ARE THEY LOVIN' IT?

SENTIMENT ANALYSIS OF FAST FOOD REVIEWS ON YELP

Springboard Data Science Career Track
By Sinead O'Connor
April 2020

I. Introduction

In the age of the Internet, people are constantly looking online to see what others think of restaurants, venues, hotels, etc. to try to make informed decisions about where they want to spend their money. With this in mind, negative reviews and comments on the internet can greatly affect a restaurant's bottom line. It is important for restaurant owners or managers to have an understanding of what customers are saying about their establishment online.

Rather than sifting through thousands upon thousands of tweets or comments, social media departments can use the results from predictive analytics models to quickly determine the public's perception of their restaurant, and use these results to inform actions to take to improve their services. For instance, businesses would be able to quickly identify negative reviews and respond to unhappy reviewers in a timely and professional manner. This will be especially useful for fast food establishments, many of which are regional, national, or even chains with many franchises.

In this project, I address the problem of building supervised models to estimate the probability of a given comment to be rated with a number of stars varying from 1 to 5.

II. Approach

A. Data Wrangling and Acquisition

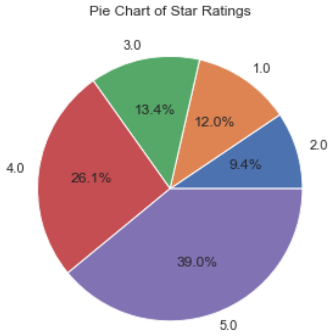
I will be using data from the 2019 Yelp Challenge. The Yelp Dataset consists of several files, including a business file, a review file, a user file, and a photo file. For the purposes of this project, I will only be using the business and review files. The business file contains information such as name, address, number of reviews, and the categories under which the business falls, for 192,609 businesses in the United States and Canada. The review file includes the user id of the reviewer, the business id, the rating (out of 5 stars), and the text of the review for 6,685,900 Yelp reviews.

I first downloaded the Yelp Dataset tar file and extracted the individual json files. Due to the size of the review json file, I installed Java Development Kit and pyspark, and loaded the business and review json files. I filtered the business data frame on the "categories" column to obtain only businesses with "Restaurant" included in the list, and I further filtered out any categories that seemed to be mislabeled, for example one of the remaining restaurants was also listed under the "Laser Hair Removal" category and turned out to be CanadaMed Laser. The resulting data frame had information on 53,324 restaurants. Using the business_id column which was common to the business and review datasets, I filtered the review data frame to obtain the 3,643,450 reviews for the restaurants.

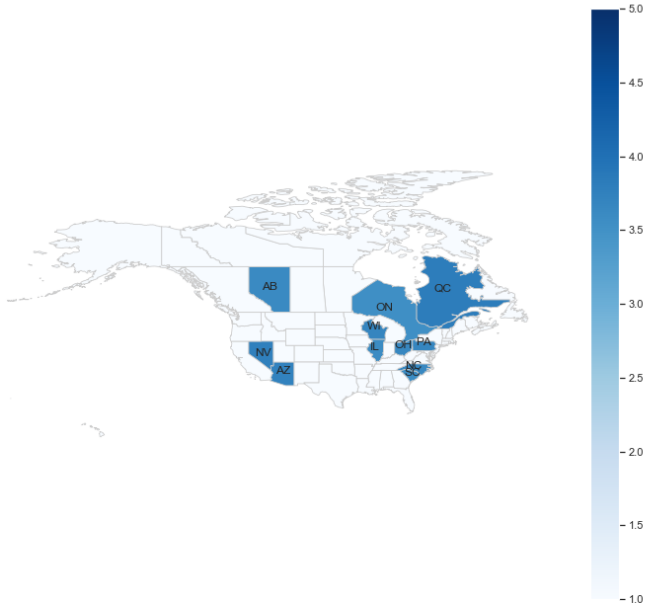
B. Exploratory Data Analysis - Data Storytelling

To initially explore the data, I examined various questions about all of the restaurants (both fast food and not.

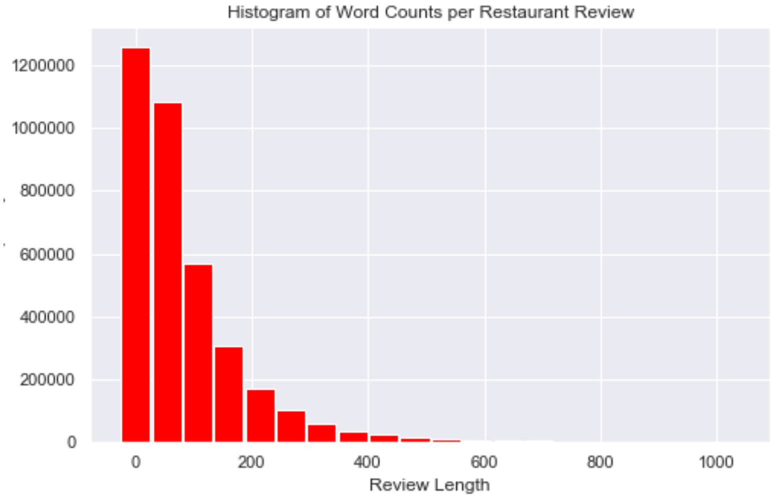
As seen in the below pie chart, of all the restaurant reviews, 65.1% were 4- and 5-star ratings. It is possible that restaurant patrons want to reserve 1- or 2-star ratings for restaurants that are particularly bad. Another reason the ratings may tend to be more favorable is because people are less likely to go to bad restaurants, so there would in turn be fewer reviews.



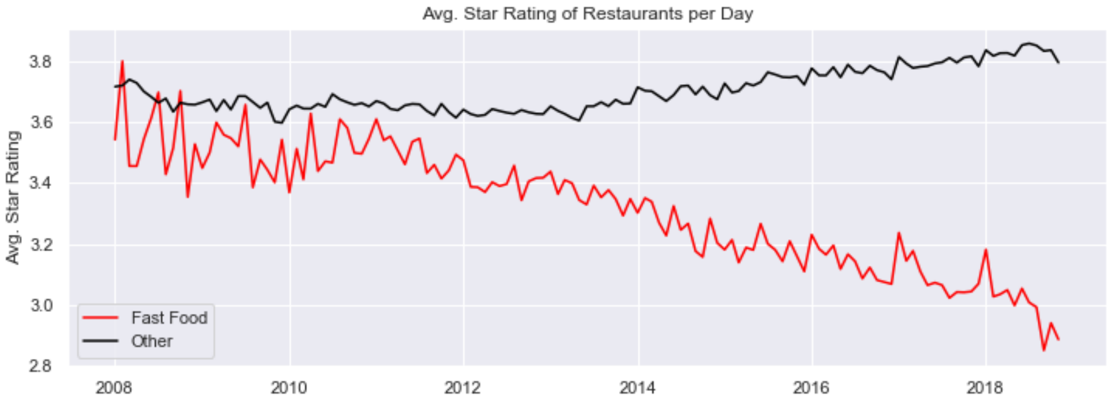
In examining the average star-rating by US state and Canadian province, we see that average rating does not appear to differ greatly from state to state. The lowest possible rating is 1 and the highest possible is 5, and we see on the below map, which was made using geopandas, that it is difficult to distinguish between the shades of blue. However Nevada and Quebec seem to have higher ratings, while areas like Ontario and Illinois appear to have lower ratings. The lowest average rating is 3.54 (Ontario) and the highest is 3.82 (Quebec).



