Restaurant Collaborative Recommendation System

Immanuel Umenei

https://github.com/Mokonzi4n/Capstone_Project_2

Project Motivation

- *In an era where we are inundated with a lot of choices and information, filtering and sorting through to what is relevant and useful to a particular objective is usually time consuming and tasking. Having a system that narrows down the choices to only very applicable and related information provides us with:*
  - *More time for other tasks*
  - *improved productivity*
  - *opportunity to learn, appreciate and enjoy new aspects of a particular objective*

Problem

- *On Business trips, family vacations or city explorations, knowing where to eat what you consider good food is always a challenge in especially big cities. There are usually a myriad of choices to pick from. It is always easier if someone recommended a restaurant they have eaten in before and liked.*

- *The objective is to have a recommendation system that takes into consideration restaurants you would like, recommended by other users who have the same taste.*

Problem

- *Consider a set of restaurants with business ids, biz_id_1, biz_id_2, ..., biz_id_6, and users with user ids, usr_id_1, usr_id_2, ..., usr_id_10, the recommendation problem reduces to replacing the question mark entries in the ratings matrix below.*
- *We want to be able to replace the "?" marks with ratings based on what other users have given as ratings.*

user_id\business_id	biz_id_1	biz_id_2	biz_id_3	biz_id_4	biz_id_5	biz_id_6
usr_id_1	?	?	3	?	?	5
usr_id_2	2	?	4	4	5	3
usr_id_3	5	4	?	?	4	?
usr_id_4	5	1	5	?	?	?
usr_id_5	?	?	?	?	2	4
usr_id_6	?	5	4	?	5	3
usr_id_7	2	?	1	5	?	?
usr_id_8	5	3	4	3	2	5
usr_id_9	?	?	?	4	2	1
usr_id_10	4	5	4	?	?	3

Approach

- *Pandas is used to load, wrangle, analyze and model the data*
- *The data used is filtered from four main yelp data set files*
- *The python natural language processing library nltk is used to filter restaurant businesses based on their category string*
- *sklearn.model_selection library is used to split the data into a training set which is used to model the data a test set used to evaluate the data.*
- *The Pearson correlation is used as the similarity metric to compare reviewers.*
- *The performance metric used is the Root Mean Square Error (RMSE) which makes use of the mean_squared_error function from the sklearn.metrics library*
- *The IMDb formula is introduced to score the restaurants by popularity and overall average rating.*
- *An average of the predicted rating and score are taken into consideration to recommend a restaurant*

Data set

- *The Yelp data set which has 1.637M users with 6.685M reviews for 192,609 businesses of all types in 10 metropolitan areas is used.*

- *The data used is got by merging filtered information from the following 4 yelp data set files; yelp_business.csv, yelp_business_attributes.csv, yelp_user.csv and yelp_review.csv.*

Wrangling the Data

- *The yelp_business.csv and yelp_business_attributes.csv are loaded in pandas and the nltk library used on the "categories" column of the yelp_business file to determine which businesses were restaurant by searching for the word restaurant or food.*
- *States are examined and filtered only to the US and subsequently to Illinois IL. The two files are then merged and all 'na' entries converted to 'nans'*

```
1 business_attributes = pd.read_csv('yelp_business_attributes.csv')
2 yelp_business = pd.read_csv('yelp_business.csv')
```

```
1 yelp_business.columns
```

```
Index(['business_id', 'name', 'neighborhood', 'address', 'city', 'state',
       'postal_code', 'latitude', 'longitude', 'stars', 'review_count',
       'is_open', 'categories'],
      dtype='object')
```

```
1 yelp_business.categories[0]
```

```
'Dentists;General Dentistry;Health & Medical;Oral Surgeons;Cosmetic Dentists;Orthodonti
sts'
```

```
1 print(len(restaurant_ids))
2 restaurant_ids[:5]
```

```
69079

['PfOCPjBrlQAnz__NXj9h_w',
 'o9eMRCWt5PkpLDE0gOPtcQ',
 'EsMcGiZaQuG1OOvL9iUFug',
 'XOSRcvtaKc_Q5H1SAzN20A',
 'xcgFnd-MwkZeO5G2HQ0gAQ']
```

```
restaurants_IL_atrbt.head(2)
```

