**Capstone project 2 milestone report**

This project will use Yelp review data and there are three major goals I am trying to accomplish. First, I will do a clustering of all the comments about good restaurants (either 4 or 5-star ratings). This will find interesting topics when a restaurant is considered good. Second, I will build a classifier to predict star rating of a restaurant given a specific review text. As raw text can contain rich information to extract, we will be able to see whether a user likes a restaurant or not. Third, I will build a recommender system to make restaurant recommendations to users. Recommender systems are very powerful and useful these days. They will sure come in handy when people try to pick a place to eat.

The client of this analysis will include first, end users who use Yelp to either review or find helpful reviews. The clustering of comments/reviews will tell them what are the aspects that are more often associated with a good restaurant. The recommender system will come in handy when they are searching for a good eatery. I will also create an adjusted rating system which can be used by both Yelp website and their users.

The Yelp data set is available for download from Yelp.com. The zipped file contains five different json files: 1. Business contains all information about the business such as name, address, city, state, and location. Most important features in this file are: review count (total number of reviews the business has received so far), attributes (such as parking lot, catering etc.), categories (such as Mexican, Burgers etc.), and hours (operation hours). 2. Review contains the star rating of each business, the date, and most importantly, the text of each review. This will be my major feature to process. 3. User contains information about all the users such as name, number of reviews made, and their friends on Yelp. 4. Tip contains shorter comments by users serving as quick suggestions. 5. Photos contain metadata about pictures which I will not use for this project.

Besides the json files I can download from Yelp website, I won’t need other data sets for this project. This is a well-known data set and the information in it is very rich. But is also has some limitations. First,

some attributes are missing and there is no way of imputing them because they are True/False binaries. For example, whether the business has wifi or has a parking lot. Second, most businesses rated are from two states and other states are underrepresented. Third, the review text is raw so there may be misspelled words.

I first read in the review json file using pandas as data frames. To process the raw text, I used NLTK library. The json file was too large to read in as a whole so I had to process by chunks. Once all the records have been imported, I examined the data. The star rating column had mixed data types, so I converted them into integers first. I then tokenized the text column from the 4 or 5-star ratings. A wordcloud clearly shows that the most frequent words associate with good ratings are words like: Amazing, best, great, good, love etc.

To get more meaningful tokens, I used a stemmer to reduce similar words into the same form. This is necessary because many words can mean the same thing but can be present in different forms. For example, like and liked would mean the same thing but be tokenized into 2 different tokens. The stemming works by trimming words to a base form thus reducing the number of repeating tokens/features. It’s essential in natural language processing and feature engineering. After getting all the tokens, I transformed them into a Tfidf (term frequency inverse document frequency) matrix for machine learning models. The Tfidf model reduces the weight of words that appear in most documents which usually are not very important words such as we, you etc. It’s common practice to use the Tfidf transformation when generating features from texts.

After the Tfidf matrix was generated, I tried two different machine learning algorithms to try to do the clustering: KMeans and NMF. Both models require a parameter which is the number of centers for KMeans and the number of components for NMF. KMeans would calculate the distances between all samples to the different centers while NMF would decompose the original matrix into two matrices: weight matrix and feature matrix, to find latent factors.

After fitting the data using KMeans and NMF models. I was able to generate 5 different topics (note this is arbitrary since I specified the number of centers/components to be 5 for these two algorithms) from all the comments. The two algorithms gave similar results. The top 5 topics for good restaurants are:

Cluster1: Good chicken food  
Cluster 2: Great service, good staff  
Cluster 3: Great pizza place  
Cluster 4: Best coffee shop  
Cluster 5: Good cheese burger

There may be more topics in all the reviews but that could be something to be completed later on.

During the process of manipulating the data, I tried to visualize and find important features other than text for my star rating prediction models. Many attributes for the restaurants could serve as additional features for my classifier such as properties of the restaurant, year of the review, and operation hours. More details will be in my final report document.