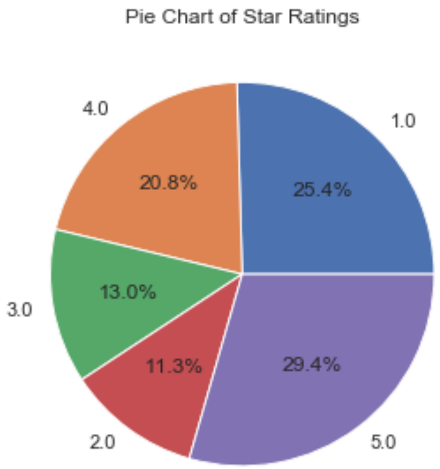
In this dataset, the smallest review length is 0 words, and the highest review length is 1,067 words. We see a strong right-skew in the below histogram with most reviews being under about 100 words. The frequency decreases as the length of the review (in words) increases. This makes sense because most people just write a few sentences on Yelp reviews, rather than paragraphs and paragraphs.



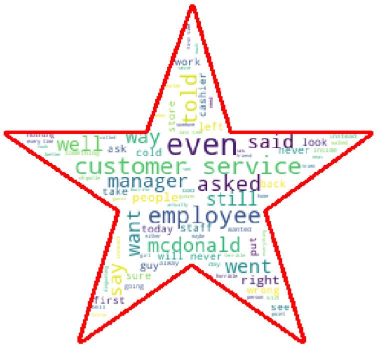
As we begin to compare fast food establishments to the non-fast food, we see that the average rating for the 220,507 fast food reviews (3.175 stars) is about 0.56 stars lower than the average rating of other 3.4 million other restaurant reviews (3.742 stars). The below time series plot also illustrates that non fast food restaurants have consistently been higher than those of fast food restaurants. In addition, fast food ratings seem to have decreased overall from about 3.6 in 2011 to about 2.9 by the end of 2018. This may be because Americans have realized that fast food is not healthy. For other restaurants, we see a slight overall increase in average reviews per month from about 3.6 in 2013 to about 3.8 at the end of 2018. There is more volatility in the fast food monthly ratings because the sample size is much smaller.



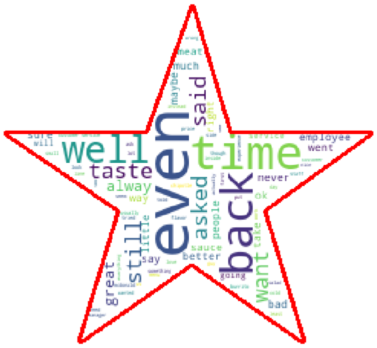
The most frequent rating of fast-food restaurant is 5-stars as it is with the all restaurants, however, fast food restaurant reviews were also given 1-star ratings 25.4% as seen in below pie chart.



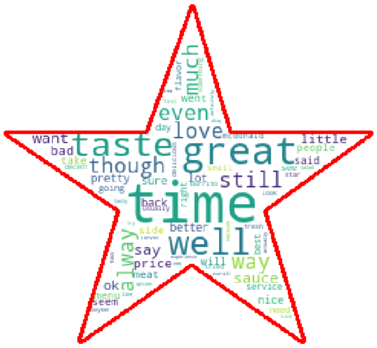
As a final step in this portion of Exploratory Data Analysis, I made word clouds for each star-rating to explore the different words used by reviewers based on their opinion of the restaurant.



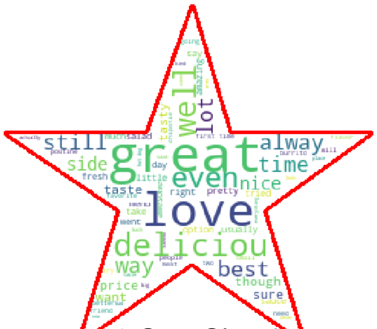
1-Star Cloud



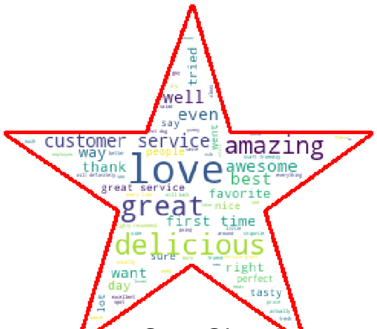
2-Star Cloud



3-Star Cloud



4-Star Cloud



5-Star Cloud

In the word-cloud for 1-

star fast food reviews, we see words such as "customer service", "employee", "staff", and "manager." This indicates that people who are giving 1-star ratings are often unhappy with service. We also see the word "manager" in only the 1-star review, which makes sense because customers only typically need to speak with a manager if they are very dissatisfied. We also see the word "service" in the 2-star cloud. We also see the word "cold" in the one-star and two-star clouds. Fast food customers typically expect hot food.

The 3-star cloud, has the word "great", but keep in mind it could be "not great." Note that I removed the word "good" because all 5 clouds had it prominently featured, but before I removed it one of the words was "pretty good." There are also the words "ok" and "bad." The word "though" appearing in the word cloud also indicates that the customer may have liked some things and disliked others. The 4-star cloud has many positive words prominently featured including "love", "delicious", "nice", and “best."

The 5-star could clearly has the most positive words. It has all the same words I mentioned in the 4-star cloud as well as adjectives like "amazing", "awesome" and "favorite", and "perfect".
It is also interesting to note that the 5-star word cloud has the phrases "customer service" and "great service." It shows that good or bad service may really make or break a review. It may be one of the things that brings a 3-star review down to a 1- or 2-star review, or what brings a 4-star review to a 5-star review.

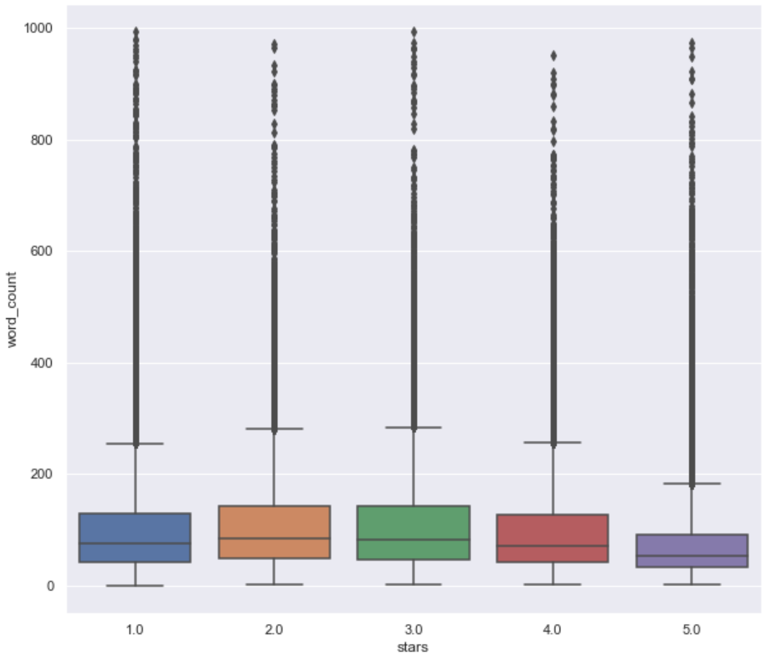
Going forward, I will only be using the fast food reviews and businesses.