city	state	postal_code	latitude	longitude	stars	...	Corkage	DietaryRestrictions_dairy-free	DietaryRestrictions_gluten-free	DietaryRestrictions_vegan	Dietar
Champaign	IL	61820	40.109986	-88.233777	3.0	...	Na	Na	Na	Na	
Champaign	IL	61820	40.110085	-88.229304	4.0	...	Na	Na	Na	Na	

```
restaurants_IL.head(2)
```

city	state	postal_code	latitude	longitude	stars	review_count	...	DriveThru	DogsAllowed	BYOB	DietaryRestrictions_gluten-free	DietaryRestrictions_ve
Champaign	IL	61820	40.109986	-88.233777	3.0	4	...	NaN	NaN	NaN	NaN	
Champaign	IL	61820	40.110085	-88.229304	4.0	109	...	NaN	NaN	NaN	NaN	

Wrangling the Data

- *Load relevant user information from the 1.6GB yelp_user.csv file in chunks of 10000 rows and also load from the 3.79GB yelp_reviews.csv file, reviews of users for restaurants in IL only.*
- *Check for missing data, rename columns with same name, replace some entries with wrongly spelled city name from Mohamet to Mahomet, fill up entries with missing 61801 postal codes.*
- *Merge all data frames filtered from the 4 four yelp files and save it as restaurants_all_info_IL.csv*

```
user_info.head()
```

	user_id	name	review_count	average_stars
0	JJ-aSuM4pCFPdkfoZ34q0Q	Chris	10	3.70
1	uUzsFQn_6cXDh6rPNGbIFA	Tiffy	1	2.00
2	mBneaEEH5EMyxaVyqS-72A	Mark	6	4.67

```
restau_reviews_IL.head(2)
```

	review_id	user_id	business_id	stars	text
1470	dfN6CDt6GVSOQjq6u8lYlw	4hnBlZWXN7fWoaP1HHNfgA	CpNMXASiwtJv5eCDf0n63g	4	Solid steakhouse. Great atmosphere, can cook y...
2095	aM9YAAnEy0g-htXRWzQtOg	PVyZXgOkVtnU6966FDFhuw	FTky74MxFIMvAJepeUUzEQ	3	The price is right and so is the location. Coo...

```
# Determine if all the restaurant/eatery ratings are not having missing values.

restau_reviews_IL.stars.isnull().sum()
```

```
0
```

```
#Rename rating in restaurant review to user_restau_rating
restau_reviewz_IL = restau_reviews_IL.rename(index=str, columns={'stars':'user_restau_rating'})

restau_reviewz_IL.columns
```

```
Index(['review_id', 'user_id', 'business_id', 'user_restau_rating', 'text'], dtype='object')
```

```
#Rename review_count in user data to user_review_count
user_data = user_info.rename(index=str, columns={'review_count':'user_review_count',\
                                                 'name':'user_name','average_stars':'user_avg_stars'})
user_data.columns
```

```
Index(['user_id', 'user_name', 'user_review_count', 'user_avg_stars'], dtype='object')
```

```
#Rename review_count in restaurantz_IL to restau_review_count
restaurantsz_IL = restaurantz_IL.rename(index=str, columns={'review_count':'restau_review_count',\
                                                            'stars':'restau_rating', 'name':'restau_name'})
restaurantsz_IL.columns
```

```
Index(['business_id', 'restau_name', 'city', 'state', 'postal_code',
       'restau_rating', 'restau_review_count', 'categories'],
      dtype='object')
```

```
# Determine if the postal code for Mahomet and Mohamet are the same.
# If true then it is the same city.

restaurants_IL[restaurants_IL.city=='Mahomet'].postal_code.unique() ==\
restaurants_IL[restaurants_IL.city=='Mohamet'].postal_code.unique()
```

