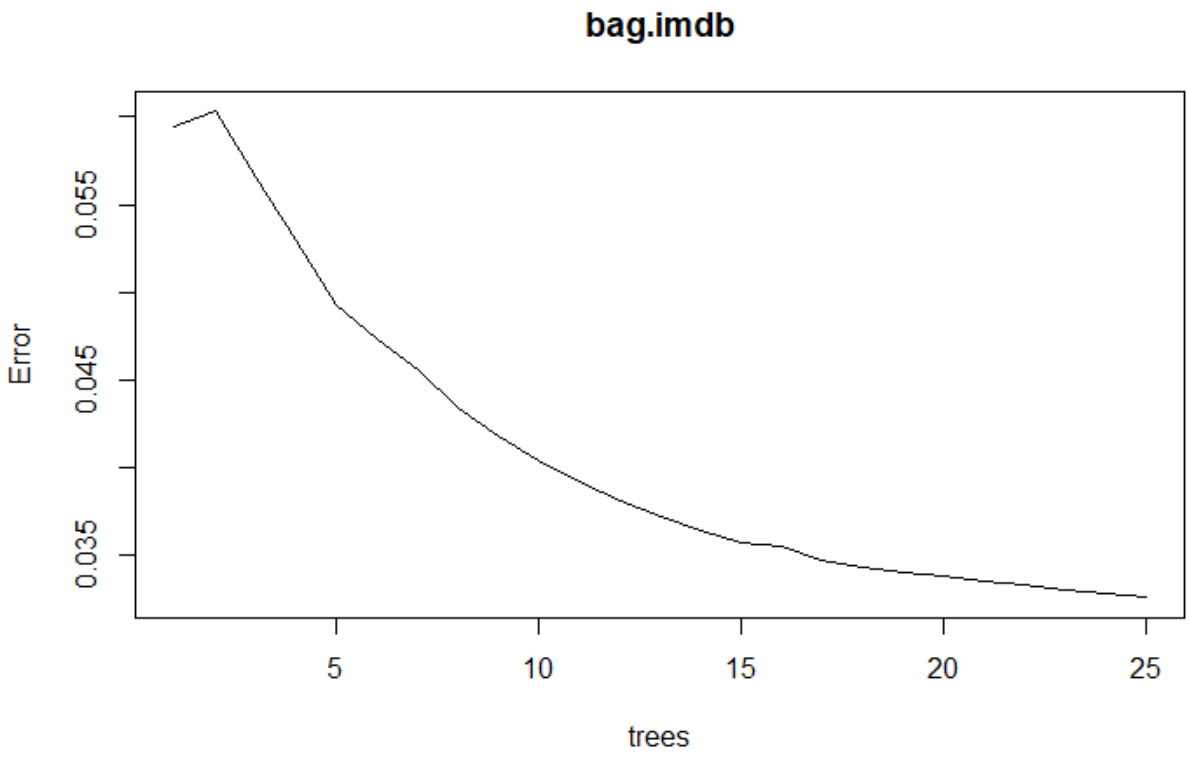
Random Forest - Bagging

Bagging, where we use all predictors, gave us a lower test MSE than all alternatives we had seen before. Using mtry=26 (all predictors), the Bagging algorithm was able to explain 97.82% of all variance in data. The ntree variable was set to 25 as the volume of data was much larger, and the computation time was exceeding both time and memory constraints.

