IMDB Data Mining Project

Richard Joerger
Rochester Institute of Technology
raj2348@g.rit.edu

Yifei Sun
Rochester Institute of Technology
ys8800@g.rit.edu

Ajeeta Khatri
Rochester Institute of Technology
ak6038@g.rit.edu

Zhuo Liu
Rochester Institute of Technology
zl9901@g.rit.edu

1. OVERVIEW

The purpose of this paper is as an update of our progress on the data mining portion of the assignment. It will contain: Our revised thoughts on our data set, the progress we've made in quantizing features, what our plan is moving forward. These things boil down quite simply. We think switching our data set will aid in advancing our progress, we've quantized the portions of the data which make sense, and we plan to switch our focus to categorizing our data as soon as possible.

2. DATA SET CHANGE

The original data set we had in mind was the same data set that we used for our data management section, the IMDb archives. We were reminded quite quickly about how messy that data set is. We were also recommended to go out and try to find more data which could be used in tandem with our existing data set to get a greater number of attributes to try and use as additional features. Instead we've decided to switch data sets entirely. The main reason for this is that the data set we're switching to, The Movie Database data set, has a greater number of attributes. Even though the data set is much smaller, with only 5000 entries, it is also readily apparent that the data is organized in a far more logical fashion which should also help us avoid getting stuck in a mess of cleaning and preprocessing our data and instead allow us to focus on the actual mining portion. The data set is very similar to the IMDb data set in that it has information about the title, the original language of the film, release date, run time, average rating, cast and production information, and vote count. What we believe makes this data set better is the keyword section, popularity, production company, budget, and revenue attributes. These added attributes provide us with a different domain to work with when it comes to actually classifying a movie with it's estimated rating.

3. DATA PREPARATION

In order to analyze and use the dataset we choose, we need to preprocess the data. We converted the original data into the way we want. Because in the original data, many attributes are stored in the form of json data.

3.1 Data Importing

At first, We need import the dataset in R from csv file.

```
setwd(" /tmdb-movie-metadata")
movie = read_csv("tmdb_5000_movies.csv",
col_names = TRUE, na = "NA")
credits = read_csv("tmdb_5000_credits.csv",
col_names = TRUE,na = "NA")
```

3.2 Data Cleaning and Arrangement

Since we observed that there are bad data in the database, data cleaning have to be done. The first part of data cleaning involves removal of spurious characters (Â) from a the movie title, genre and plot keyword columns. This might be because we have scrapped the data from the net.

```
movie$title <- (sapply(movie$title,gsub,pattern =
"\\Â",replacement = ""))
```

Next step included removing duplicate data. Duplicate data will skew our analysis hence needs to be removed.

```
movie <- movie[!duplicated(movie$title), ]
```

I tried creating different dataframes for extracting data from the json object. I use jsonlite library to extract the data. Then created comma seperated columns for 'keywords', 'genre', 'production_countries', 'production_companies', 'spoken_languages' for each movie object and combined them in the main movie data. You can check the code in our script file. Here shows how we create a genre dataframe.

```
genres <- movie %>% filter(nchar(genres) > 2) %>%
mutate( js = lapply(genres, fromJSON)) %>%
unnest(js,.names_repair = "check_unique") %>%
select(id, title, genres = name)
genres <- aggregate(genres  .,data = genres, paste,
collapse = ",")
```

In the end, we combine these columns in the main movie data.

```
movies <- subset(movie, select = -c(genres, keywords,
production_companies, production_countries, spoken_languages))
movies <- movies %>%
full_join(keywords, by = c("id", "title")) %>%
full_join(genres, by = c("id", "title")) %>%
full_join(production_companies, by = c("id", "title")) %>%
full_join(production_countries, by = c("id", "title")) %>%
full_join(spoken_languages, by = c("id", "title"))
```

4. REVISED DATA SET CONJECTURES

One of the things that happens as you begin to understand a data set is you start connecting the dots between the different aspects of the data. For example, since time has gone on more people have probably engaged with newer

entertainment products which means that more people will have a greater variety of opinions and therefore those entertainment products will have a score that is far closer to the middle. Additionally, we believe there will be a correlation between average rating and various other factors. Looking at figure 1 we can see the distribution that shows the relationship between different attributes of the dataset. Further in figure 2. we can observe that the Count of the movies is normally distributed across Average vote(rating) with a mean of around 6. In figure 4 we see a very interesting relation between the popularity and vote count for various titles. We can also observe the relationship between the avg ratings for the movies based on a particular original language. Finally we also see the summary of the entire dataset.

