Daniel Timmermann

Final Report:
IMDb Movie Review Classification

Objective
Build a model that can 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 they align with.

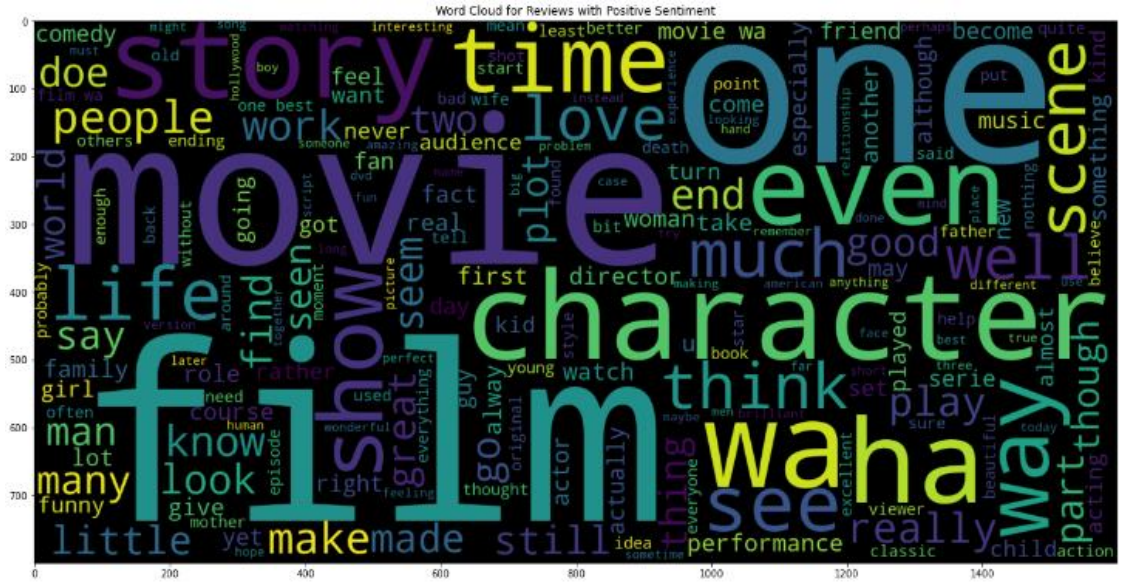

Figure 1

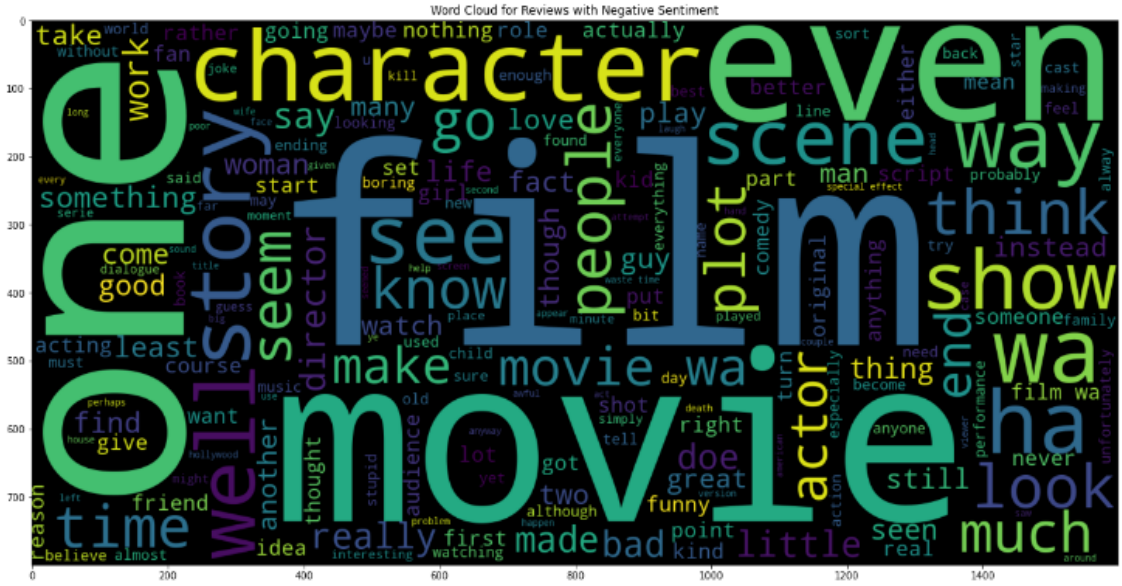

