Chef's Special

For my last project at Metis I wanted to explore recommender systems. There are a few popular movie and book datasets to get started with, however due to my interest in all things food related I chose one from Yelp. It was also a fitting option since many restaurants have text reviews. In the past I have used natural language processing to analyze product reviews from Amazon and I wanted to bring it into the current project as well.

I subset my data to include only restaurants with more than 20 reviews and users who have left 15 or more ratings. This ensured that my models had enough data to learn from. I then tried a number of different approaches to modeling my data.

I started by building a collaborative filtering (CF) system, a method to predict a rating for a user-item pair based on the history of ratings given by the user and given to the item.

My baseline guess as to what score a user would give a restaurant was compiled with the following formula:

$$\textit{baseline guess for user } m \textit{ on restauran } n = \mu + m\mu + n\mu$$

$$\textit{where } \mu \textit{ is the average rating for all of restaurants by all users}$$

$$m\mu \textit{ is the difference between } \mu \textit{ and user } m\textit{'s average rating}$$

$$n\mu \textit{ is the difference between } \mu \textit{ and the restaurant } m\textit{'s average rating}$$

I also computed cosine similarities between both users and restaurants and ran a few different KNN models, with my best results for predicting restaurant ratings by a user coming from k = 5.

Many collaborative filtering algorithms are build on top of user-item rating matrices where each row represents a user, each column an item and the entries of such a matrix are the ratings given by users to items. This results in a very wide and sparse matrix which is memory intensive when it comes to computing. Ultimately, the model I choose the work deeper with was Singular Value Decomposition (SVD). SVD is a matrix factorization technique which is useful when trying to reduce the number of features of a dataset by reducing space dimensionality. It is therefore great for working with the kind of matrices that arise in recommenders.

For the natural language processing portion of modeling, I used latent Dirichlet allocation (LDA) for topic modeling. LDA is a generative statistical model useful for discovering the abstract “topics” that occur in a collection of documents. In my case, the documents were the restaurant reviews and tips users write and I decided to extra 50 different topics. The topics LDA picked up on ranged from explicit foods that could be used to categorize the restaurant to adjectives that referred to the atmosphere of the place.

By combining the two I hope to make it possible to generate results from ambiguous queries. Moving forward, I would also like to add in a way for a new user to rate popular restaurants in their area so that the model could also start making recommendations for them.
The final step will be to build out a UI for interacting with the model.