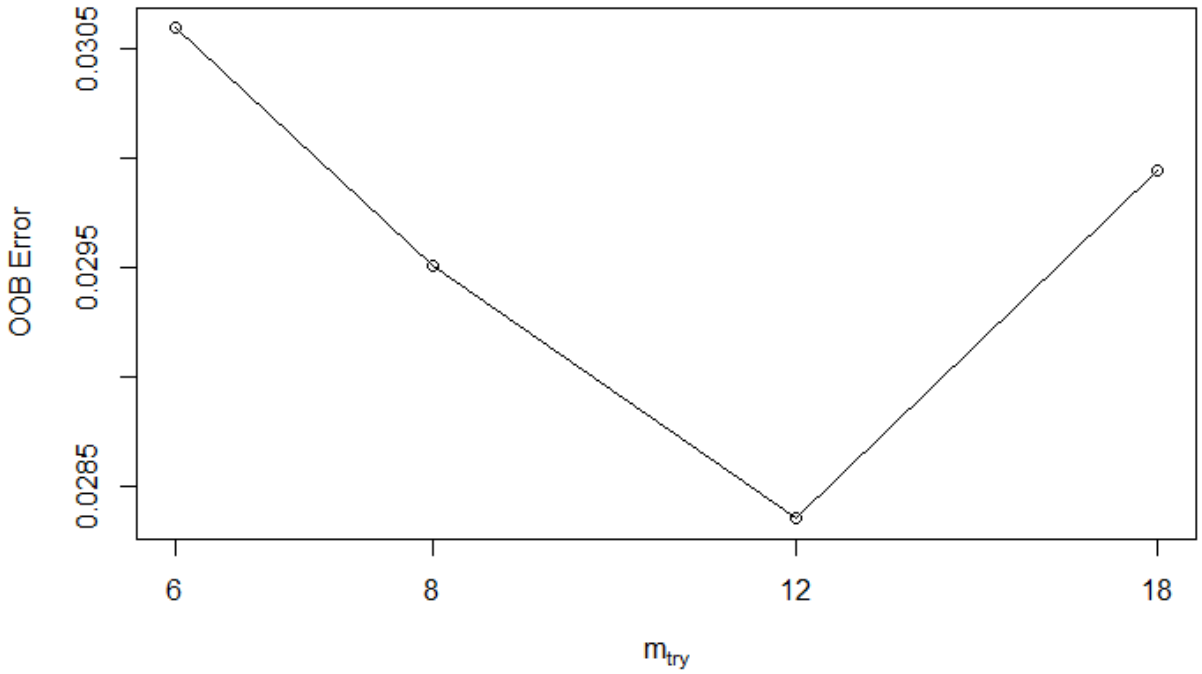
```
> bag.imdb

Call:
 randomForest(formula = avg_vote ~ ., data = imdb, mtry = 26,      ntree = 25, importance = TRUE, subset =
 train)
               Type of random forest: regression
                     Number of trees: 25
No. of variables tried at each split: 26

         Mean of squared residuals: 0.03201631
                   % Var explained: 97.82
```