Figure 2

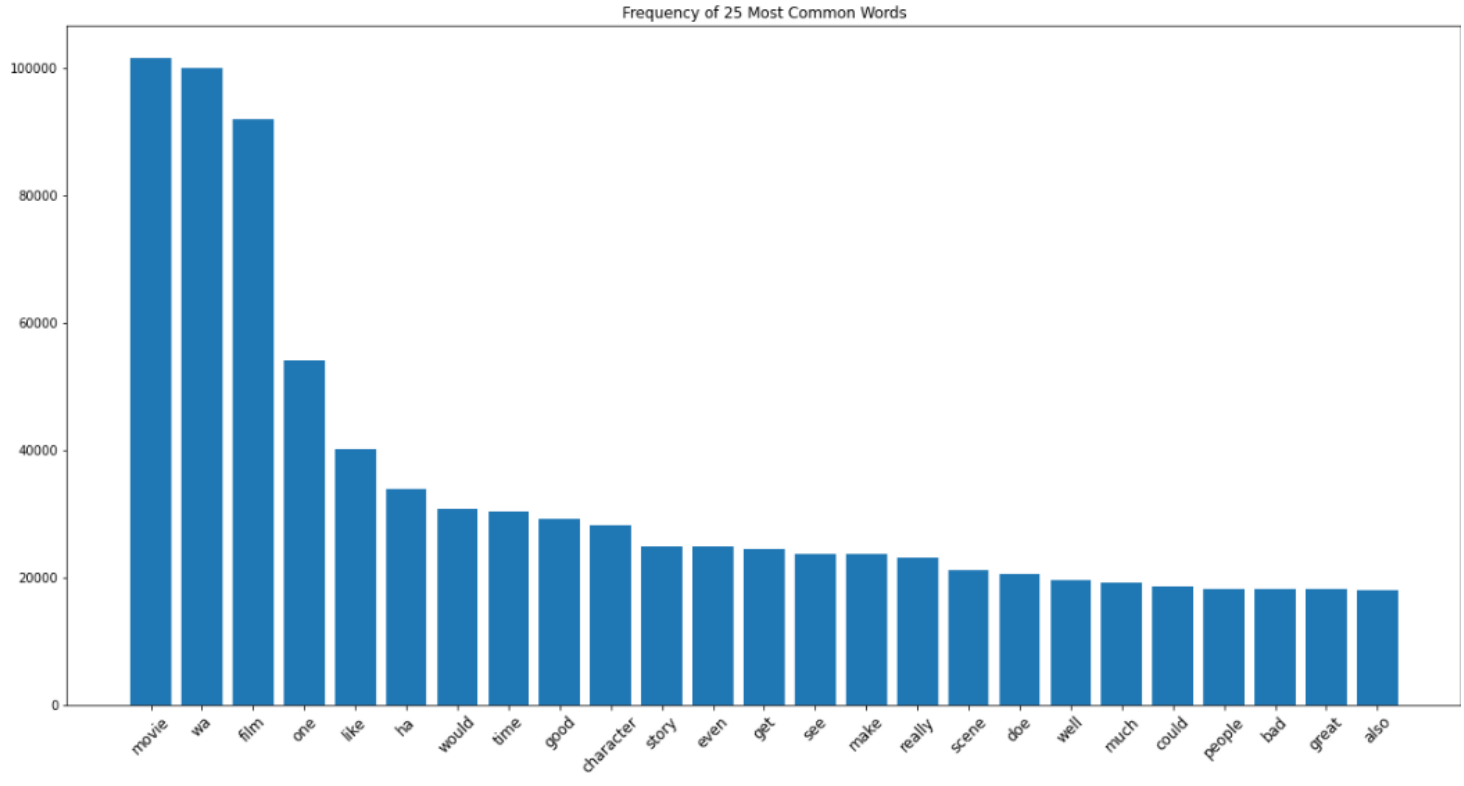

Figure 3

In Figure 4 and Figure 5, I visualized the word and character counts in reviews. From the plots, I can conclude that really short reviews tend to be labeled more as positive than negative.

