Springboard Data Science Career Track | Nikki Seegars
Capstone Project II: Yelp Restaurant Recommendation Engine

Problem Statement

There is a need to have Yelp user personalized collaborative recommendation engine for restaurant choices based on the historical data that has been gathered about each user's past star reviews. The recommendations will be based on the similarities to other users in their network by predicting their preference by using matrix filtering.

Data Set Overview

Yelp is a popular online business directory with crowd-sourced user reviews. Many types of business have yelp pages including restaurants, shopping stores, and personal services such as tax preparation. The restaurants tend to have more user inputs than other types of businesses. Yelp supplies a subset of their user data for academic purposes and to be used in company sponsored challenge problems. The available data can be downloaded from their website and is taken from 10 metropolitan areas across two countries. There are six json files with information about businesses, reviews, and users. See Table 1 below which is a summarization of the contents in each json file.

Json File	Record Count	Number of Features	Features
business	192,609	14	business_id, name, address, city, state, postal code, latitude, longitude, stars, review_count, is_open, attributes, categories, hours
checkin	161,950	2	business_id, date
photo	200,000	4	photo_id, business_id, caption, label
review	6,685,900	9	review_id, user_id, business_id, stars, date, text, useful, funny, cool
tip	1,223,094	5	text, date, compliment_count, business_id, user_id
user	1,637,130	22	user_id, name, review_count, yelping_since, friends, useful, funny, cool, fans, elite, average_stars, compliment_hot, compliment_more, compliment_profie, compliment_cute, compliment_list, compliment_note, compliment_plain, compliment_cool, compliment_funny, compliment_writer, compliment_photos

Table 1.

Data Collection & Wrangling

Each of the json files were loaded into DataFrames to review the features and have basic descriptive statistics calculated. In addition the files were saved to .csv files in order to easily load into SQL tables for data aggregation. Although the data set is marketed as being a subset of 10 metropolitan areas, the review.json file has 36 different states included. For simplicity

purposes, the top 5 states are being used for building the recommendation engine. The states are shown along with their review counts in Figure 1.

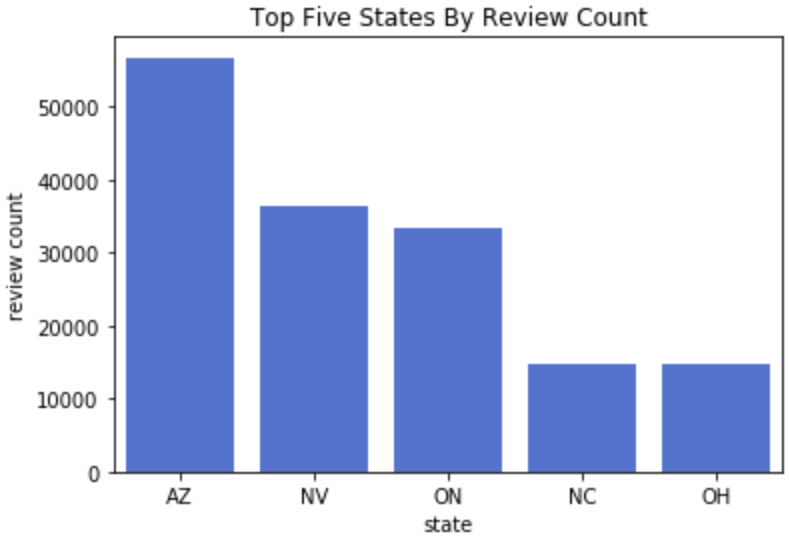

Figure 1.

Some of the businesses in the Yelp data set have been identified as being closed, these businesses will not be included in the recommendation model.

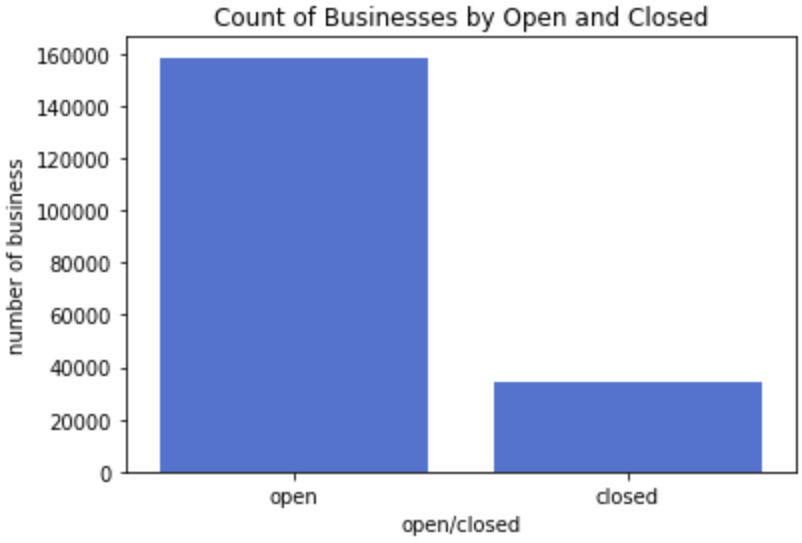

Figure 2.

