Restaurant Recommender

Milestone Report

Viraj P Modak

February 03 2020

CONTENTS

1. Problem statement:

Recommender systems have become extremely popular in today's world in social media and ecommerce industries. Some examples include, Facebook recommending friends, Netflix recommending movies or Amazon recommending products. A similar approach can be used to match restaurants to consumers. Typically when users look for restaurants, they go by star ratings alone. This approach is fraught with multiple shortcomings. The reviews may be highly skewed toward users with different preferences than the user who is looking for restaurant options. Furthermore, given the sheer number of restaurants in a large city, it is possible some really good options may never receive any consideration. The proposed recommender system will aim to tackle these issues. The problem statement can be defined as follows: Design a recommender system based on existing star ratings, reviews and user profiles, which will recommend a set of restaurants to a user. It is assumed here that the user will not have visited these restaurants before.

The obvious target client in this case is the entire population of a big city as they will be the direct users of this product. It can also serve as an advertisement platform for new restaurants. In addition, it can also be marketed to restaurant review apps or app based companies such as Yelp and Google. Because of the ubiquity of the business it targets i.e. the food industry, this product can add value in a variety of different ways.

2. Data to be used:

For this purpose, the Yelp restaurant review database will be used. The raw data is in the form of multiple json files including user, business and review information. The dataset can be found at the following location:

https://www.kaggle.com/yelp-dataset/yelp-dataset

This data will need to be cleaned up and represented in a matrix form where the rows are users and the columns are restaurants. The cells will include (quantified) information about the ratings and reviews.

3. Problem solving approach:

Defining the problem solving approach at such an initial stage is tricky because this is not a traditional supervised or unsupervised learning problem. Some options are listed as follows:

1. Standard recommendation system (forced supervised learning problem):
    - Reduce the data to a user by restaurant matrix with cells housing just the star ratings.
    - Divide the data into a train set and test set.
    - Train a ML model using a content-based, collaborative-based or a hybrid approach using just the training set. Optimize using cross-validation
    - Test performance on the test set
    - Ask user to enter ratings for select restaurants which he/she has visited. Provide recommendations based on new data - i.e. restaurants which according to the model, the user will give high star ratings for
2. Recommendation system with reviews (forced supervised learning problem): The approach here is similar to the previous option. But in this case, in addition to the star ratings, reviews and key-words will also be incorporated. Subsequent training, optimization and testing approach will be similar
3. Unsupervised learning based on clustering:
    - Reduce the data into feature space which purely consists of ratings and reviews
    - Identify clusters based in this feature space
    - Ask a new user to rate and review a handful of restaurants that he/she has visited
    - Recommend restaurants based on proximity to clusters

4. Data Wrangling/Clean-up

Since this is not a traditional X vs Y machine learning problem, it is anticipated that the data wrangling will not be more continuous and not a one-time effort. In Particular, there will be subsequent clean-up involved as part of Exploratory Data Analysis as well. Following is a brief summary of the preliminary data wrangling process.

- The data consisted of separate json files for user info, review info and business info. The json files were converted into Pandas dataframes.
- The file for business info included information about other services as well. This was filtered to include only Restaurants
- The dataframes were then saved as csv files and these csv files will then be used for all subsequent analysis
- The info files consisted of data from multiple metropolitan cities. Based on the problem statement, it made sense to develop a system for a city and the same methodology can be implemented for other cities as well
- The rating matrix with users as rows and restaurants as columns with the star ratings as cells was extracted which will be the starting point for training the recommender system based on collaborative/content-based/hybrid filtering.
- However, the data in the info (csv) files will be used to perform Exploratory Data Analysis and Inferential Statistical Analysis

5. Exploratory Data Analysis

The EDA approach is described in a fully executed notebook and the link for the same is provided in the Appendix. Key highlights and plots are shown here.

5.1 Distribution of ratings - raw data

The star ratings from the review_info dataset were plotted to explore how the ratings are distributed. It is shown in Figure 1. It can be seen that:

- 5-star reviews are more than 1-4 star reviews
- More than half of the reviews imply a positive experience (4-5) stars

This shows that in most of the cases, the reviewers have been satisfied because of the service. However, this might mean that only the "good" restaurants get reviewed often and may not mean the general quality of the restaurants in the city of choice is good. For that how the restaurants distributed across the star-rating metric needs to be explored.