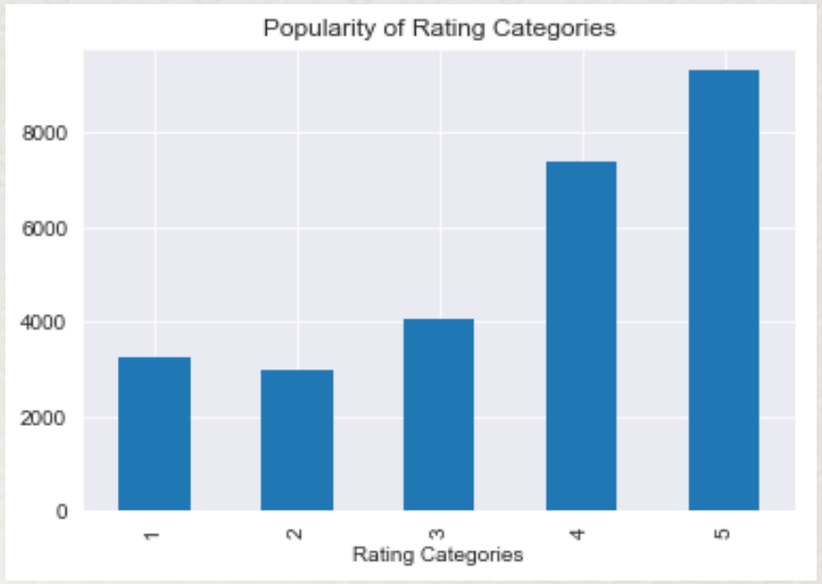
```
array([ True])
```

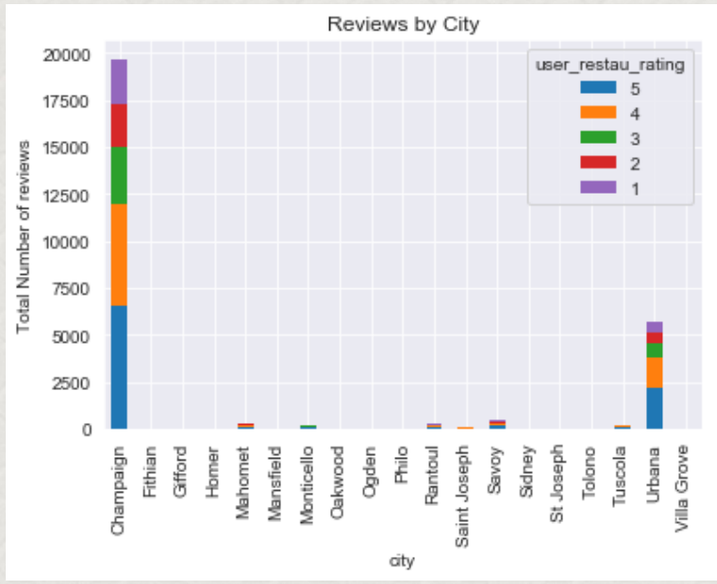
```
# Replace Mohamet with Mahomet and count the number of reviews. It is 302 from 297
restaurants_data = restaurants_data.replace('Mohamet','Mahomet')
restaurants_data[restaurants_data.city=='Mahomet'].city.count()
```

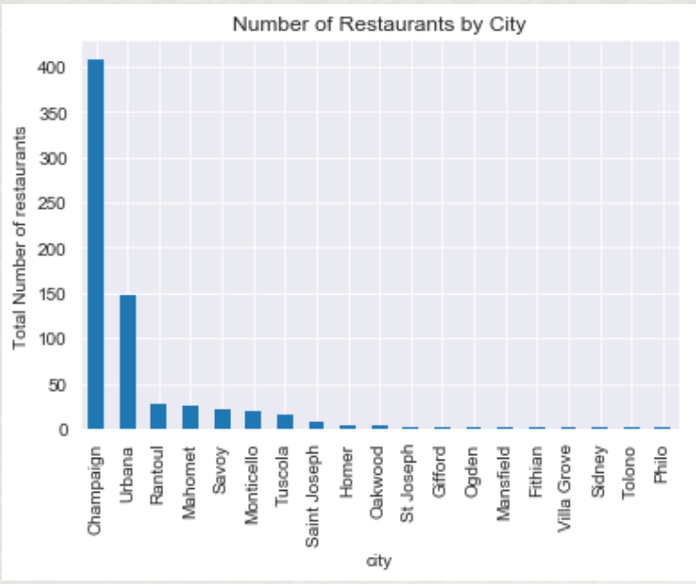
```
302
```

Data Story Telling

- *Ratings: Most popular ratings are 5 stars and least popular are 2 stars. Implying people would rather give a 1 when the think a restaurant is subpar than give a 2.*
- *Reviews: The highest number of reviews by city are for restaurants in the city of Champaign because it has the highest number of restaurants.And the most reviewed of the restaurants is found in Urbana as shown below*