In order to focus only on restaurants, the categories feature will be used and any records with categories beginning with select descriptors will be assumed to be restaurants. There are 1,177 unique categories in the original data set. Of those, 218 describe food and beverage businesses. The review.json file has reviews, user id, and star ratings. The text in the reviews are very long so that column was left out of the DataFrame and .csv files since it is not needed

for this analysis. The user.json file has attributes including number of reviews and any friends' user ids. The checkin.json, photo.json, and tip.json files do not contain any necessary information for this project, so those features were not incorporated in the model.

The .csv file data was imported into PostgreSQL tables and the following view was created of a query to combine all of the pertinent features.

```
SELECT
r.date,
r.review_id,
r.user_id,
r.stars AS user_stars,
b.categories,
r.business_id,
b.name AS business_name,
b.review_count,
b.stars AS business_stars,
b.address,
b.city,
b.state_,
b.postal_code,
b.latitude,
b.longitude,
b.hours,
b.is_open,
rank() OVER (PARTITION BY r.business_id ORDER BY r.business_id, r.date) AS order_rank
FROM yelp_review_data r
JOIN yelp_business_data b ON r.business_id = b.business_id
WHERE b.is_open = 1
AND b.state_ IN ('AZ', 'NV', 'ON', 'NC', 'OH')
AND b.review_count > 4
AND b.categories SIMILAR TO '(Acai Bowls|Afghan|African|American|Arabian|Argentine|Armenian|Asian
Fusion|Australian|Austrian|Bagels|Bakeries|Bangladeshi|Bar Crawl|Barbeque|Bars|Basque|Beer|Belgian|Beverage
Store|Bistros|Brasseries|Brazilian|Breakfast & Brunch|Brewpubs|British|Bubble
Tea|Buffets|Burgers|Burmese|Butcher|Cafes|Cafeteria|Cajun/Creole|Cambodian|Canadian (New)|Candy
Stores|Cantonese|Caribbean|Caterers|Cheese|Chicken|Chinese|Chocolatiers & Shops|Cideries|Cocktail
Bars|Coffee|Colombian|Comfort Food|Convenience Stores|Conveyor Belt
Sushi|Cooking|Creperies|Cuban|Cupcakes|Custom Cakes|Czech|Delicatessen|Delis|Desserts|Dim Sum|Diners|Dinner
Theater|Dive Bars|Do-It-Yourself Food|Dominican|Donairs|Donuts|Eatertainment|Egyptian|Empanadas|Ethical
Grocery|Ethiopian|Ethnic Food|Ethnic Grocery|Falafel|Farmers Market|Fast Food|Filipino|Fish &
Chips|Fondue|Food|French|Fruits &
Veggies|Gastropubs|Gelato|German|Gluten-Free|Greek|Grocery|Guamanian|Hakka|Halal|Hawaiian|Health
Markets|Herbs & Spices|Himalayan/Nepalese|Honduran Hong Kong Style Cafe|Hot Dogs|Hot Pot|Hungarian|Ice Cream
& Frozen Yogurt|Imported Food|Indian|Indonesian|International|Internet Cafes|Irish|Irish
Pub|Italian|Izakaya|Japanese|Juice Bars & Smoothies|Kebab|Kids Activities|Korean|Kosher|Laotian|Latin
```

The PostgreSQL query is imported into python by creating a function that uses the pd.read_sql to retrieve the filtered data.

```
# Set up a connection to the postgres server.
conn_string = "host="+ creds.PGHOST +" port="+ "5432" +" dbname="+ creds.PGDATABASE +" user=" + creds.PGUSER \
+" password="+ creds.PGPASSWORD
conn=psycopg2.connect(conn_string)

# Create a cursor object
cursor = conn.cursor()

def load_data(schema, table):

    sql_command = "SELECT * FROM {}.{};".format(str(schema), str(table))
    print (sql_command)

    # Load the data
    data = pd.read_sql(sql_command, conn)

    print('data shape', data.shape)
    return (data)
psql_data =   load_data('public', 'yelp_business_review_subset0')
```