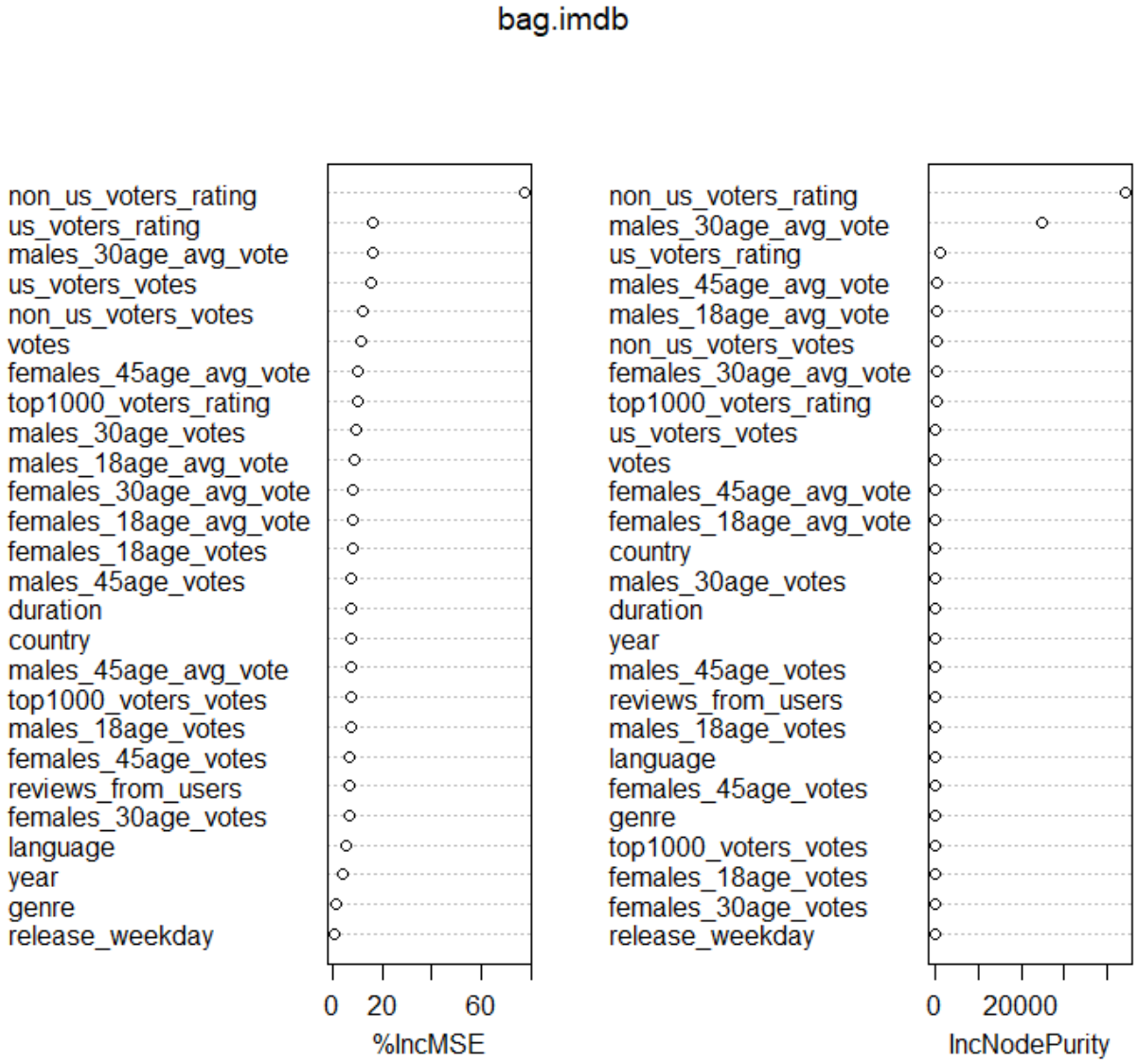


Using tuneRF tuning, the best mtry value was 12 as for this value of mtry, the OOB Error was at a minimum.




```
> bestmtry <- tuneRF(x, y, stepFactor=1.5, improve=1e-5, ntree=25)
mtry = 8  OOB error = 0.02950958
Searching left ...
mtry = 6        OOB error = 0.03059301
-0.03671466 1e-05
Searching right ...
mtry = 12       OOB error = 0.02834922
0.03932148 1e-05
mtry = 18       OOB error = 0.02994009
-0.0561171 1e-05
> print(bestmtry)
   mtry   OOBError
6     6 0.03059301
8     8 0.02950958
12   12 0.02834922
18   18 0.02994009
```