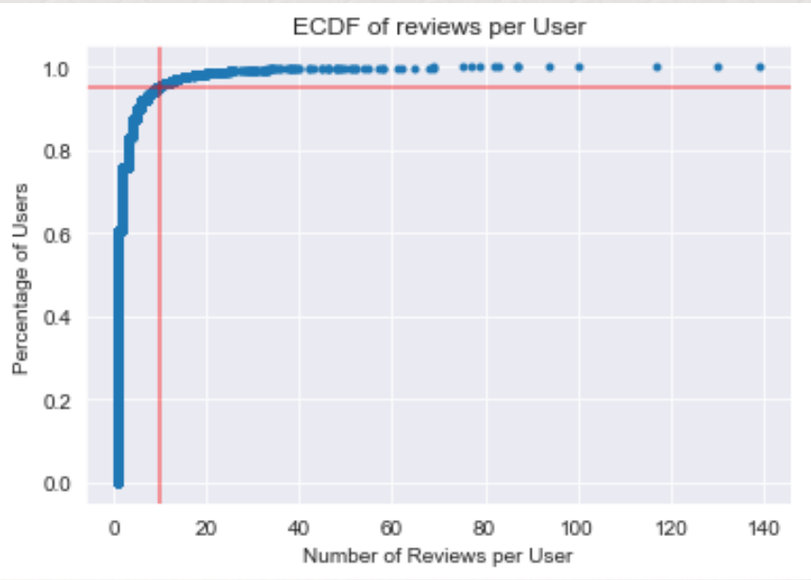


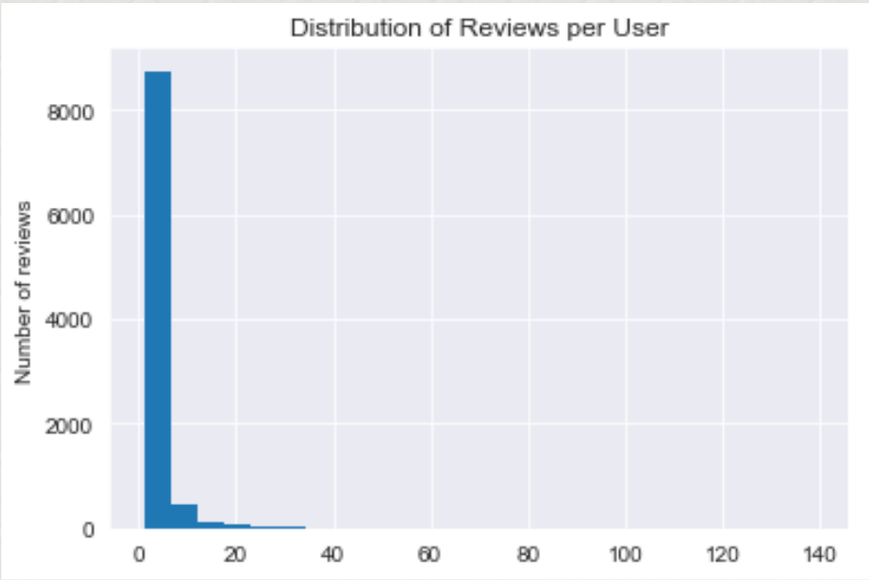
Exploratory Data Analysis

- *User review activity: 95% of the users are having about 10 or less reviews while 5% have between 10 and 140 reviews. This gives a right-skewed distribution of number of reviews per user with a mean of approximately 3.0 per user and median of 1.0 giving a steep sloped (almost right angled) ECDF.*