Each of the five metropolitan geographic sizes was approximated with the haversine formula which calculates the great-circle distance (shortest) between two points on Earth using their latitude and longitude. Python has a haversine library that will calculate the spherical difference given the latitude and longitude of two points.

```
from haversine import haversine
haversine((start latitude, start longitude), (end latitude, end longitude), unit='mi') #unit in miles
```

The five metropolitan areas of interest are Metro Phoenix, Las Vegas Valley, Greater Toronto, Charlotte Metropolitan, and Greater Cleveland. The great-circle distances calculated for each using the minimum and maximum latitude and longitude values are 69.61, 45.15, 69.42, 59.76, and 226.23 miles respectively. Greater Cleveland has an outlier with the haversine well over 70 miles. Upon closer inspection of the data it was determined that one restaurant incorrectly listed the state as “OH” instead of “ON”. After the changes were made to the DataFrame, Greater Cleveland’s distance went down to 78.08 miles and Greater Toronto remained the same.

Metro Area	Distance (miles) Before Adjustment	Distance (miles) After Adjustment
Metro Phoenix	69.61	69.61
Las Vegas Valley	45.15	45.15
Greater Toronto	69.42	69.42
Charlotte Metropolitan	59.76	59.76
Greater Cleveland	226.23	78.08

Table 2.

Each of the 215 restaurant categories had a column created for one hot encoding with a 1 indicating that the category was included and 0 indicating it is not for each record.

```
for category in category_list:
    psql_data[category] = np.where(psql_data['categories'].str.contains(category), 1, 0
```

Exploratory Data Analysis

Grouping by state, a summary of aggregate descriptive statistics give a picture of how each metropolitan area compares to one another.

	business_stars					review_count				
	count	min	max	mean	median	count	min	max	mean	median
state_										
AZ	1012812	1.0	5.0	3.801264	4.0	1012812	5	2556	388.067710	258
NC	235738	1.0	5.0	3.732587	4.0	235738	5	1572	241.455451	144
NV	1118221	1.0	5.0	3.823314	4.0	1118221	5	8348	994.148511	506
OH	197360	1.0	5.0	3.742301	4.0	197360	5	1074	161.299787	93
ON	489709	1.0	5.0	3.600254	3.5	489709	5	2121	170.063403	97

Table 3.

Yelp allows users to give a review and rate each business between 1 and 5 stars, with 1 being the least favorable and 5 being the most favorable. Each metropolitan area's mean average business star rating is fairly similar, between 3.6 and 3.8 with Toronto having a bit of lower average star rating. For this project, restaurants with 5 or more reviews were included with the median number of review count for restaurant by metropolitan area ranging from 93 to 506.

Most of the star ratings fall between 3 and 4 stars as shown in the histograms in Figure 3 with Greater Toronto being the only metropolitan area showing more 3 star ratings, than 4 star ratings.