5. DATA MINING APPROACH

What we are aiming to mine from our data is the rating of a movie given some set of features. The features we're looking to use are: the number of movies the top 5 actors have acted in, the number of movies the director has directed, the number of movies the producer has produced, the budget of the film, the revenue of the film, the popularity of the film. If we're feeling particularly ambitious we may even try to quantize the vector attribute to better understand how genre plays a roll in the rating of the movie.

We will begin by quantizing all of the data which we believe will help us to better predict what the rating of the movie will be. From there, we will split up the data into three sets. The first and the largest will be the training data set. This will be used to train our model and do some early stage testing to verify that our model, and indirectly our features, are capable of predicting the rating of a movie. The next set of data will be the validation set. This will be used to fine tune our model with new data which it hasn't seen before. The last set of data will be the testing data. This is the data we will use to asses the quality of our model for data which it hasn't seen before. Our approach for assessing the performance of our model will be rather straight forward. We will calculate the accuracy of the model, the F1 score, precision, and recall. We believe these metrics will provide us with more than enough information to guarantee that our model not only predicts values that are right but that it does so consistently without making massive errors. Up to this point we haven't discussed which model we plan to use so we've

5.1 Feature Selections

We have begun our discussion of what our feature set should be. Since our goal is to classify the rating of a movie we have to define features which do not involve rating. If we were using our original data set we would be very limited on what kind of features we could use but with the new data set we can use popularity of the movie, the revenue of the movie, and budget as features. We can then calculate some additional features from the existing attributes. For example, one of the features we've discussed is counting the number of movies each actor in the movie has been in and then using the number of movies the 3 most prolific actors have been as feature. We could also do the same with the directors and producers. We could also use the run time of the movie as an indicator. Another thing we could use, if we successfully quantize it, is the genres which the movie is. We haven't finalized the features as we haven't finalized exactly to what extent we're going to quantize the data.

6. TRAINING MODEL

In statistics, the logistic model is used to model the probability of a certain class or event existing. This can be extended to model several classes of events. Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression is estimating the parameters of a logistic model.

Logistic regression is a widely used technique because it is very efficient, does not require too many computational resources, it doesn't require input features to be scaled and doesn't require any tuning. Logistic regression is easy to regularize and it outputs well-calibrated predicted probabilities. Like linear regression, logistic regression does work well when remove attributes that are unrelated to the output variable as well as attributes that are very similar to each other. The disadvantage of logistic regression is that we can't solve non-linear problems since it is decision surface is linear. Logistic regression is also not one of the most powerful algorithms out there and can be easily outperformed by more complex ones. Since its outcome is discrete, logistic regression can only predict a categorical outcome and it is known for vulnerability to overfitting.

A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resources costs, and utility. It is one way to display an algorithm that only contains conditional control statements.

Compared to other algorithms decision tree requires less effort for data preparation during pre-processing. A decision tree does not require normalization and scaling of data. Missing values in the data also does not affect the process of building decision tree to any considerable extent. A decision tree model is very intuitive an easy to explain to technical teams as well as stakeholders. But a small change in the data can cause a large change in the structure of the decision tree causing instability. For a decision tree sometimes calculation can go far more complex compared to logistic regression. It often involves higher time to train the model so it is relatively expensive as complexity and time taken is more. Decision tree is inadequate for applying regression and predicting continuous values.