- *This is an indication that most of the users are not very active reviewers*

- *The user with the highest amount of activity is Nelson.*



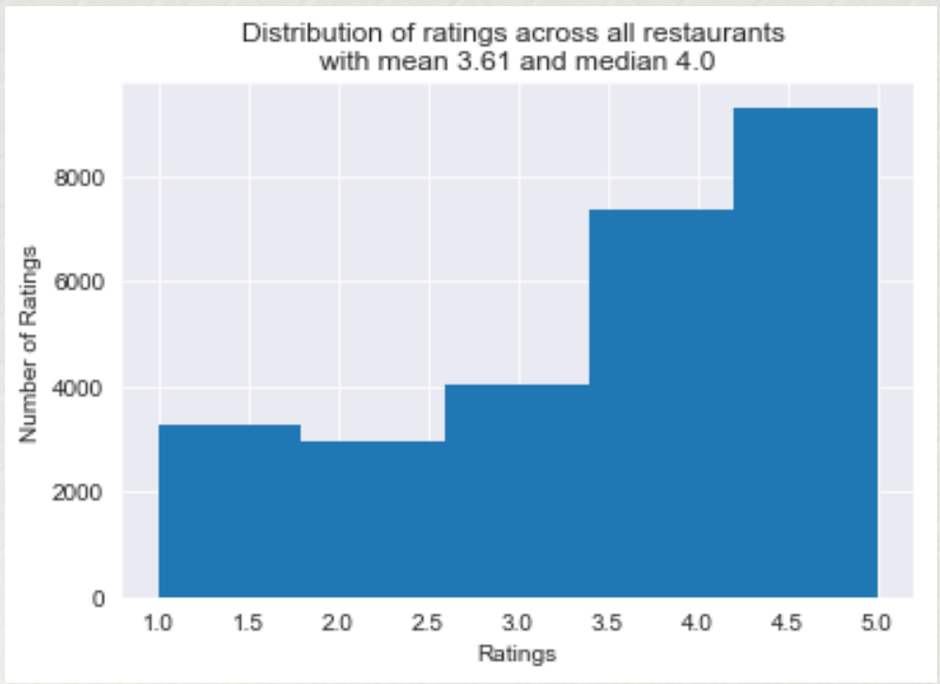


user_name	
Nelson	139
Lisa	130
Mark	117
Ashley	100
Teej	94
Daniella	87
A	87
Alex	83
Jamie	82
Kent K.	79
Jay	77
Tim	75
Anne	69
Sarah	69
Melissa	68
Jessica	65
Natalie	62
Yum	61
Sherry	58
Seth	58

count	9496.000000
mean	2.838880
std	5.840909
min	1.000000
25%	1.000000
50%	1.000000
75%	2.000000
max	139.000000

Exploratory Data Analysis

- *A distribution of the ratings is left-skewed with a majority of the ratings being 3 to 5 as earlier indicated and a mean of 3.61 and median of 4.0. This indicates that most of the reviewers would give a positive review to the different restaurants*



count	26977.000000
mean	3.612781
std	1.368265
min	1.000000
25%	3.000000
50%	4.000000
75%	5.000000
max	5.000000

Ratings Recommendation Process

- *The ratings are 'recommended' (filled) by considering the ratings given by another user or a combination of other users.*
- *For a ratings prediction to be close to that which the user in question would give, the other user or users should be very similar in their ratings profile of restaurants that this user has visited*
- *If we want to determine the rating of biz_id_5 by usr_id_10. The most similar user is usr_id_6. The system would recommend a rating of 5 stars for biz_id_5 by usr_id_10. Similarly using usr_id_2, a 4 stars rating for biz_id_4 is recommended for usr_id_6.*

user_id\business_id	biz_id_1	biz_id_2	biz_id_3	biz_id_4	biz_id_5	biz_id_6
usr_id_1	?	?	3	?	?	5
usr_id_2	2	?	4	4	5	3
usr_id_3	5	4	?	?	4	?
usr_id_4	5	1	5	?	?	?
usr_id_5	?	?	?	?	2	4
usr_id_6	?	5	4	?	5	3
usr_id_7	2	?	1	5	?	?
usr_id_8	5	3	4	3	2	5
usr_id_9	?	?	?	4	2	1
usr_id_10	4	5	4	?	?	3