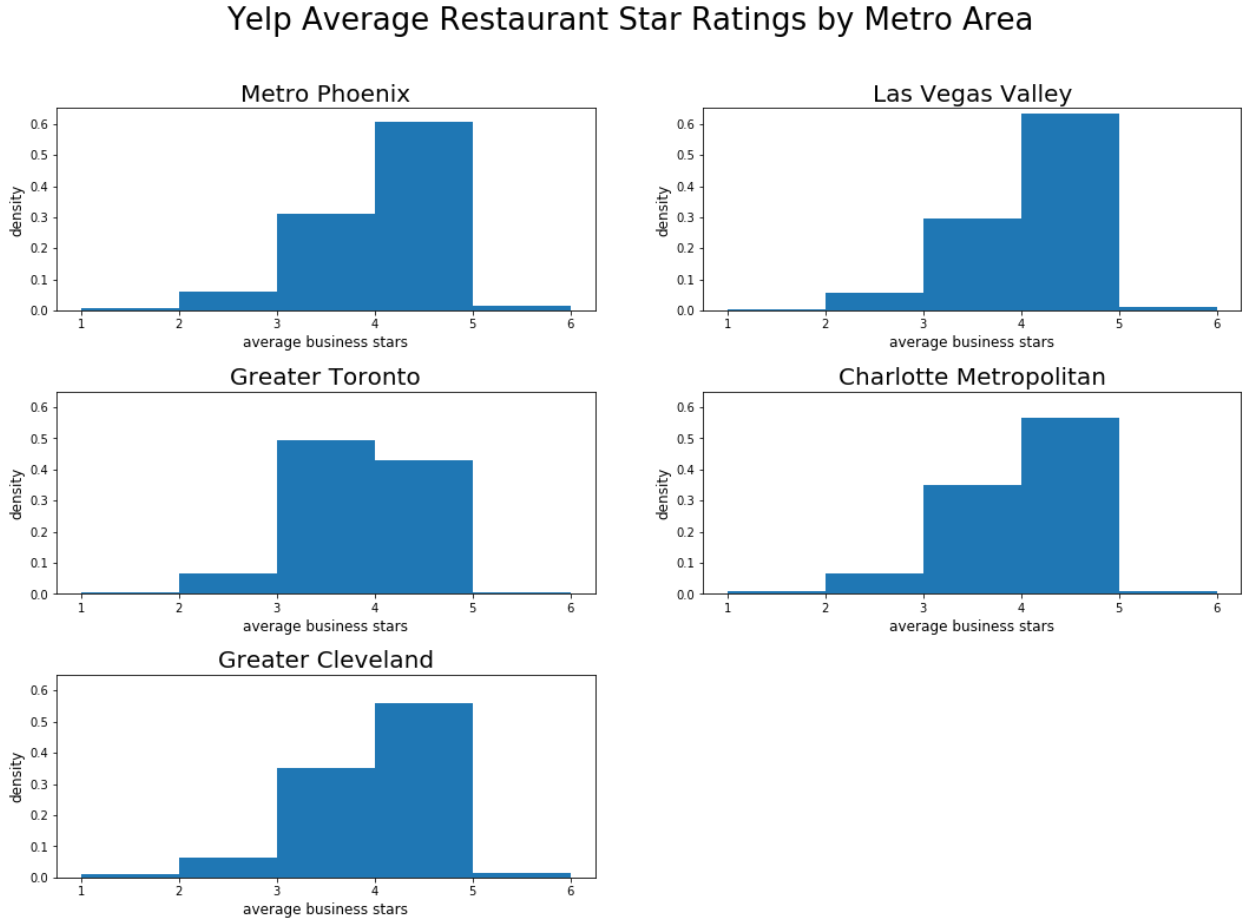

Figure 3.

The top 10 restaurant categories are similar for each of the metropolitan areas although Greater Toronto show higher counts categorized as Asian inspired (Japanese, Chinese, and Sushi Bars).

