# Milestone Report

Yelp provides a challenge for people to conduct research or analysis using their dataset and is currently in the 13 round of the challenge. For this iteration, they’ve updated the dataset and it includes information about 156,000 local businesses in 12 metropolitan areas. For this capstone project, I will build a restaurant recommendation system for users utilizing a combination of two methods: content-based filtering and collaborative filtering.

An example of a question being explored could be: “what is the best steakhouse to recommend to a user?” Using content-based filtering, we can recommend a steakhouse based upon that user’s previous reviews and preferences. But what if we have a brand new user to Yelp? Using collaborative filtering, we can recommend a steakhouse based upon similar user’s ratings.

This recommender system will provide insight from current users to recommend restaurants to both new and current users.

**UPDATE**:

Due to the size of the data and sparsity, I took a different approach for my project using the Yelp dataset. My goal is to create a recommender system that would give us insight as to whether a user would like a restaurant (greater than 3 stars) and if a user would

# The Data and More

## The Data

Using Yelp’s provided dataset for the challenge appeared to be straightforward and the data needed was in abundance. The data encompassed large files (over 8gb in total) and was zipped into a TAR file. Once I extracted the individual files from the TAR file, there were multiple JSON files to be used. For this project, we only needed a few of these JSON files, the ones that were most relevant.

These files were: business.json, review.json, and user.json.

The three files to be used combined to be about 7.7gb and gave me a lot of trouble in a few ways. Initially, reading in the data due to computational constraints. Using a home-built computer, I was able to read in the files, but it took a few tries of crashing. This was completed using a function that converted the JSON string into a flat python dictionary which could then be passed into Pandas. Another issue was the time it took to run many different lines of code on files of this size.

The business dataset broke down into many columns, the majority of which were business attributes.

The user dataset broke down in fewer columns and gave us data regarding the Yelp users.

The review dataset broke down into just nine columns.

The key to these datasets was the unique id’s for each business, user, and review that we would eventually use.

## Cleaning the Data

The difficulty I ran into most was working with such large files as many times the code that was needed to run took a lot of time. For the goal of this project, we wanted to recommend restaurants to both new and current users. That being said, I began with removing any data where the user\_id was null. To eventually predict what users would like, we need to begin with identifying what current users already like and it would be useless to include anything that we can’t trace back to a user.

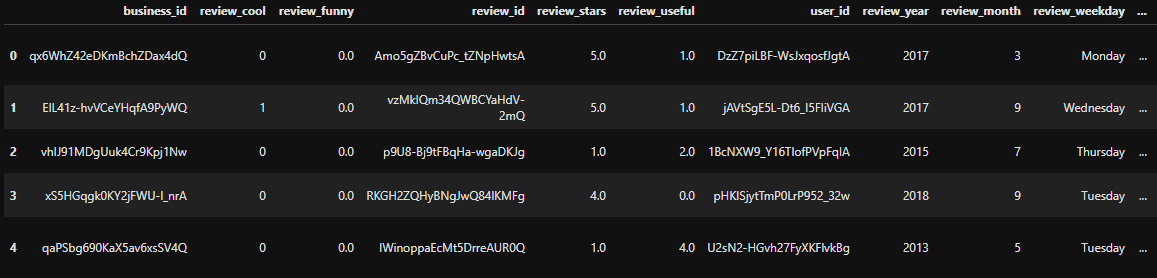
We also want to only be working with business data that is related to restaurants or food, therefore the removal of any business data that didn’t include “Restaurants” or “Food” as a category was completed. It is very difficult to accurately predict what a user would like if we only know one thing about them, ie. Only have left one review. The next step was to only keep review data of users who had at least 2 reviews associated to them.

Next steps included reducing the number of Restaurant Categories as there are many specialty restaurants that could be grouped together into a larger “food group.” This was completed by creating a finite number of cuisines and grouping together the restaurant type by the cuisine type. Something unique that I did for the purpose of hopefully increasing prediction accuracy long term was to group together the Asian cuisine restaurants that offered more than one type of Asian cuisine into Asian Fusion.