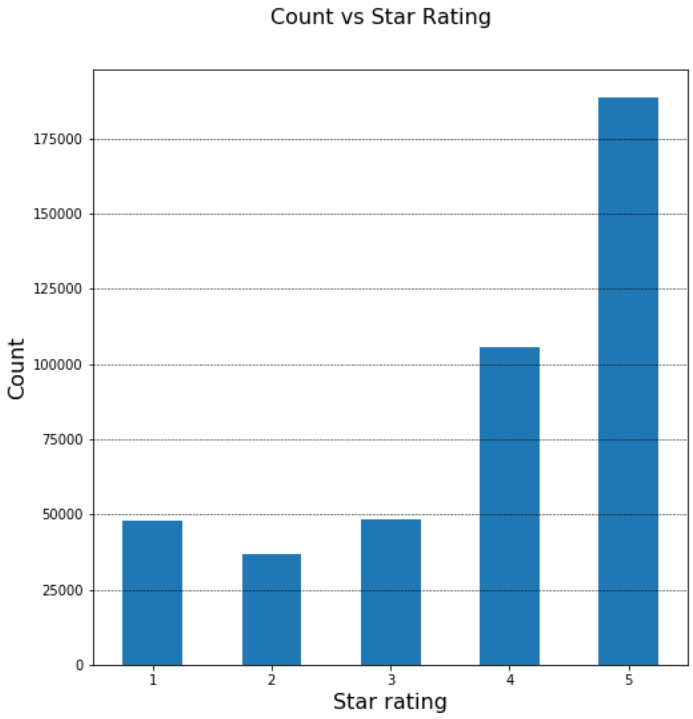

Figure 1: Count of reviews plotted as a function of the review star rating

5.2 Distribution of restaurants by star reviews

The restaurant count can be plotted as a function of the average star rating for those restaurants - shown in Figure 2. It sheds more light on what was seen in the previous plot. Overall in the city of Phoenix, there are a little over 500-600 restaurants which have an average rating of above 4. However, the majority of the restaurants are rated 3-4. This in a way confirms what was hypothesized earlier, that only "good" restaurants are reviewed more often. This can mean either people don't visit the not-so-good restaurants as often or that after people visit these restaurants, they do not feel like writing a review. Overall though, it can be concluded that in Phoenix, people are satisfied with their restaurant experience - (assumption: above 3 star is satisfied).

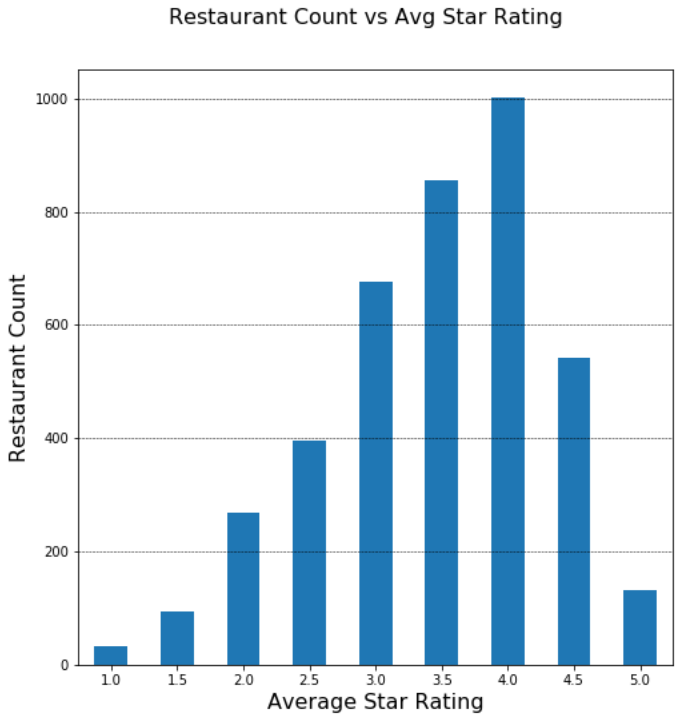

Figure 2: Restaurant count as a function of average star rating

5.3 Ratings per user

This analysis will help identify how active a user is while reviewing and shown in Figure 3. This makes for an interesting viewing. It is interpreted as follows:

Almost 100000 users have written only ONE review for restaurants in Phoenix. On the other end there are only 20 reviewers who have written more than 100 reviews for restaurants in Phoenix.