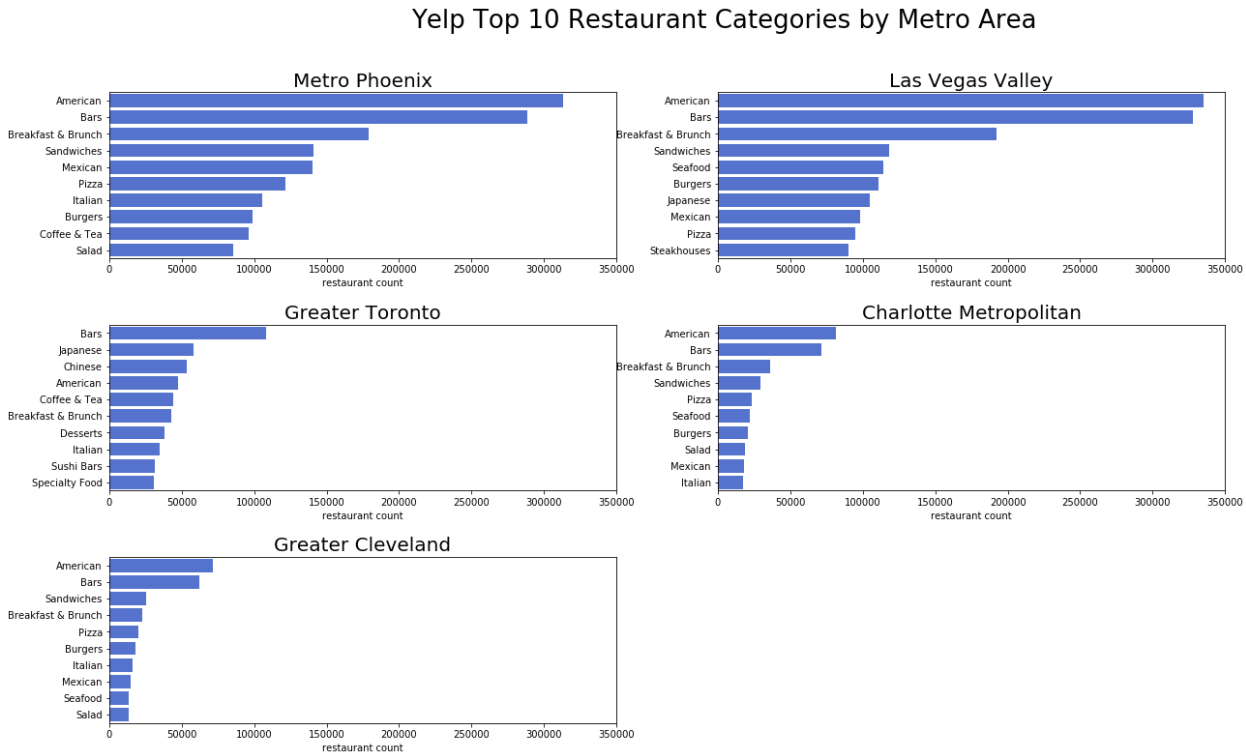

Figure 4.

Initially, on average, the star ratings start off relatively high and modestly decrease as the rating counts go up. However, after approximately having 5-15 reviews the star rating starts to increase.

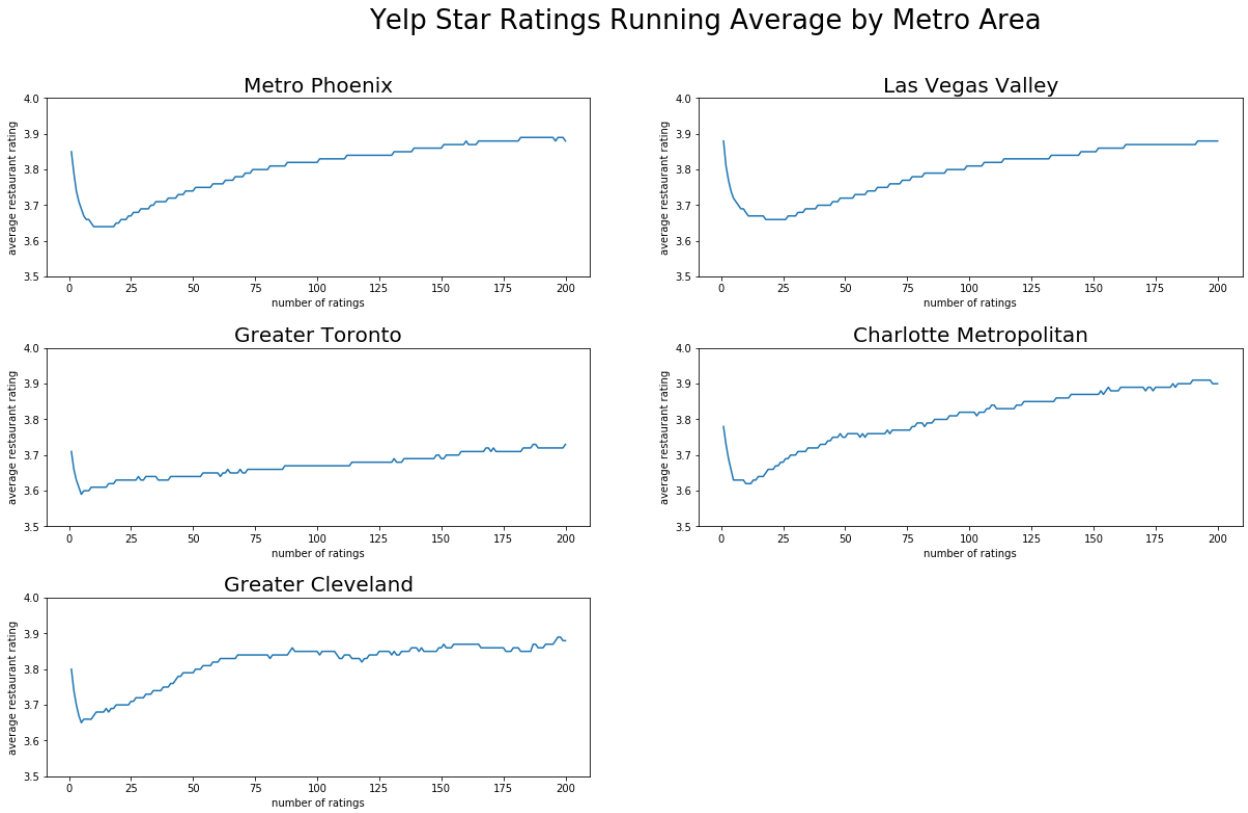

Figure 5. **NOTE**: the y scale does NOT begin at 0, but instead begins at 3.5.