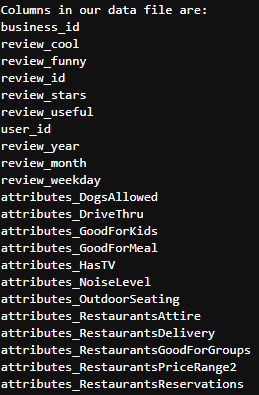
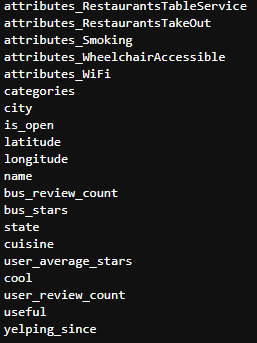
The business dataset included many features that weren’t relevant to restaurants to begin with, so I manually identified those and removed them.

Final cleaning step was to merge together the datasets in a usable format where each user’s review data for a business, that business’ attributes, and the user’s attributes were all in one row. This was done by merging on the unique id’s within each dataset.

By the end of the cleaning process, we had a dataframe that looked like this:



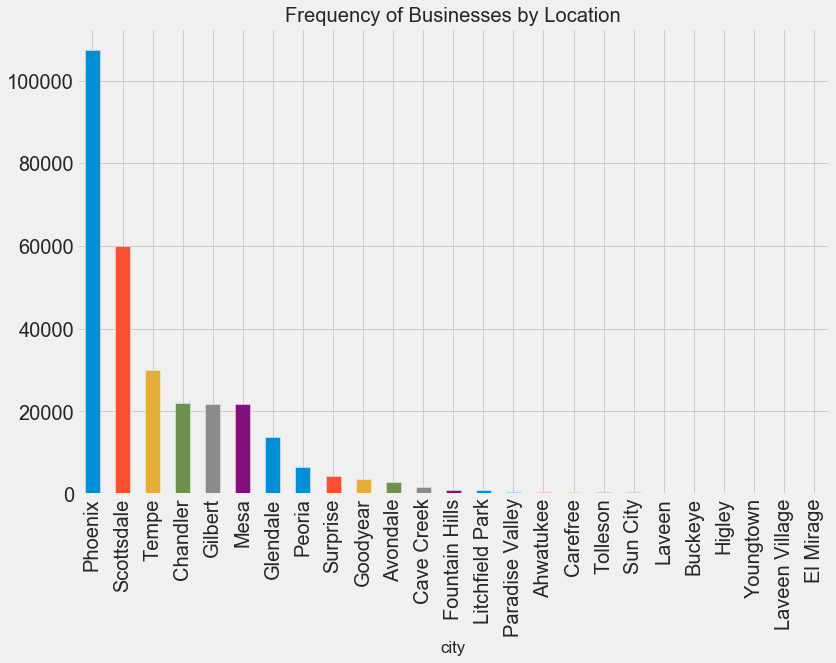
Here is a list of the columns:



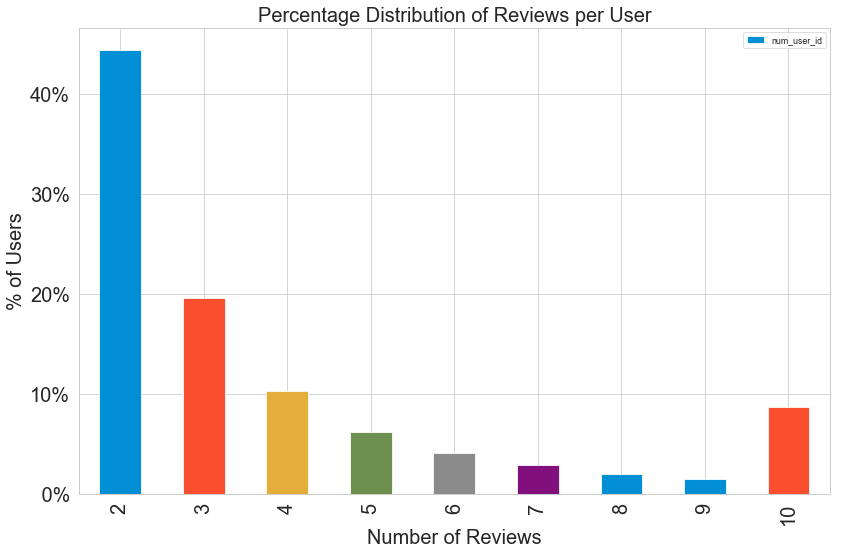
At this point I began exploring the data, and realized that this was too much data to be working with. I made an adjustment in my project to work only with restaurants in Arizona to help reduce the size and to get a better understanding of the restaurants where I live. This was fairly simple and I went back to the initial business data and selected only businesses that were in Arizona. The rest of the code that followed was hardly changed.

