Project 4

Objective:

The main goal for this project was to use NLP to analyze Yelp reviews and extract features from the text to categorize the reviews into useful and not useful categories. Successfully categorizing reviews as useful has many business applications. Users would spend less time filtering out unhelpful reviews and would have a better experience using Yelp's product. Business owners would gain invaluable insight about their customer base and the aspects of their business that could be changed to have the highest impact on improving customer satisfaction. Finally, Yelp could implement this feature behind a paywall to generate revenue from business owners using their service or offer it for free and increase traffic to their website and app.

Data:

I downloaded review, business, and reviewer information from the Yelp dataset. The format for the data was JSON. From this large dataset, I used 3500 reviews from 12 In-N-Out Burger Restaurants in Las Vegas.

*I started off looking at mexican restaurants in Las Vegas but found that the my topics were revolving around different food items such as taco, burrito, quesadilla which wasn't helpful for the goal of my project. I narrowed it down to restaurants that sold burgers but that was still too general. I ended up using In-N-Out Burger because they were the most highly represented restaurant in the dataset and would provide me with the largest number of reviews .

Analysis:

I started off with a linear pipeline and wanted to use Multinomial Naive Bayes to try and predict usefulness by using a count vector of unigrams and bigrams of the text. For preprocessing, I tokenized the text first and removed stopwords. To add custom stopwords I made a count for the most common words and removed the ones that showed up in the majority of the reviews. After creating bigrams and adding it to the text, I used a count vectorizer and used these vectors as the only features. I tried several different iterations with just unigrams and with trigrams instead of bigrams but all the models had low predictive power.

My next step was to use topic modeling to reduce the dimensionality of the data and to use the topics of each review as features in various classification algorithms. I used LDA and LSI with the TF-IDF of the words in the reviews. I eventually chose 3 topics as my final features based on the explained variance of the topics and also by choosing the number of topics that resulted in the highest AUC scores. The algorithms I tested were Naive Bayes, Logistic Regression, and Gradient Boosting. I performed a grid search with a cv of 5 to determine the optimal hyperparameters. None of my models were overfitting and often had test scores better than train scores. Of the algorithms I tried, I chose to move forward with Logistic Regression because it performed the best and is simpler than Gradient Boosting.

With just the topics of each review as features, I wasn't able to predict usefulness much better than baseline models of guessing the mode. I decided to add the length of the review as a feature and this proved to improve the AUC by about 0.1. Some other features that I tried to

include was the polarity of the review, parts of speech used in the review, and also reviewer information. I used TextBlob for sentiment analysis and to count parts of speech. For reviewer information I included: total number of posts of each reviewer, number of fans, and number of useful votes the reviewer has received. None of these features improved my model's predictive power in a significant way so I excluded them from my final model. I suspect that I may have had a bug in my code when adding reviewer information and that I didn't match the correct reviewer with the correct review.

Conclusion:

For this project, I wanted to maximize the precision of my model because I wanted to have high confidence in the usefulness of the reviews that were categorized as useful. By setting the threshold of useful to 0.65 my model had a precision of 70% and a recall of 10%. My model can essentially identify 1/10 useful reviews with 70% confidence! This is promising looking into the future because I think there are several ways in which I could drastically improve both precision and recall.

I would like to use Word2Vec models and see if I can generate more distinct topics. I also think that properly adding reviewer data as features would be highly beneficial. Finally I think that using a CNN to analyze pictures included in the reviews would significantly improve my model.

Tools:

- Jupyter Notebooks
- Python
    - Sklearn
    - Numpy
    - Pandas
    - TextBlob
    - Gensim
    - Matplotlib
    - Nltk
    - Json
- Google Slides