Logistic regression and decision tree differ in the way that they generate decision boundaries. Decision tree bisects the space into smaller and smaller regions, whereas logistic regression fits a single line to divide the space exactly into two. Of course for high dimensional data, these lines would generalize to planes and hyperplanes.If two classes are separated by a decidedly non-linear boundary, decision tree can better capture the division, leading to superior classification performance. When classes are not well-separated, decision tree is susceptible to overfitting the training data, so that logistic regression's simple linear boundary generalizes better.

7. WORK LOAD DISTRIBUTION

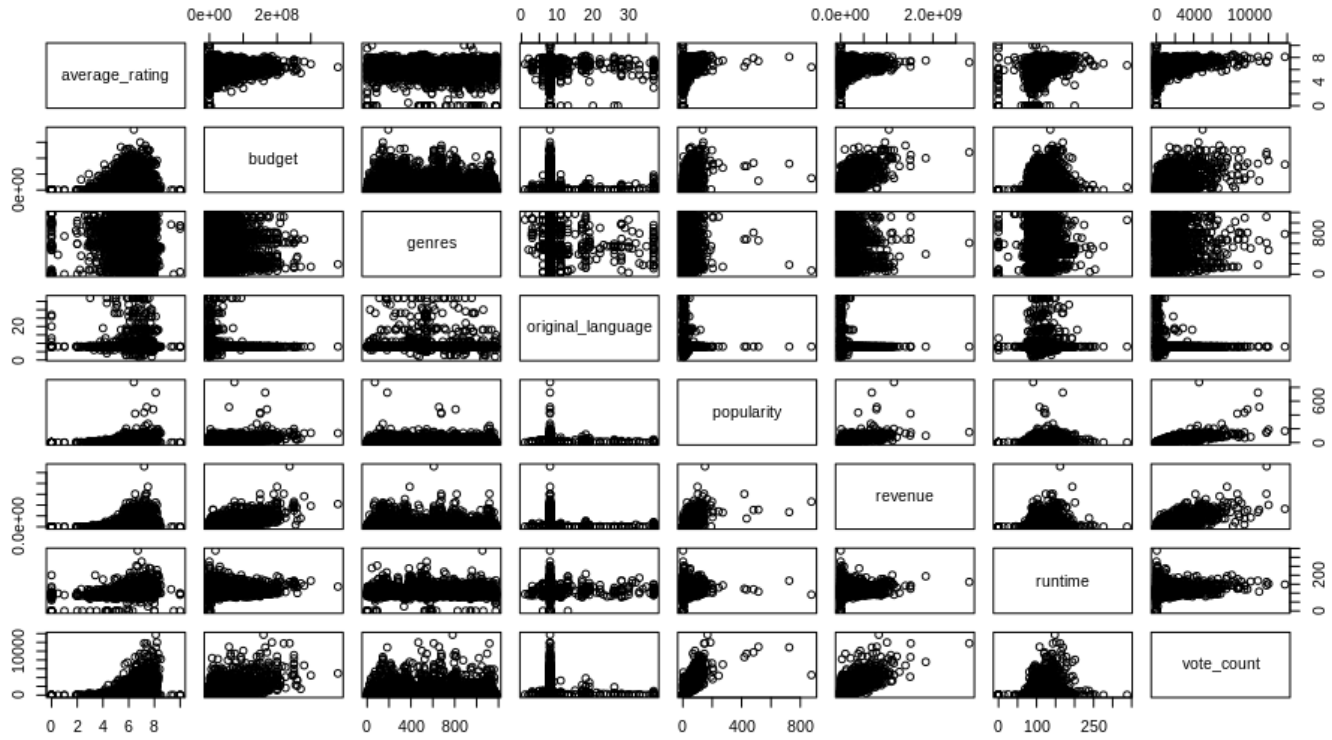

Figure 1: Pairs of comparison

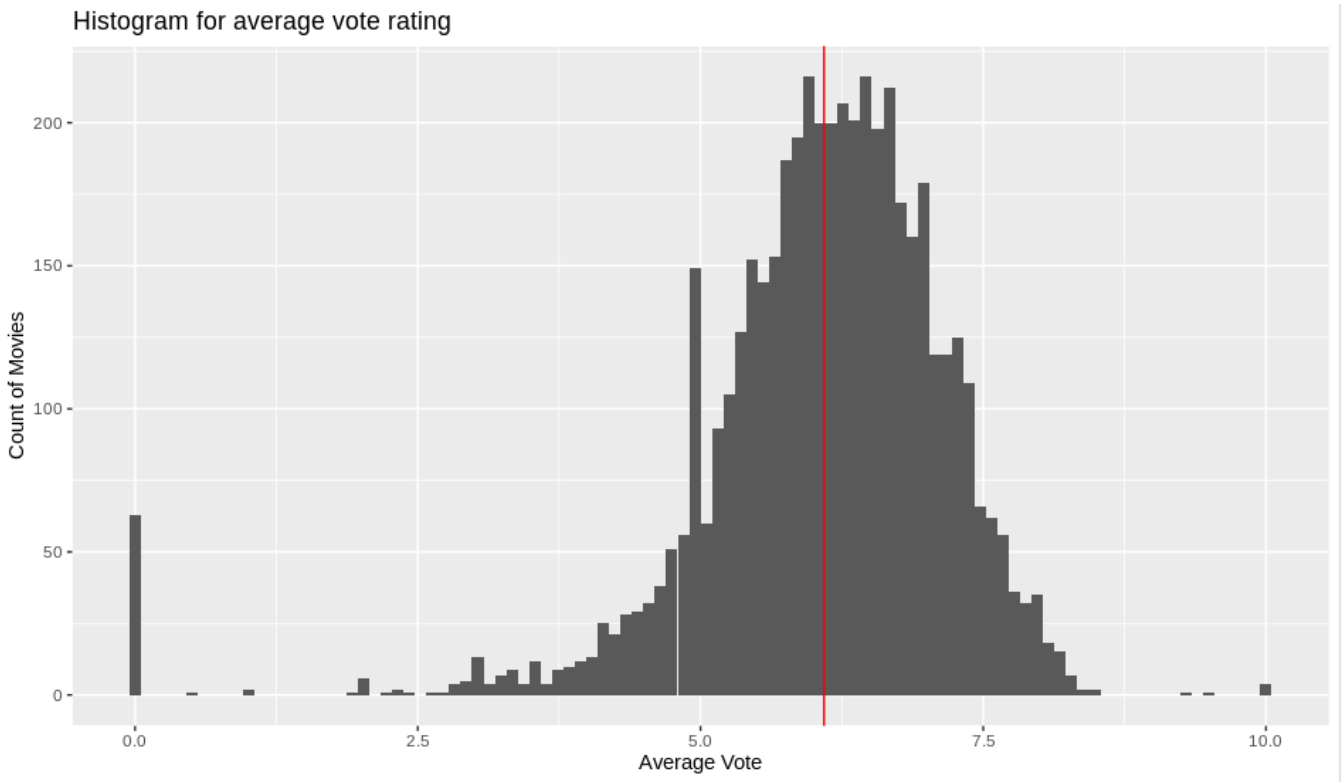

Figure 2: Histogram for average vote rating

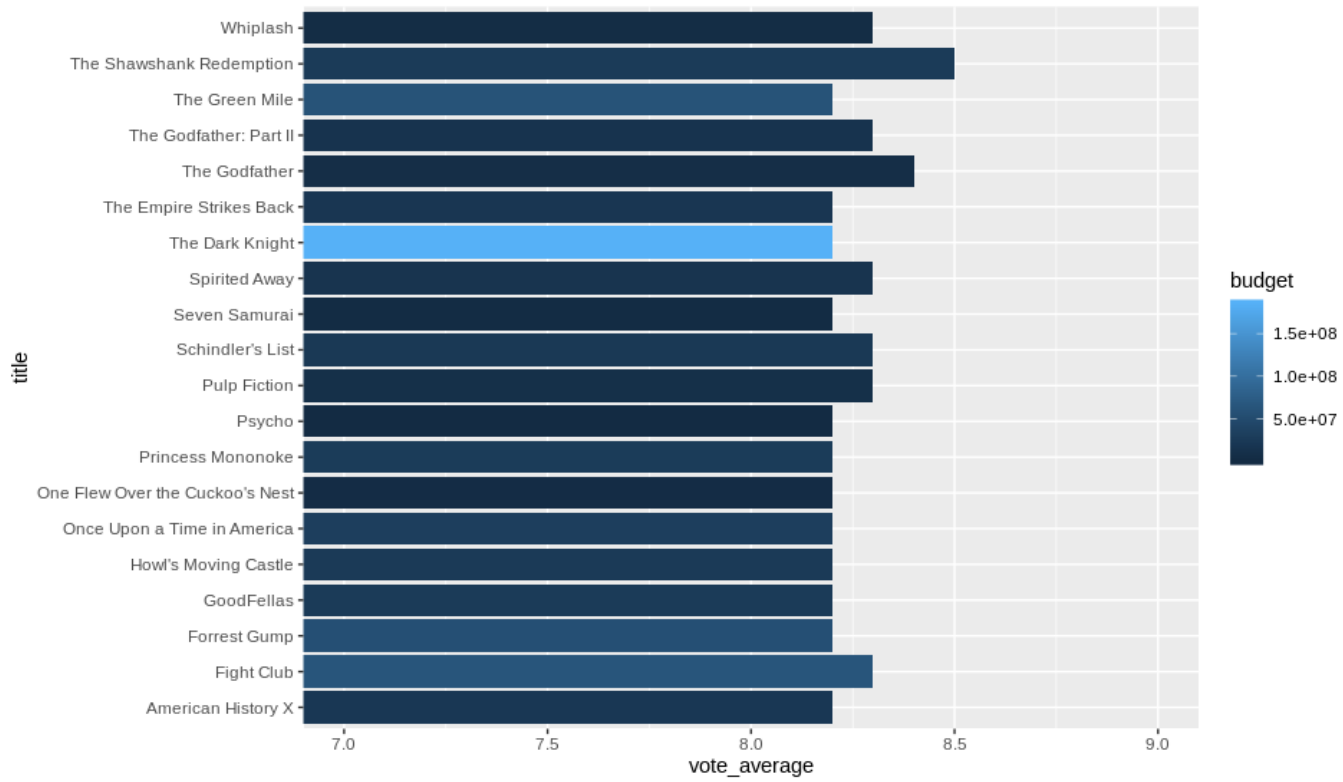

Figure 3: Plot for average vote rating and budget for titles

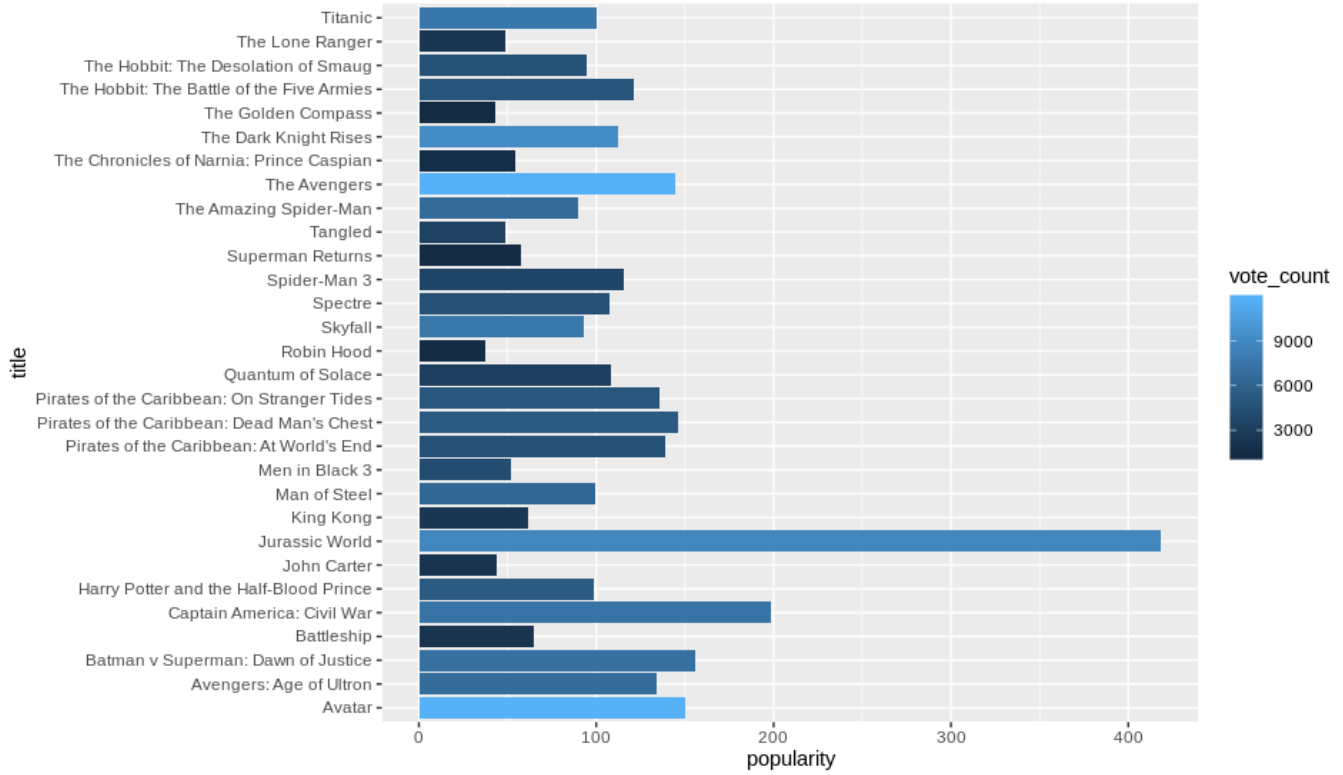

Figure 4: Plot for vote count and popularity for titles

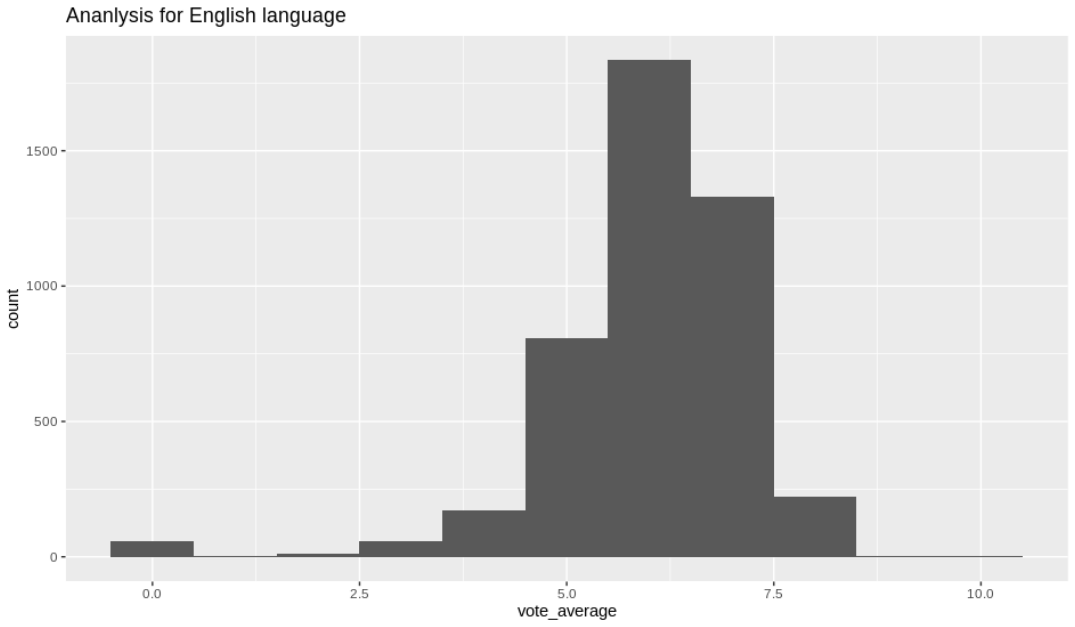

Figure 5: Plot for average vote and for titles in English language

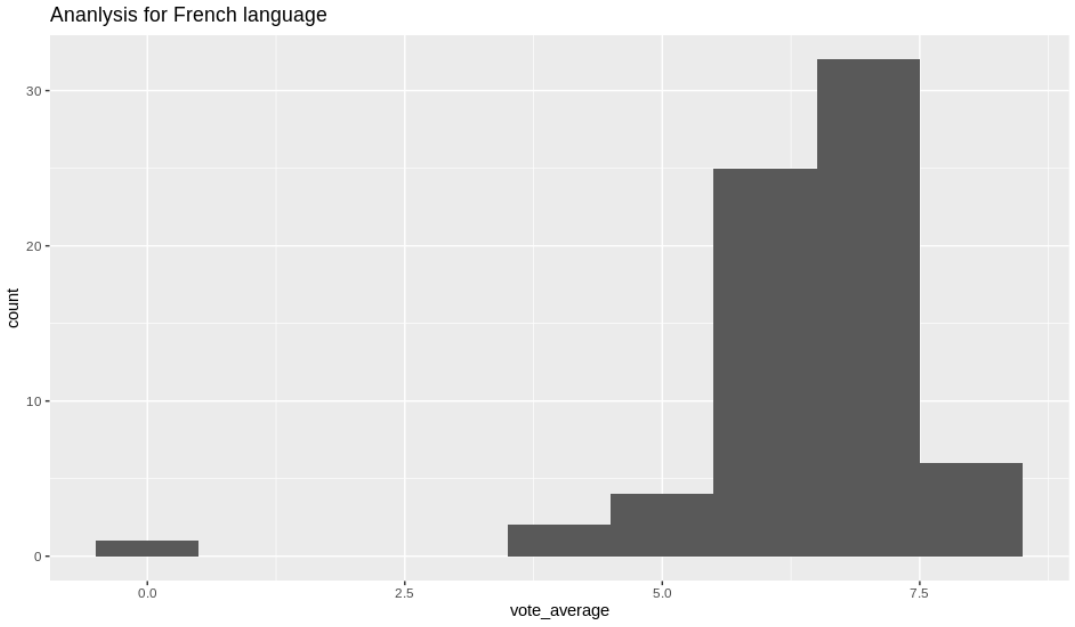

Figure 6: Plot for average vote and for titles in French language

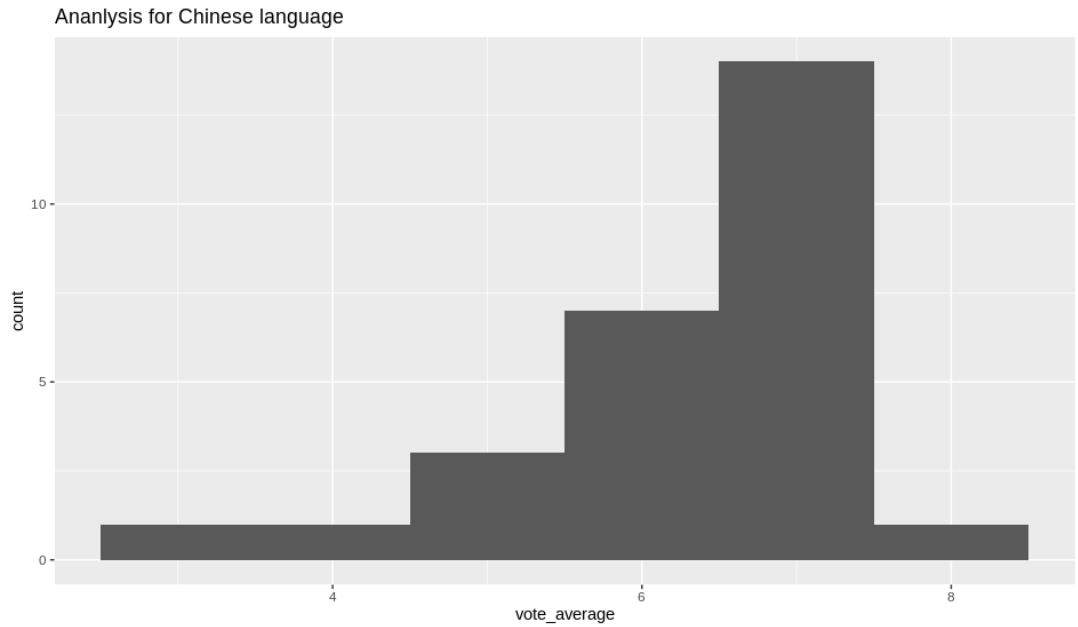