C. Exploratory Data Analysis - Inferential Statistics

In the Inferential Statistics notebook, I tested several hypotheses about the fast food reviews.

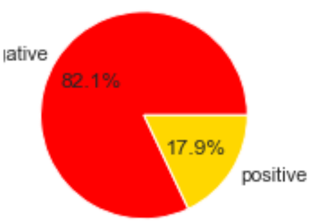
Here is a summary of the findings:

- We observed that the average rating of 29,284 reviews of Canadian fast food restaurants was 3.338. while the average of 191,223 US fast food reviews was 3.15. Using a two-sample z-test, we saw that this difference was statistically significant.
- There is a weak positive correlation of 0.23 between the length of a review in words and the number of useful votes the review received. The correlation is statistically significant, but it is important to remember that it could be statistically significant due to the large sample size of 220,507.
- From the below boxplots we see that, the five-star reviews seem to have a lot less spread than the others, and that, overall, the reviews tend to be shorter. The two-star and 3-star reviews have the highest average review lengths, 110.11 and 109.02, respectively. It makes sense that 2-star and 3-star reviews tend to be longer because there may be more to explain if did not have a great time at the restaurant or had a mixed review. Through an ANOVA test, we concluded that there is a statistically significant difference in at least one of the star-ratings. We rejected the null hypothesis that the average review length for one-star, two-star, three-star, four-star, and five-star reviews were all the same.

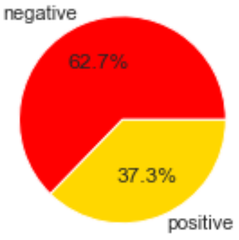


- In the fast food dataset, the proportion of positive (4- or 5-star) reviews for McDonald's (17.9%) was less than half the proportion of positive Subway reviews (37.9%), as seen in the below pie charts. There is a statistically significant difference in these proportions.

Proportion of Positive McDonald's Reviews



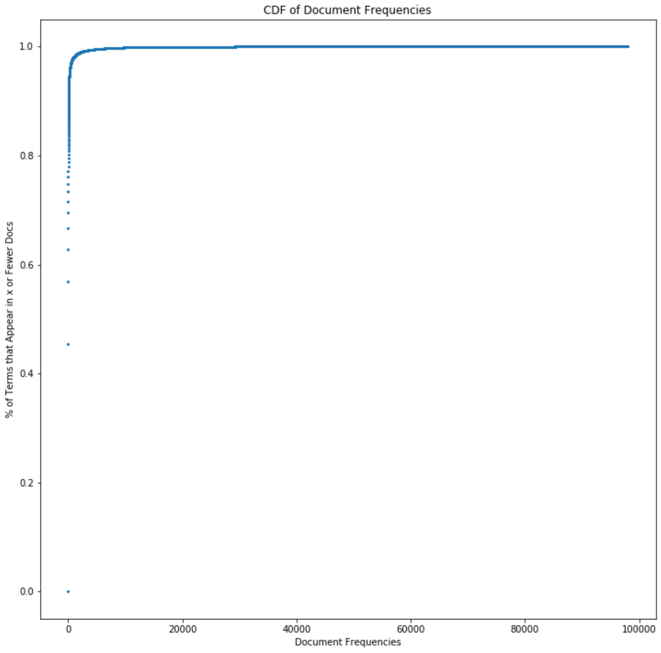
Proportion of Positive Subway Reviews

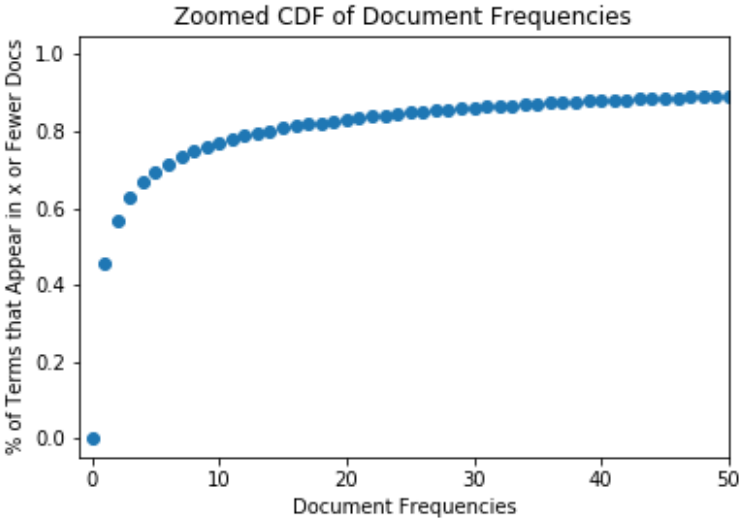


D. Baseline Naive Bayes & Logistic Regression Models:

Prior to feature engineering and training the baseline models, I preprocessed the reviews text. I tokenized the reviews using NLTK's wordpunct_tokenize. I also noticed that not all of the reviews were in English, so I decided to use the langdetect library to filter out non-English reviews. Given my computer was not powerful enough to use langdetect on the full set of 220,507 reviews, I first filtered reviews that had non-ascii characters and that contained French, Spanish, or German stopwords. This left us with 23,180 reviews to further examine, and using langdetect, I filtered out 843 non-English reviews and dropped them from the fast food data frame.

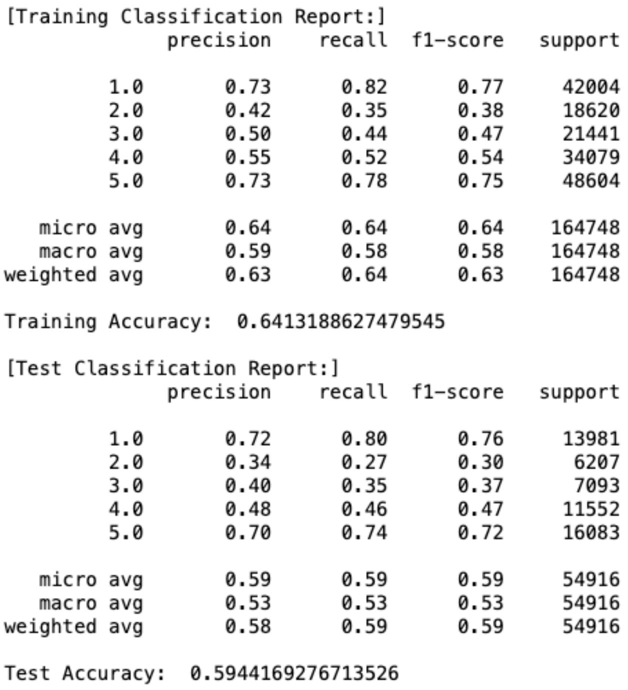
I first tried using Multinomial Naive Bayes. To engineer the features, I used a CountVectorizer on the tokenized reviews, supplying stop words, but without providing a minimum or maximum document frequency. The training accuracy was 0.653 and the test accuracy was 0.595 and I observe that there was imbalance among the classes. Given the 0.058 difference between the training and test accuracy it seems there is overfitting, so I tried tuning hyper parameters, using a CDF of document frequencies to choose a min_df and max_df. The word that appeared in the most documents was “food” which makes sense given they are fast food reviews.