Importance variable would highlight the best variables and provide us insights into which variables could be discarded further.



```
> importance(bag.imdb)
                         %IncMSE IncNodePurity
year                    3.563946      54.74074
genre                   1.439497      35.13889
duration                7.424174      56.52144
country                 7.394808      67.31613
language                5.335146      37.30570
votes                  10.959760      82.58048
reviews_from_users      6.685813      44.44191
males_18age_avg_vote    8.563422     128.52275
males_18age_votes       6.948380      39.31786
males_30age_avg_vote   15.983450   24708.30725
males_30age_votes       9.250100      57.81922
males_45age_avg_vote    7.032717     275.07701
males_45age_votes       7.554419      51.13200
females_18age_avg_vote  8.131263      71.74989
females_18age_votes     7.689121      28.27596
females_30age_avg_vote  8.207202     113.35721
females_30age_votes     6.307385      25.24234
females_45age_avg_vote  9.910235      79.45592
females_45age_votes     6.686750      35.94383
top1000_voters_rating   9.864484     108.01633
top1000_voters_votes    6.962332      30.48576
us_voters_rating       16.088613    1261.45352
us_voters_votes        15.381458      85.91538
non_us_voters_rating   77.463324   44122.10311
non_us_voters_votes    12.085409     121.32510
release_weekday         0.612606      13.02207
```

Here, we see that variable release_weekday and genre has little to no significance in determining the success of a movie. The variable genre not having any effect is as expected, however, release_weekday being the variable of least importance in the dataset is a new discovery. Lastly, the Root Mean Squared Error is 0.16 which is considerably better than Decision Tree based models.

Strengths

Unlike Decision Tree, as Random Forest implements multiple Decision Trees and averages values to reduce variance in its results, it gives a much better predictive performance. This can also be seen from the low RMSE value of 0.16.

Weaknesses

However, the predictive performance comes with a byproduct of the model losing its interpretability.

Model Comparison

On comparing the metrics of the models: Root Mean Squared Error (MSE), we can draw the below table of its values.

Sr. No.	Predictive Model	Root Mean Squared Error
1.	Linear Regression	0.23
2.	KNN Regression	0.90
3.	Decision Tree	0.48
4.	Random Forest - Bagging	0.16

Hence, from this single metric, we can conclude that Random Forest offers the lowest Root Mean Squared Error in comparison to the other models followed closely by Linear Regression. KNN Regression and Decision Tree based models do not offer high enough predictability in this implementation and are thus to be rejected.

With the difference between Linear Regression and Random Forest being so little, we have concluded that as of this moment, we will consider Random Forest as the solution for the moment. However, with future iterations of the project, Linear Regression will once again be tested with coming iterations as it can offer higher interpretability as well, which is an added benefit.

Next Steps

In the next steps for our implementation, we shall work exclusively on resolving the issues with factors. To use columns such as actor, director, writer as a factor, the levels of the factors are to be decreased. As of now, there are 1000+ levels for each of these columns. As there are so many levels of these factors, the algorithms cannot use these predictors to predict the IMDb score of any movie.

Additionally, there were 15,000+ records discarded as there were NA values present in them. The next step here is to perform imputations in these records to ensure usage of more data records. This means the model's efficiency improves with more data available for it to predict with.

Lastly, further data analysis is required to determine if any other columns can be derived and added into the dataset by the virtue of Feature Engineering. With this, there is also the opportunity to replace or remove current variables that add unnecessary bias to the prediction. In the future, shrinkage methods can also be applied to test and reduce variance.

Conclusion

The objective of being able to predict IMDb ratings using a publicly available dataset was effectively achieved. We deployed, tested, compared, and analyzed four major predictive modeling methods: Linear Regression, KNN Regression, Decision Trees, and Random Forest algorithms. The computations and data cleaning were all performed using RStudio and R.

Before comparing these algorithms, the problem was recognized as a regression problem and hence, the metric that was chosen to compare these algorithms was: Mean Squared Error (MSE). Using this metric, we were able to determine the best algorithm to be Random Forest, followed closely by Decision Tree.

This study also recognized which variables in the dataset were of high importance for usage in predictive modeling. This offers insights not just for application of predictive modeling but also for other stakeholders who want to shape the public opinion by the way of online ratings of movies on IMDb.

References

1. https://www.kaggle.com/stefanoleone992/imdb-extensive-dataset
2. https://www.gartner.com/en/information-technology/glossary/predictive-modeling