IMDB Movie Rating Predictions

Laura Contreras

4/27/2023

COMP 4449

I. Introduction:

Many movies are coming out now that the world is back open from COVID-19. With competition in the movie industry, a model to predict how well a movie will be rated may become necessary in Hollywood. The goal of this project is to determine out of the 3 models, Lasso, Ridge, and RandomForestRegressor, which one is the best model to predict movie ratings, and can a highly be predictive model from the best selection?

II. Dataset:

The dataset was obtained from Kaggle and contained roughly 5k observations with 28 attributes. Some of the attributes were director names, IMDB score, movie title, and genre. In the dataset, it was decided to drop the following columns as in the dataset they were noise: title year, director and actor names, plot keywords, IMDB link, aspect ratio, and face number in the movie poster. This dataset contained a lot of null values with the gross column containing over 880 missing data values, so this column was ultimately dropped from the dataset. In this project, the target variable used was the continuous variable imdb_score.

III. Data Preparation and Exploration:

In preparing the data, the null values were first observed. The gross column was dropped due to containing almost 18% missing values, while the rows with null values in other columns were dropped. This resulted in a dataset of 3828 observations.

The continuous variables were observed first. The distribution of the IMDB score showed that the majority of the data lay in the score range from 5.0-8.0, making this a left-skewed distribution. When comparing the IMBD score to the duration of the movie, we see the left skewed data and notice that the majority of the movies tend to range from 50-150 minutes.

A 3-D plot was then created to compare the number of user votes to the duration of the movie to the IMDB score. It was shown that the number of while the duration tends to range from 50-150, as the number of users for reviews increases, so does the score. After this plot, the data were scaled using a StandardScaler which removes the mean and scales each feature/variable to unit variance.

A pair plot was then performed, and showed in many continuous variables such as the budget, number of user ratings, and number of critics reviews that the distribution when compared with the IMDB score shows that the data is left-skewed. The comparison of cast total Facebook likes to all other variables shows so outliers that appear very large. The same occurs for the budget variable. In the pair plot, we see a large data point at 50 on the y-axis, however, the other points tend to lie below 10 for this variable.

After seeing this, a correlation matrix was created to determine whether the continuous variables had a positive or negative correlation to the IMDB score. It was found that the 4 variables with the strongest positive correlation were the num_voted_users with 0.48, then duration with 0.37, then num_critic_for_reviews with 0.35, and last num_user_for_reviews with 0.32. The budget variable had almost no correlation with the IMDB score. It obtained a score of 0.03.

From here, the outliers were removed from the dataset for these variables and boxplots were created to show their distribution per score rating.The majority of the distributions appeared normal, however, for some, there was a positive and negative skew the more the rating increased.

The categorical variables were then observed. Using a histogram, the distribution of the IMDB score against the color of the movie showed that the majority of the movies in color tend to range in the 6-7.5 rating, while black and white movies have a more widespread range of movie ratings. We do see that there are more color movies than black-and-white movies in the data set. However, when looking at the average rating black and white has a higher movie rating than a color movie. Color averaged a movie rating of 6.35, while black and white averaged a movie rating of 7.04.

In the column content rating, the ratings were combined to create only the classic G, PG, PG-13, and R+. The column originally had 15 different rating categories, so they were combined based on the age a viewer could watch the movie. Histograms were created to see how the distribution of movie ratings was across movie content ratings. They all appeared to follow the distribution of the IMDB score with the histograms showing a left skew.

For the final data preprocessing, dummy variables were created for the content ratings, language, and country columns. The color column was transformed to 0 and 1, with 0 being for black and white, and 1 representing color. For the movie title column, a CountVectorizer was implemented to grab the count of the non-stopwords. This however did increase the dimensionality of the data frame significantly. After this transformation, the columns increased by 3791. The genre was gathered by splitting the multiple genres in the column value and creating dummy variables for them. There were a total of 23 genres represented in the data set. These were then all combined into a final movie data frame.

VI. Model Selection, Tuning, and Results

Lasso, Ridge, and RandomForestRegressor were the three models selected to be tested. The dataset was split on an 80/20 train, test split, with the IMDB score being the target variable. During the first fit of the models, the mean squared error was utilized to determine which model is best. From the scores, Lasso did the worst with an MSE of 1.039, then Ridge with 0.563, and lastly RandomForestRegressor with 0.44. However, it was noticed that the train data set was overfitting the data as it had significantly better scores than the test sets for Ridge and RandomForestRegressor.

The models were then hyper-tuned with various combinations. Lasso and Ridge were hyper-tuned to find the best alpha, while RandomForestRegressor was tuned to find the best n_estimators, max_features, max_depth, and bootstrap. From hyper tuning with a GridSearchCV, it was found that Lasso's best parameter had an alpha of 0.1, Ridge’s default parameters were best, and RandomForestRegressor was best with n_estimators set to 200 and the rest as default. During the second fit of the models, the mean squared error and R2 score were utilized to determine which model is best. From the scores, Lasso still did the worst with an R2 of 0.27 and MSE of 0.75, then

Ridge with an R2 of 0.0.45 and MSE of 0.56, and lastly RandomForestRegressor with 0.577 and MSE of 0.436. RandomForestRegressor was determined to be the best model.

The best model was then further hyper-tuned to find better parameters, however, it was found that the n_estimators set 200 and the rest as default were the best. The R2 score calculated was 0.577, and the MSE was 0.436. In the Predicted vs Actual plot for the test predictions, we saw that the higher the rating the better the model performed, however when it came to a low rating it was not good at predicting movie scores. In the Residuals plot, the test set residuals did have some in the -1 to 1 range, however, there were points that were far away from the 0 line. The train set residuals did far better with a majority of the distribution between the -1 to 1 range. This is an indication that there is still overfitting in the model.

V. Discussion

By comparing the three models, Lasso, Ridge, and RandomForestRegressor, it was found that the RandomForestRegressor was the best model for predicting the IMDB score, however, it was a weak model. While a low MSE score was obtained (0.436), the R2 was very not very large (0.577), indicating the model is weak.

A lesson learned during the hyper-tuning is that expanding on the parameters for a GridSerachCV can make the code run for a long period of time. It was attempted to expand further on the hyperparameters during the second tuning, however, the code ran for over a day before selecting the best estimator. My time spent on the project ended up being 50% data cleaning and 50% model building/tuning. If one has a timeline due date, it is important to realize how many different combinations the GridSearchCV will look through.

To further improve the R2 and MSE, more outliers could be looked at and removed in the continuous variables. For the movie title variable, instead of using the CountVectorizer, sentiment analysis could be used to transform the movie into a 0 or 1 for indicating if the title is positive or negative. This would help reduce the dimensionality of the dataset.