Synopsis

Problem Statement

What was once considered one of Amazon's biggest selling points has become almost the opposite of what it claims to be. Jeff Bezos was one of the first e-commerce sellers to allow public reviews on his products. While many others would be scared to allow customers to be blunt about their experience with a product or service, Jeff embraced it. Amazon reviews allowed top products to rise to the top and lesser quality products to fall off. Although harsh on Amazon sellers, this was one of Amazon's key points on their rise to the top of the e-commerce industry. The need for other real reviews and honesty is so important in an industry where the actual product cannot be seen until one has it in hand. Unfortunately, today, Amazon reviews are losing their value. Sellers are creating hundreds, or even thousands of fake reviews to boost their rating to 5 stars, basically diminishing the worth of the reviews in the first place.

Using machine learning we are looking to fix this problem and bring Amazon reviews back to the trustworthy feature they once were.

Data Wrangling

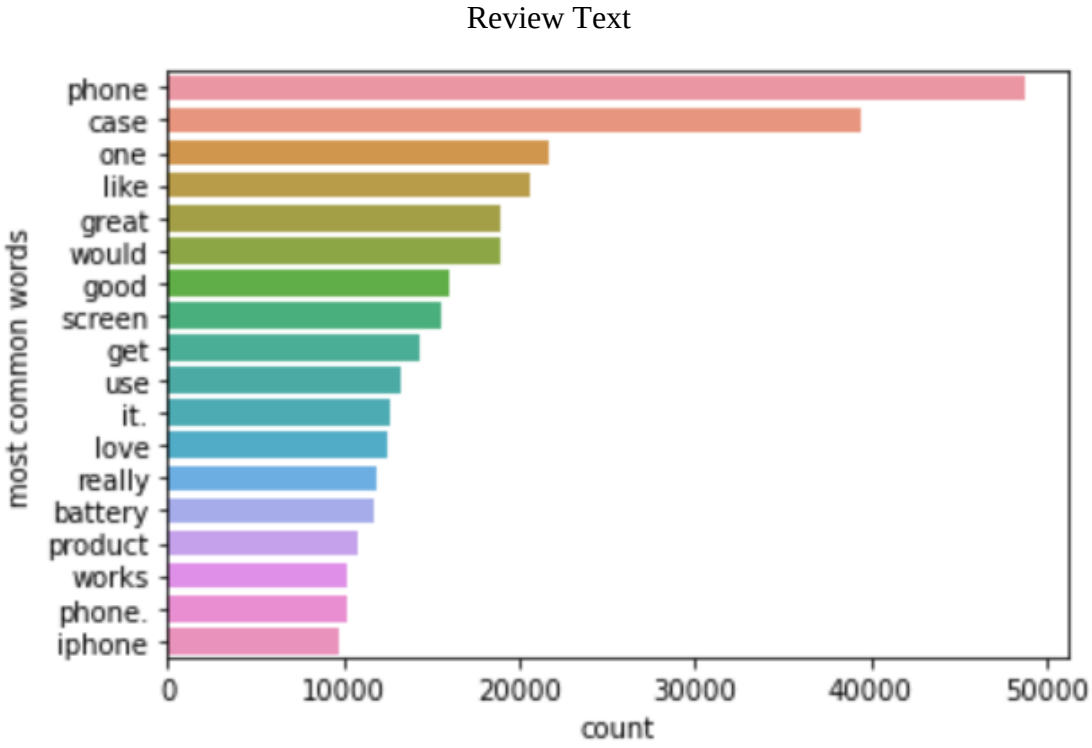
The Spam data that I used was from Kaggle and contained 3 million rows and 13 columns. In this section I wanted to get a good understanding of my data and see any patterns or concerns I should have before any modeling or feature engineering is done. The data included the reviews, summary title, time, id, and some other ID based data as well. This data was already prelabeled with a simple binary 0 or 1 for non-spam or spam, respectively. I first checked the data for any missing values or abnormalities. I found just 1 row with an integer label in a review box. This was likely someone leaving a number review in the description box. This row was removed and then we were able to focus on more high-level construction. I then changed column