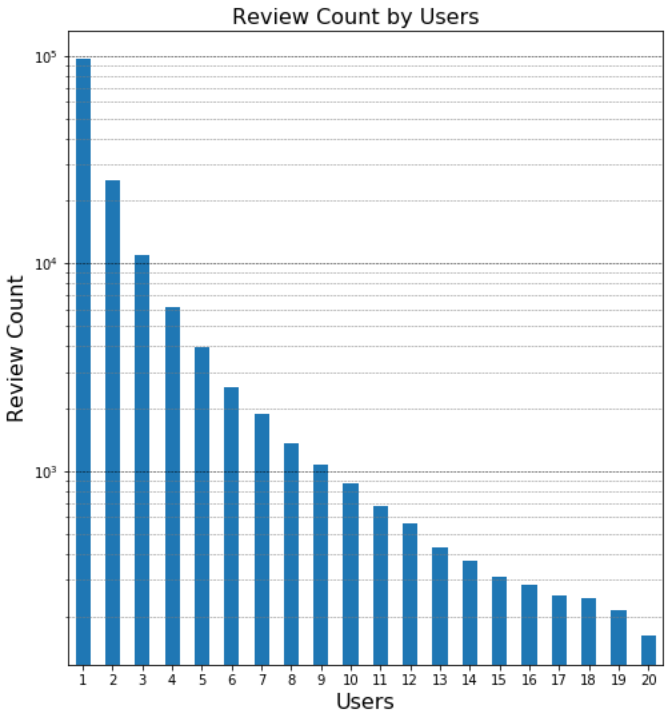

Figure 3: Review count per bin of user count

This analysis will be useful in the future because collaborative filtering is based on user characteristics and reliable characteristics about likes and dislikes can be extracted only if a user frequently writes reviews. Getting information from people who have reviewed only once or twice does not make much sense

5.4 Ratings per year and per month

Time trends in review writing habits can also be explored to answer questions like is there a particular year(s) or period of the year when the reviews are more frequent? The frequency of reviews by month and by year are shown in Figure 4 and Figure 5 respectively. We can see that the number of reviews submitted has definitely increased over time. This can of course be explained by an increases in connectivity, reliance on

online reviews and their prominence. As far as review month is considered, we do not see a large variation between months, but indeed, during the Nov/Dec holiday season, the frequency of reviews decreases, possibly because people prefer to stay indoors with family.