Model

- *A class object called RatingReco is created. It is composed of two methods "learn" and "estimate" which take advantage of Pandas' data munging and modeling capabilities.*
- *The "learn" method creates a pivot table that transforms the data into a very sparse ratings matrix which is used by the "estimate" method to predict or estimate ratings. The ratings matrix is shown below*
- *The pearson function is called within the estimate method as a similarity metric based on how correlated two user rating profiles are for all the restaurants in the data set*

```
The shape of the ratings matrix is: (763, 9512)
```

user_id business_id	-06IbFmaohdrg7kQKc3A4A	-0e6xyw_4zyg-2YtqSlS_g	-0xEqfbgJFmbXh53qSqEww	-3agoL-p87vZteiDzrz5og	-4GjoEvMHZlG8DQWy8xqtA	-5HYPAQFgtX4
-05uZNVbb8DhFweTEOoDVg	NaN	NaN	NaN	NaN	NaN	
-2q4dnUw0gGJniGW2aPamQ	NaN	NaN	NaN	NaN	NaN	
-5NXoZeGBdx3Bdk70tuyCw	NaN	NaN	NaN	NaN	NaN	
-5dd-RjojGVK9hjAMCXVZw	NaN	NaN	NaN	NaN	NaN	
-7PuYohz9dR80iGfVR_kLA	NaN	NaN	NaN	NaN	NaN	
-865Ps6xb3h1LP67JcQ3mA	NaN	NaN	NaN	NaN	NaN	
-A4suUjxa7gNaiUMDjO42g	NaN	NaN	NaN	NaN	NaN	
-B1en4UZJzJEBiFjp1OJSQ	NaN	NaN	NaN	NaN	NaN	
-Jhlh8Scjy669NdtCfKSSg	NaN	NaN	NaN	NaN	NaN	
-LfTBo0oa_uD454ScEW2XA	NaN	NaN	NaN	NaN	NaN	

10 rows × 9512 columns

Model Validation and Evaluation

- *We split the data using the train_test_split function from the sklearn.model_selection library into a training and test data set in a ratio 4:1 respectively*
- *The training dataset is used to "train" the model after which it is applied to the test data set*
- *The performance of the recommendation is measured by the RMSE of the model defined by the function "compute_rmse", which makes use of the "mean_square_error" function from the "sklearn.metrics" library.*
- *Applying the model to the test data we have an* ***RMSE = 1.05****. Which indicates on average the recommendation system is approximately 1 star off per predicted rating as shown below:*

```
%time print ('RMSE for RatingsReco: %s' % evaluate(reco.estimate))
```

```
RMSE for RatingsReco: 1.0477057896764816
CPU times: user 13min 32s, sys: 7.63 s, total: 13min 40s
Wall time: 13min 56s
```

	user_name	restau_name	user_restau_rating	pred_rating
0	Sheryl	"Alexander's Steakhouse"	4	3.0
1	Michael	"Alexander's Steakhouse"	2	2.0
2	Taffy	"Alexander's Steakhouse"	5	4.0
3	Zewditu	"Alexander's Steakhouse"	4	4.0
4	Jake	"Alexander's Steakhouse"	5	3.0
5	Melissa	"Alexander's Steakhouse"	2	3.0
6	Harry	"Alexander's Steakhouse"	5	3.0
7	Herb	"Alexander's Steakhouse"	2	2.0
8	Jay	"Alexander's Steakhouse"	4	5.0
9	Chad	"Alexander's Steakhouse"	5	3.0

Score Metric

- *Considering ratings only a 5 star rated restaurant reviewed by one person and a 5 star restaurant rated by say 200 people could be considered the same. However one is more popular than the other.*
- *To also take popularity into consideration the weighted rating (score) calculated by the function "weighted_rating" (based on the IMDb formula) shown below is added as column "score" to the data frame*

$$score = (\frac{v}{v+m}R) + (\frac{m}{v+m}C)$$

Where:

- v is the number of restaurant reviews;
- m is the minimum number of reviews required to be recommended;
- R is the average rating of the restaurant
- C is the mean rating across the whole report

restau_name	restau_review_count	score
"El Oasis"	77.0	4.947979
"Old Time Meat & Deli Shoppe"	45.0	4.913299
"Caribbean Grill Restaurant"	41.0	4.905417
"Fernando's Food"	29.0	4.869948
"Krannert Center for the Performing Arts"	27.0	4.861278
"Prairie Fruits Farm & Creamery"	16.0	4.780965
"Fresh International Market"	16.0	4.780965
"Salad Meister"	13.0	4.739896
"Wines At the Pines"	12.0	4.722556
"Haymakers"	10.0	4.679873
"Natural Gourmet"	8.0	4.621668
"Breaking Taco"	6.0	4.537594
"Bent Bean"	6.0	4.537594
"Maize Mexican Grill"	470.0	4.494373
"Golden Harbor Authentic Chinese Cuisine"	362.0	4.492708
"Sakanaya Restaurant"	314.0	4.491604
"Huaraches Moroleon"	159.0	4.483570
"Homer Soda Company"	5.0	4.479793
"Cakes By Lori"	5.0	4.479793
"Pickwick Coffee Roasting"	5.0	4.479793

City Restaurant Recommendation

- *A function "restau_recommend" is defined which takes as inputs a "user_id" and "city" and calls reco.estimate an instantiated object of the RatingRepo class. It recommends a list of restaurants based on a rating that is the average of a 5 stars predicted rating and the score.*

- *As shown below for the city of Champaign Amanda has 6 more recommendations than J. and Mindy*

	USER NAME	CITY	RESTAURANT NAME
0	J.	Champaign	"Aldi"
1	J.	Champaign	"Cafe Sababa"
2	J.	Champaign	"Leadbelly's"
3	J.	Champaign	"Edible Arrangements"
4	J.	Champaign	"Teamoji"
5	J.	Champaign	"Kung Fu Tea"
6	J.	Champaign	"Grovestone"
7	J.	Champaign	"Wedding Cakes by Rosie"

	USER NAME	CITY	RESTAURANT NAME
0	Mindy	Champaign	"Aldi"
1	Mindy	Champaign	"Cafe Sababa"
2	Mindy	Champaign	"Leadbelly's"
3	Mindy	Champaign	"Edible Arrangements"
4	Mindy	Champaign	"Teamoji"
5	Mindy	Champaign	"Kung Fu Tea"
6	Mindy	Champaign	"Grovestone"
7	Mindy	Champaign	"Wedding Cakes by Rosie"

	USER NAME	CITY	RESTAURANT NAME
0	Amanda	Champaign	"Fresh International Market"
1	Amanda	Champaign	"Aldi"
2	Amanda	Champaign	"Cafe Sababa"
3	Amanda	Champaign	"Wood N' Hog Barbecue"
4	Amanda	Champaign	"Leadbelly's"
5	Amanda	Champaign	"Edible Arrangements"
6	Amanda	Champaign	"Teamoji"
7	Amanda	Champaign	"Kung Fu Tea"
8	Amanda	Champaign	"Pie's the Limit"
9	Amanda	Champaign	"Binny's Beverage Depot"
10	Amanda	Champaign	"Grovestone"
11	Amanda	Champaign	"Wedding Cakes by Rosie"
12	Amanda	Champaign	"Crave Truck"
13	Amanda	Champaign	"Chester's BBQ"

Conclusion

- *This recommendation system provides a user with a list of restaurants in a city that other users similar in taste and judgement to this specific user would recommend to him.*
- *A specific case of 19 Illinois cities extracted from the yelp data set was used. This could be scaled to different cities nationally*
- *Such a system is useful in recommending restaurants to a user when he gets into a new city not based on distance but on its popularity and rating by other users similar to him/her*

Recommendations
(no pun intended)

- *This analysis was purely collaborative using explicit data (ratings) and as such would not offer any recommendations in a restaurant cold start problem. In that case, a content based recommendation system which looks at the similarity in the attributes of one restaurant compared to another reviewed by the user would be a solution to a more personalized rating. Using the business attributes as independent variables and the rating as the dependent variable a Regression model could be implemented.*

- *A user cold start problem would be solved by considering additional meta data of the user like age, sex, race, dietary restrictions or inclinations , etc., which are currently not available in the yelp dataset. This additional information could improve the accuracy of a Machine Learning algorithm in recommending a restaurant*