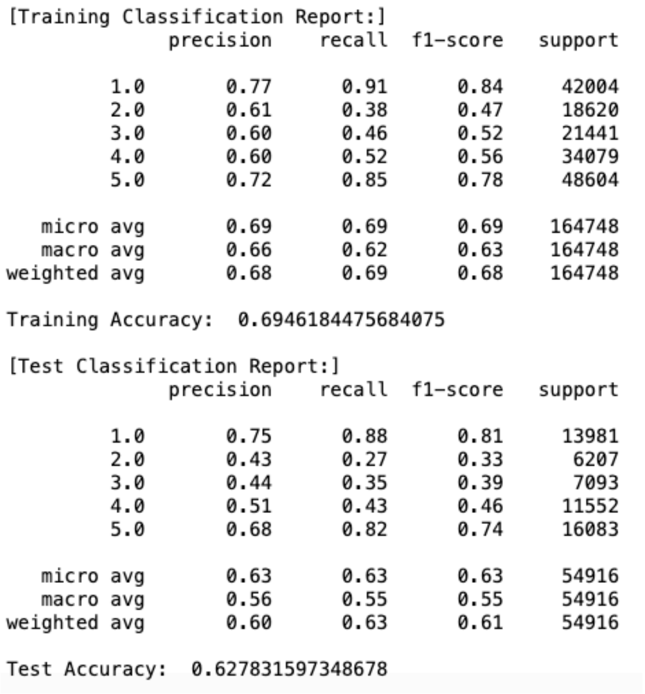
Based on the above Cumulative Distribution Function and zoomed version of it, I chose min_df = 3 and max_df = 70,000. I then trained both a Naive Bayes and Logistic Regression model, with the below results:

Naive Bayes:



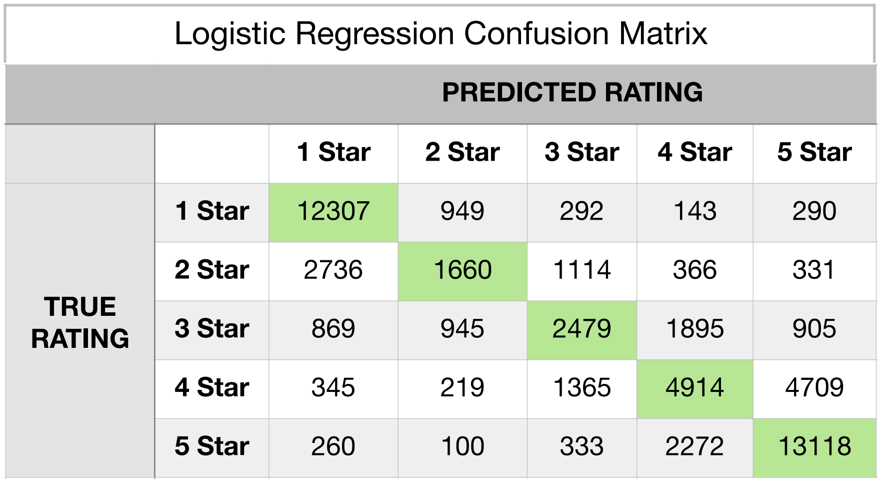
```
[Training Classification Report:]
              precision    recall  f1-score   support

         1.0       0.73      0.82      0.77     42004
         2.0       0.42      0.35      0.38     18620
         3.0       0.50      0.44      0.47     21441
         4.0       0.55      0.52      0.54     34079
         5.0       0.73      0.78      0.75     48604

   micro avg       0.64      0.64      0.64    164748
   macro avg       0.59      0.58      0.58    164748
weighted avg       0.63      0.64      0.63    164748

Training Accuracy:  0.6413188627479545

[Test Classification Report:]
              precision    recall  f1-score   support

         1.0       0.72      0.80      0.76     13981
         2.0       0.34      0.27      0.30      6207
         3.0       0.40      0.35      0.37      7093
         4.0       0.48      0.46      0.47     11552
         5.0       0.70      0.74      0.72     16083

   micro avg       0.59      0.59      0.59     54916
   macro avg       0.53      0.53      0.53     54916
weighted avg       0.58      0.59      0.59     54916

Test Accuracy:  0.5944169276713526
```

Logistic Regression:



```
[Training Classification Report:]
              precision    recall  f1-score   support

         1.0       0.77      0.91      0.84     42004
         2.0       0.61      0.38      0.47     18620
         3.0       0.60      0.46      0.52     21441
         4.0       0.60      0.52      0.56     34079
         5.0       0.72      0.85      0.78     48604

   micro avg       0.69      0.69      0.69    164748
   macro avg       0.66      0.62      0.63    164748
weighted avg       0.68      0.69      0.68    164748

Training Accuracy:  0.6946184475684075

[Test Classification Report:]
              precision    recall  f1-score   support

         1.0       0.75      0.88      0.81     13981
         2.0       0.43      0.27      0.33      6207
         3.0       0.44      0.35      0.39      7093
         4.0       0.51      0.43      0.46     11552
         5.0       0.68      0.82      0.74     16083

   micro avg       0.63      0.63      0.63     54916
   macro avg       0.56      0.55      0.55     54916
weighted avg       0.60      0.63      0.61     54916

Test Accuracy:  0.627831597348678
```

From the above classification reports, we see that the Naive Bayes model with min_df = 3 and max_df = 70,000 decreases overfitting slightly, however test performance is also slightly worse than when using the default parameters. The Logistic Regression model has a higher training and test accuracy, of 0.605 and 0.628, respectively, and has better precision, recall, and f1-scores. However, there is still overfitting for some of the classes. In both models, we see there is imbalance among the classes in the dataset, and the models are much better at classifying the larger categories (1-star and 5-star reviews) than the other reviews. This could also be because it's easier to identify extreme reviews.



Logistic Regression Confusion Matrix

		PREDICTED RATING				
		1 Star	**2 Star**	**3 Star**	**4 Star**	**5 Star**
TRUE RATING	**1 Star**	12307	949	292	143	290
	2 Star	2736	1660	1114	366	331
	3 Star	869	945	2479	1895	905
	4 Star	345	219	1365	4914	4709
	5 Star	260	100	333	2272	13118

From the above confusion matrix for the Logistic Regression model, we also see that most of the 1-star reviews that are misclassified, are misclassified, as 2-star review, and vice versa. Similarly, most of the 5-star reviews that are misclassified, are misclassified as 4-star reviews and vice versa. This is good because it's less concerning to misclassify a 4-star rating as a 3- or 5-star rating than as a 1-sat rating, for example.

I also tried running a Logistic Regression model on the same features with the parameter class_weight = 'balanced' to address the class imbalance, however it performed worse in each metric than the original logistic regression model, and the imbalance was even slightly more extreme.

E. Extended Modeling:

To address the problem of imbalance, I first redefined the problem as a binary classification problem, by labeling each review as "positive" or "negative" rather than as a function of the number of stars associated with it. I defined "positive" as any review with a 4- or 5-star rating and "negative" as any 1-, 2-, or 3-star review. I created a new column called "pos," with 1 indicating a positive review, and 0 indicating a negative review. The positive class now has 110,318 observations, while the negative class has 109,346 which has better balance between the classes.