# Data Exploration

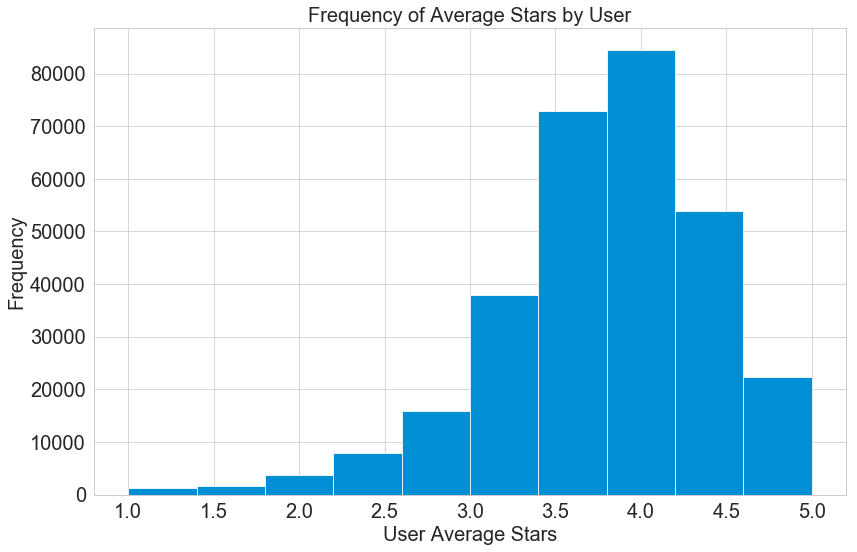
I began with getting a better understanding of the data through a local geographic standpoint, and that histogram plot is below.



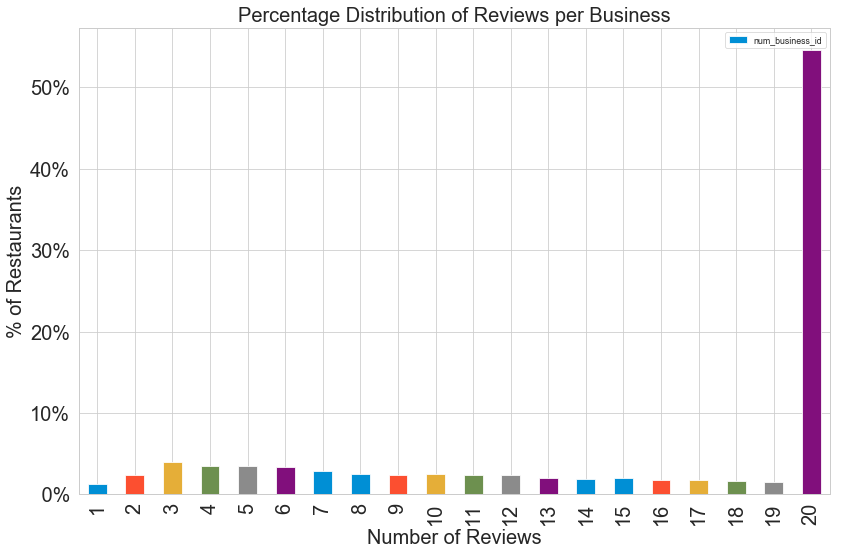
I was also curious about the users we were working with and just how active most Yelper’s were. I created a function that allowed for an easy plot that broke down how many reviews our users left.



