Capstone 2 FinalReport

Daniel Weissberger

Introduction

Being able to recommend a restaurant to a user could be a very useful tool for Yelp. It could help them tailor advertisements to users and could improve their search algorithm. The more accurate restaurant recommendations become the more likely a user will be to return for another search. The goal is to utilize the dataset to develop a recommendation system for yelp users.

Data Wrangling

A subset of the data will be selected for the initial phases of the project. This will suffice since the goal is to design the overall pipeline and fine tune the processes involved before testing it on the entire dataset. Due to this a subset of the data was loaded for preliminary EDA and other analyses. The following data files were provided in csv format:

File Name	File Size
yelp_business.csv	31,017 KB
yelp_business_attributes.csv	40,408 KB
yelp_business_hours.csv	13,542 KB
yelp_checkin.csv	4,935 KB
yelp_review.csv	3.7 GB
yelp_tip.csv	144,614 KB
yelp_user.csv	1.3 GB

Data Selection

Initially a network analysis will be performed which will focus on the business, business attributes, review and user tables only.

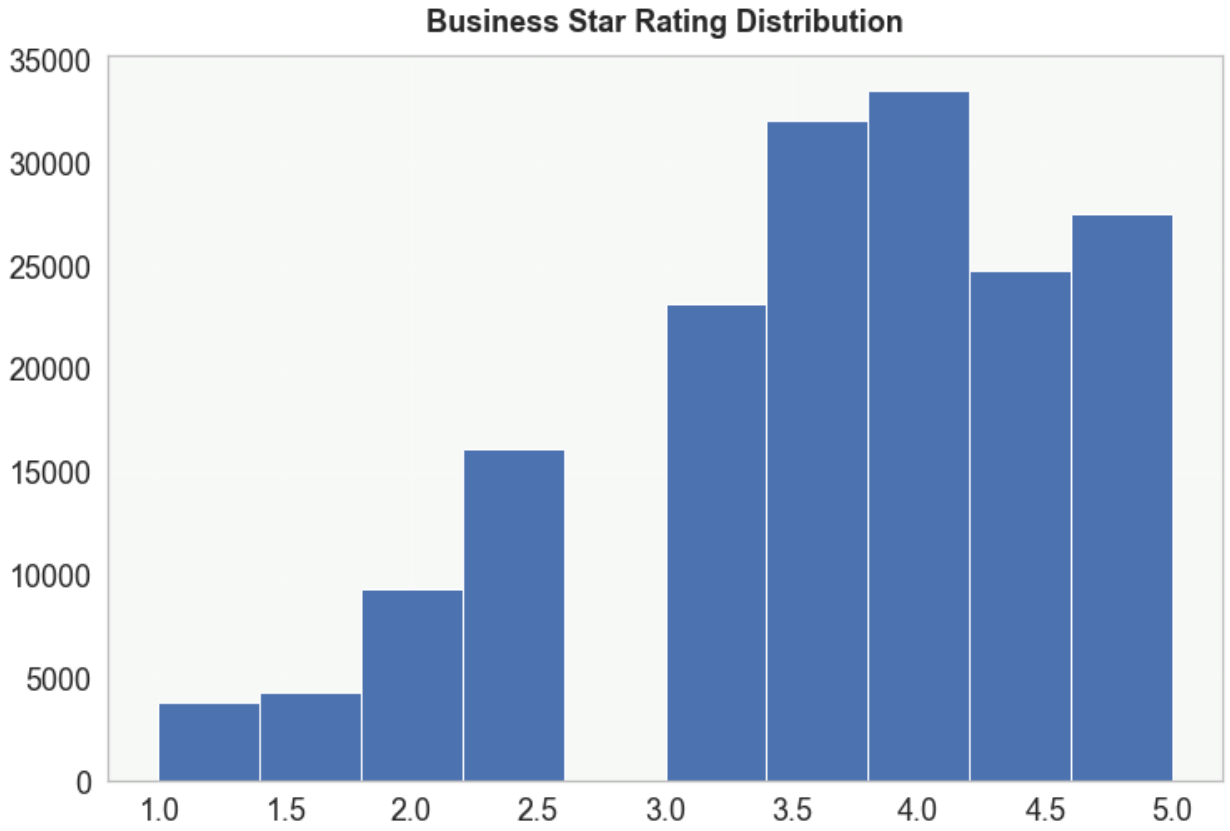
To deal with the large amount of data initially, I will select a random subset of users. From this random subset of users (about 10%) I will retrieve only the reviews that these users have

submitted. Since the business and business attribute files are small in size, these will be loaded entirely.

An alternative approach which will also be used is to focus on a particular city choosing all restaurants located in that city. From these restaurants aggregate their reviews and finally identify the users associated to those reviews. This will allow for a zonal analysis which in the initial phase of development will likely be ideal, since recommended restaurants should be located a reasonable distance from the user.

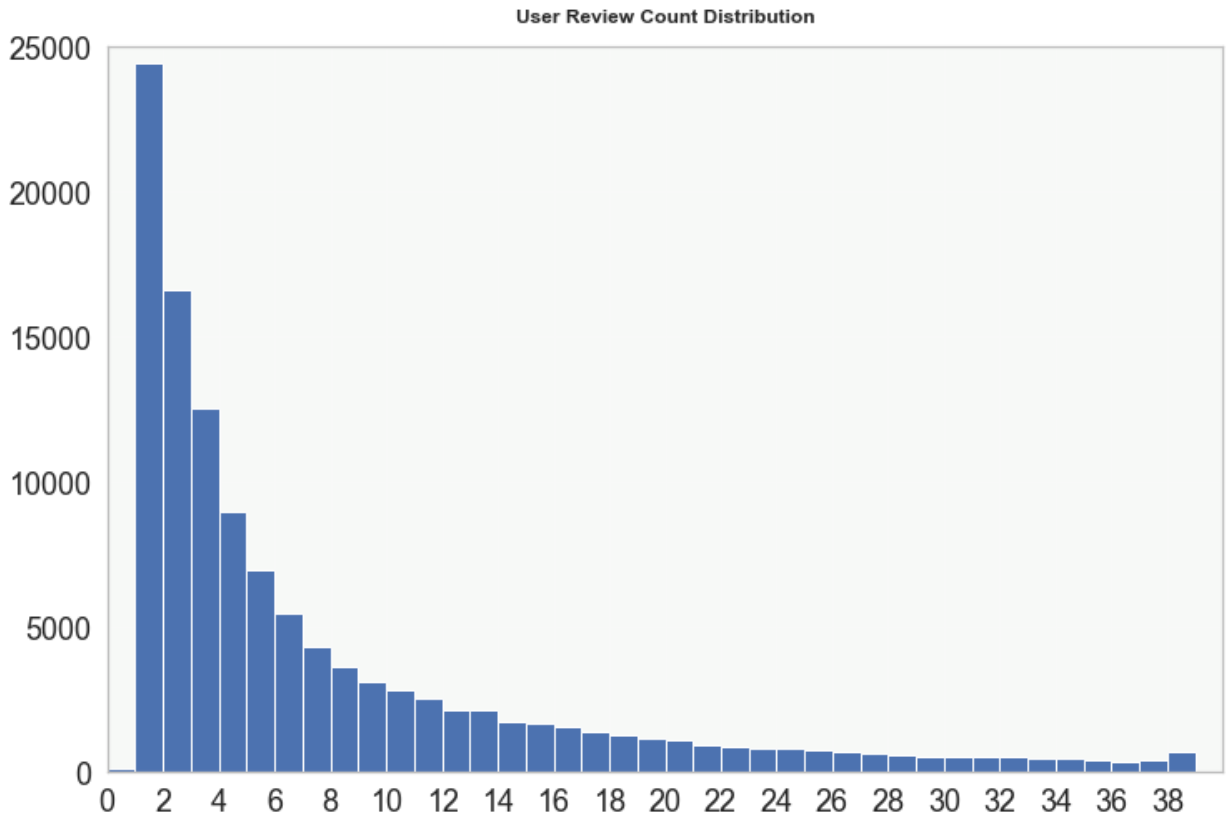
EDA

Using the subset of selected users we perform some exploratory data analysis. First let's take a look at the star rating distribution (for all businesses in the dataset):



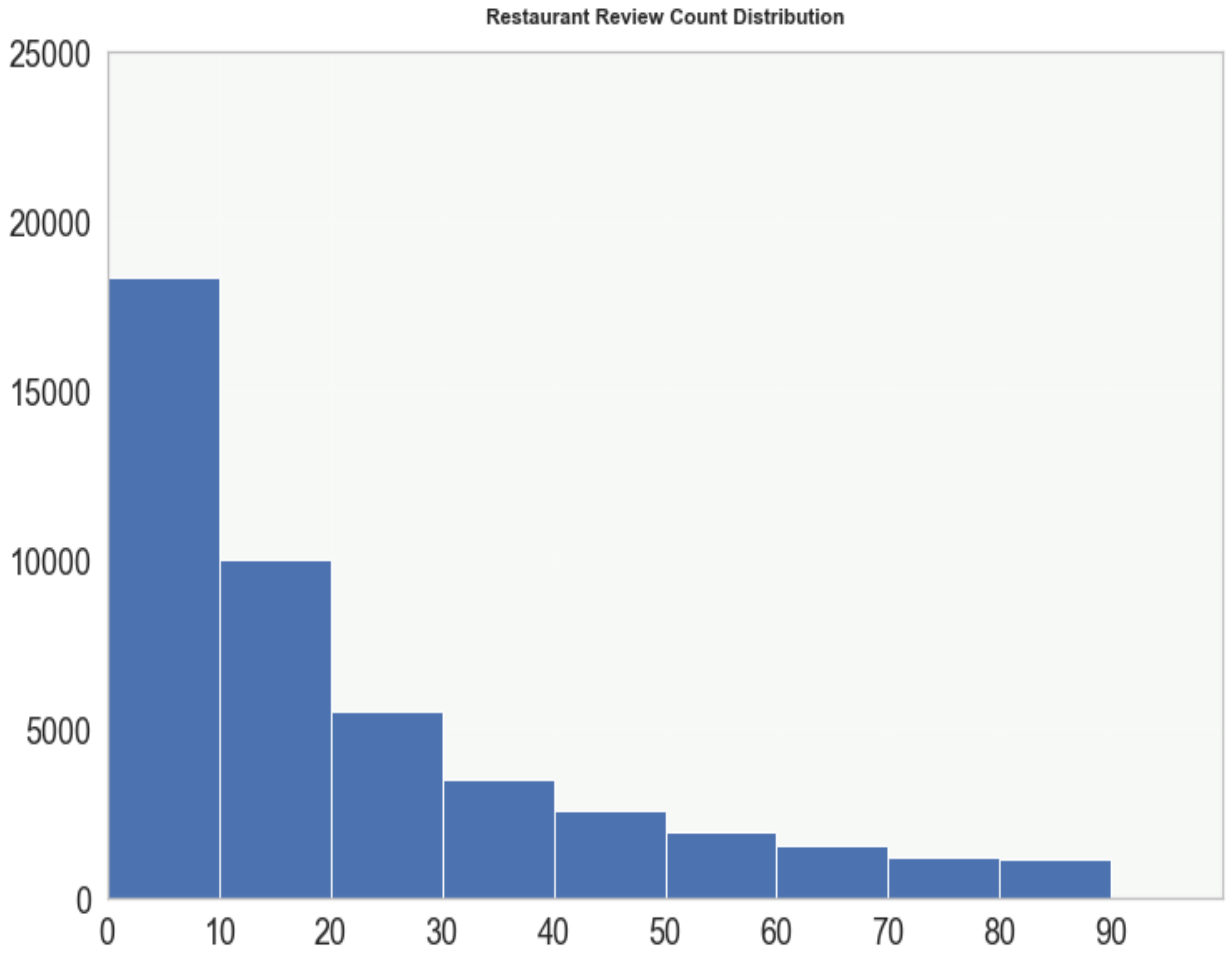
Note the interesting gap there are actually no businesses reviewed between 2.6 and 3.0 stars in the dataset. Quite an interesting result which shows the polarity of the data. Also interesting to note that a majority of star ratings lie within the 3-4 star range.