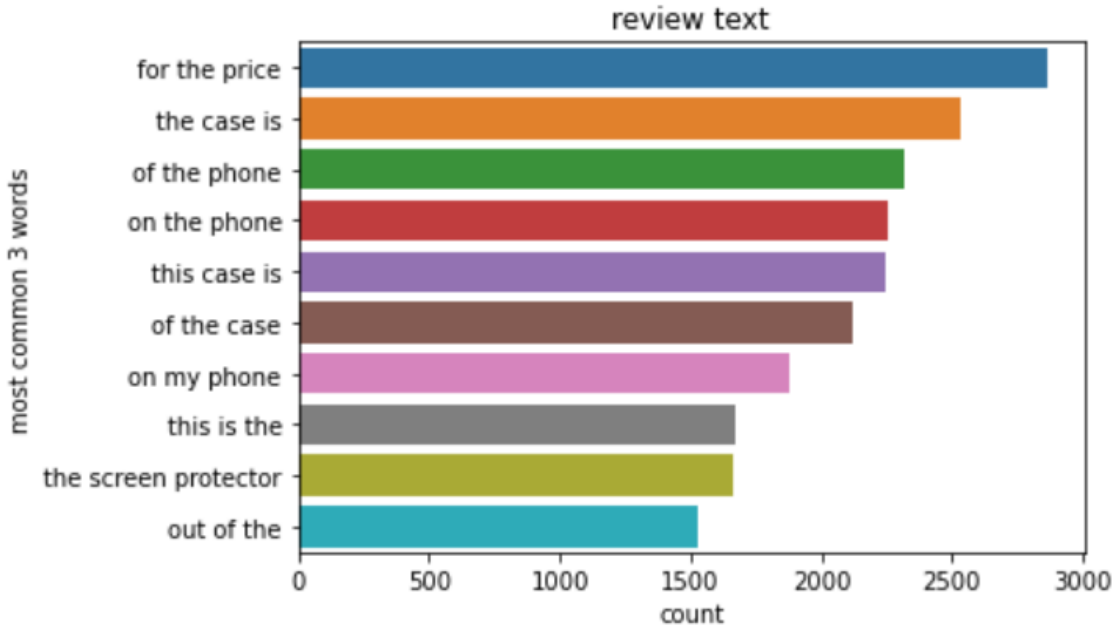
data types to the appropriate labels that way we can start manipulating the data the way we want to. There was no need to fill null values as there was only 1 row, but I could have used the median rating if that were missing.

Exploratory Data Analysis

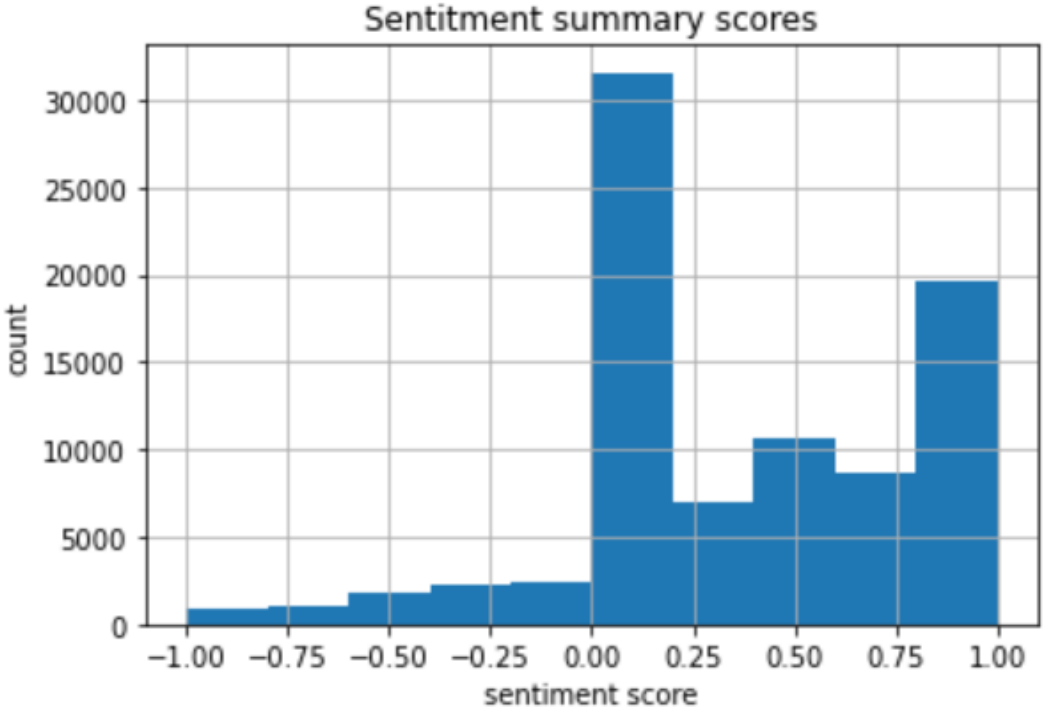
In this section we focus on plotting our data to allow us to see high level patterns or concerns with the data.

