Daniel Timmermann

Final Report:
IMDb Movie Review Classification

Objective

Build a model that could best predict a movie review on IMDb's website as positive or negative.

Problem

IMDb currently does not have a metric that quickly identifies the ratio of bad to good movie reviews and would like to add this to their website, so visitors to their site can quickly identify a movie that has a bad or good reputation with movie goers.

Solution

My goal for this project was to build a model that could detect if a movie review is positive or negative. Once the reviews are correctly identified then the company can take the ratio and add this to the movie statistics. Visitors to the site are looking for a quick metric to determine how a film was received by the audience so they know if they should add it to their watchlist or avoid wasting their time. This can be a difficult task though because some reviews can be labeled as positive but contain words that might be perceived as negative (or vice versa).

Data Wrangling

The raw dataset from "Learning Word Vectors for Sentiment Analysis", by Maas, Andrew L. and Daly, Raymond E. and Pham, Peter T. and Huang, Dan and Ng, Andrew Y. and Potts, Christopher, contains 50,000 reviews with sentiment ratings of positive or negative. I started by looking at a few of the reviews to see some examples of the text that needed to be cleaned. I then built a function to clean all the reviews. I removed accents, special characters, digits, new lines, line breaks, extra whitespace, and quotations. I also expanded all contractions and lowercased all text. Then I performed preprocessing steps, such as lemmatizing the reviews and removing stop words, so I could perform exploratory data analysis.

Exploratory Data Analysis

I first created word clouds for both positive and negative reviews. The word clouds show the 200 most frequent words and can be seen in Figure 1 and Figure 2. I then created a plot of the frequency of the 25 most common words throughout all reviews, which can be seen in Figure 3. After looking at the 25 most common words, it is obvious that certain words made the list, and this verifies my intuition. In Figure 4 and Figure 5, I visualized the word and character counts in reviews.