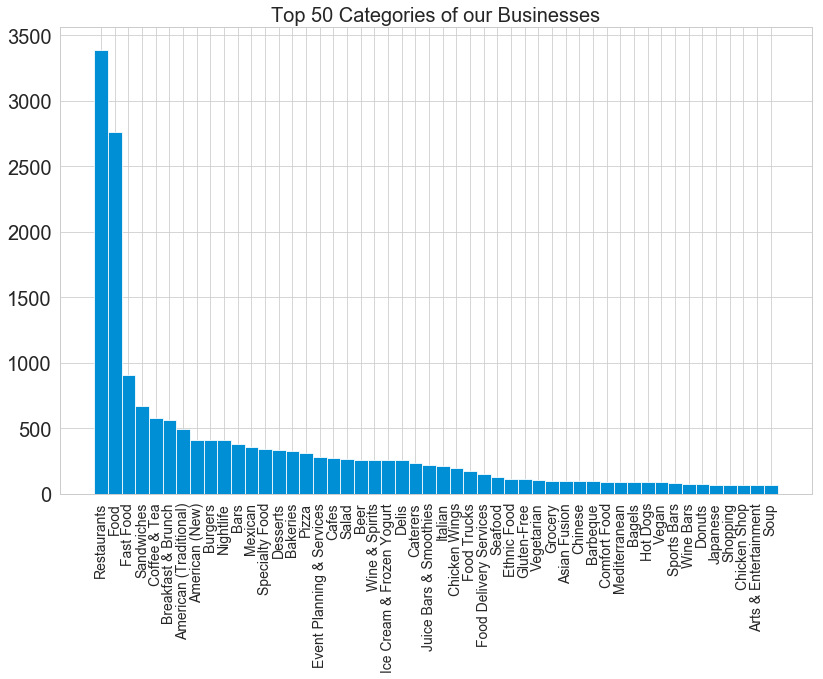
We can see here that the majority of Yelper’s have only left 2 reviews, but interestingly there are quite a few that have left 10 or more. It could be those that are active in Yelp, are very active in Yelp, but the majority aren’t. We must keep this in mind for our eventual prediction model.

Another point of interest was the average star rating a user gives. We can see in this histogram plot that the majority of User’s are generous and average 3.781 stars per review. This is something we need to keep in consideration as a potential that many users tend to give higher ratings than others.

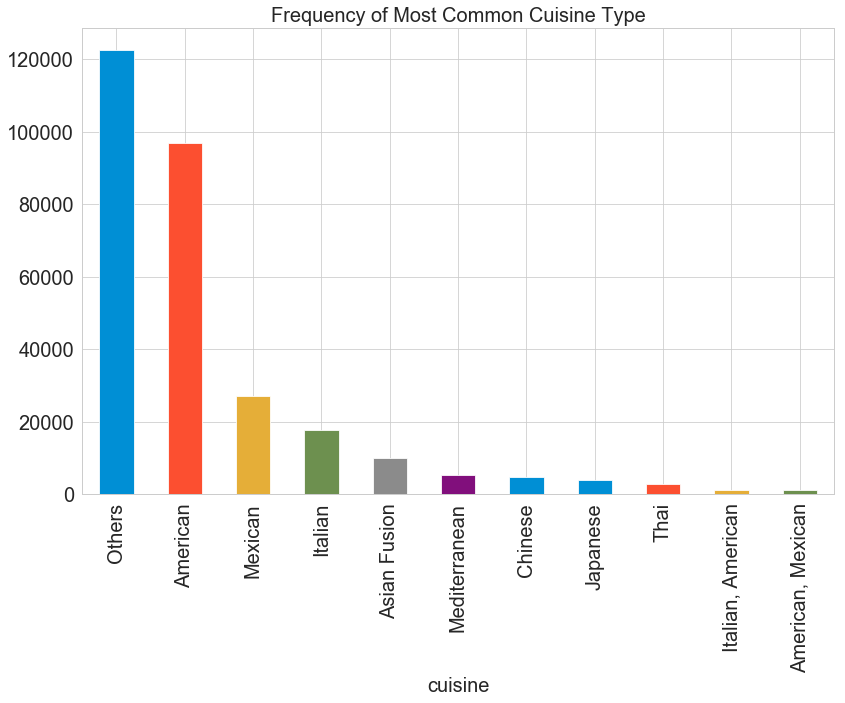
At this point, we’ve looked at some of our User data and Review data, but not our Business data. The plot below gives perspective into how many businesses have 20 or more reviews on their Yelp page.