How about review count per user? Let's examine the distribution:



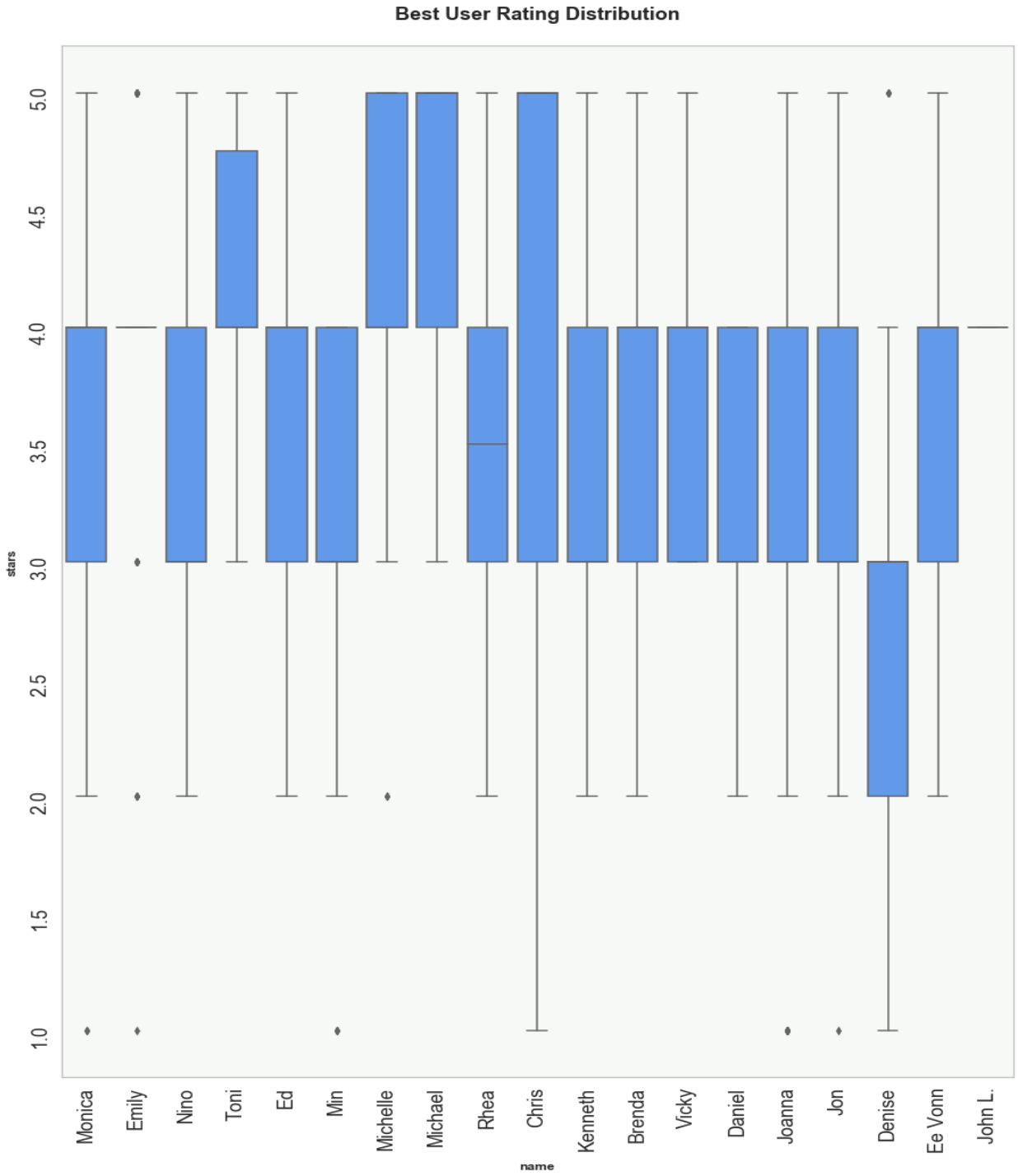
Clearly this is a highly skewed distribution with most users only contributing 1-6 reviews at most. This will make things more challenging to suggest new restaurants to users since for many it will be a cold start.

How about the restaurant review count distribution? Let's examine if the same skew is observed with restaurants and their reviews:



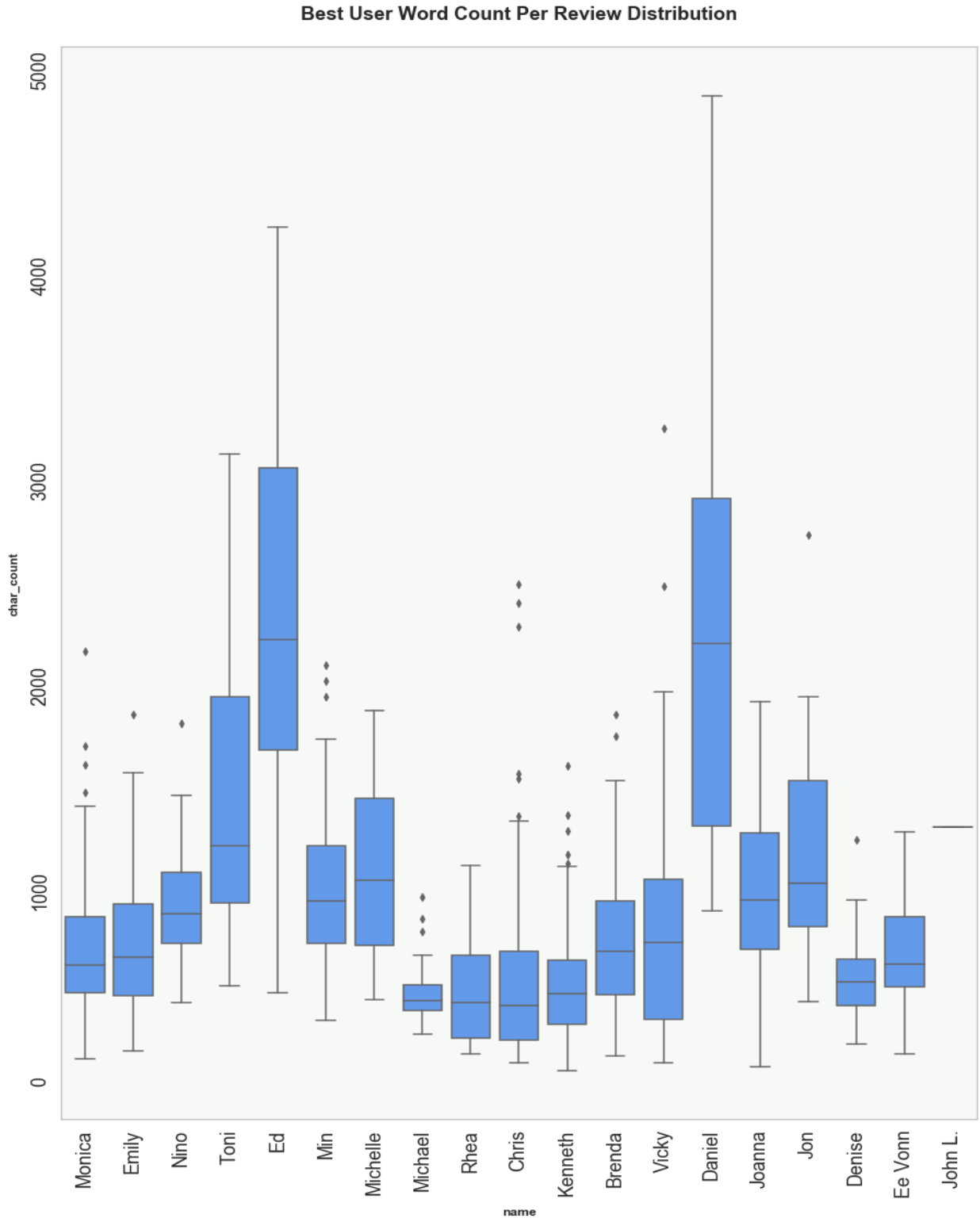
We observe similarly skewed data indicating that a large number of restaurants only have 0-30 reviews.

How much do the reviews differ per user? Let's examine this by taking the 10 most active users and comparing their restaurant star rating distribution:



This is a very interesting result. Firstly we can observe that for most users their star rating distribution seems to lie within a range of 1 star. Additionally almost all users seem to lie within the 3 to 4 star range which matches the results we observed from the business rating distribution.