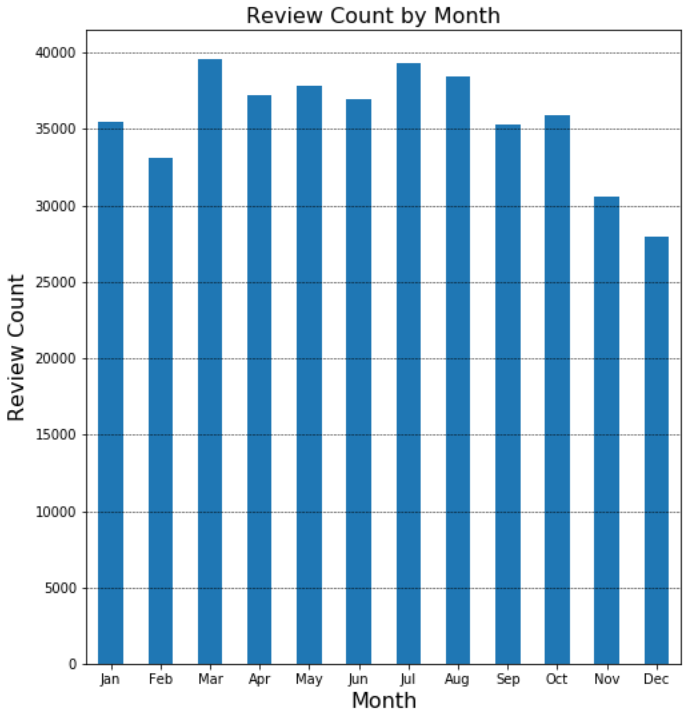

Figure 4: Review count by month

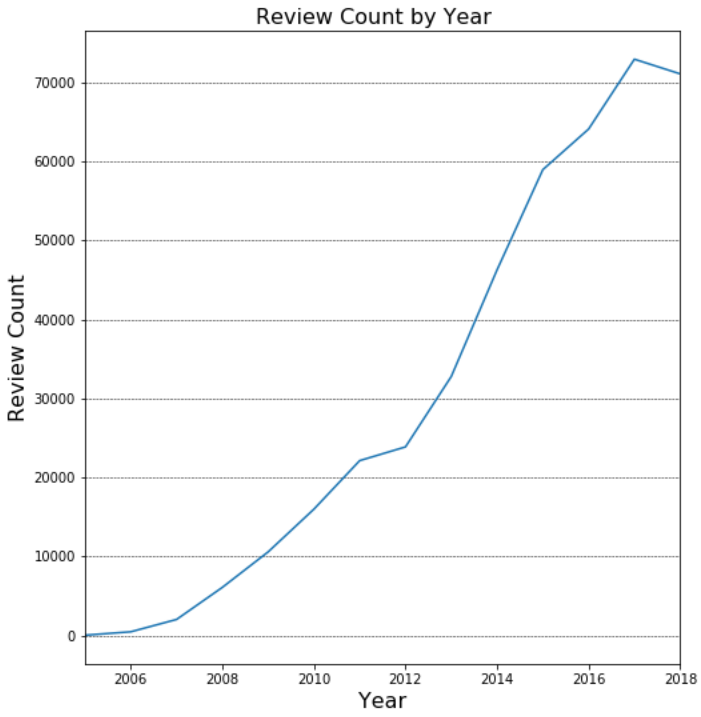

Figure 5: Review count by year

5.5 Restaurant count by category

Content based filtering is a common way to build a recommendation system. In this case, multiple objects (or in this case, restaurants), will be compared based on their characteristics - or content. In our case, one such characteristic is "category". In the next few plots, the trends between category and reviews are explored for the city of Phoenix. Figure 6 shows how many restaurants serve some typical cuisine. Mexican has the highest with >700 restaurants with Indian being the lowest with only ~50 restaurants. The data is not scrubbed for being exclusive. For example, there might be some restaurants which serve both Mexican and Italian. However, the y-axis will still reflect the correct number of restaurants serving that particular cuisine.

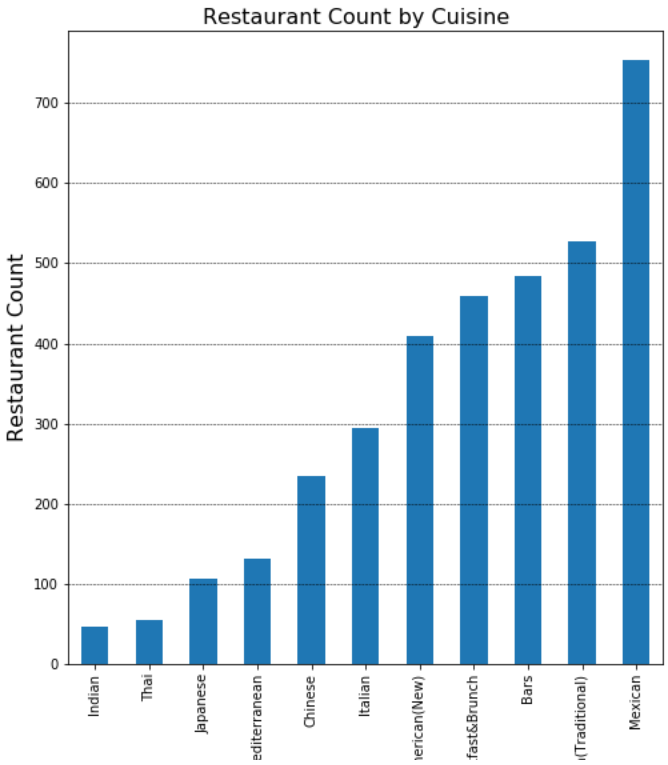

Figure 6: Restaurant count by cuisine

5.6 Restaurant count by category - segmented by star rating

We can extract how the restaurants are rated based on the category (or cuisine) shown in Figure 7. The resulting graph has a lot of information which needs to be unpacked. The general trend we see is expected i.e. a bell shaped plot of ratings - 1-5 stars. With peaks around either 3.5 or 4. We can extract an average rating by category as well, which will be seen in the next plot