From the graphs below we can see the most common words used in a review and the common 3 consecutive words from a review. While this data alone will not be great in predicting spam vs. non-spam data this allows us to understand the data better. People who make fake reviews may use data like this to make their reviews fit more in line with real reviews. Knowing what the opposing side will do is just as important as our defense on trying to stop them

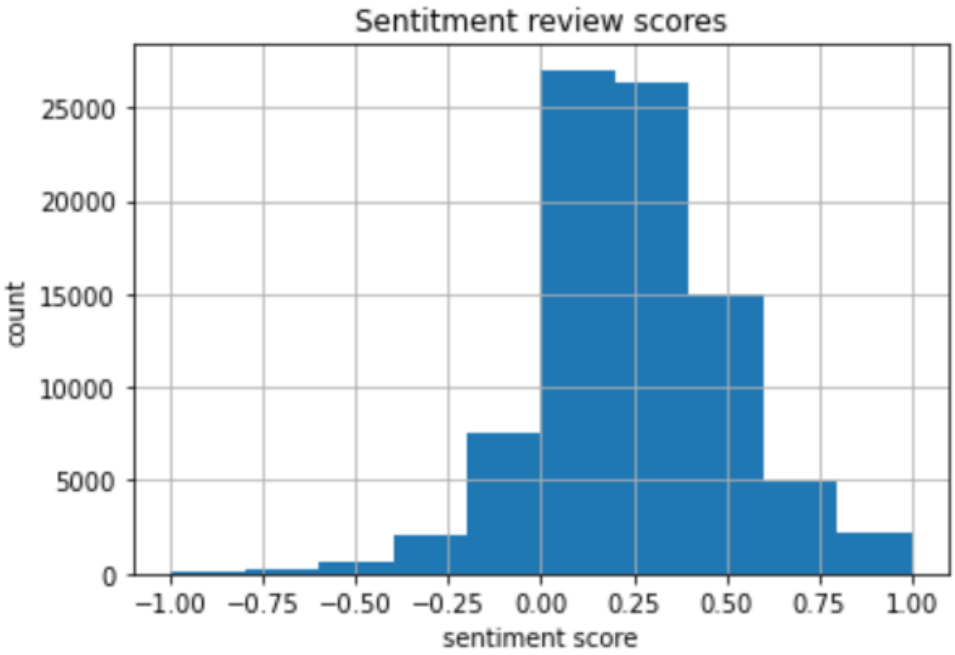




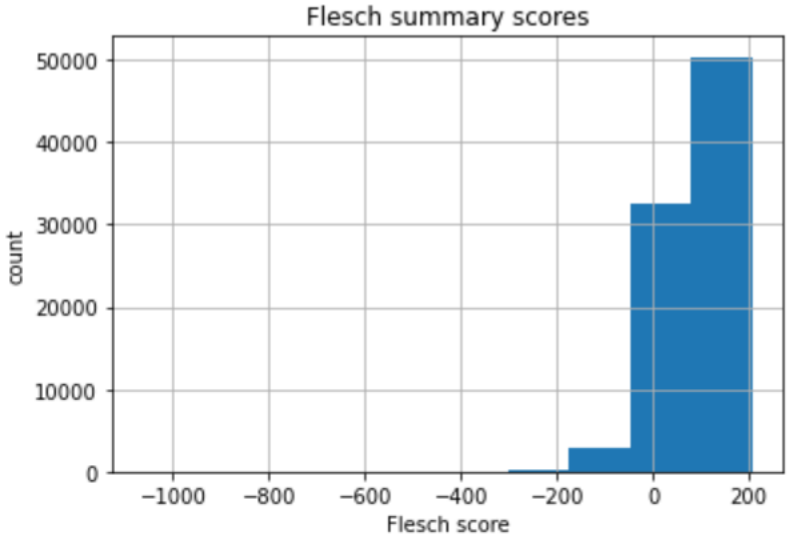
Another aspect we looked at was sentiment scores, we used TextBlob's sentiment analysis library to help score our reviews and summaries on sentiment. In this case a score of 1 would be the best review you could give and a score of -1 would be the worst score you could receive. Looking at the charts below we can see that in the summary most of the scores' center around 0 and 1. Now the summary is only allowed to be one sentence which make explain why many score around neutral ratings. Seeing the two peaks, makes us interested in what prediction power the sentiment scores may hold and will be something we want to include in our analysis.



Next, we looked at the sentiment scores for the reviews. This data looks to be normally distributed, and reviews tend to give a slightly above average review.



Another aspect that was interesting to look at was the Flesch scores. This library provides a score based on the reading comprehension ability needed to understand what was written. We suspected that most fake reviews would score very lowly on this scale and would be something to look out for when modeling. As we see from the graph below, there were many reviews that scored extremely low, which makes sense since even real reviews rarely have difficult language.



Preparation and Analysis