How wordy are the best users we selected from the dataset? Let's examine the word count distribution for the reviews of the selected users:

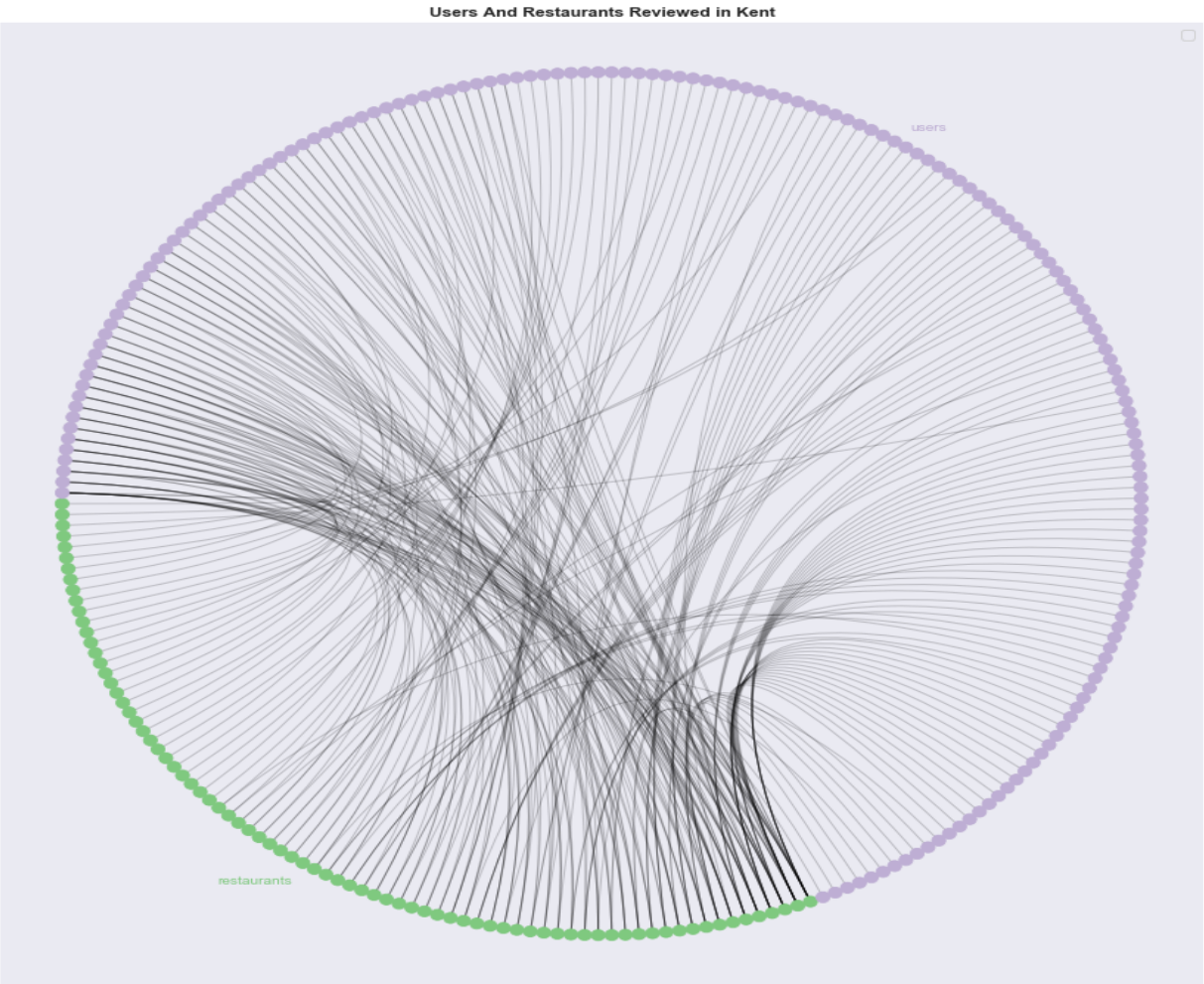


Although our users seem to agree when it comes to rating, they seem very unique in terms of how they convey their restaurant opinions. Analyzing the content of these reviews may aid in profiling the similarity of these users.

Network Analysis

For the baseline restaurant recommendation system a network graph analysis will be conducted. In this analysis users and restaurants will be treated as nodes in a bipartite graph with restaurant reviews as the edges in the graph. For the initial analysis the city of Kent Ohio was selected since it contains 111 restaurants, a number that will be manageable for visual representations.

First we selected the subset of restaurants from the data, extract the users and reviews from those restaurants, and construct a networkx graph linking restaurants and users. The following is a CircosPlot of the network:



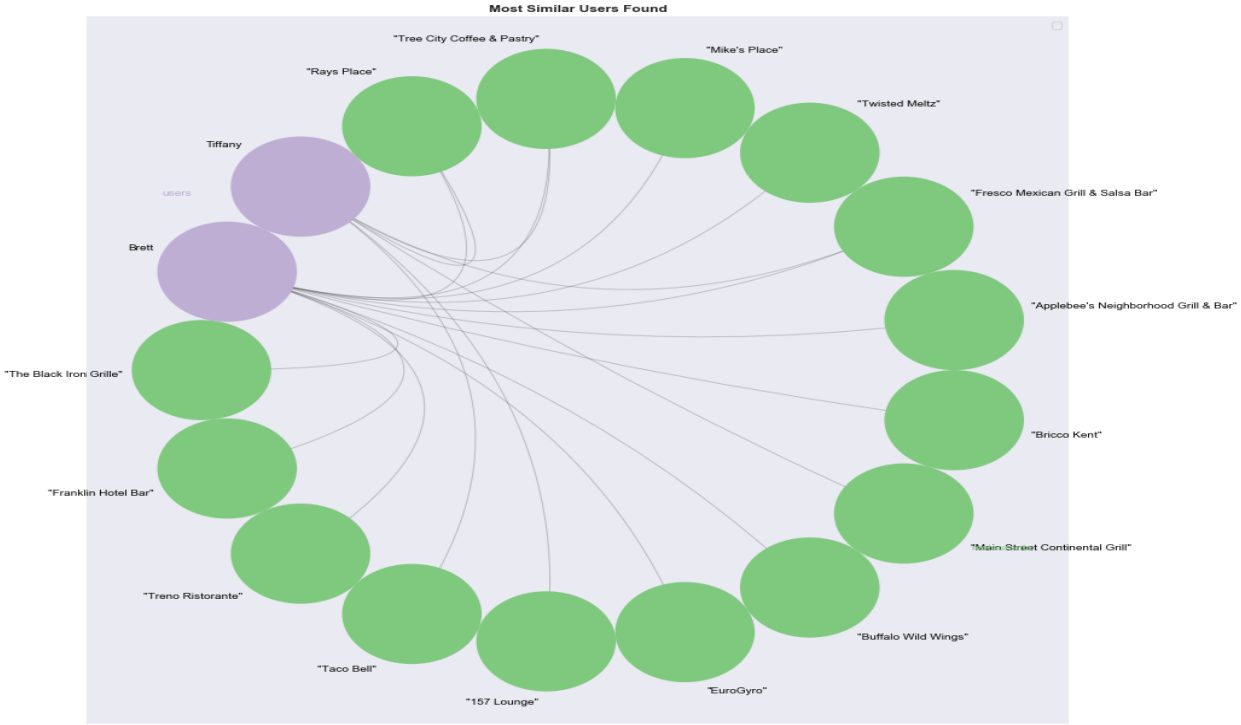
User Similarity

For the baseline recommendation system, user similarity will be analyzed by examining relationships in the network. A user similarity score will be computed as follows:

1) For a given pair of users (user1, user2)
2) Identify the restaurants that user1 and user2 have reviewed
3) Count the number of reviews where the general sentiment was shared (at this stage this is either positive or negative if the review is greater than or less than 3 respectively)
4) Take this count and divide by all other nodes that the two users could potentially share (in this case all other nodes in the restaurant partition)

Using this user similarity score, recommendations can be made to users. For instance if the score is above a given threshold we can look at recommending restaurants to user 1 that user 2 has visited and vice versa.

To test the similarity score, we compute the similarities of all users against each other in the graph of Kent restaurants. We then take the users who shared the highest similarity score. Finally we can plot the result in a CircosPlot showing the two most similar users and their node connections:



We see that there are some shared connections and likely they shared the same feeling about those restaurants. For the future the following enhancements can be made to user similarity:

1) Sentiment analysis can be conducted on reviews to determine if two users really have shared feelings about a restaurant
2) User checkins can be examined to gain more information about user patterns
3) A clustering approach can be taken, in which user properties are vectorized and users grouped into clusters of similar users.