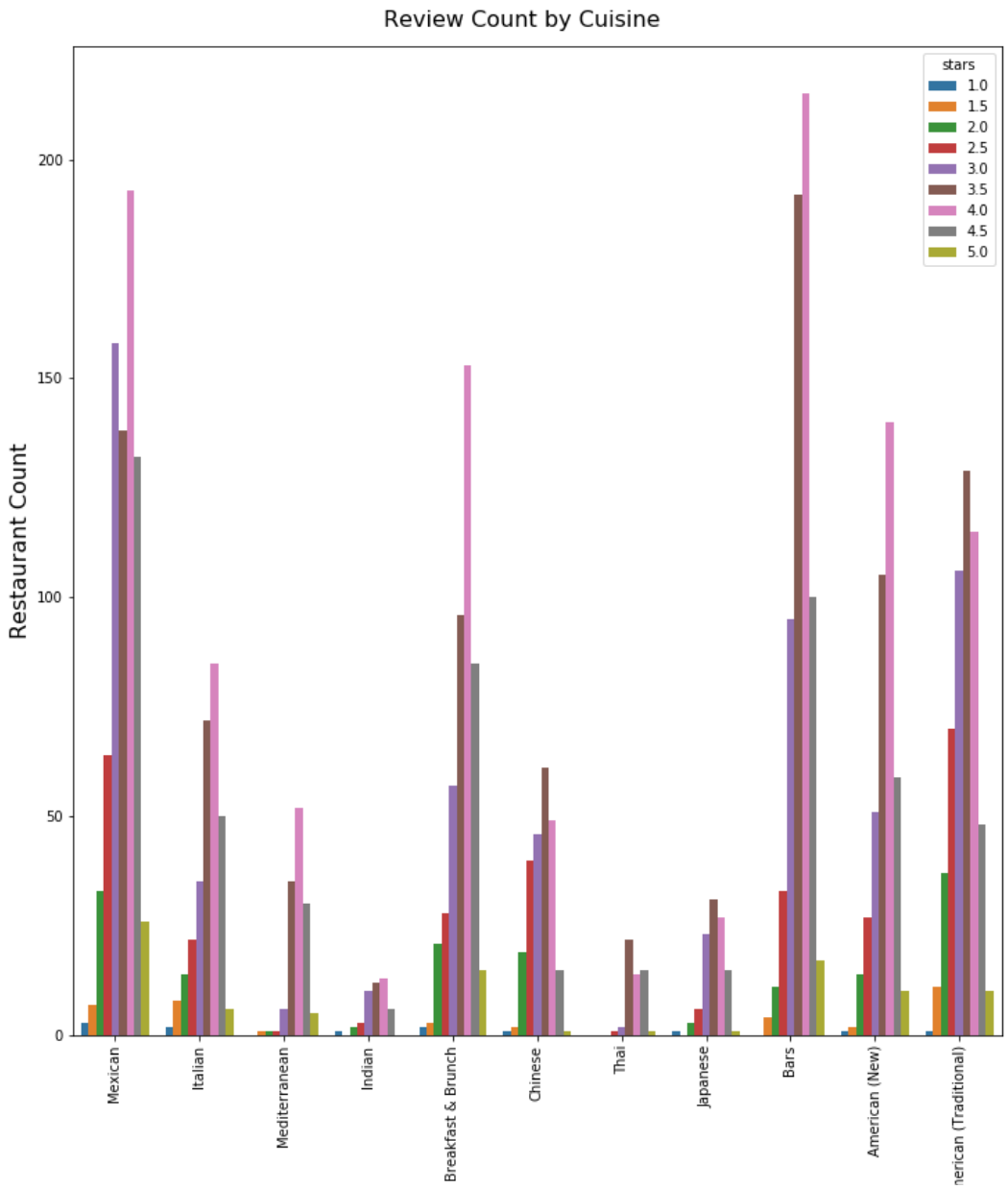

Figure 7: Restaurant count by category/segmented by star rating

5.7 Average Rating by category

As noted before, most of the categories (or cuisine) average between 3.5 and 4 stars as shown in Figure 8. An interesting thing to note here is with the Mediterranean and Thai restaurants - we saw that these two cuisines offer the least choice in terms of number of restaurants. Even so, they are two of the best rated cuisines in the city of Phoenix.