Preparation for this data took several steps. Since the data we were looking at was text based we need to use Natural Language Processing libraries to breakdown our data into meaningful numbers or categories. We feature engineered several columns to help extract meaning from the data. We took character and word count from both the reviews and summaries, we took the sentiment analysis from two different libraries, and acquired ease of reading ratings from the Flesch library. These scores were then standardized since they were all on different scales. Last for our Vader sentiment analysis we created dummy variables to create columns for neutral, positive, and negative scores. We split the data between train and test data and just had one last issue to solve before Modeling could occur. With over 3 million rows I was having difficult running cells in Jupyter notebooks, so for the sake of time I randomly removed about 70% of the data. While this would be usually be a bad choice in a real-world scenario, my computer does not have the computing power, so we had to make do.

We are now at the modeling process where I decided to choose three models for our data: Random Forest, Naïve Bayes Gaussian classifier, and Logistic Regression. Random Forest was a good choice in this scenario as it works well with categorical data and handles data with high dimensionality well. Naïve Bayes Gaussian Classifier was a good choice as it uses little computing process and is a simple model to understand what is going on "under the hood." Lastly, logistic regression is a great choice for classifier models with low computational cost and good results if the data does not have too much dimensionality.

The best metrics to rate these models would be precision, recall and f1-score since we a looking at a categorical problem with a simple yes/no answer. We focused more on recall because we want to remove as many spam reviews as possible. While there will be some real

reviews removed, false negatives, they are not as harmful in this scenario since we want to curate the reviews as best as possible to customers and removing some real reviews will not be detrimental. We ran each model with default hyperparameters and then tested on the test data. After our initial algorithms ran we can see that logistic regression provided the best recall by a good margin with an impressive 93.5%.