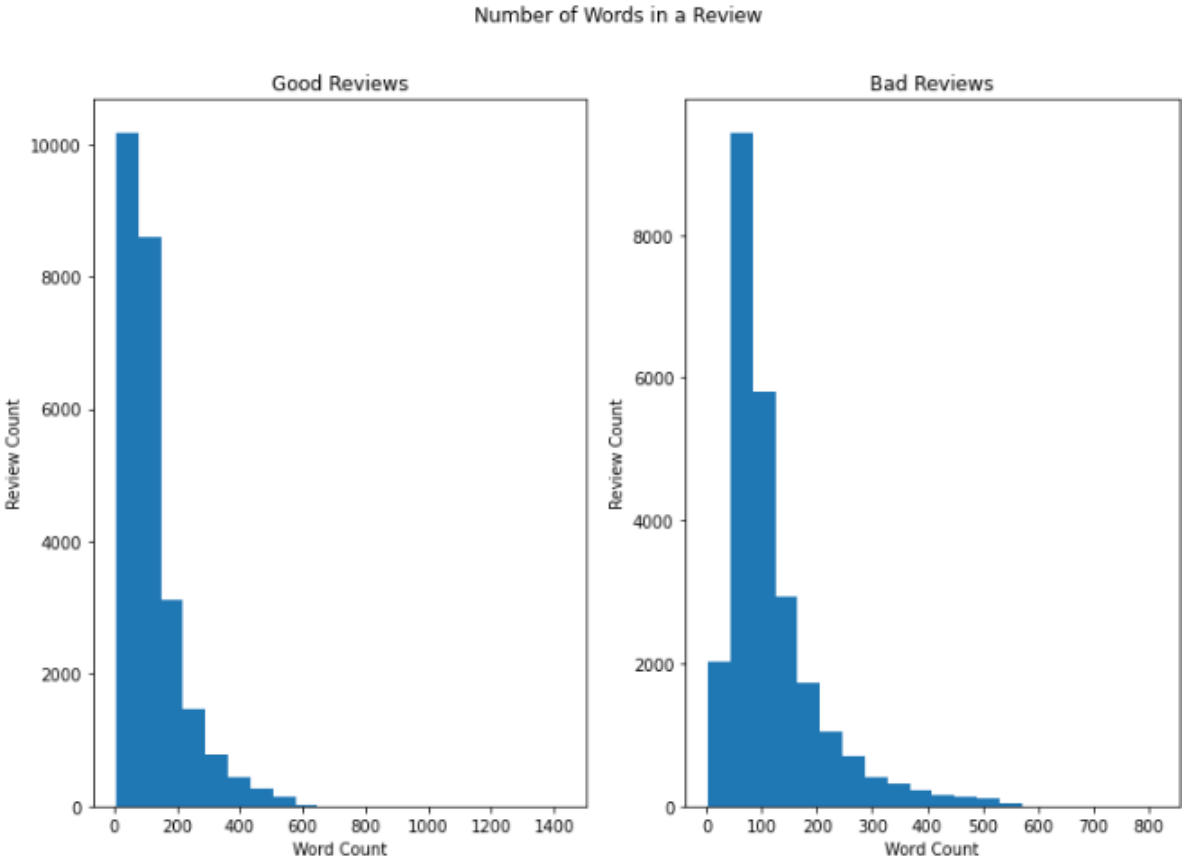

Figure 4

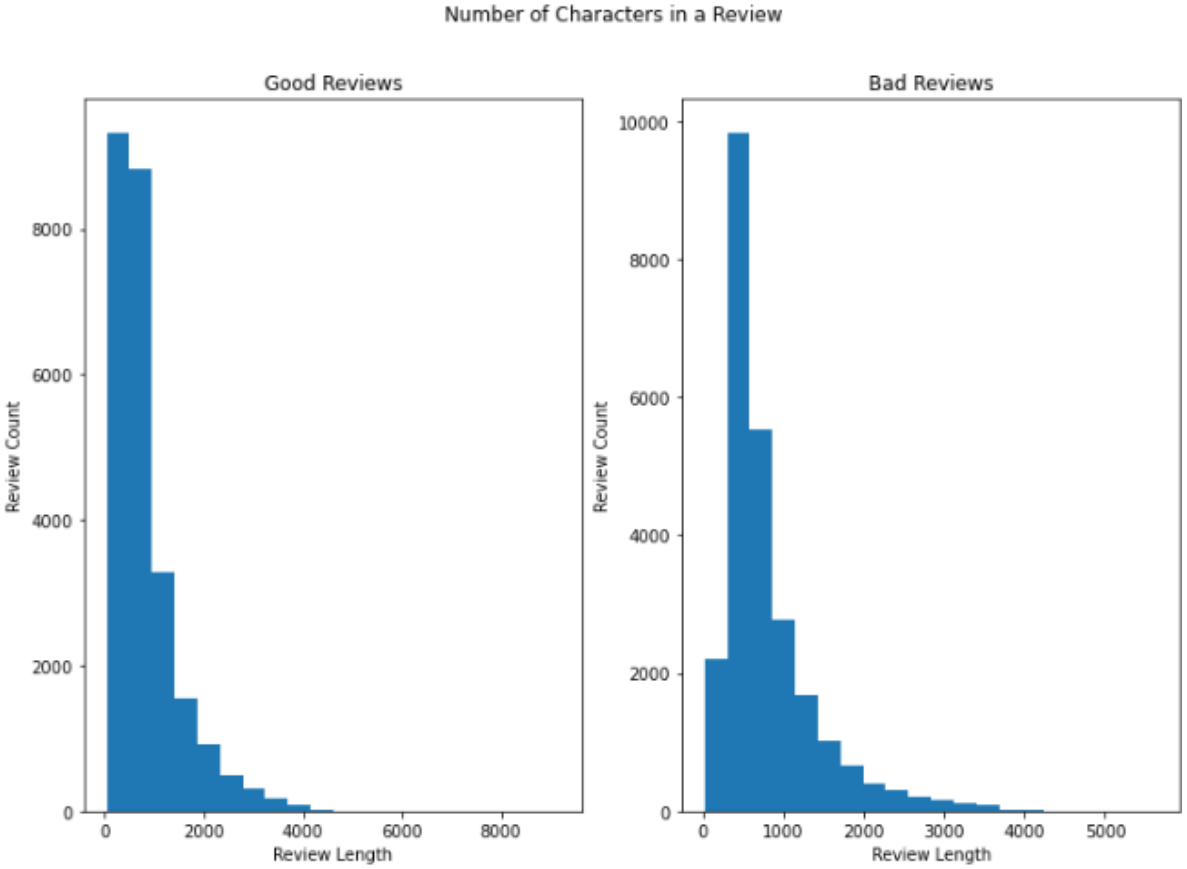

Figure 5

Data Preprocessing

Now with data exploration complete I moved on to preprocessing the data to prepare it for use in machine learning models. Usually when preprocessing text data you first need to remove stop words, lemmatize, and/or perform stemming, but for this project I removed stop words and lemmatized the text in the data cleaning step so I could explore the data.

Next, I decided how I wanted to create a matrix from the text to use as input for my models. The two most popular options I had to choose between were Bag of Words and Term Frequency-Inverse Document Frequency (i.e. TFIDF). Bag of Words is a representation of text that describes the occurrence of words within a document and TFIDF evaluates how relevant a word is to a document in a collection of documents. I decided to implement bag of words because of its simplicity and I chose to look at unigrams and bigrams. A unigram each word is considered as a separate item and a bigram is two words next to each other are grouped together and become a single item. Bag of Words can be implemented with n-grams, but when I included trigrams the execution time became too long.

Modeling

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

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

Due to execution time issues, all the models I compared were trained and tested with default parameters. The models were trained with 40,000 reviews and 10,000 reviews were left out for testing. The scores I compared were accuracy, precision, recall, and F1.

After training and testing my models a few times, logistic regression was the clear winner. Figure 6 shows the models scores to easily compare. Logistic regression performed better in every scoring category, except for precision. Even though the precision was higher for support vector machine and random forest, it was obvious logistic regression was more consistent.

Model	Accuracy	Precision	Recall	F1
Logistic Regression	0.6795	0.7296	0.5561	0.6311
Support Vector Machine	0.5129	0.8409	0.0150	0.0295
Random Forest	0.5151	0.9457	0.0176	0.0346

Figure 6

Final Model Review

Below is what I have concluded from the logistic regression model.