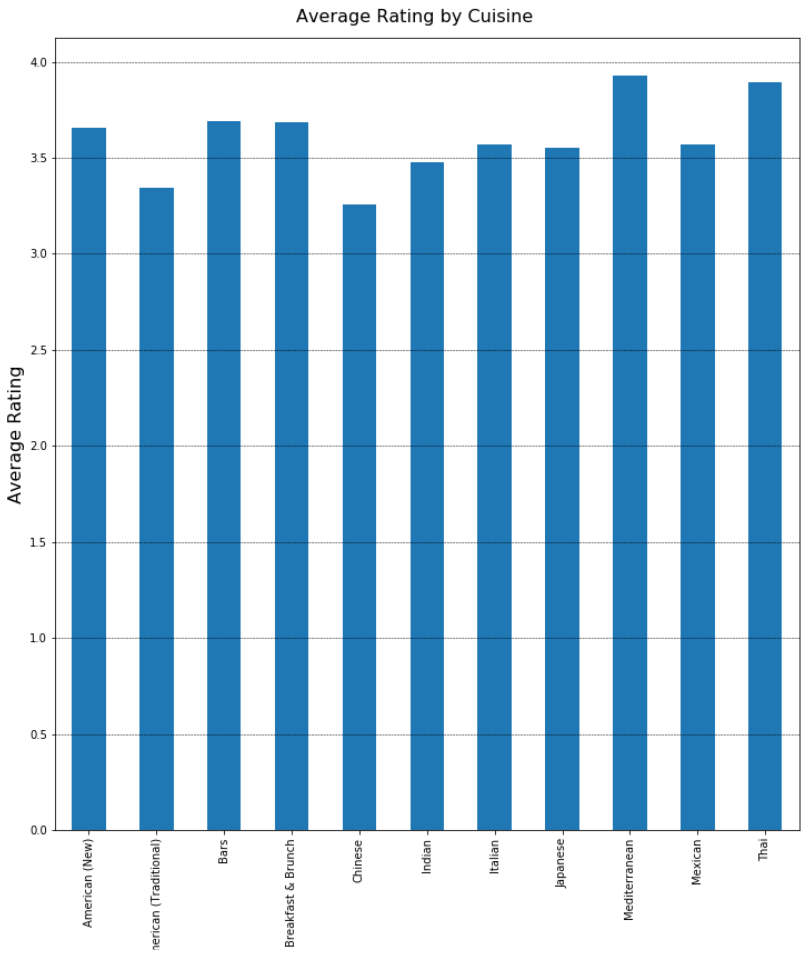

Figure 8: Average rating by Category

5.8 Review count vs useful

In the previous plots, it was explored how the number of ratings vary as a function of number of users. But how many of them are actually helpful and if there are any outliers is yet to be seen. The "useful" count vs review count can be plotted to explore that relationship - shown in Figure 9. Intuitively we expect this to be a positive correlation which is what we see.

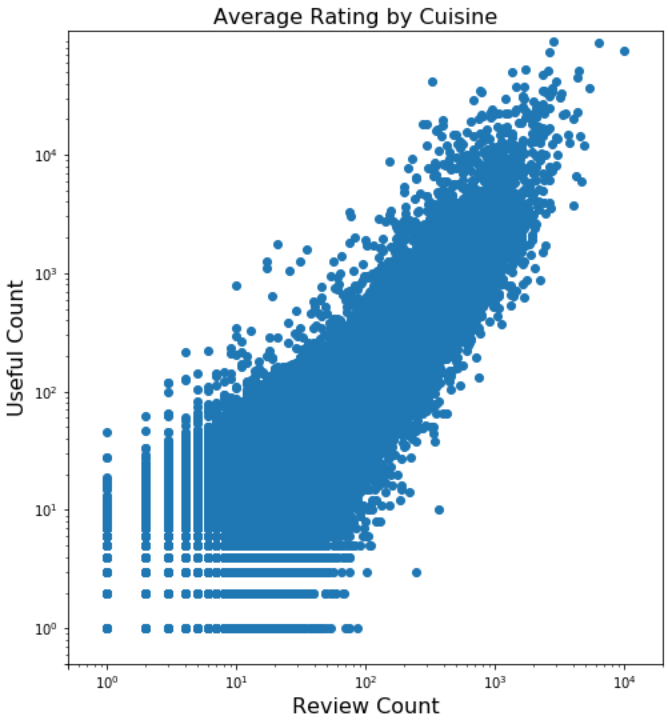

Figure 9: Usefulness of user reviews vs review count

5.9 Exploring review content

Now that the raw trends are explored, information from the review text can be extracted. The objective of this is to investigate if there is any similarity between two reviews which have similar star ratings and if there is significant difference between reviews which have largely different ratings. For this purpose the following approach was adopted.

Let us consider a pair of reviews with a 1-star and a 5-star rating. The review text of these reviews was reduced to key words using PorterStemmer and after excluding English stopwords. Two of the reduced reviews were converted into a vectorized feature array and cosine similarities were calculated between those. This calculation can be repeated for and bucketed into all possible couplings of reviews - 1&4, 2&4, 3&4 and so on. Because of limited computational power only 1000 reviews were considered out of 427491, which still yielded 499500 pairs. The cosine similarities were then plotted as a function of star-rating pairing which is shown in Figure 10. However, we don't find significant difference between reviews of similar star rating and a very different star rating.

Whether a review has a positive or negative sentiment can be understood. However that will require using actual ML techniques and either supervised or unsupervised learning. That is not the primary objective of this section or of this project as a whole and was not explored further.