Algorithm	Preicision	Recall	F1-score
Random Forest	0.826740	0.894474	0.859274
Naive Bayes	0.829291	0.896779	0.861715
Logistic Regression	0.826272	0.935411	0.877461

Below we can see the ROC-AUC curve to see how much of the data each algorithm covered. We show both train and test score to ensure that no overtraining occurs, as we can see train and test scores are very similar and produce respectable scores. Logistic regression led the way with an 84% coverage score on both the training and testing data.

	Algorithm	ROC-AUC train score	ROC-AUC test score
0	Random Forest	0.813558	0.813338
1	Naive Bayes	0.829508	0.827569
2	Logistic Regression	0.836712	0.837393

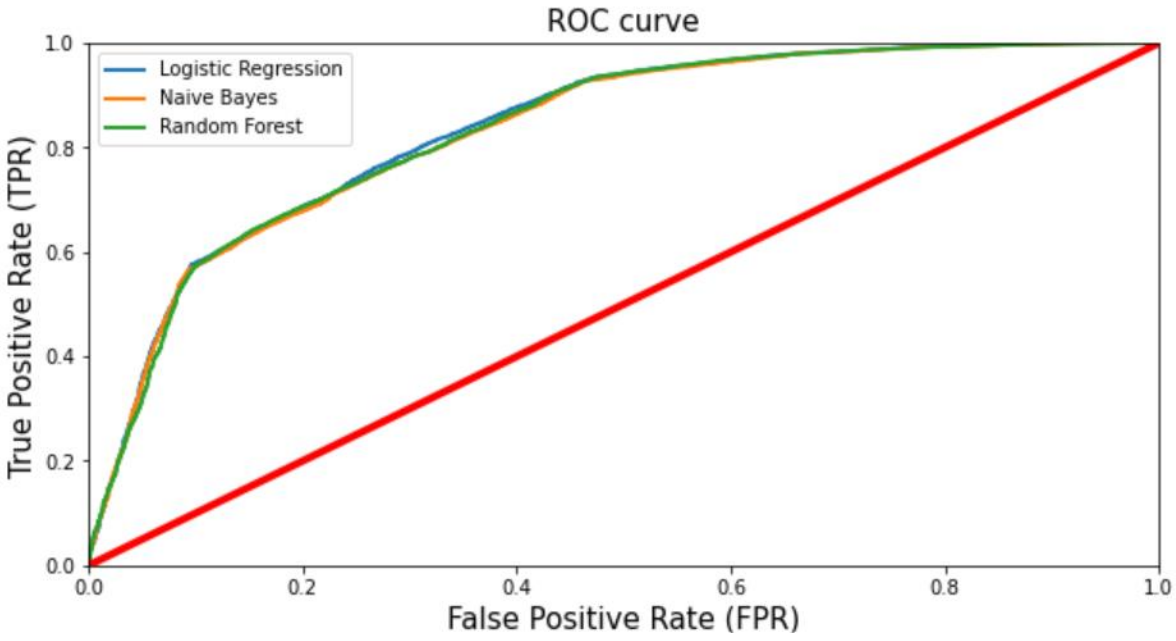
Now that we have a baseline and know that our models are not overfitting, we can begin to tune hyperparameters. We used GridSearchCV with a three-fold cross validation to check for the best hyperparameters for each model. At this point we see our recall scores have increased for Random Forest and Naïve Bayes, but only a marginal increase in Logistic regression.

Algorithm	Preicision	Recall	F1-score
Random Forest	0.824917	0.936728	0.877274
Naive Bayes	0.828958	0.927125	0.875298
Logistic Regression	0.826178	0.935576	0.877480

