Project Midterm Report

Authors: Jessie Wang (yw696), Anil Vadali(arv42)

Abstract

With over 600 movies being made in the US each year and increasing competition from new companies, the keys to film success has become more important than ever. As a result, in this project, we hope to explore and predict what features of movies make them more successful than others. Specifically, we are developing a model that will analyze and predict the IMDB score for a certain movie, which is the baseline for how successful a film is in the movie world.

Dataset Description

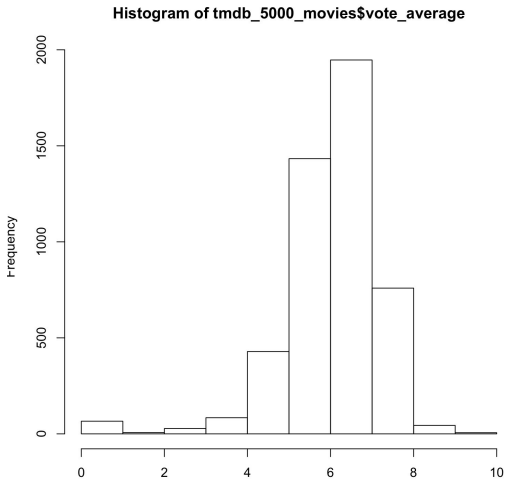
The original dataset has 4803 data points and 22 features in tmdb_5000_movies and 3 features in tmdb_5000_credits.

After data preparation, we include features as follows: budget, popularity, revenue, runtime, status, vote_count, release_date_year, release_date_month, the first three genre id, the first three keywords, the first three production_companies.

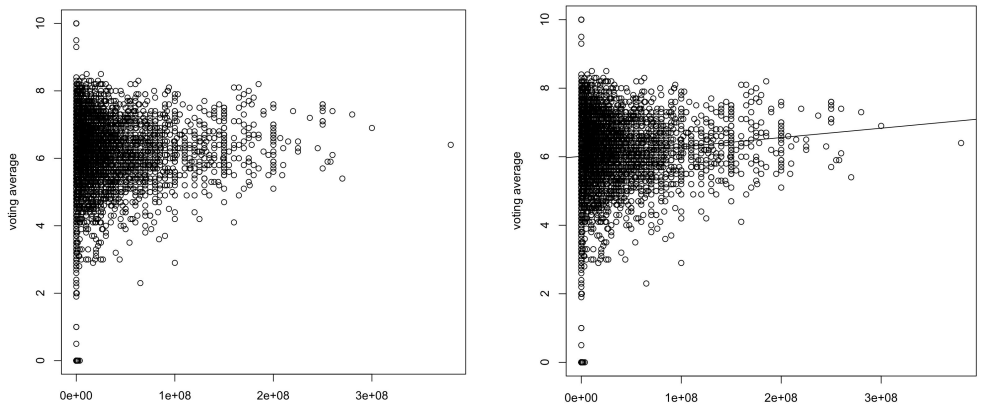
The feature we discarded for midterm report included homepage, id, original language, title, overview, tagline. Some of the features we are discarding because we think they are irrelevant to the prediction. These features include homepage url, id. We are discarding original language because the majority of them are in english and the sample for non-english movies are not big enough to give reasonable insights into the data. We are discarding title, overview, tagline, etc for now because in order to make use of these features, we need more advanced techniques such as NLP.

Visualization of potential important data features

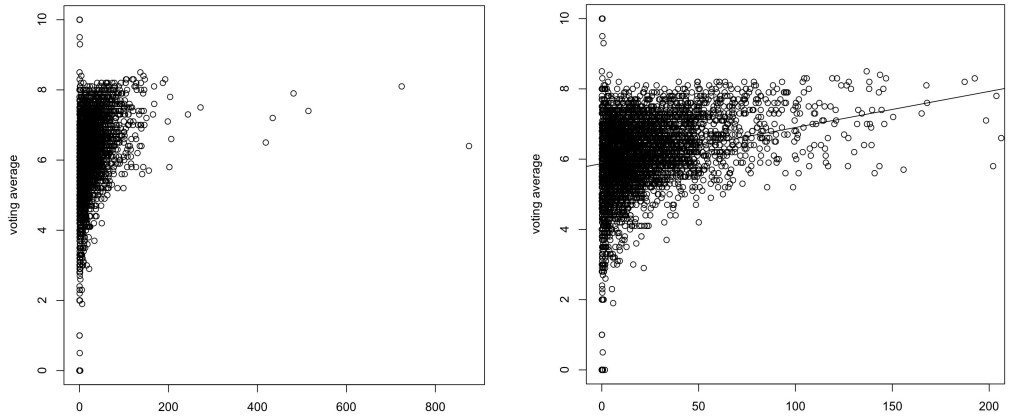
Histogram of average rating: The histogram gives us a visualization representation of the voting average range. As we can see, the histogram gives a bell shaped data, which means the vote_average of all movies are pretty close to normal distribution.