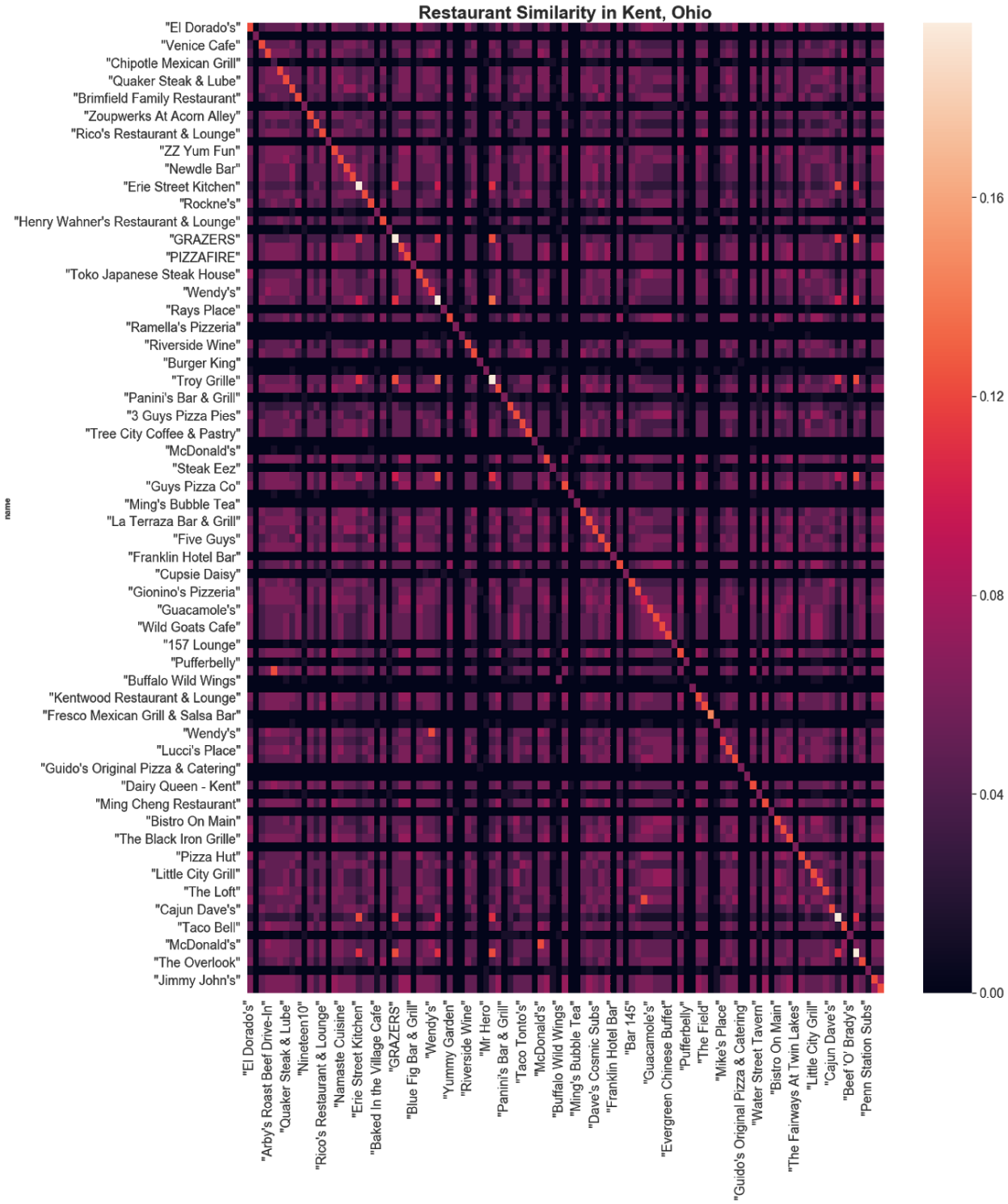
Restaurant Similarity

Restaurant similarity is the concept of taking two given restaurants and establishing a similarity score for those restaurants based on a number of given features. Inputs to this score include the business attributes as well as the reviews of a business, business hours.. Etc.

This score could be very useful since every user has different tastes, yet a user's taste profile could be categorized by examining the restaurants she has visited and enjoyed. Therefore the score could be used to qualify a set of potential restaurant recommendations answering the question, do these recommendations fall within the taste profile of the user? Possible approaches for calculating restaurant similarity:

1) A simple comparison of restaurant features and categories (baseline implementation)
2) Clustering approach in which restaurant information is vectorized and restaurants are grouped into N clusters. If I know that a user dines in only clusters 1,2,3 i will make sure to recommend restaurants that fall within those clusters.
3) Regression can be applied to predict the rating a user will give to a restaurant given their historic preferences. Recommendations can be validated and selected via this regression algorithm's predictions.

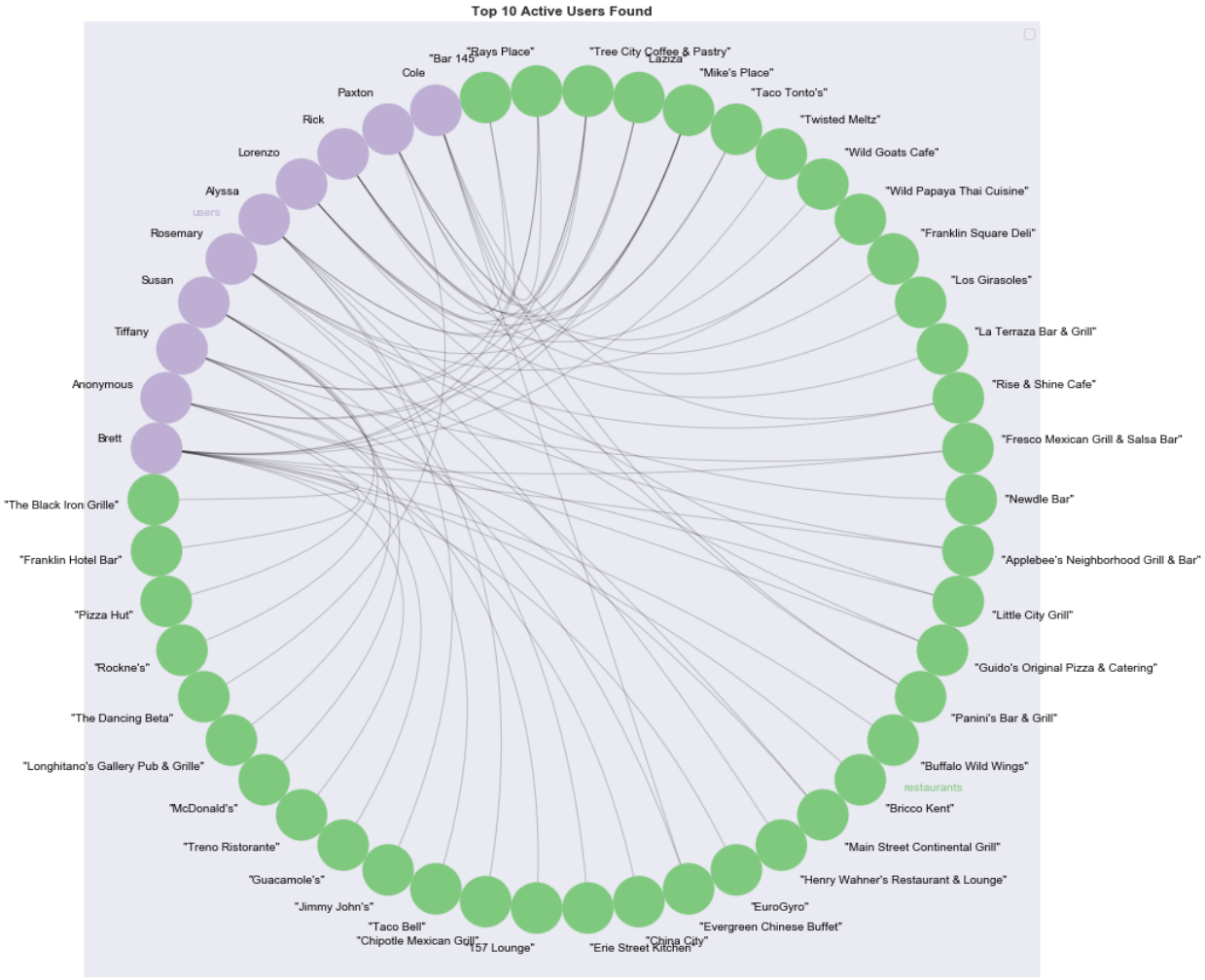
For the baseline restaurant similarity a simple approach will be taken in which all restaurant attributes will be compared. The score will be calculated by taking the number of matching features divided by the total number of features (note this is flawed and will be improved upon in the future). For the initial analysis of Kent, Ohio the following visual representation of restaurant similarity was created using the baseline approach:



Degree Centrality

Degree centrality is a measure of the connectedness of a particular node in the network. As a result it can aid in identifying the most active users in a network. These users may be quite

interesting to examine since they will likely be the ‘trend-setters’ of the network visiting new restaurants first to ensure their review is heard. It may be interesting to include recommendations by examining the behavior of some of the most active users. As a preliminary analysis degree centrality is calculated for each node in the Kent, Ohio network. Finally the 10 users with the highest degree centrality are shown in a CircosPlot along with their neighboring nodes/edges:



It’s clear that lots of activity is going on amongst the active users even in this relatively small network.

Recommendation System

The results of the network analysis show that even in a relatively small city, there is a great degree of overlap between users and the restaurants they visit. A recommendation system could leverage this data to make restaurant recommendations to users. Several approaches to recommendation systems can be applied. Collaborative filtering, content filtering and matrix factorization are three common approaches which will be applied and evaluated on the yelp dataset.

Content Filtering

Content filtering also known as cognitive filtering leverages the knowledge of the items a user has liked and the features or “content” of those items. Essentially computing an item similarity score for all items in the network. In this case the items are restaurants and such a system would typically suggest the top n items or restaurants with the highest computed similarity score. Results for the content-based filtering system are below:

Accuracy	
Precision	
Recall	

Collaborative Filtering

Collaborative filtering is another approach to a recommendation system which uses the similarity of users in a network to recommend restaurants to a given user. Similar to content filtering, in this approach a similarity score is computed for each user based on their taste profile (in this case which restaurants the user has liked/disliked). This similarity score is then used to generated a weighted average of all user scores for a given item (weighted by the similarity score itself).

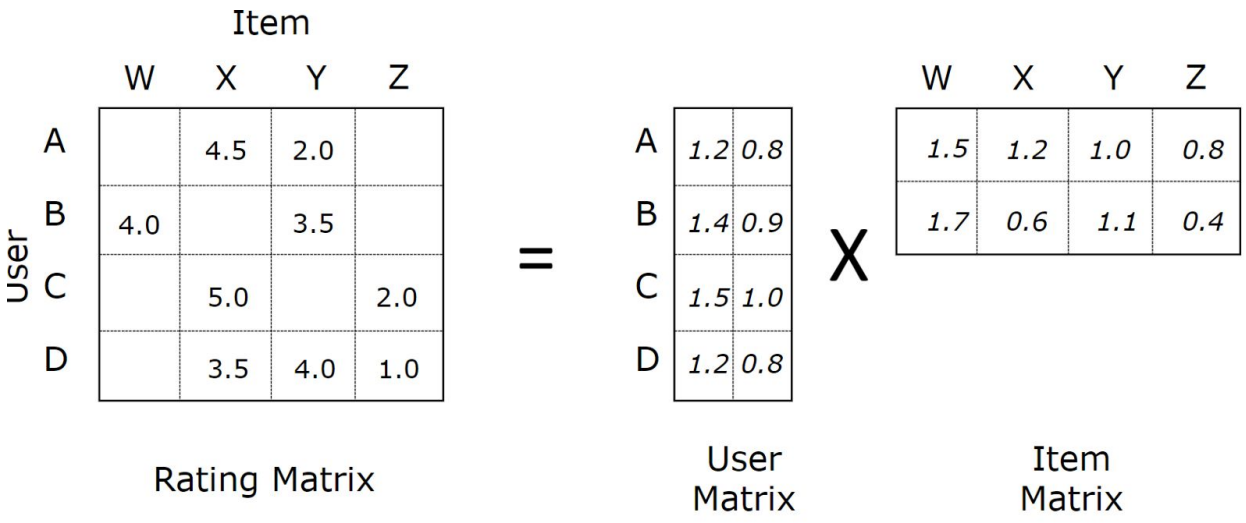
Collaborative filtering significantly outperformed content filtering. Perhaps an indication that the content filtering system needs some modification. Results for the collaborative filtering approach are below:

Accuracy	.526
Precision	.742
Recall	.546

Results indicate a low accuracy and high precision which indicates that the model is underfitting the data. Many incorrect predictions are driving a lower accuracy score while superficially boosting the precision score. Perhaps this model could be optimized further to improve these scores, yet for now another type of recommendation system will be investigated.