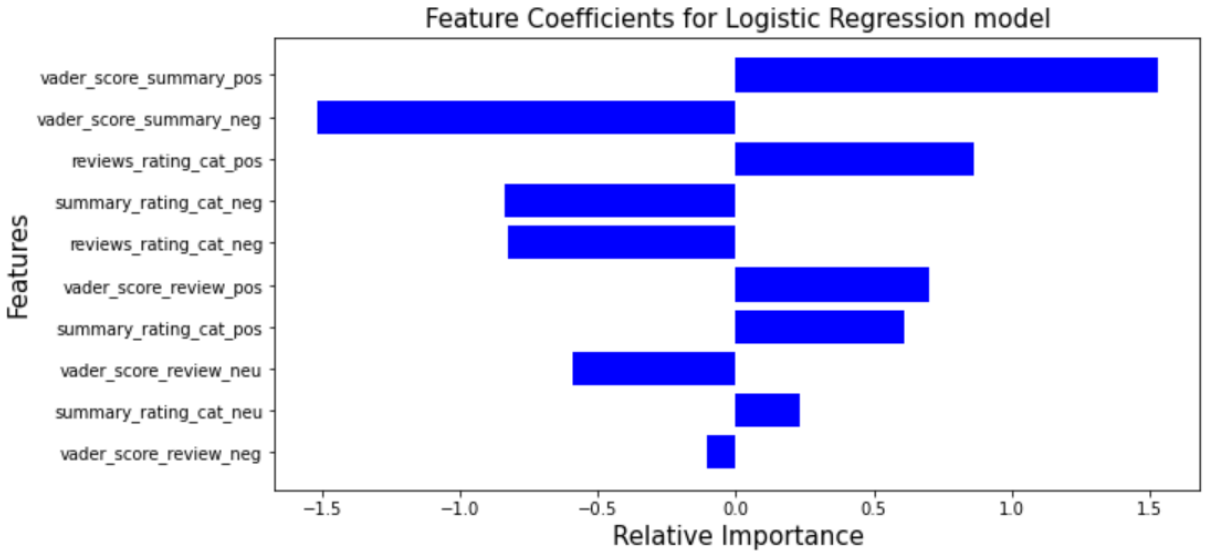
Next, we look at ROC-AUC scores where once again we see an increase in Random Forest and Naïve Bayes, with a small increase for Logistic Regression. A possible reason for this is that the final amount that the algorithm cannot get right are real reviews that are worded very similarly to fake ones.

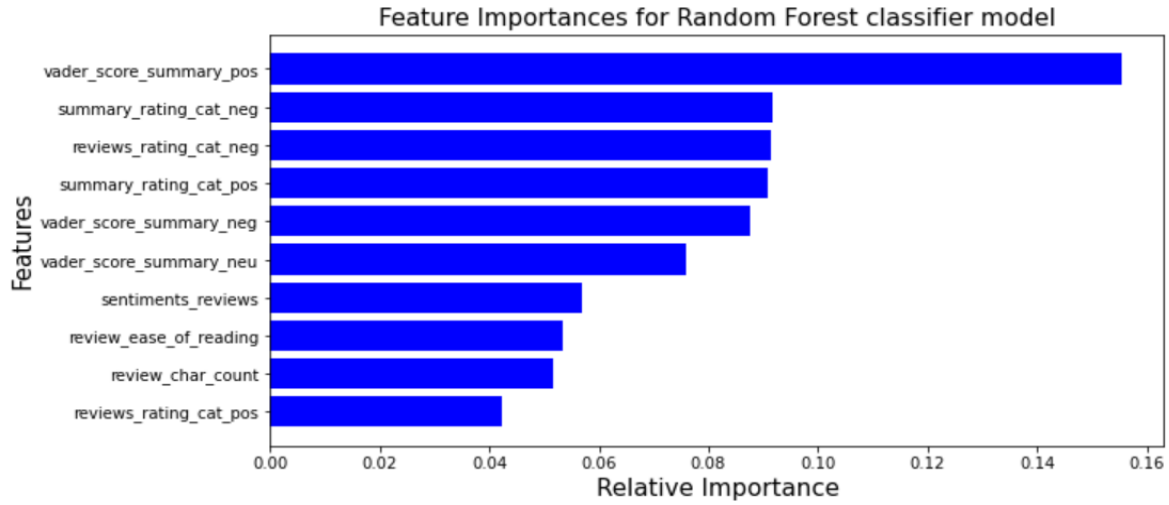
Algorithm	ROC-AUC train score	ROC-AUC test score
Random Forest	0.832839	0.833538
Naive Bayes	0.832150	0.827686
Logistic Regression	0.836347	0.836520