Another histogram plot below shows the top categories in our Business data. Something to note here is that a Business may have multiple categories, so this shows the frequency for those categories amongst our Businesses.



Of course, after breaking down our most popular categories, it’s insightful to understand more about the business’ cuisine types. We can see here the breakdown of cuisine types in Arizona. Anything that didn’t fall under one of the pre-determined cuisine types was labeled as “Others” and surprisingly was the most common type. I think this can attest to the large variety of restaurants in Arizona.



From here it made sense to take a closer look at some of the statistics of our data.

The target variable for this project was “review\_stars” and we had a total of 301,731 in our dataset. A breakdown of our target variable:

* Statistics of the target variable:

|  |  |
| --- | --- |
| Median value: | 4.000000 |
| Mean value: | 3.771028 |
| Standard deviation value: | 1.363917 |

* Insight to the number of users, reviews, and ratings:

|  |  |
| --- | --- |
| Number of users : | 62446 |
| Number of reviews : | 301731 |
| Number of ratings : | 301731 |
| Average reviews/ratings per user: | 4.83187 |

The final piece to get set moving forward was to see how sparse our data was. I created a matrix and the result was: 0.10%

This number was devastating and resulted in a change in direction.

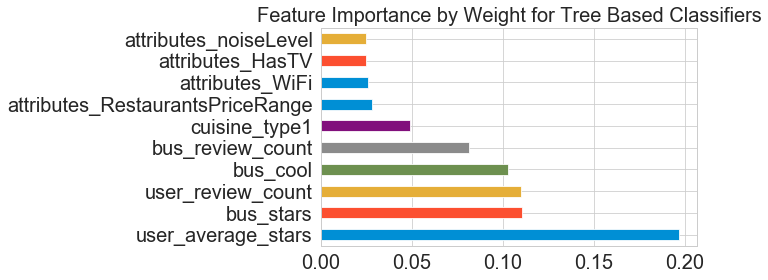
## Machine Learning:

This project was aimed at providing a recommendation system utilizing the Yelp Dataset. This would provide insight into how likely a user is to like a restaurant (rating greater than 3) and if they would give a rating greater than the business’ average. This also could provide some additional insight to be used in other functions of the business.

To gain this insight, I chose to utilize a classification-based set of prediction models including Logistic, KNN Classifier, ADAboost, SVM, Random Forest, and Decision Tree. Comparing the results of the various models in terms of accuracy would allow me to identify which of the prediction models was best to take a deeper exploration into.

**The first question to answer**: Would a user like a restaurant? (Give a rating greater than 3)

Once choosing the means to complete this prediction system, it was a matter of determining which features would provide the greatest impact. I completed the feature reduction for Tree Based Classifiers and the results are as follow:



This allowed me to identify the 10 most important features to utilize in my prediction systems.

The next step was to create and test the various models and being sure to fine tune the parameters for each model. Below is a summary of the results:



Consistently, the ADAboost Classifier was the best performing based upon its accuracy score determining how often it predicted correctly whether a user would give a rating greater than 3.