Matrix Factorization

One final approach is matrix factorization which uses matrix decomposition to generate a User-Rating Matrix (prediction matrix). Below is an image depicting this approach:



The left most matrix is the matrix we would like to predict which is in this case a matrix of n Users and m Items. The idea here is that the missing values of this matrix can be computed by the dot product of 2 matrices; A user-feature matrix of dimension MxK and a feature-item matrix of dimension NxK. At this point you may ask, what is K? What are these features? These

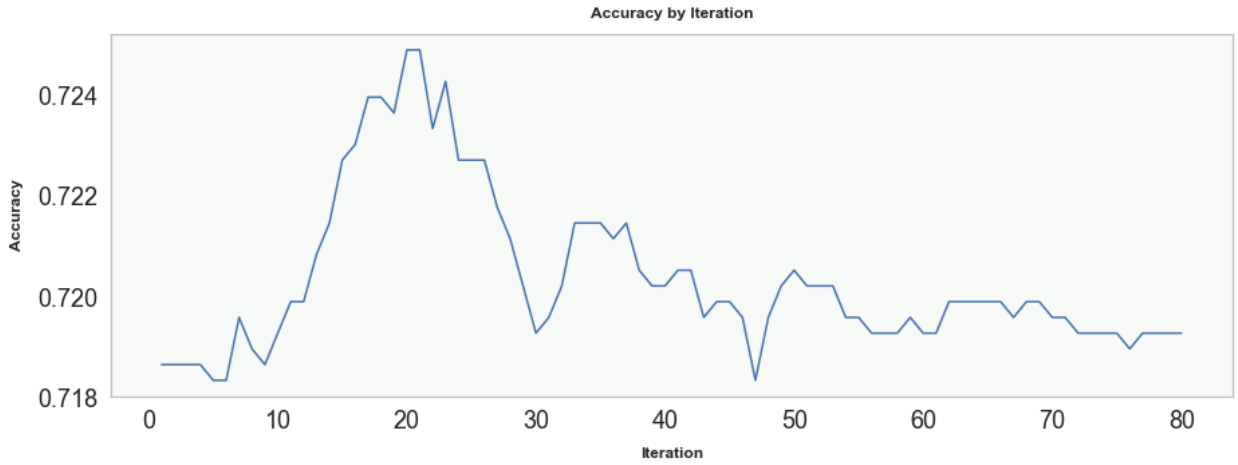
features in matrix factorization are called ‘latent’ features since they are hidden and can’t be found by directly analyzing the data. For instance two users may like the same movie because it fits a genre they are into or has actors which they both like. These type of features are not captured in typical collaborative filtering or content filtering models.

The process of matrix factorization essentially becomes a guessing game where the two ‘feature’ matrices are filled in iteratively. On each iteration, predictions are evaluated against known values (in this case restaurant star ratings). This evaluation then feeds back into the next iteration with the goal to minimize error on prediction of known User-Item matrix values. For this implementation of Matrix Factorization, SGD or stochastic gradient descent will be used to implement the error-minimization algorithm.

The results for Matrix Factorization predictions on yelp data are below:

Accuracy	.719
Precision	.735
Recall	.952

Results show a significant improvement from previous systems with accuracy and precision in the low 70 percentile. However its clear from the recall score that very few false negatives were flagged which indicates that the model is likely overfitting. In other words the model is making too many false positive predictions. Perhaps a hybrid model could help with these numbers or another approach could be to use a different minimization function to make predictions. Below is a plot showing the accuracy between iterations of the matrix factorization function:



Clearly the accuracy peaks at around 20 iterations, yet the peak is not much higher than the settling point.

Conclusion

Results indicate some successes in predicting restaurants for users. This could be a great utility for Yelp to help drive business by leveraging this knowledge. With accuracy and precision in the 71-72 percent range, these predictions are fairly reliable, yet this could most likely be fine tuned further in the future. Additionally methods such as factorization machines could be applied to add in other feature/user data which could also drive an improvement in the recommendation system.

Introduction

Being able to recommend a restaurant to a user could be a very useful tool for Yelp. It could help them tailor advertisements to users and could improve their search algorithm. The more accurate restaurant recommendations become the more likely a user will be to return for another search. The goal is to utilize the dataset to develop a recommendation system for yelp users.

Data Wrangling

A subset of the data will be selected for the initial phases of the project. This will suffice since the goal is to design the overall pipeline and fine tune the processes involved before testing it on the entire dataset. Due to this a subset of the data was loaded for preliminary EDA and other analyses. The following data files were provided in csv format:

File Name	File Size
yelp_business.csv	31,017 KB
yelp_business_attributes.csv	40,408 KB
yelp_business_hours.csv	13,542 KB
yelp_checkin.csv	4,935 KB
yelp_review.csv	3.7 GB

yelp_tip.csv	144,614 KB
yelp_user.csv	1.3 GB

Data Selection

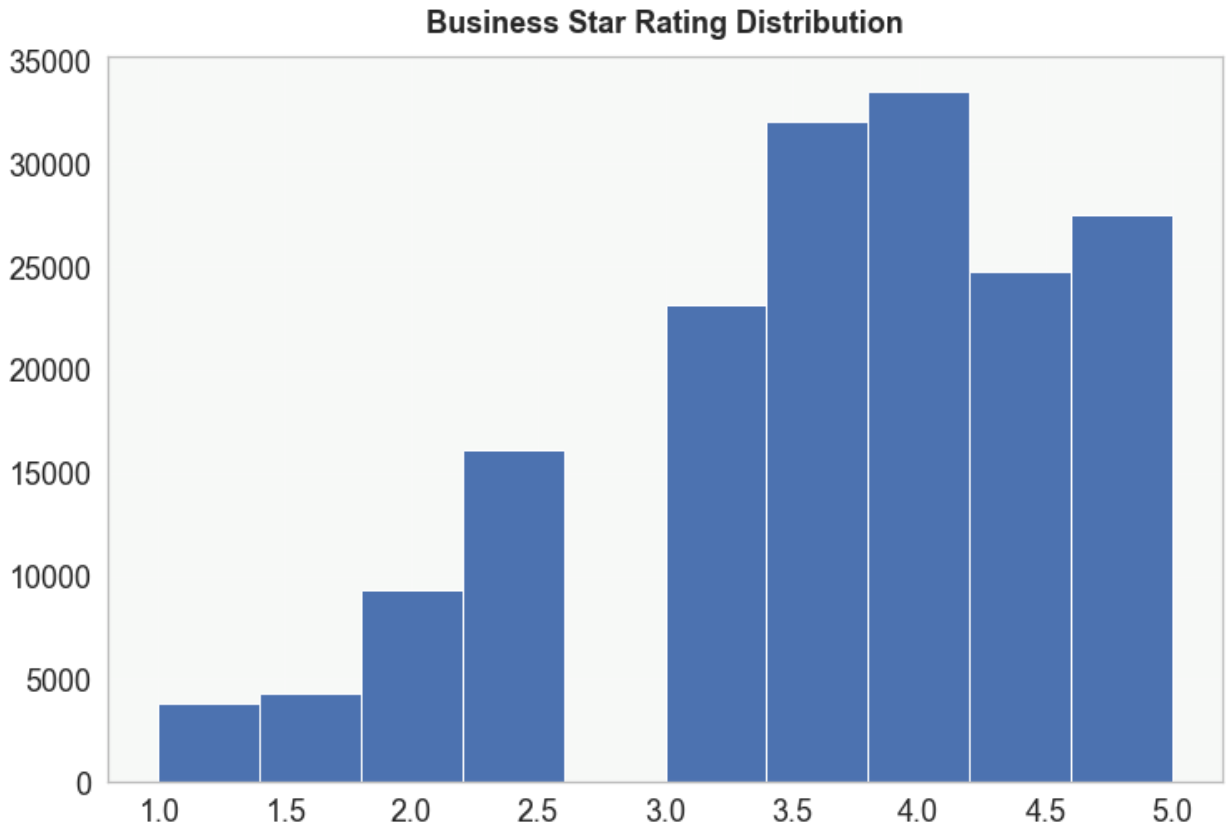
Initially a network analysis will be performed which will focus on the business, business attributes, review and user tables only.

To deal with the large amount of data initially, I will select a random subset of users. From this random subset of users (about 10%) I will retrieve only the reviews that these users have submitted. Since the business and business attribute files are small in size, these will be loaded entirely.

An alternative approach which will also be used is to focus on a particular city choosing all restaurants located in that city. From these restaurants aggregate their reviews and finally identify the users associated to those reviews. This will allow for a zonal analysis which in the initial phase of development will likely be ideal, since recommended restaurants should be located a reasonable distance from the user.