Figure 7: Plot for average vote and for titles in Chinese language

```
    budget            homepage              id         original_language  original_title       overview
Min.   :        0   Length:4800        Min.   :     5   Length:4800        Length:4800        Length:4800
1st Qu.:   795000   Class :character   1st Qu.:  9020   Class :character   Class :character   Class :character
Median : 15000000   Mode  :character   Median : 14633   Mode  :character   Mode  :character   Mode  :character
Mean   : 29060614                      Mean   : 57198
3rd Qu.: 40000000                      3rd Qu.: 58640
Max.   :380000000                      Max.   :459488

  popularity       release_date          revenue             runtime         status            tagline
Min.   :  0.000   Min.   :1916-09-04   Min.   :0.000e+00   Min.   :  0.0   Length:4800        Length:4800
1st Qu.:  4.668   1st Qu.:1999-07-14   1st Qu.:0.000e+00   1st Qu.: 94.0   Class :character   Class :character
Median : 12.925   Median :2005-10-01   Median :1.917e+07   Median :103.0   Mode  :character   Mode  :character
Mean   : 21.498   Mean   :2002-12-30   Mean   :8.229e+07   Mean   :106.9
3rd Qu.: 28.351   3rd Qu.:2011-02-17   3rd Qu.:9.294e+07   3rd Qu.:117.8
Max.   :875.581   Max.   :2017-02-03   Max.   :2.788e+09   Max.   :338.0