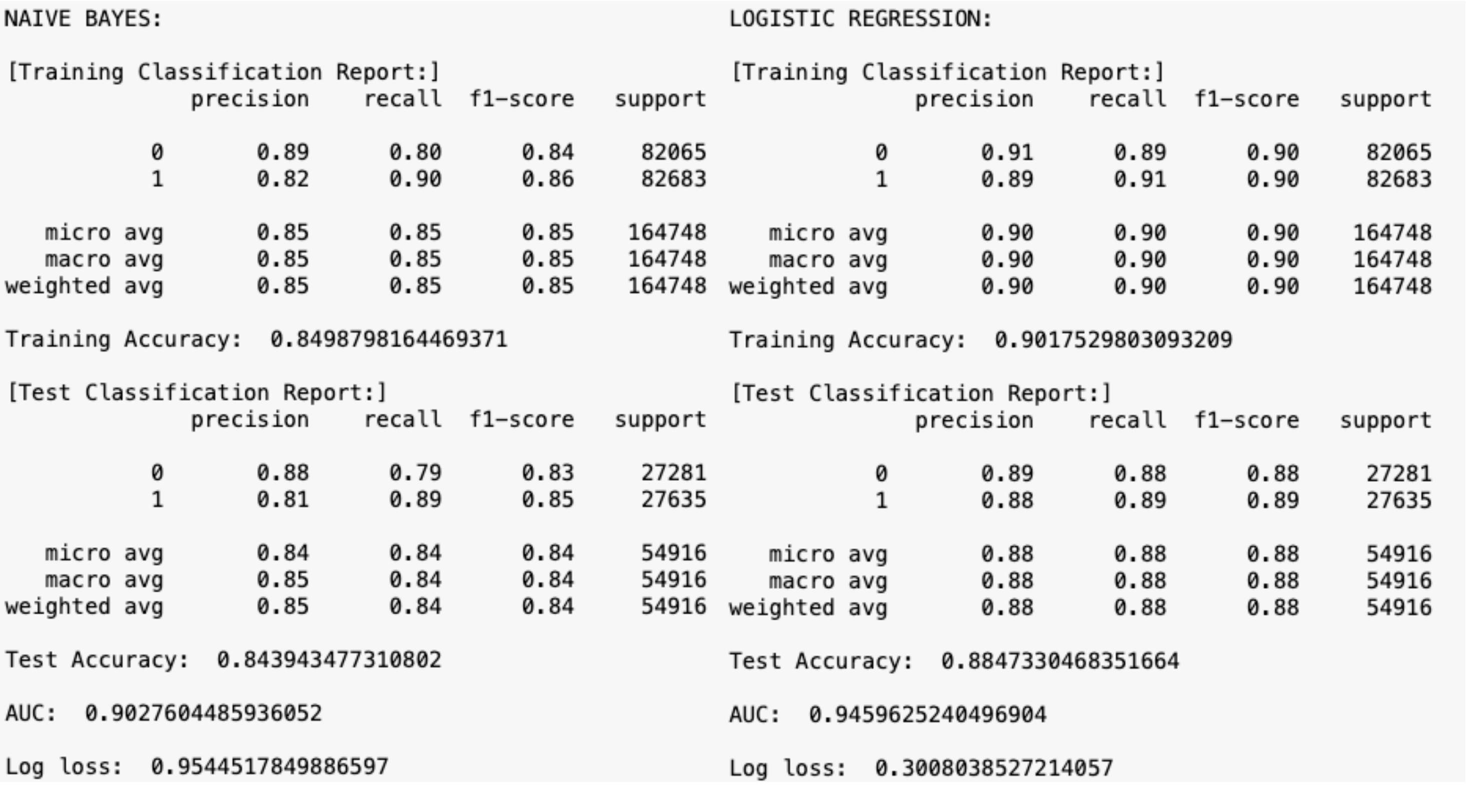
To get a new baseline, I completed the following steps:

- Split the data into a training set and test set.
- Used a CountVectorizer with a list of stop words, a minimum document frequency of 3, and maximum document frequency of 70,000 to extract features from the tokenized reviews. This resulted in 27,621 terms/features, and the one that appears in most documents/reviews was "good."
- Trained a Naive Bayes Model and a cross-validated Logistic Regression model with the column "pos", whether to not the review was positive, as the target vector.

From the classification reports above we see that the Logistic Regression model performed better than the Naive Bayes model with respect to every metric. We also note that class imbalance is less of a problem than in the baseline 5-class classification, especially in the binary logistic model. In both the Naive Bayes and Logistic Regression models, we see that there is not as much overfitting.

To further improve the binary classification performance, I did the following, using the same train/test split for each one:

- Ran a Random Forest Classifier using the same exact features as in the binary Naive Bayes and Logistic Regression models.
- Extracted features using Term Frequency - Inverse Document Frequency (Tfidf) vectorization still using min_df = 3 and max_df = 70,000, and ran a new cross-validated Logistic Regression model.

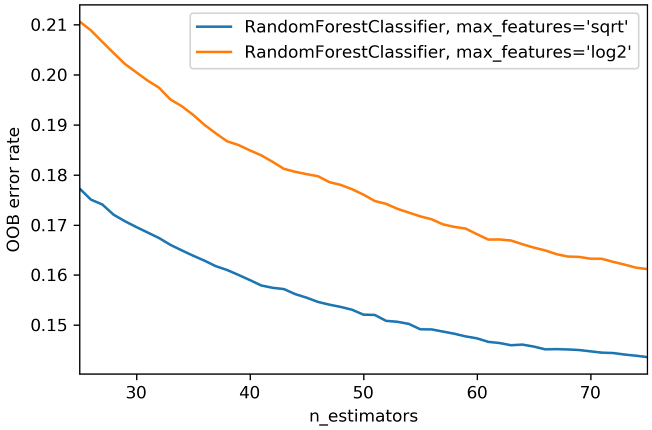


```
NAIVE BAYES:                                          LOGISTIC REGRESSION:

[Training Classification Report:]                     [Training Classification Report:]
              precision    recall  f1-score   support               precision    recall  f1-score   support

           0       0.89      0.80      0.84     82065            0       0.91      0.89      0.90     82065
           1       0.82      0.90      0.86     82683            1       0.89      0.91      0.90     82683

   micro avg       0.85      0.85      0.85    164748    micro avg       0.90      0.90      0.90    164748
   macro avg       0.85      0.85      0.85    164748    macro avg       0.90      0.90      0.90    164748
weighted avg       0.85      0.85      0.85    164748 weighted avg       0.90      0.90      0.90    164748

Training Accuracy:  0.8498798164469371                Training Accuracy:  0.9017529803093209

[Test Classification Report:]                         [Test Classification Report:]
              precision    recall  f1-score   support               precision    recall  f1-score   support

           0       0.88      0.79      0.83     27281            0       0.89      0.88      0.88     27281
           1       0.81      0.89      0.85     27635            1       0.88      0.89      0.89     27635

   micro avg       0.84      0.84      0.84     54916    micro avg       0.88      0.88      0.88     54916
   macro avg       0.85      0.84      0.84     54916    macro avg       0.88      0.88      0.88     54916
weighted avg       0.85      0.84      0.84     54916 weighted avg       0.88      0.88      0.88     54916

Test Accuracy:  0.843943477310802                     Test Accuracy:  0.8847330468351664

AUC:  0.9027604485936052                              AUC:  0.9459625240496904

Log loss:  0.9544517849886597                         Log loss:  0.3008038527214057
```