Word Cloud for Reviews with Positive Sentiment

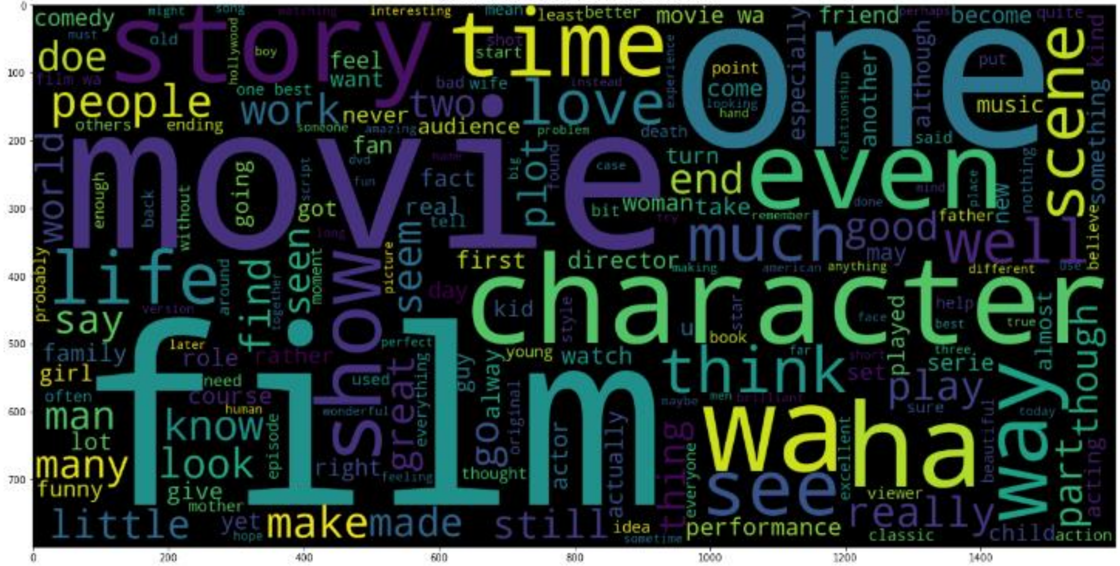


Figure 1

Word Cloud for Reviews with Negative Sentiment

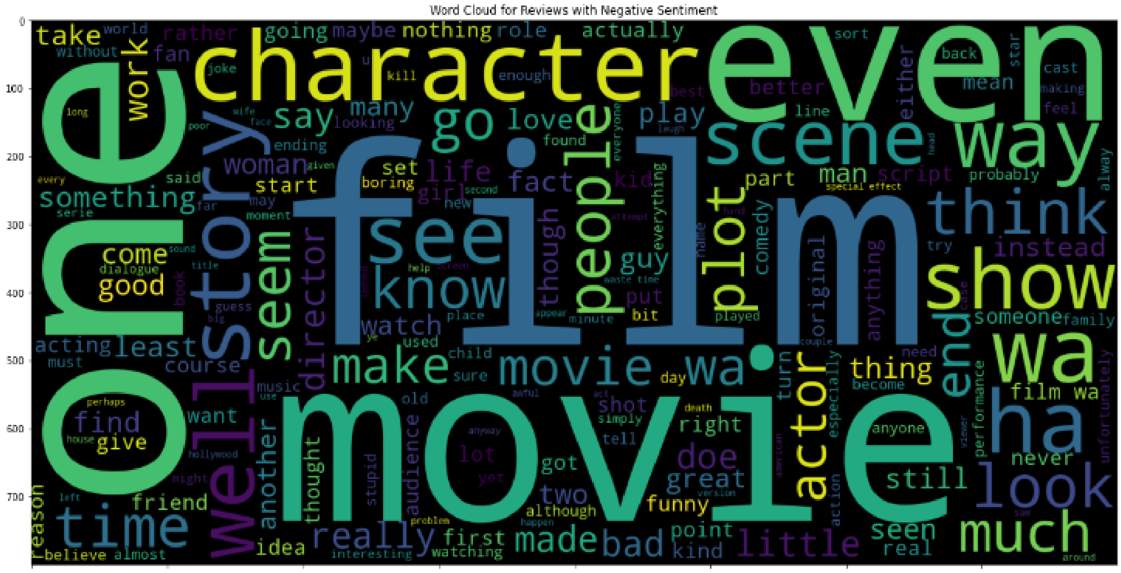
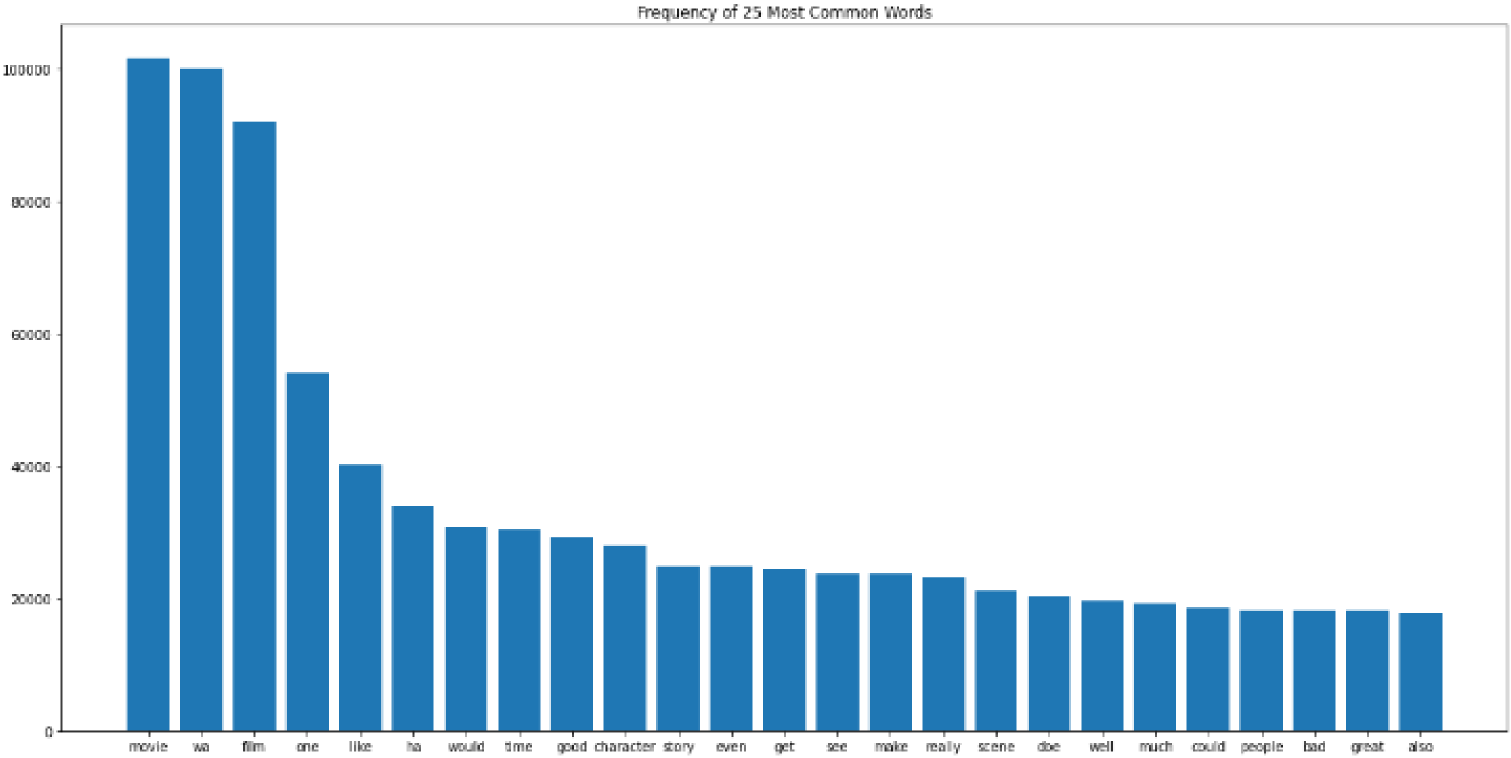


