CAPSTONE PROJECT #1

In-Depth Analysis

Author: Jerome Gonzaga

Problem Statement

Current Issue: It can often be very difficult to find a good restaurant, good mechanic, good barber shop, good cafe, etc. People can find a business on yelp and see their associated star rating but it's hard to trust the star rating. Many times I find that people look at reviews and photos to determine if the business fits what they are looking for and even this method can result in mixed conclusions about a business.

Problem to solve: Is there a way to optimize searching for a business based on other characteristics? If so, what sort of patterns in the dataset can we find in order to base the searches on?

Data Used:

1. https://www.yelp.com/dataset

Documentation for the available data: https://www.yelp.com/dataset/documentation/main

In-Depth Analysis

As a way to start the in-depth analysis, the features considered for machine learning were the review texts written by users. The target for the machine learning model was to correctly predict the user's Yelp rating score given the review text that's fed to the model. The idea of the machine learning model is that the model gets all of the user's review text from a particular business category of interest and the associated Yelp score, analyzes the review text to determine the top words used by the user per Yelp rating score, then after being trained with this data, predict the Yelp rating score of the review text of that same particular business category from other user's. Based on the predicted Yelp rating scores, a list of recommended businesses is provided.

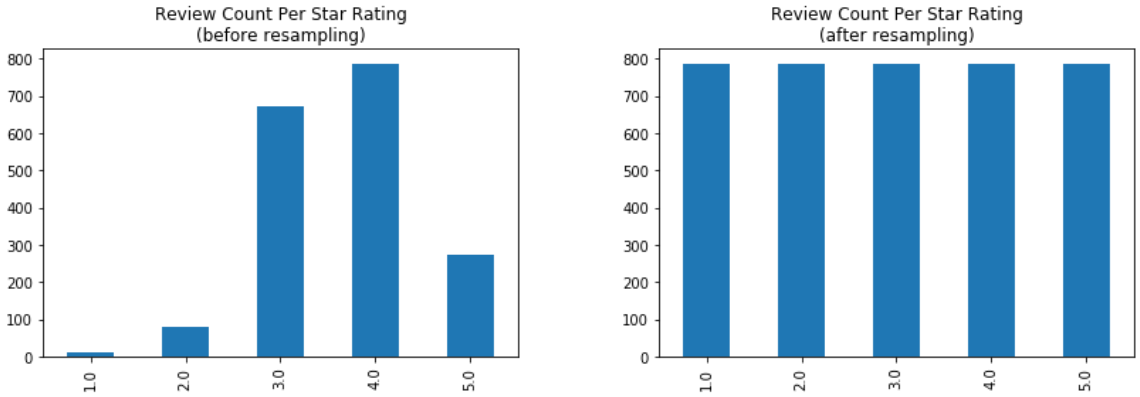
Building the Model

Since the Yelp rating scores associated with the review texts are provided, this is a supervised machine learning. There were a large number of features that could be considered for the model to predict the Yelp rating score. Some of the potential features that can be explored include:

- The number of check-ins a business received over a period of time
- The number of reviews received by businesses
- The text from the tips provided by users for businesses
- The businesses that a user has in common with friends and/or fans
- The attributes offered by a business

There are a large number of features that can be explored but for the purposes of this project, only a user's review texts were considered. In the future, one or even more of these other features can be considered in order to better predict a user's Yelp rating score.

Before the model analyzes the review texts, one significant step that was implemented was to ensure that each classification has the same amount of samples. Before performing this step, I had originally provided the model the unaltered review text dataset. Often, there would be more reviews associated with one or two of the Yelp rating scores (ie. more reviews for a rating of 3 versus a rating of 5). As a result, the model predictions were heavily skewed by the review texts for one or two particular rating scores. So, a resampling step was performed before the review text dataset was fed to the model so that for rating scores that had less review texts a random sampling was performed to duplicate review texts for a specific classification so that the number of reviews per classification would be equal.



Example of resampling review texts to equalize the number of reviews per star rating

This model takes each of the review texts, by row, and passes each review text through a pipeline. The pipeline first vectorizes the review using CountVectorizer(). During the CountVectorizer step much of the text preprocessing is done here including removing stopwords that are very common in the english language, stemming the words to get rid of variations in word tense, and also tokenizing review text phrases using 1-gram and bigrams, After each of the review texts are vectorized, the output matrix is scaled to now be heavily skewed by some of the outlier tokens. Then the matrix is then passed onto the TruncatedSVD step to find the features have a more significant impact on the prediction of the Yelp rating score. Lastly in the pipeline, the OneVsRestClassifier and LogisticRegression. The review texts are split into a training set and a test set in order to very the performance of the model.

Evaluating the Model

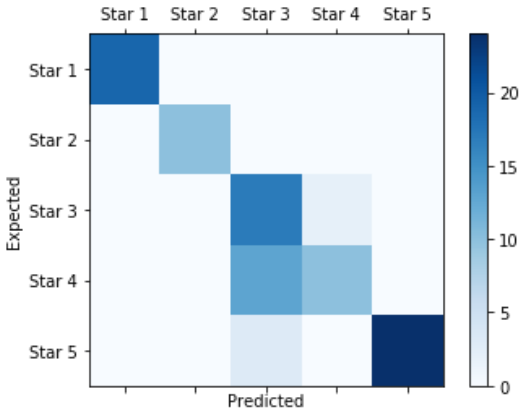
As an example, here are the output metrics and recommendations for one user for the business category of “breakfast, food, restaurant”:

Training Accuracy: 1.0
Testing Accuracy: 0.816

Classification Report:

	precision	recall	f1-score	support
1.0	1.00	1.00	1.00	19
2.0	1.00	1.00	1.00	10
3.0	0.52	0.89	0.65	19
4.0	0.83	0.43	0.57	23
5.0	1.00	0.89	0.94	27
accuracy			0.82	98
macro avg	0.87	0.84	0.83	98
weighted avg	0.87	0.82	0.82	98

Confusion Matrix:



The model determined that the top 20 tokens used as features for predicting Yelp rating scores were the following:

1. churro tot
2. 2 shrimp
3. 2 slice
4. 18 brunch
5. beauti
6. 10 awhil
7. 7 pm
8. abl walk
9. 10 clam
10. 6 ashley
11. allow use
12. 50 great
13. addendum
14. 3 dip
15. onlin went
16. 10th floor
17. almond orang
18. adjust
19. 3 salsa
20. 3 salsa

The reason for duplicates for the tokens is possibly due to the resampling preprocessing step performed on the review texts. The truncated words are a result of the stemming preprocessing.

Finally, the next step that will be part of the final report is using this model to provide recommendations to the user for businesses that most probably will fit the user's likings (based on the user's review text from other similar businesses). One consideration to note, however, for this recommendation system at this time is that the recommendations are based on both the quality and the quantity of the user's reviews. If a user does not have many reviews for a specific category of businesses, then the model will not have any past data to base the recommendations off of.