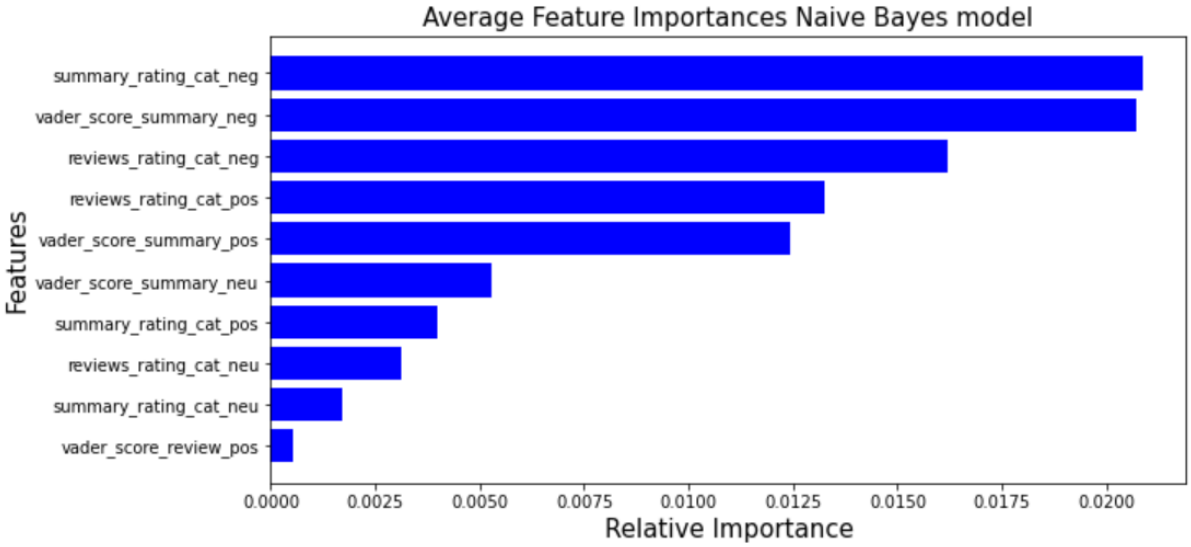
Illustrated below are the ROC-AUC test scores for all three algorithms. As one can see, even though the algorithms work in different ways, their coverage of the data is almost identical.



Below are the feature importance's for all three models and the weight for the top 10 features.







Takeaways

In this study we were able to get respectable scores and our models predicted spam vs. non-spam data with high accuracy. It was interesting to look at how the models ranked the features and how each of them had slightly different features in their top 10 rating. The feature that could provide some of the highest insight is a positive Vader score. It ranked first for Random Forest and Logistic Regression and ranked fifth for Naïve Bayes. This was taken from the Vader sentiment analysis rating, which is specifically designed for deciphering social media content, which amazon reviews are. This makes sense since spam reviews will rank the product five stars. We can infer from this that the models found that highly rated Vader scores was heavily related to a fake review. It also makes sense to see Negative summary scores to weigh heavily, as that could be a clear indicator or a real review, since fake spam reviews are focused on creating good scores.

Future Research

While our models produced a score over 90% there is still room to improve our model. Being able to use more data would be the most helpful, redoing this study with a better computer might produce even more accurate data. Furthermore, using the time and name features could have also been useful looking back on this study. There maybe certain hours that reviews are more likely to be written by humans compared to how the fake reviews are created. Also, when inputting the name feature, the spam reviews usually included a real name while real reviews would sometimes leave that blank. Humans are more likely to answer questions incorrectly or not at all, which could have helped us decipher from the two even more. Another way to improve this study would be to include entity types from the NLP library. It would have interesting to look at

how often names were brought up in the reviews and if held any relation to the prediction fake reviews.