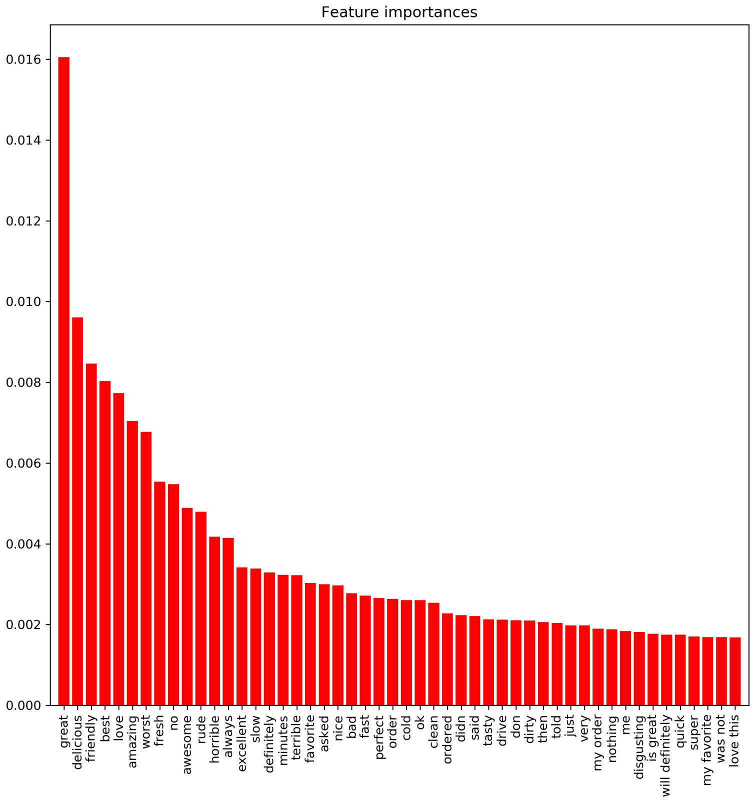
- Extracted features using a TfidfVectorizer, this time with ngram_range = (1,2), to extract both unigrams and bigrams from the reviews, rather than just unigrams. Using a similar Cumulative Distribution Function plot as the one in the baseline, I chose max_df to be 45,000 as the plot seemed to level off there. I originally chose the minimum document frequency to be 6, but this resulted in overfitting when a fit a cross-validated Logistic Regression model, so I increased it to 35. The only stop words I removed were "a," "an", "and", "the", and punctuation. This resulted in 51,515 features/terms and the term that appeared in most documents was "service." I did not want to remove too many stop words when adding bigrams as features because "good" and "not good" have very different meanings, but if we removed "not" which is in the list of NLTK stop words, we would not be able to capture that difference in meaning.

For both Random Forest models, I selected a value for the "max_features" parameter by creating a plot of out-of-bag error rate versus number of trees. In both cases, max_features = 'sqrt' minimized the OOB error rate so I chose max_features to be square root of the total number of features and I chose n_estimators to be 500.

Due to computing power, I was not able to test to see the OOB error rate for max_features = None, because it took my computer too long to run. Below is the plot generated for the RandomForestClassifier built using features from the TfidfVectorizer with unigrams and bigrams.



In addition, I used this Random Forest feature importance method to identify the 50 most important features (unigrams and bigrams) in making the prediction, as seen in the graph below.

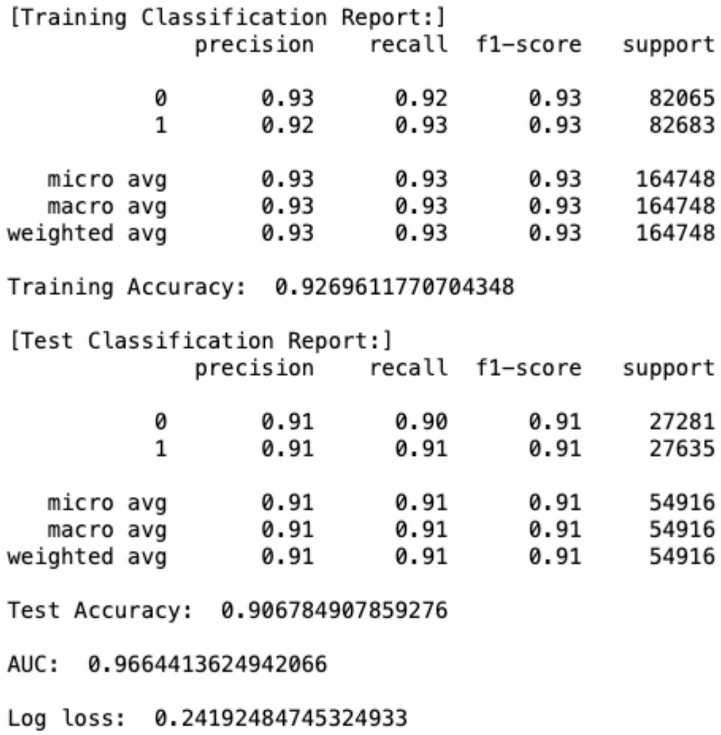


We see that the most important words include many adjectives and superlatives with positive connotations such as “great”, “delicious,” “friendly,” and “best,” as well as words with negative connotations in the service industry such as “rude,” “bad,” “slow,” and “dirty.” I was surprised to see words such as “said” in the list because it does not seem like it would be predictive.

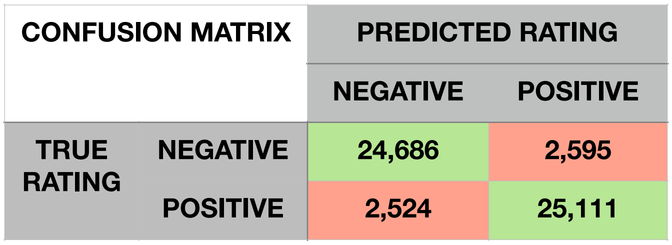
The below table summarizes results for each model. For each test metric, I have highlighted the best value in green, and the worst value in red.