The confusion matrix, seen in Figure 7, can be used to calculate the precision, recall, accuracy, and f1-scores I mentioned before. The confusion matrix is from a different run of the model though, so the numbers do not perfectly match Figure 6. The model misclassified 2,314 out of 5,029 positive reviews as negative and 939 out of 4,971 negative reviews as positive. This confirms the scores received in Figure 6 and tells me the model does a better job at distinguishing negative reviews. I'll comment further in the last section of the report.

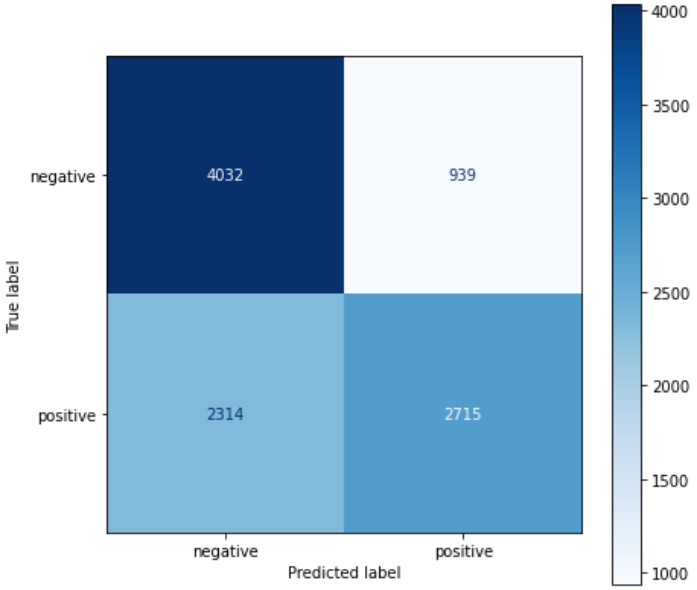

Figure 7

The receiver operating characteristic (ROC) curve, seen in Figure 8, 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 76%, I can conclude the model has some predictive power, but there is still a lot of room for improvement.

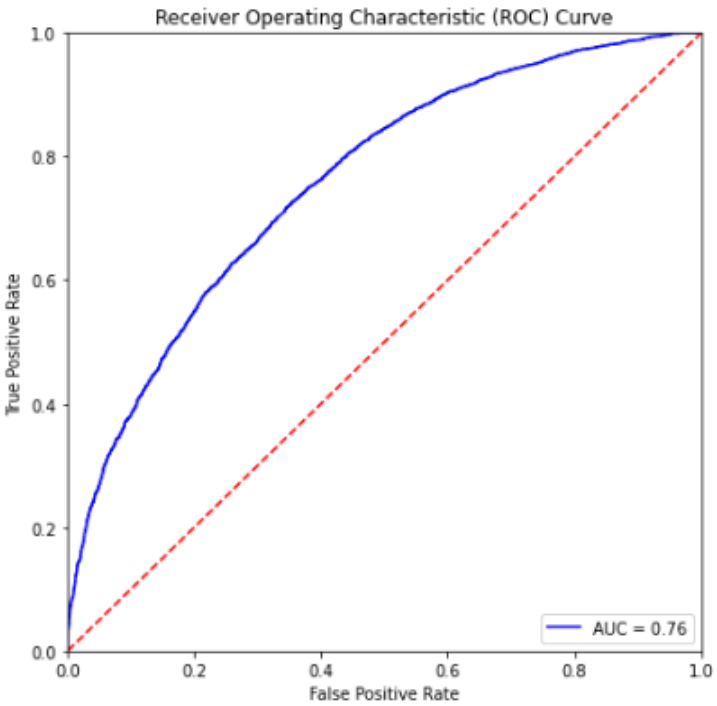

Figure 8

Since recall is the score the model is struggling with, I next wanted to look at the precision-recall curve. The precision-recall curve, seen in Figure 9, shows there is an inverse relationship between the precision and recall scores and relationship looks very linear. Almost everywhere along the plot precision decreases as recall increases. There are rarely any instances where precision stays the same as recall increases.