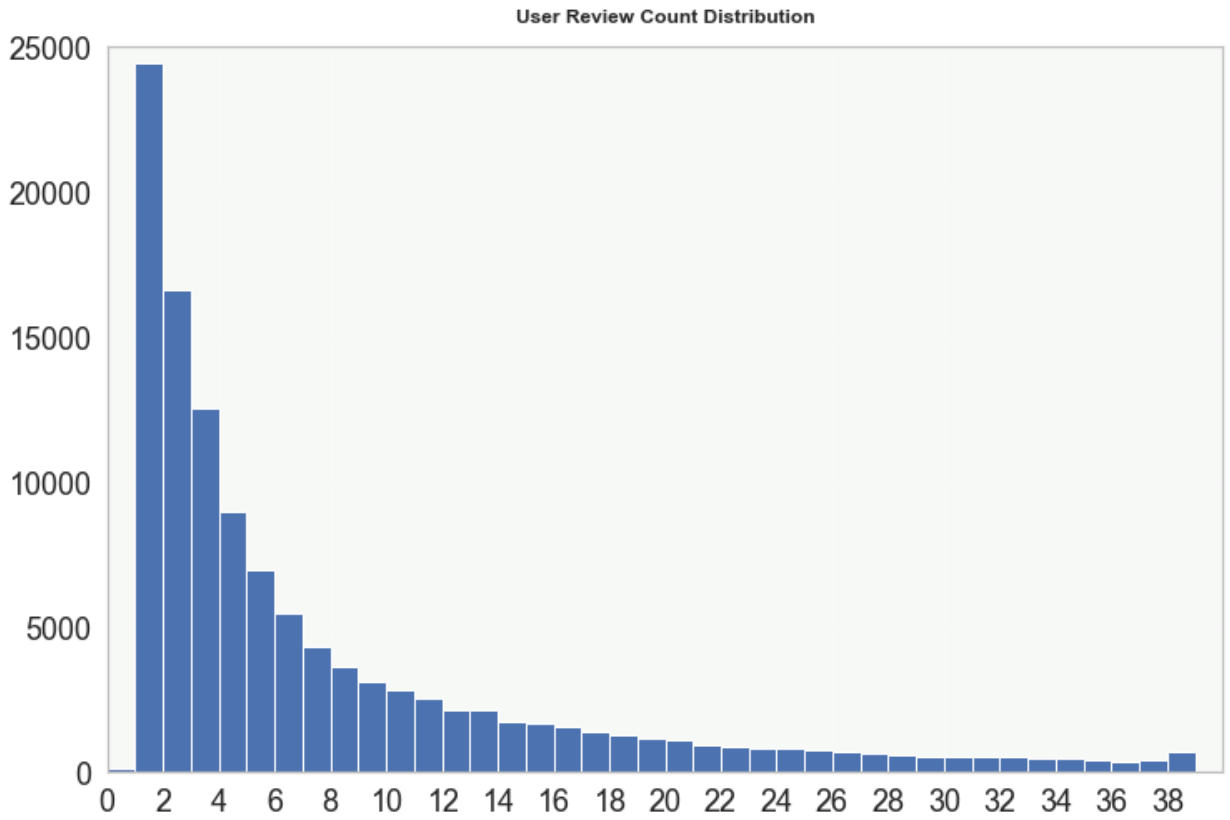
EDA

Using the subset of selected users we perform some exploratory data analysis. First let’s take a look at the star rating distribution (for all businesses in the dataset):



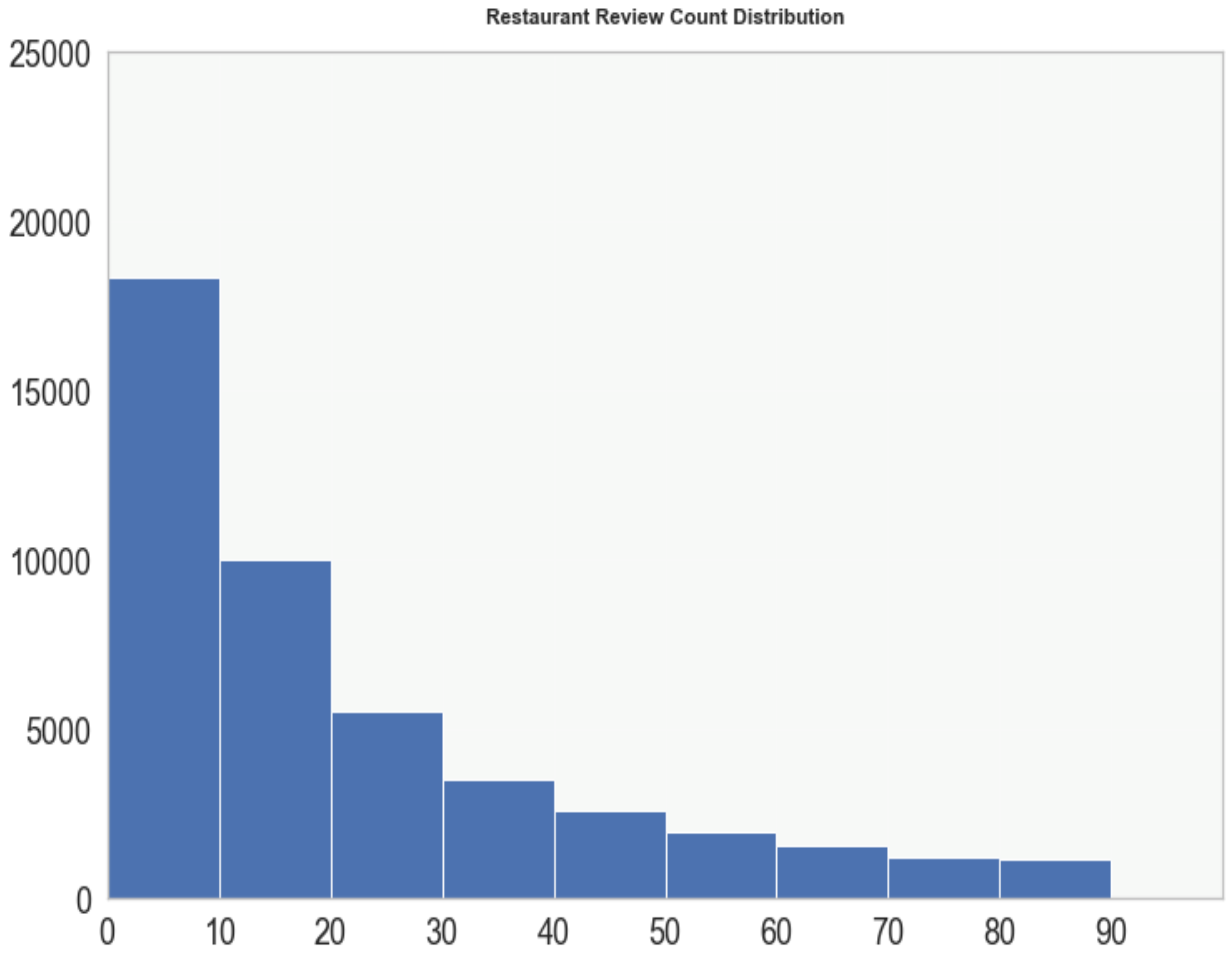
Note the interesting gap there are actually no businesses reviewed between 2.6 and 3.0 stars in the dataset. Quite an interesting result which shows the polarity of the data. Also interesting to note that a majority of star ratings lie within the 3-4 star range.

How about review count per user? Let's examine the distribution:



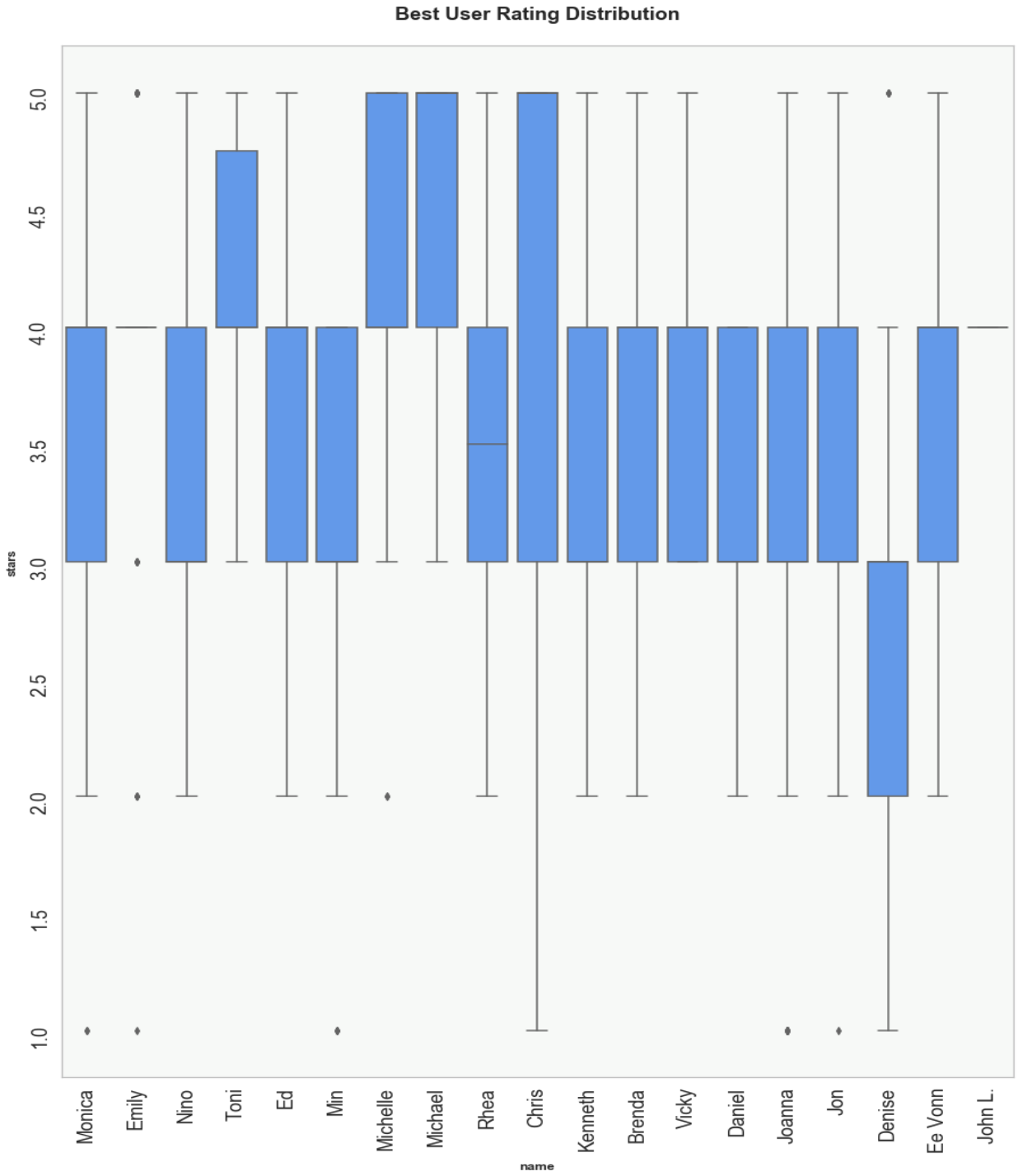
Clearly this is a highly skewed distribution with most users only contributing 1-6 reviews at most. This will make things more challenging to suggest new restaurants to users since for many it will be a cold start.