MODEL	CLASS	PRECISION	RECALL	F1	SUPPORT	LOG LOSS	ACCURACY
NaiveBayes-CountVec, Unigrams	Negative	0.88	0.79	0.83	27281	0.9544	0.8439
	Positive	0.81	0.89	0.85	27635		
LogReg-CountVec, Unigrams	Negative	0.89	0.88	0.88	27281	0.3008	0.8847
	Positive	0.88	0.89	0.89	27635		
RandForest-CountVec, Unigrams	Negative	0.87	0.87	0.87	27281	0.3818	0.8667
	Positive	0.87	0.87	0.87	27635		
LogReg-TfidfVec, Unigrams	Negative	0.89	0.88	0.89	27281	0.277	0.8864
	Positive	0.88	0.89	0.89	27635		
LogReg-TfidfVec, Uni & bigrams	Negative	0.91	0.90	0.91	27281	0.2419	0.9067
	Positive	0.91	0.91	0.91	27635		
RandForest-TfidfVec, Uni & bigrams	Negative	0.88	0.88	0.88	27281	0.3690	0.8772
	Positive	0.88	0.88	0.88	27635		

Of all the binary classification models that I built, the one that performed the best was the Logistic Regression model with unigrams and bigrams built using Tfidf vectorization. It performed the best with respect to every test metric that I measured, including precision, recall, F-1 score, accuracy. Below is the classification report and the confusion matrix:



```
[Training Classification Report:]
              precision    recall  f1-score   support

           0       0.93      0.92      0.93     82065
           1       0.92      0.93      0.93     82683

   micro avg       0.93      0.93      0.93    164748
   macro avg       0.93      0.93      0.93    164748
weighted avg       0.93      0.93      0.93    164748

Training Accuracy:  0.9269611770704348

[Test Classification Report:]
              precision    recall  f1-score   support

           0       0.91      0.90      0.91     27281
           1       0.91      0.91      0.91     27635

   micro avg       0.91      0.91      0.91     54916
   macro avg       0.91      0.91      0.91     54916
weighted avg       0.91      0.91      0.91     54916

Test Accuracy:  0.906784907859276

AUC:  0.9664413624942066

Log loss:  0.24192484745324933
```



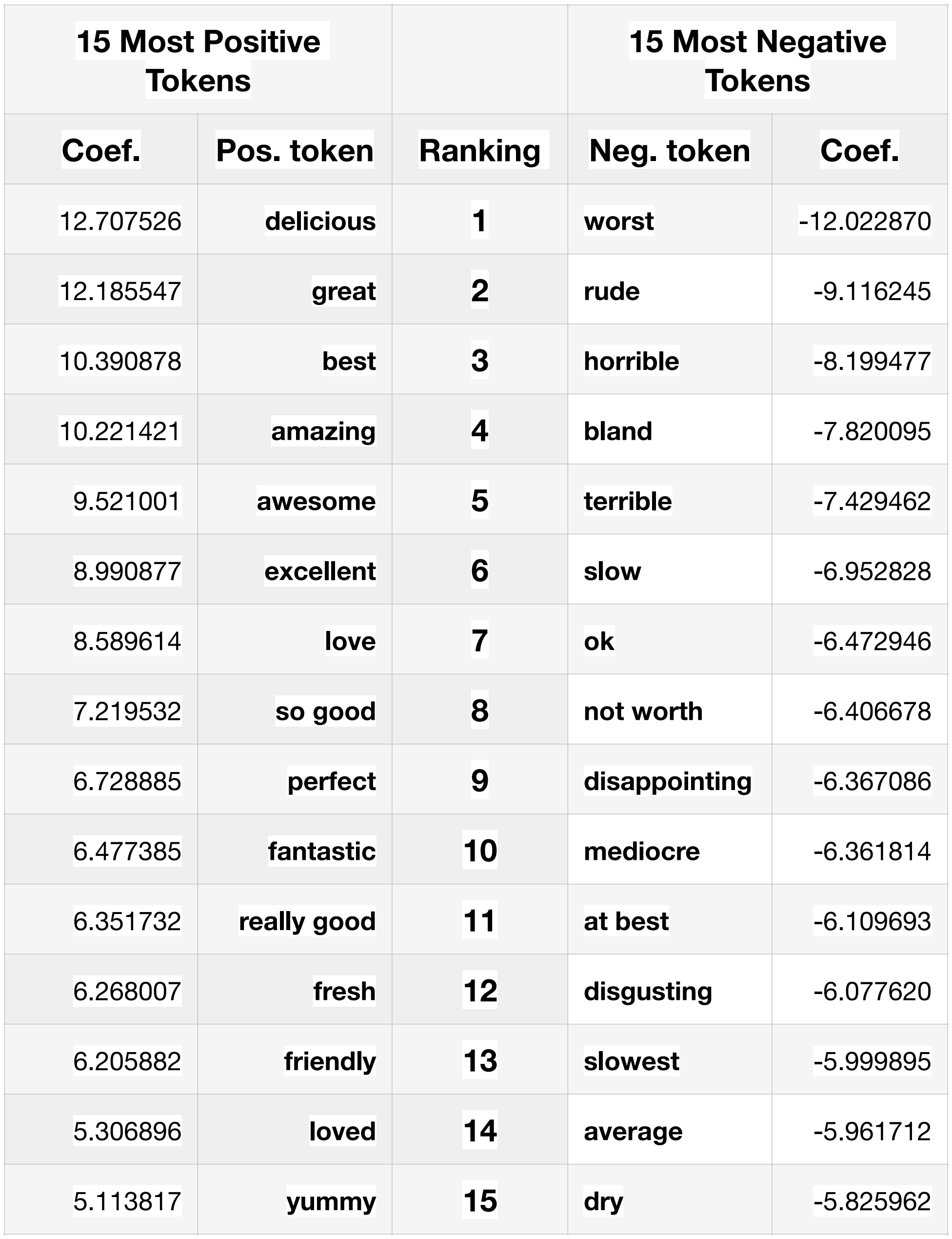
CONFUSION MATRIX		PREDICTED RATING	
		NEGATIVE	POSITIVE
TRUE RATING	NEGATIVE	24,686	2,595
	POSITIVE	2,524	25,111

F. Discussion of Results

Given that the Logistic Regression model trained on features extracted from Tfidf vectorization with unigrams and bigrams performed best with respect to each metric, it is not difficult to decide which model to use. However, in the event that we develop models in the future that perform better with respect to certain metrics and not all, I will provide a table to show the business interpretation of choosing the model based on certain test metrics.

Test Metric	Current Best Value	Business Interpretation
Accuracy	0.9068	90.68% of total predictions were correct. It may be reasonable to choose the model with the highest accuracy since the classes are pretty balanced.
Precision	0.91	Both the positive and negative class had precision of 0.91 meaning when the model predicted the review was positive it was correct 91% of the time, and when it predicted the review was negative, it was also correct 91% of the time.
Recall	0.91	The model has 0.90 recall on the negative class (class 0) and 0.91 on the positive class (class 1), meaning when the review was truly negative, the model correctly predicted it was negative 90% of the time, and when the review was truly positive, the model was correct 91% of the time.
F1 Score	0.91	This is a combination of precision and recall.

By identifying the features with the largest positive coefficients we are able to identify the tokens that the Logistic Regression model is finding to be the most positive, since the larger the positive coefficient is, the more the feature/token will contribute to a higher predicted probability of a positive review. Similarly, we are able to identify the tokens most indicative of a negative review by identifying the features with negative coefficients that have the largest absolute values. As seen in the below list of the 15 most positive and 15 most negative unigrams and bigrams, the determined mathematically by the model seem to align quite well with intuition. The most positive words include "delicious," "positive," and "best," while the most negative are "worst," "rude," and "horrible." None of the words in the below list are particularly surprising. We also see the benefit of including bigrams, for example "best" is the 3rd most positive token and "at best" is the 11th most negative token.