                  NA's   :1                                NA's   :2
   title            vote_average      vote_count        keywords           genres          production_companies
Length:4800        Min.   : 0.000   Min.   :    0.0   Length:4800        Length:4800        Length:4800
Class :character   1st Qu.: 5.600   1st Qu.:   54.0   Class :character   Class :character   Class :character
Mode  :character   Median : 6.200   Median :  235.5   Mode  :character   Mode  :character   Mode  :character
                   Mean   : 6.092   Mean   :  690.5
                   3rd Qu.: 6.800   3rd Qu.:  737.2
                   Max.   :10.000   Max.   :13752.0

production_countries spoken_languages
Length:4800          Length:4800
Class :character     Class :character
Mode  :character     Mode  :character
```

Figure 8: Data Summary

Table 1: Timeline

Week 11	Phase 3	Submitted as a team
Week 12	Quantize Data	Yifei - Avg rating vs Budget, Zhou - Avg rating vs Popularity,Richard - Avg rating vs cast and crew members, Ajeeta - Avg rating vs different Languages
Week 12	Final Feature Selection	Full team as this will require discussion
Week 13	Train classifier	Rich and Yefei
Week 13	Test and Validate classifier	Zhou and Ajeeta
Week 13	Write Report	Full Team
Week 13	Final Submission of Data Mining Component	Full Team