How about the restaurant review count distribution? Let's examine if the same skew is observed with restaurants and their reviews:



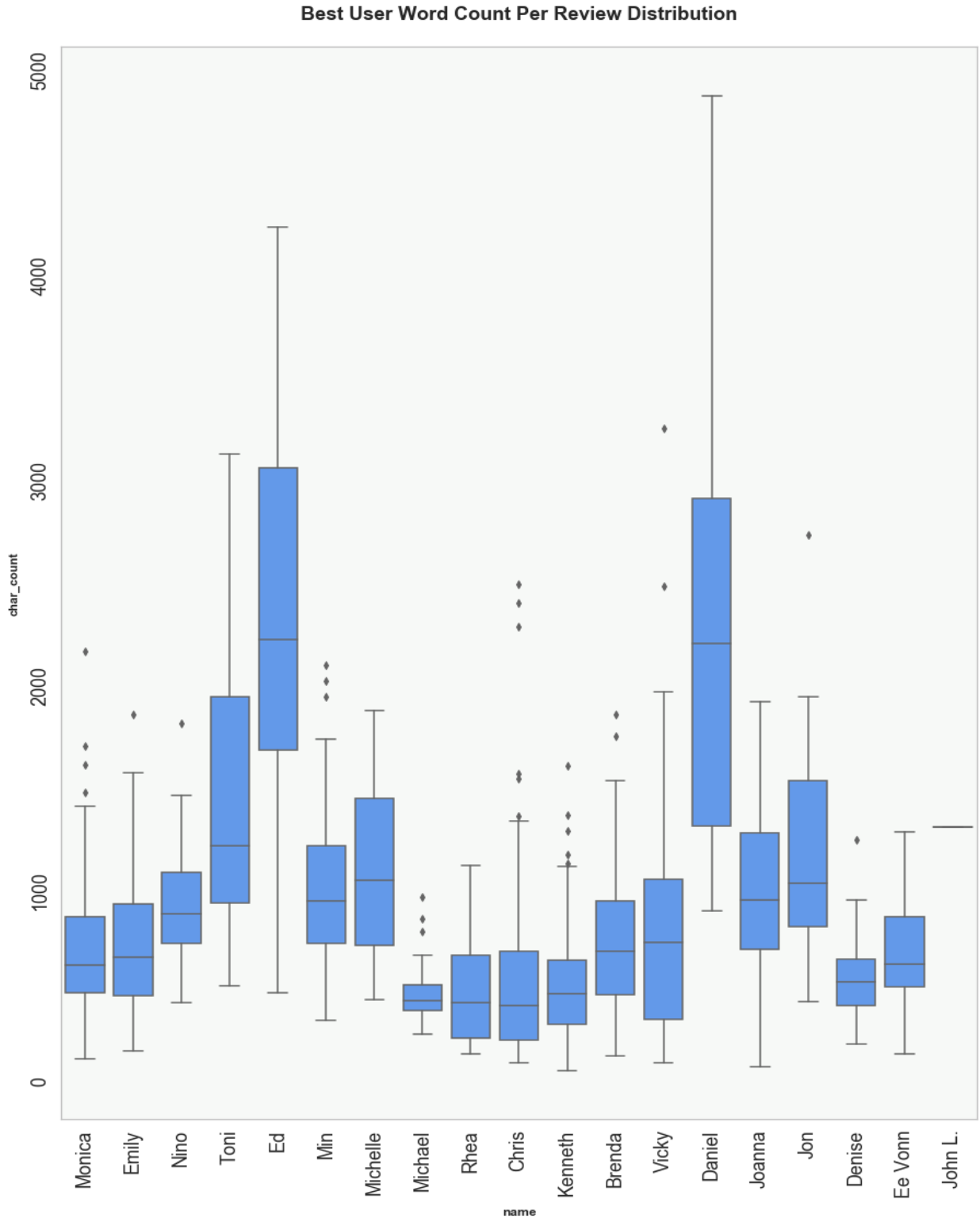
We observe similarly skewed data indicating that a large number of restaurants only have 0-30 reviews.

How much do the reviews differ per user? Let's examine this by taking the 10 most active users and comparing their restaurant star rating distribution:



This is a very interesting result. Firstly we can observe that for most users their star rating distribution seems to lie within a range of 1 star. Additionally almost all users seem to lie within the 3 to 4 star range which matches the results we observed from the business rating distribution.

How wordy are the best users we selected from the dataset? Let's examine the word count distribution for the reviews of the selected users:



Although our users seem to agree when it comes to rating, they seem very unique in terms of how they convey their restaurant opinions. Analyzing the content of these reviews may aid in profiling the similarity of these users.

Recommendation System

The results of the network analysis show that even in a relatively small city, there is a great degree of overlap between users and the restaurants they visit. A recommendation system could leverage this data to make restaurant recommendations to users. Several approaches to recommendation systems can be applied. Collaborative filtering, content filtering and matrix factorization are three common approaches which will be applied and evaluated on the yelp dataset.

Content Filtering

Content filtering also known as cognitive filtering leverages the knowledge of the items a user has liked and the features or “content” of those items. Essentially computing an item similarity score for all items in the network. In this case the items are restaurants and such a system would typically suggest the top n items or restaurants with the highest computed similarity score.

Collaborative Filtering

Collaborative filtering is another approach to a recommendation system which uses the similarity of users in a network to recommend restaurants to a given user. Similar to content filtering, in this approach a similarity score is computed for each user based on their taste profile (in this case which restaurants the user has liked/disliked). This similarity score is then used to generated a weighted average of all user scores for a given item (weighted by the similarity score itself).

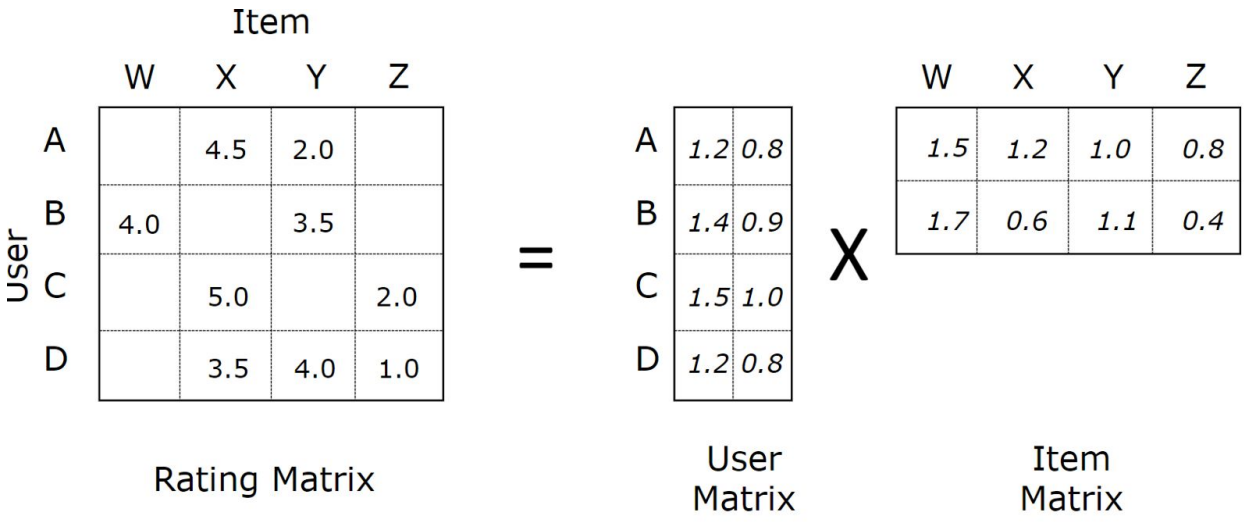
Collaborative filtering significantly outperformed content filtering. Perhaps an indication that the content filtering system needs some modification. Results for the collaborative filtering approach are below:

Accuracy	.526
Precision	.742
Recall	.546

Results indicate a low accuracy and high precision which indicates that the model is underfitting the data. Many incorrect predictions are driving a lower accuracy score while superficially boosting the precision score. Perhaps this model could be optimized further to improve these scores, yet for now another type of recommendation system will be investigated.

Matrix Factorization

One final approach is matrix factorization which uses matrix decomposition to generate a User-Rating Matrix (prediction matrix). Below is an image depicting this approach:



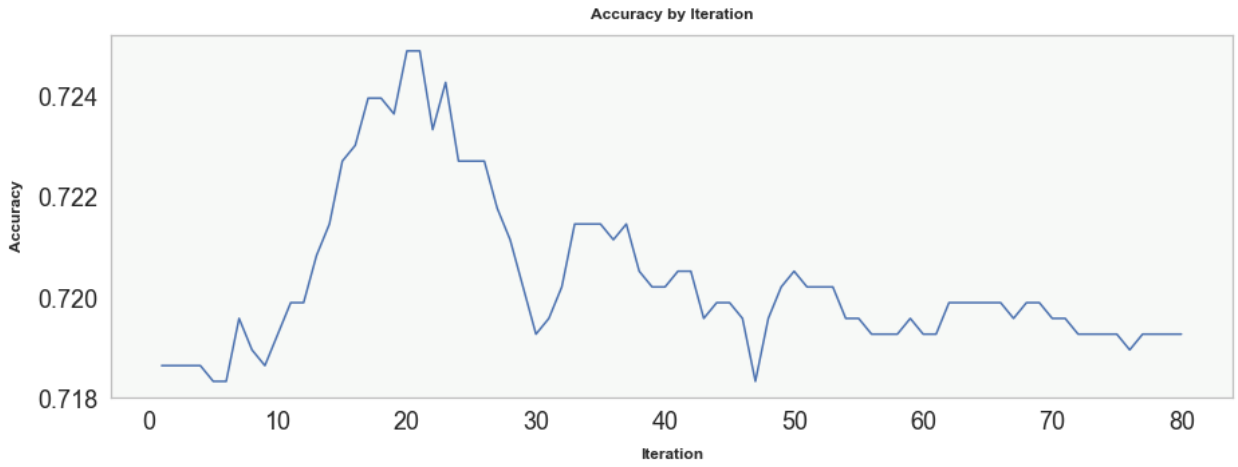
The left most matrix is the matrix we would like to predict which is in this case a matrix of n Users and m Items. The idea here is that the missing values of this matrix can be computed by the dot product of 2 matrices; A user-feature matrix of dimension MxK and a feature-item matrix of dimension NxK. At this point you may ask, what is K? What are these features? These features in matrix factorization are called ‘latent’ features since they are hidden and can’t be found by directly analyzing the data. For instance two users may like the same movie because it fits a genre they are into or has actors which they both like. These type of features are not captured in typical collaborative filtering or content filtering models.