15 Most Positive Tokens			15 Most Negative Tokens	
Coef.	**Pos. token**	**Ranking**	**Neg. token**	**Coef.**
12.707526	**delicious**	**1**	**worst**	-12.022870
12.185547	**great**	**2**	**rude**	-9.116245
10.390878	**best**	**3**	**horrible**	-8.199477
10.221421	**amazing**	**4**	**bland**	-7.820095
9.521001	**awesome**	**5**	**terrible**	-7.429462
8.990877	**excellent**	**6**	**slow**	-6.952828
8.589614	**love**	**7**	**ok**	-6.472946
7.219532	**so good**	**8**	**not worth**	-6.406678
6.728885	**perfect**	**9**	**disappointing**	-6.367086
6.477385	**fantastic**	**10**	**mediocre**	-6.361814
6.351732	**really good**	**11**	**at best**	-6.109693
6.268007	**fresh**	**12**	**disgusting**	-6.077620
6.205882	**friendly**	**13**	**slowest**	-5.999895
5.306896	**loved**	**14**	**average**	-5.961712
5.113817	**yummy**	**15**	**dry**	-5.825962

We may also examine examples of reviews that were misclassified to try to understand why the models were misclassifying certain ones. For, example:

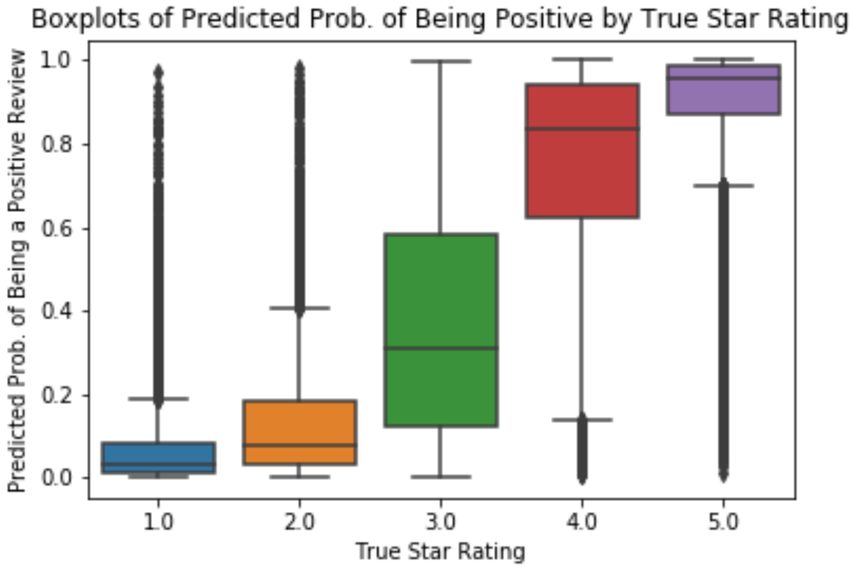
- 5-star/positive review that was misclassified as negative (with only a 0.0088 probability of being positive): `"Worst fatburger in the whole city, always smells like burnt grease, employees all look dirty, few times eating there can't get order right”`
  - It seems like this review may have either been fake or the Yelp reviewer may have accidentally selected the wrong star-rating.
- 5-star/positive review that was misclassified as negative (with only a 0.0282 probability of being positive): `"I like the Protein Source and find the staff very nice, there seems to be a big turnover so if some staff don't seem up to par then maybe it's because`

of that. Some yelper named Edith was extremely critical , almost referring to staff members as psychologically damaged, possible jealousy on this woman's behalf perhaps. I just hope the manager didn't dismiss people because of her review, you're a cruel and sick woman you probably complain everywhere you go! Do you like getting young people fired, you basically suggested they do that, thats extremely vindictive ; what were you like when you were a teenager around 18 or 19. You gave backhanded complements and referred to staff as lost souls, to me you are the one with no soul. You made personal attacks on these young people and some of the things you said are disgusting. If the upper management fired people because of your reviews alone then I wont be back there especially if I don't see the same people I won't go back there anymore! As for the managers of this place, if you take what one person says as the overall opinion of everyone I think you should go back into training!"

  - A human reader may be able to easily detect that this review is clearly a criticism of another reviewer, Edith, who left a negative comment rather than the restaurant, however, a computer would likely see all the negative words and classify it as negative.

- 1-star/negative review misclassified as being positive (with 0.9741of being positive): "Was one of the best parts of this aging casino until some marketing genius from Boyd Gaming decided to close this hidden gem. I'm glad they listened to their hotel guests who rave about this place. Great job boyd gaming."
  - To me this review is quite sarcastic, but a computer may miss the sarcasm because many of the words/bigrams have positive connotations- "hidden gem", "rave", "great job" for instance, however the reviewer seems to be expressing disappointment with the restaurant closing. It may make sense to filter out reviews for closed restaurants in the future for this reason.

Although we used binary classification, we can also examine the predicted probability of the review by the underlying true rating. From the below box plots, we can clearly observe that overall reviews that were given 5-star ratings tended to have the highest probability of being positive, 4-star the second highest, 3-star the 3rd highest, 2-star the 4th highest, and 1-star the lowest which is a good sign. We also see that the 3-star reviews have the largest variance/spread which would be expected since it may be more difficult to classify a 3-star review as positive or negative.



III. Conclusions

A. Summary

In conclusion, I have wrangled business reviews from the Yelp Challenge dataset, and filtered them to obtain only the fast food reviews. I tokenized the text of the reviews and removed non-English reviews prior to fitting models as non-English reviews may have negatively affected performance. I started with a 5-label classification problem, training both a Multinomial Naive Bayes and a Logistic Regression model. To address the class imbalance observed in both models, I redefined the problem as a binary classification problem - trying to predict whether a review is positive (4-5 stars) or negative (1-3 stars).

I then fit one Naive Bayes model and several Random Forest and Logistic Regression models using a combination of different text feature extraction techniques - CountVectorizers and TfidfVectorizers. Some models were built using just unigrams while others were built using both unigrams and bigrams. Each of the models had the same train/test split, so we were able to compare the results of the models being trained and tested on the same underlying set of reviews.

B. Future Work

There are several other Natural Language Processing and Machine Learning techniques I would like to apply to the problem, in the future in an attempt to improve the classification model, including but not limited to:

- Implementing different algorithms to build models, such as a Support Vector Machine and a Simple Neural Network.
- Lemmatizing tokens so that words such as “loved” and “love” or “has” and “have” and will be recognized as having the same base meaning.
- Revisiting the 5-class problem. I will first try using a stratified train test split to ensure the proportions of each class are the same in the training set and the test set. I will then try using resampling methods and the imblearn package to deal with imbalanced classes. It may also make sense to create 3 classes- “positive”, “neutral,” and “negative.”
- Applying the models to other restaurants besides fast food restaurants to see how well it generalizes.

IV. Recommendations to Client

Given that this model performed best with respect to each metric, it is not difficult to decide which model to use for now. I recommend using the cross-validated Logistic Regression model fit using term frequency-inverse document frequency vectorization with unigrams and bigrams, as it performs best with respect to every test metric measured. Clients can use this model to get a good idea of how their customers are responding to their fast food chains or individual restaurants on social media.

In addition, by filtering comments with very low predicted probability of the comments being positive will allow clients to quickly identify very negative feedback that may require rapid responses from the business's social media department. It will also allow clients to identify reviews with high probabilities of being positive, so they can acknowledge the commenter's kind words.