Figure 2

Frequency of 25 Most Common Words



Data Preprocessing

Now with data exploration complete I moved on to preprocessing the data to prepare it for use in machine learning models. With the Time and Amount features having such a wide range of values compared to V1-V28, I first scaled all of the features. I standardized the features, meaning I subtracted the average of the values for each feature and then divided the values by the standard deviation for each feature.

Next I decided how I wanted to handle the highly imbalanced target class. I have 283,253 non-fraudulent and 473 fraudulent transactions. If I were to feed all of the data to the model, then the model will classify almost every if not all transactions as not fraudulent. I had to choose between the following three options: Random Undersampling, Random Oversampling, and Synthetic Minority Oversampling Technique (SMOTE). I decided to implement random undersampling which would collect a sample of 473 non-fraudulent transactions, so there would be an even amount of transactions identified as non-fraudulent to pass into the model. Now I am ready to split my 946 rows of data into training and testing sets.

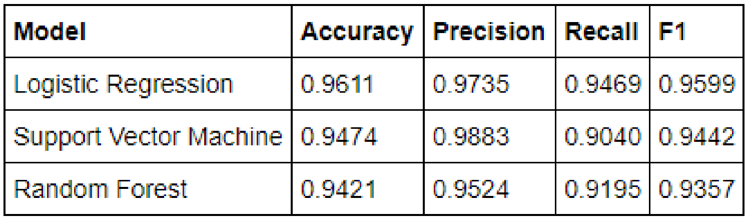
Modeling

For the modeling step I choose to compare the performance of 3 models.

1. Logistic Regression
2. Random Forest
3. Support Vector Machine

I collected 5 random samples of data to feed into the logistic regression and support vector machine models and then I averaged the scores for each sample. The scores I compared were accuracy, precision, recall, and F1. Both logistic regression and support vector machine were implemented as is and I did not specify and parameters. For the random forest model, I used a grid search method with cross validation to tune the estimators, criterion, and max_depth parameters. Since I used cross validation with 5 folds, this meant I did not need to feed 5 random samples to the model. Every time the model was trained with a different 80% of the 1 random sample I fed to the model and a different 20% was used to test.

After running my models a few times, logistic regression was the clear winner. Figure 3 shows the models scores the first time I tested the models. Logistic regression typically performed better in every scoring category, but the score I wanted to focus on the most was recall. Recall is popular to score to focus on when tackling fraud. One thing you want to minimize with fraud is false negatives. Recall shows how well the model can identify fraudulent transactions without mislabeling transactions as not fraudulent. The impact from credit card fraud will be minimized, but if the model mislabels transactions as not fraudulent then credit card fraud will still be present.