The process of matrix factorization essentially becomes a guessing game where the two ‘feature’ matrices are filled in iteratively. On each iteration, predictions are evaluated against known values (in this case restaurant star ratings). This evaluation then feeds back into the next iteration with the goal to minimize error on prediction of known User-Item matrix values. For this implementation of Matrix Factorization, SGD or stochastic gradient descent will be used to implement the error-minimization algorithm.

The results for Matrix Factorization predictions on yelp data are below:

Accuracy	.719
Precision	.735
Recall	.952

Results show a significant improvement from previous systems with accuracy and precision in the low 70 percentile. However its clear from the recall score that very few false negatives were flagged which indicates that the model is likely overfitting. In other words the model is making too many false positive predictions. Perhaps a hybrid model could help with these numbers or another approach could be to use a different minimization function to make predictions. Below is a plot showing the accuracy between iterations of the matrix factorization function:



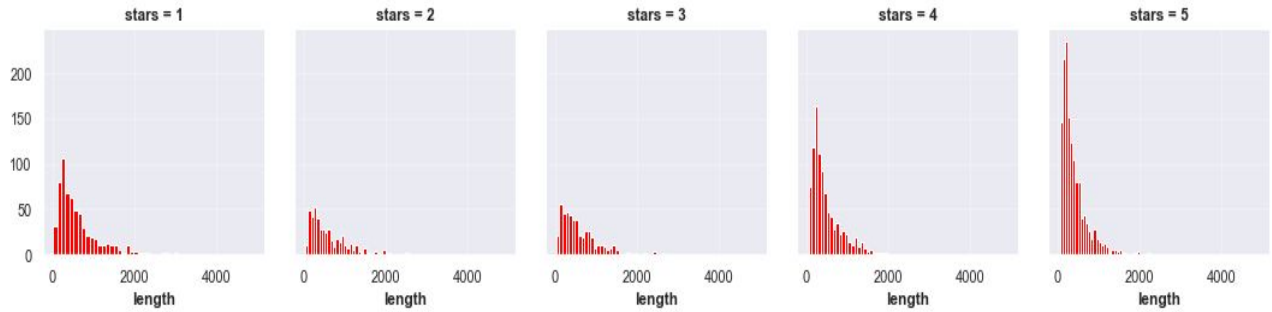
Clearly the accuracy peaks at around 20 iterations, yet the peak is not much higher than the settling point.

Sentiment Analysis

Included in the dataset are user reviews of restaurants which can also be very useful to yelp in understanding its users and there preferences. In an effort to gain more information about their users they may want to classify the reviews into different categories. As a proof of concept, the yelp review data in this project will be analyzed using a variety of NLP and machine learning techniques in order to make binary predictions on the user's sentiments.

Sentiment Analysis Data Wrangling

Before the data can be inserted into a machine learning classifier, several steps must be taken to clean and rearrange the data into a digestible form. The main reason behind this is that machine learning algorithms can't interpret words leading to the need for vectorization of the review data. Firstly let's take a look the the review word count distribution for each star category:



There seems to be an interesting trend on either extreme showing that when users really like or dislike a restaurant they tend to write lengthier reviews.

Data Wrangling

As mentioned earlier the data must be vectorized. For the purpose of this analysis we will focus on three categories: Negative, Neutral and Positive reviews. To do this, the dataset is filtered by stars on 1 star, 3 star and 5 star reviews only.

Following this, we remove punctuation from the reviews as well as stopwords: words which are generally low information and do not contribute to the analysis such as 'the', 'is' and 'are'.

Finally, we then use the **CountVectorizer** from the sklearn library to generate a vocabulary object which essentially creates a dictionary for all of the words in the review dataset. The vocabulary object has a transform object which will take in the cleaned reviews and map them to vectors each with the same length as the vocabulary dictionary and representative of the

word counts for each review. Below are the results of analysis for a variety of machine learning techniques (numbers show the weighted averages for precision recall and f1-score for 1, 3 and 5 star reviews respectively.

Multinomial Naive-Bayes		
Precision: .79	Recall: .77	F1-score: 514
Random Forest		
Precision: .79	Recall: .77	F1-score: 514
Decision Trees		
Precision: .65	Recall: .64	F1-score: 514
Gradient Boost		
Precision: .77	Recall: .75	F1-score: 514
Multilayer Perception Classifier (Best Performer)		
Precision: **.80**	Recall: **.79**	F1-score: 514

Making Predictions with the Multilayer Perception Classifier Model

Since the MLP model was by far the best performer, we will use this model to make predictions on some of the yelp review data. Below is an example of a negative, neutral and positive review each showing the actual and predicted ratings.

Negative Review - MPC

“Stopped in for cocktails and a appetizer at the bar and both were not good at all. Cocktails were so weak we had to order a double and it seemed to me the bartenders had no clue. Most likely won't be back in the near future.”

Actual Rating: 1
Predicted Rating: 1

Positive Review -MPC

“Came to Fourk for lunch today, I work right down the road and have driven by here as they have re-modeled it. A co-worker and I ordered the Lobster Empanadas as an appetizer, and they were taken out of the fryer to early and were still soggy.

I order the meatball hoagie and it was pretty good, only complaint was that it was drenched in sauce and ended up everywhere. The meal came with fries, which were fried well but they do the thing that Ruby Tuesday used to do and bring out a small thimble of ketchup to you when you ask.

Co-worker got the pork sub, which came as a seasoned pork with no sauce. He stated it was good but requested BBQ sauce for it.

The service was just fine our server was prompt and polite which is all you can ask, and the inside of the restaurant was clean and well kept.

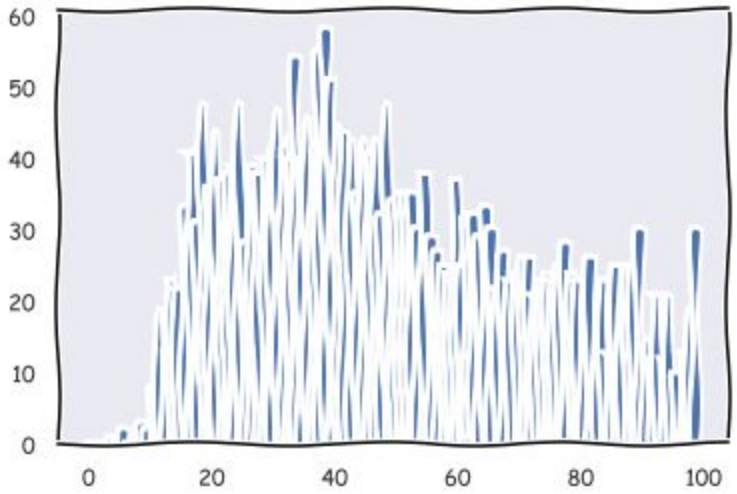
Overall it was a nice vibe, I could see going there for dinner but it was a bit pricey for the quality of food that you get.”

Actual Rating: 3
Predicted Rating: 3

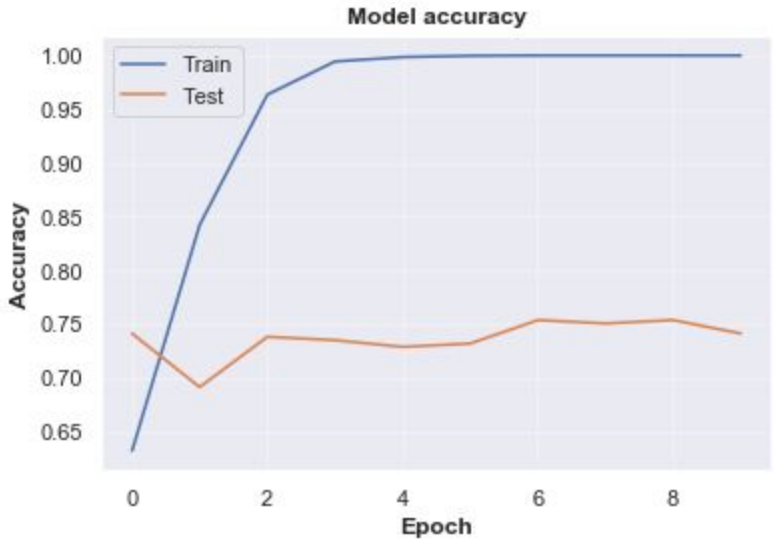
RNN - KERAS

One final approach to improving the NLP model would be to attempt to use a neural network, in this case an RNN. To implement this model first TensorFlow and Keras must be installed using python PIP commands. Following this a very similar process to the data wrangling steps in the previous section is applied however in contrast this is done with KERAS function library instead of SKLEARN.

Below are the results from the RNN word count distribution



Below are the results from the model training process:



Conclusion

Results indicate some successes in predicting restaurants for users. This could be a great utility for Yelp to help drive business by leveraging this knowledge. With accuracy and precision in the 71-72 percent range, these predictions are fairly reliable, yet this could most likely be fine tuned further in the future. Additionally methods such as factorization machines could be applied to add in other feature/user data which could also drive an improvement in the recommendation system.