A more in depth exploration of the ADAboost model can be seen below:

**Confusion Matrix:**

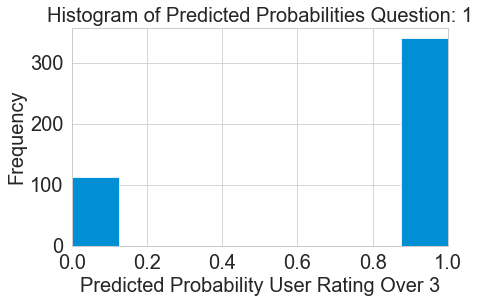
[[ 77 75]

[ 36 265]]

|  |  |
| --- | --- |
| 77 True Negatives | 75 False Positives |
| 36 False Negatives | 265 True Positives |

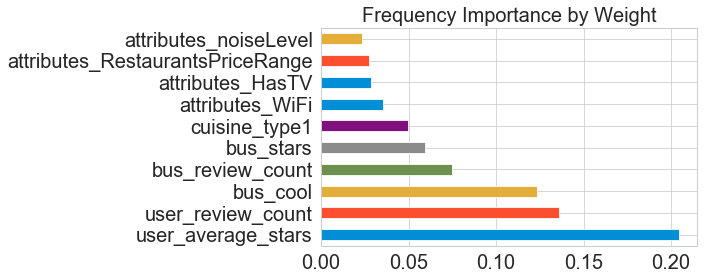
|  |  |  |
| --- | --- | --- |
| Attribute | Result | Description |
| Accuracy | .755 | How often classifier is correct |
| Error (misclassification rate) | .245 | How often classifier is incorrect |
| Sensitivity | .8804 | When actual value is positive, how often is classifier correct |
| Specificity | .5066 | When actual value is negative, how often is classifier correct |

Our classifier is highly sensitive and not very specific. Meaning that it does a great job of predicting when the actual value is positive, but struggles with when the actual value is negative. This greater sensitivity may also be due to the fact that majority of users give higher ratings, if you recall the plot showing user average star distribution from above.



**Question 2:** Would a user give a rating greater than the businesses average?

Let’s explore our second question in greater depth. It began with identifying the features that would provide the greatest impact on our prediction model. This can be seen below:

After understanding which features were most important, I built the same set of models with their individually tuned parameters. The results are below:



We see here that the ADAboost Classifier was the most accurate prediction model again. Let’s take a deeper exploration into this model to understand even more.

**Confusion Matrix:**

[[ 136 93]

[ 68 156]]

|  |  |
| --- | --- |
| 136 True Negatives | 93False Positives |
| 68 False Negatives | 156True Positives |

|  |  |  |
| --- | --- | --- |
| Attribute | Result | Description |
| Accuracy | .645 | How often classifier is correct |
| Error (misclassification rate) | .355 | How often classifier is incorrect |
| Sensitivity | .696 | When actual value is positive, how often is classifier correct |
| Specificity | .594 | When actual value is negative, how often is classifier correct |

Our classifier is just fairly sensitive and specific when predicting if a user would give a rating greater than the business’ average. The second question we targeted to predict wasn’t as accurate as our first question, and this may still be due to the user high rating bias we saw earlier.

# Conclusion

The initial goal of this project was to predict what a user would rate a restaurant that they hadn't previously been to. Given the sparsity of the data and the difficulty working with the massive files to begin with, this project took a different route. We had to pose a different question. I aimed to work with only data that involved the state of Arizona, where I currently reside.

The new direction was still related to the original goal. I aimed to use business attributes and user data to predict whether a user would like a restaurant (greater than 3 stars) and if they would rate a restaurant higher than the business' average rating.

This was setup as a classification problem. I built a variety of models using the most correlated attributes to the target variable. From here, there were multiple models built and tested to identify which provided the best results.

In both cases, the best model was the AdaBoost Classifier. While answering the first goal, whether a user would give a rating over 3, the model had an accuracy score of ~75%. The second goal, whether a user would give a rating greater than the business' average, the AdaBoost Classifier had an accuracy score of ~65%.

Looking at the results of the first AdaBoost model, I wasn't pleased with seeing such a high sensitivity and relatively low specificity. The second model wasn't as sensitive but also had a slightly higher sensitivity.

## Follow Up work:

If given more time and more computational power, I would move all of the data provided and utilize Spark or Hadoop in conjunction with AWS to provide a more in depth analysis and likely yield a better prediction model. I would also like to test for a user average star rating bias as this would impact the Yelp Challenge on a larger scale.