Model	Accuracy	Precision	Recall	F1
Logistic Regression	0.9611	0.9735	0.9469	0.9599
Support Vector Machine	0.9474	0.9883	0.9040	0.9442
Random Forest	0.9421	0.9524	0.9195	0.9357

Final Model Review

Below is what I have concluded from the logistic regression model.
The confusion matrix, seen in Figure 4, can be used to calculate the precision, recall, accuracy, and f1-scores I mentioned before. The model misclassified 10 out of 102 fraudulent transactions. This is something that I would want to continue to improve given more data.

The receiver operating characteristic (ROC) curve, seen in Figure 5, is a visualization used to evaluate the "predictive power" of the model. A curve that is closer to the top left corner means the model has good predictive power. The red dashed line represents a model that has no predictive power. The AUC score in the bottom right corner is the area under the curve represented as a percentage. The higher the score the better the model. Based on the ROC curve and an AUC score of 97%, I can conclude the model has good predictive power.

Now that the overall performance of the model has been assessed I want to view the effect each predictor variable had on the target variable. I created a bar plot, seen in Figure 6, that would visualize the model coefficient for each feature. Later I will mention what can be done with this information to continue to improve the model.

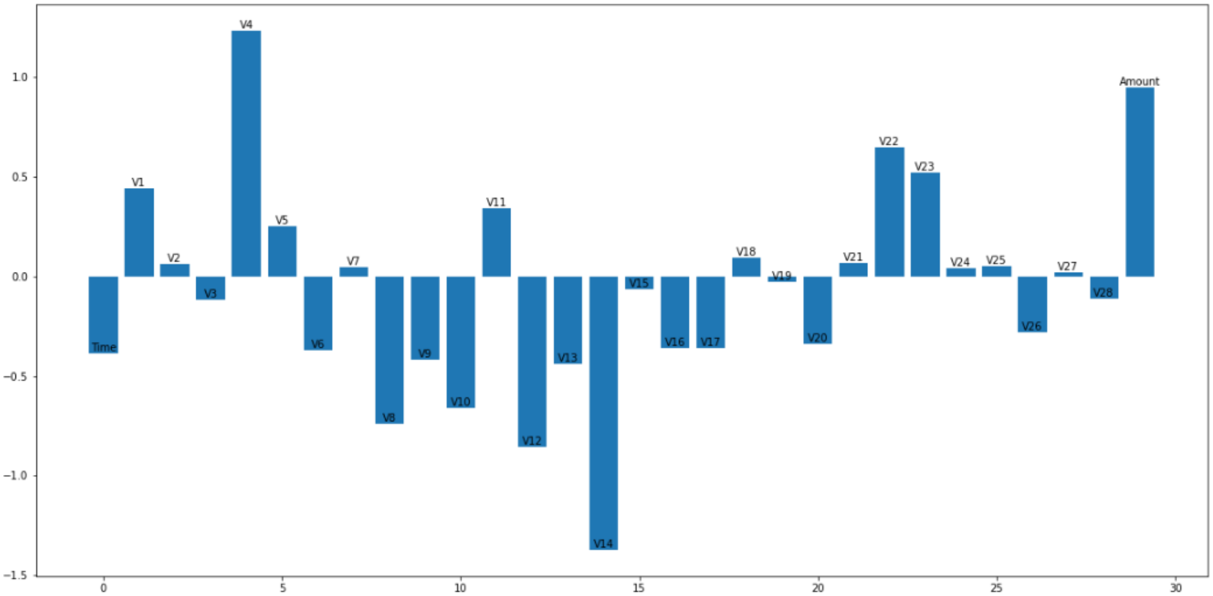


Figure 6

Next I wanted to visualize the decision boundary for this binary classification problem. It is very difficult to do with many features, so I implemented principal component analysis to reduce the features down to 2. Now my visualization for the decision boundary would be 2-dimensional with 473 rows of data and can be seen in Figure 7. Sadly, I do not know which features were kept, so it is difficult to have certain conclusions based on the visualization. I also visualized the decision boundary using the entire dataset to show that, without using an sampling technique to combat the high class imbalance for the target variable, the model would almost always predict transactions as not fraudulent. This decision boundary can be seen in Figure 8.