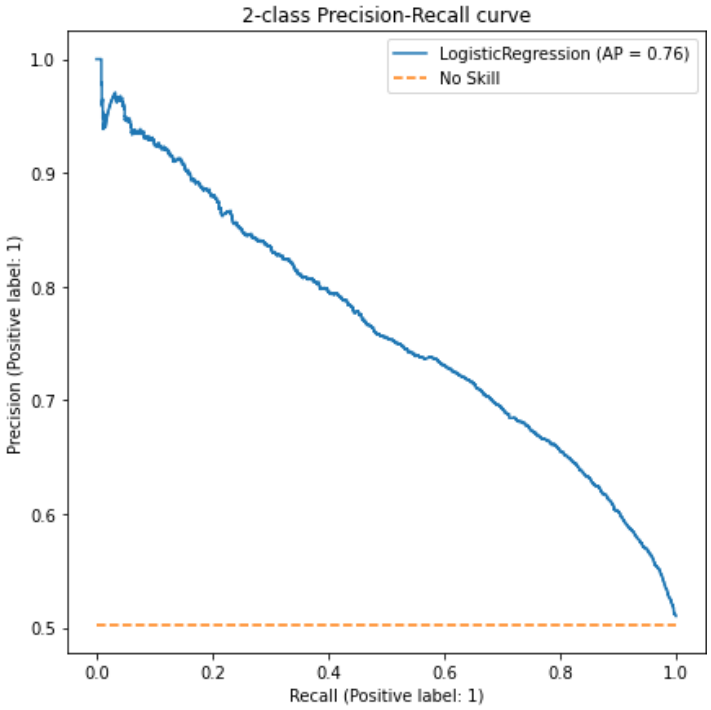

Figure 9

For the last metric, I have plotted the change in the mean absolute error as the dataset size increases for the test and training sets. From the plot, seen in Figure 10, it can be seen that as the dataset size increases the mean absolute error decreases. This metric was important because I wanted to know if I was feeding too much data into the model. Also, this is intriguing and makes me wonder how large the dataset would have to be for there to be an increase in error for the test set. Last but not least it is important to check to see if the plot was made correctly and this can be done with the last subset. The training set size is 39,999 with 0.325 test error and this aligns with the confusion matrix (FP+FN)/10000 = (939+2314)/10000 = 0.325.

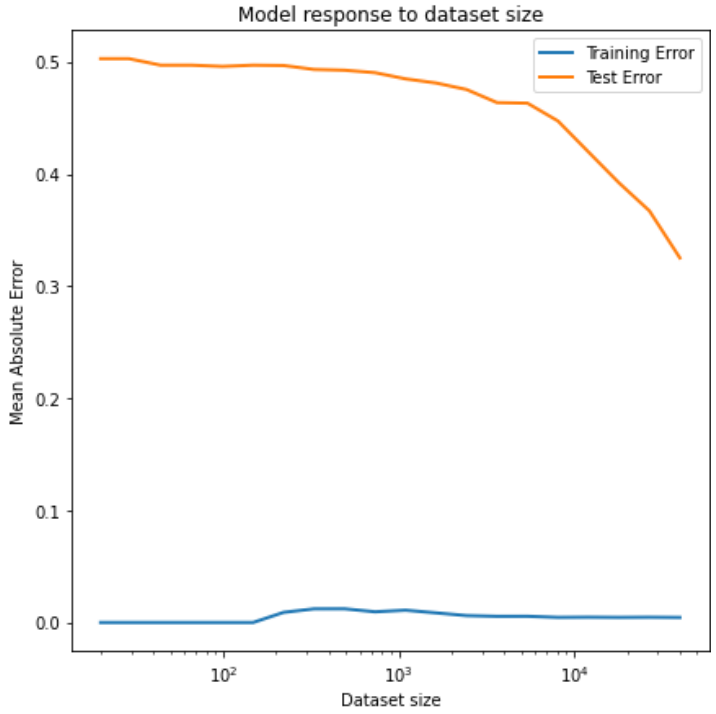

Figure 10

Takeaways and Future Research

As mentioned before, all models were used with default parameters. It would be interesting to test and see how all three models compared if I were to tune the parameters and had more processing power to perform a grid search for the random forest model. I now want to make comments on the misclassifications made by the logistic regression model. My hypothesis for why the model does a poor job classifying positive reviews is because a lot of positive reviews have a mixture of words with bad and good connotation. Negative reviews might be easier to identify because there aren't many really short reviews and reviewers use more words with negative connotation.

When creating features, one thing I want to attempt is TFIDF instead of Bag of Words and see how model performance is affected. I'd also like to see how model performance changes if I were to add word count, character length, and most common word in every review as features. I could also further clean the reviews by removing the most frequently used words in both negative and positive reviews. The last step for further research would be to collect a larger dataset.

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.