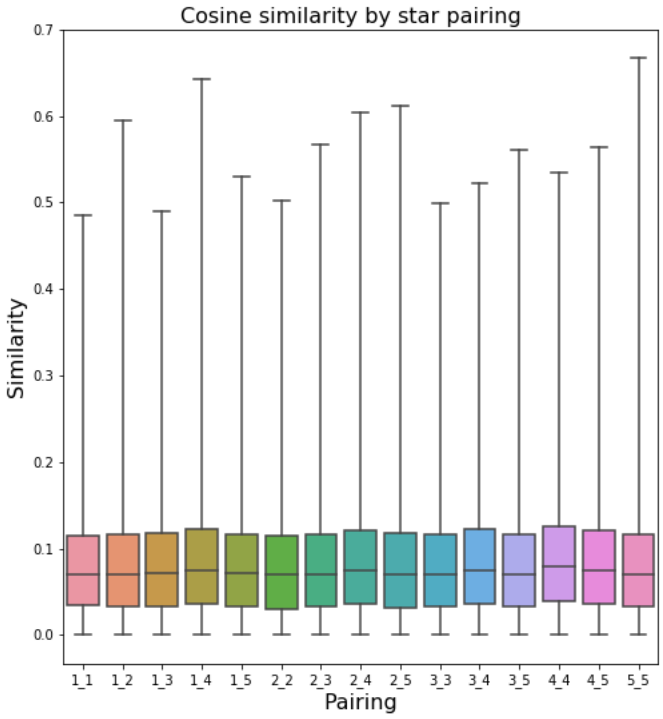

Figure 10: Review similarity vs review rating pairing. No significant similarity noted with respect to parings

6. Statistical Analysis

As mentioned before, this is not a standard machine learning problem with a set of independent variables and a response variable. As a result, standard inferential statistics and statistical tests may neither be relevant nor add any value to the end goal. Even so, based on the variables we have explored, we can still obtain basic statistical parameters such as mean and standard deviations wherever applicable. A few relevant cases are illustrated below.

6.1 Rating mean and stdev by cuisine

This data was shown in Figure 8 but listed along with the standard deviations in Table 1. We see no significant differences in the spread of the ratings. All of them lie between 0.5 and 0.8. Italian, is the only noticeable cuisine which has a larger spread considering the number of Italian restaurants are not as high as some of the other cuisine.

Table 1: Mean and stdev of restaurant ratings by cuisine

Cuisine	Mean Rating	Stdev
American (New)	3.656479	0.68597
American (Traditional)	3.343454	0.7755
Bars	3.691904	0.63072
Breakfast & Brunch	3.684783	0.74579
Chinese	3.25641	0.72985
Indian	3.478723	0.74424
Italian	3.568027	0.80557
Japanese	3.551402	0.67619
Mediterranean	3.927481	0.54815
Mexican	3.570955	0.78113
Thai	3.890909	0.50636

Furthermore, one-way ANOVA can be also performed on star ratings by cuisine to see if the mean star rating for different cuisines are statistically dissimilar. For this purpose the star ratings of the individual reviews from the review_info file were segment it by cuisine and the mean of those star ratings were subjected to a one-way ANOVA. This is a more robust check than the one before, because the data size is larger and TRUE mean of cuisines will be tested rather than mean of average star rating of the restaurants serving that cuisine. The F-value for this test was 323.8 and the P-value was 0, suggesting that the null hypothesis - that all means are equal - is rejected. Details of this calculation are presented in the IPython notebook for EDA.

6.2 Quantifying review length by keywords

In the data-wrangling and EDA sections, the review text was vectorized into keywords. It will be interesting to see how long a typical review is. The following statistics were calculated for length of reviews:

- **Mean**: 51 keywords
- **Median**: 36 keywords
- **Stdev**: 47 keywords

It can be seen that there is a large spread - mean:stdev is almost 1:1 - in terms of reviews written. In other words, there are users which are succinct in their reviewing and there are users which are more verbose. However, the distribution is heavily skewed toward to reviews which are shorter, which is explained by the median value of 36. The distribution is plotted in Figure 11.