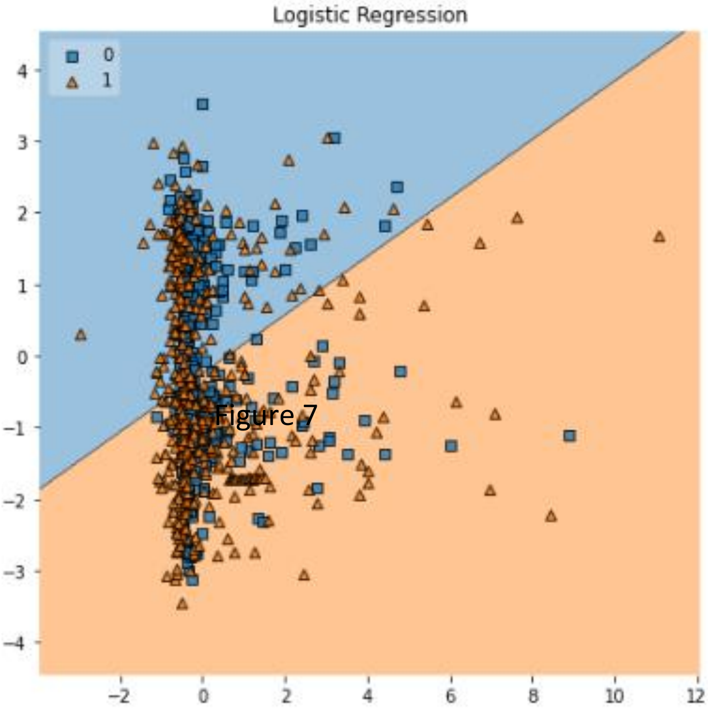


Figure 7

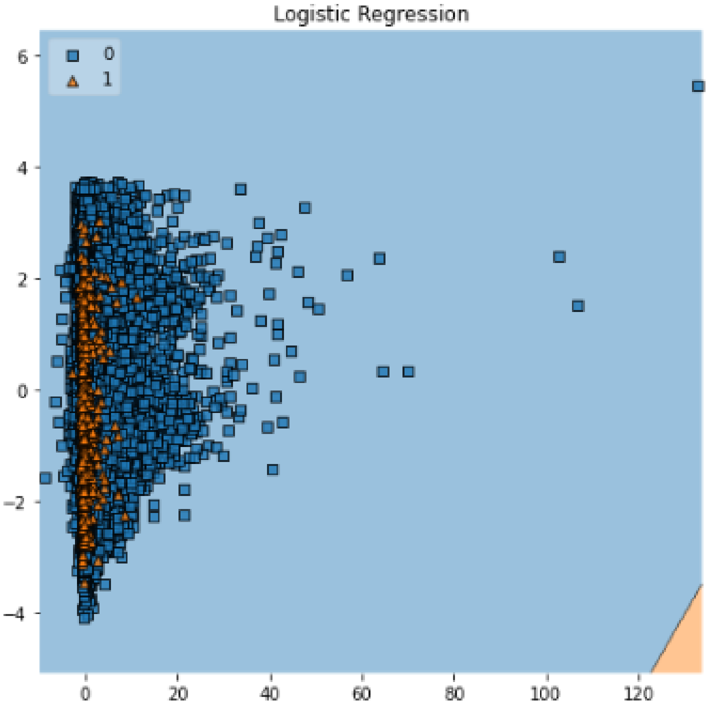


Figure 8

Takeaways and Future Research

As mentioned before, the logistic regression and support vector machine models were used with default parameters. It would be interesting to test and see how all three models compared if I were to do more tuning of the parameters and to include more values in the grid to search from for the random forest model.

Another process that can be attempted to improve model results would be to try out other sampling methods such as random oversampling and synthetic minority oversampling technique.

Lastly, I would like to add more reason to my choice for logistic regression in this project. Logistic regression is the least complex computationally, it would be the easiest to explain to interested stakeholders, and the simplest model to deploy in a business environment.