Line fit between rating and budget: The left graph is the scatter plot of vote_average vs. budget (vote_average as the y axis and budget as the x_axis), which give us a general idea of the data distribution. The right graph is an attempt to fit a linear model, which shows slight correlation between vote_average and budget.



Line fit between popularity and rating: The graph on the left is the scatter plot of the popularity ratings for each movie in the dataset we analyzed and it is compared against what rating the movie ended up getting (popularity score on the x-axis vs. voting average on the y-axis). We originally believed that the popularity feature would play a large role in determining the overall movie rating score since a movie that is often times more popular, is generally, admired more. As a result, we wanted to observe any trends between these two fields. Though there appears to be a positive linear trend between the two variable, we later realized that the popularity field is not very significant in the prediction of the overall score based on the analysis described below.



Missing/Corrupted Data

After processing the data, we check for missing data and they mainly exist in the genre_id, keyword_id and production_company_ids because different movies have different number of associated genre, keyword and production_company. The way we are handling missing data is just to discard any row with missing data, but this might be problematic since we only have 3600 data points for training and many of them have at least one missing data. Another method we can use is to replace the missing genre_id, keyword_id and production_company_id with the mode. We will experiment on different methods and choose the one with lower validation error.

CV

For cross-validation, we decide to separate the entire dataset into 75% training and 25% validation data. We will also try LOOCV and k-fold for future analysis.

Techniques to Avoid Over/Underfitting

1) Linear model with regularizer (L1 vs. L2)
2) Kernelized Linear model

Since linear model is a very inflexible model that can easily underfit important patterns and cause high bias, we want to use kernelized linear model to relax the strict linear assumption and try to decrease the bias error to a reasonable value.

3) Decision Tree with bagging

Since decision tree is a highly flexible model with a hard-to-find “sweet spot” that balances reasonably low bias and reasonably low variance, we want to avoid overfitting to the training data by applying bagged decision tree (random forest) and decision tree pruning.

Preliminary Analysis

1) Linear model: For preliminary analysis, we want to have a basic idea of how well a linear model would fit to predict voting average. For simplicity reasons, we are only using numeric values for this analysis and will incorporate categorical variables later. From the summary we get from fitting this linear model, we can see that variables such as budget, runtime, vote_count have significant contribution to prediction of voting average, X21 (release date year) and X22(release date month) also has somewhat contribution. We also find the RSE are relatively low and Adj R^2 are relatively too high, which means this linear model is not enough to predict voting averages. We will use other techniques and add more data features to make the model more accurate.

```
Call:
lm(formula = voting_avg_train ~ ., data = train_prelim)

Residuals:
    Min      1Q  Median      3Q     Max
-3.1874 -0.3735  0.0467  0.4380  2.8442

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.326e+01  5.714e+00   4.071 5.13e-05 ***
budget      -5.653e-09  6.537e-10  -8.648  < 2e-16 ***
popularity  -3.775e-04  7.484e-04  -0.504  0.61407
revenue     -2.943e-10  1.890e-10  -1.557  0.11983
runtime      1.071e-02  1.171e-03   9.140  < 2e-16 ***
vote_count   2.902e-04  2.754e-05  10.535  < 2e-16 ***
X21         -9.111e-03  2.838e-03  -3.211  0.00137 **
X22          1.429e-02  7.137e-03   2.002  0.04560 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6697 on 840 degrees of freedom
Multiple R-squared:  0.342,     Adjusted R-squared:  0.3365
F-statistic: 62.38 on 7 and 840 DF,  p-value: < 2.2e-16
```

Future Work

1) Incorporate categorical variables

We hope to approach utilizing categorical variables in many different ways. One specific approach we could try is to use one-hot encoding to represent each categorical variable feature. However, a main concern is that it may lead to a sparsity within vectors/matrices. This may lead to very small weights overall for those features at the end. Additionally, another method we hope to explore is to utilize some variant of NLP to analyze the text features. This would be another challenge that we may have to encounter in the future.

2) Add regularizer to the linear model
3) Implement a RBF kernel linear model
4) Implement a Bagged Decision Tree
5) Incorporate Facebook Likes/Twitter Favorites/Youtube Views

The latest dataset update removed the facebook likes feature. We will try to retrieve these information if we can. Another option one of our proposal peer reviewers mentioned that may useful would be to see how many views a certain movie trailer got on Youtube and utilize that information as a feature as well.