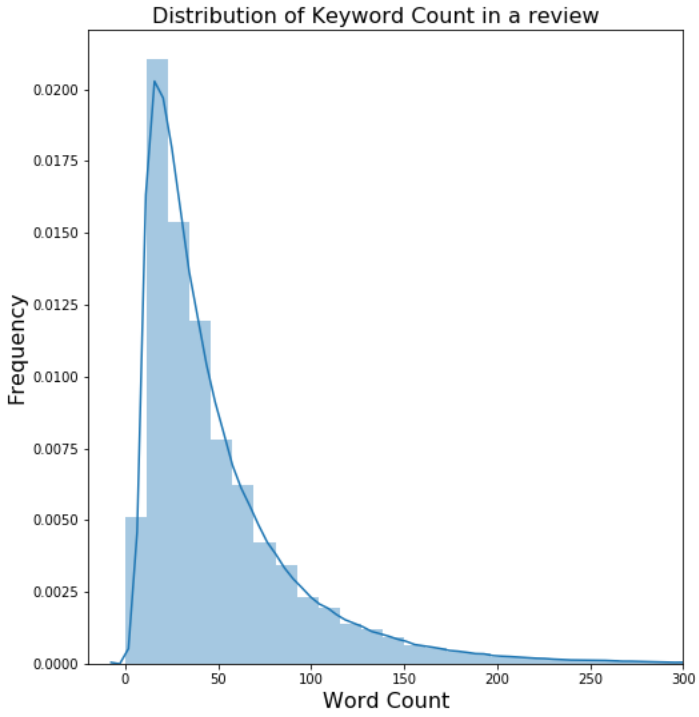

Figure 11: Distribution of review keyword count

6.3 Yelping age Statistics

In collaborative filtering, the recommendations are based on how the user is similar to previous users or reviewers. One characteristics which is worth exploring is the "Yelping Age" or how long has the user been active on Yelp. The following statistics were calculated and the distribution is shown in Figure 12. We can see that the average yelp user in Phoenix is quite mature at an age of 6-7 years

- **Mean**: 2404 days
- **Stdev**: 945 days
- **Median**: 2372 days

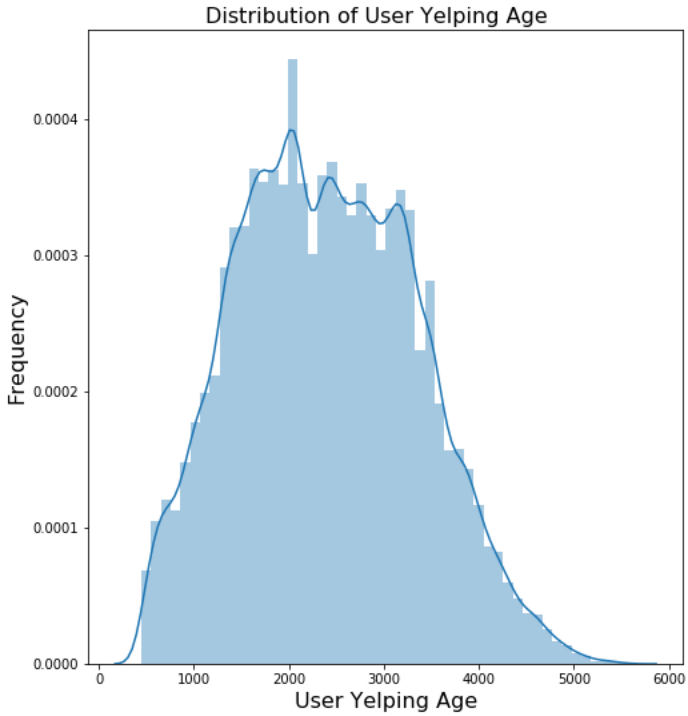

Figure 12: Distribution of Yelping Age of Users

7. Summary

We started with a problem statement of developing a Restaurant Recommender app which primarily can be used by the general population, also used as an advertising platform and finally also acquired by apps such as Google and Yelp. The raw dataset was acquired from Kaggle, which was cleaned up to a usable format. This included information about reviewers, restaurants and the actual reviews. Keeping in mind that this is not a standard ML problem with independent and response variables, EDA and Statistical Analysis were performed for informational purposes.

Review texts were vectorized. A rating matrix was also derived with User ID and Restaurant ID as the X and Y axes respectively and star ratings as cell values. This information will be critical in the next step, building collaborative/content-based and hybrid models for the recommender system.

Appendix

Cleaned up dataset is not included because of the large size. However, instructions for cleaning up the dataset from the raw data are available in the following notebooks.

Link to preliminary data clean-up notebook:

https://github.com/virajmodak16/Restaurant_Recommender/blob/master/Preliminary_data_cleanup.ipynb

Link to Notebook for EDA and Statistical Analysis:

https://github.com/virajmodak16/Restaurant_Recommender/blob/master/Exploratory%20Data%20Analysis.ipynb