Northeastern University
CS-6220 – Data Mining Techniques
Fall 2017, Praveen Singh, Anupam Sapre, Kuai Hu, Raghav Sairam

Project Report

1. Introduction

The project aims at implementing multiple data mining techniques on the Yelp academic dataset (https://www.yelp.com/dataset) and thus answer the following questions.

1. Suggest additional services for business based on user review patterns.
2. Estimate the rating given by an individual to a previously unvisited business.
3. Suggest businesses to a person based on their review patterns.
4. Predicting the geographic location of a user based on review patterns.

These different questions implement completely different algorithms and thus require the dataset to be pruned/cleaned accordingly (depending on how reactive the algorithm is to stray data and overfitting).
While the first 3 questions tackle the exploratory process in a similar fashion, the last question uses a relatively independent exploratory.
A basic description of the implementation of each question is as follows. Each implementation will be discussed in detail in this section.

- **Suggest friends for users with similar interests in category of business**

An index of how similar two people are, can be obtained by how they rate the same business. Based on the range of this index, a threshold is obtained below which two users are viable friends and thus suggested.This will be solved using content based recommendation by creating cosine matrix for user profiles.

- **Estimate the rating given by an individual to a previously unvisited business.**

The opinion of a user on a particular business is obtained by aggregating the opinions of similar users (have similar ratings for common business) and thus allows the recommender system to predict suggestions.
The underlying assumption for this question is that, people with similar opinions of particular business would share this similarity for other businesses. Such an assumption gives reason to implement collaborative filtering.

- **Suggest businesses to a person based on their review patterns.**

This problem is solved using content based recommendation. There are two approaches to this
1) Generate cosine matrix with user profiles and business profile
2) Generate cosine matrix with business profiles only & find similar businesses. Suggest businesses which are similar to businesses user has liked.

- **Predict the geographic location of a user based on review patterns.**

The dataset describes the location of individual businesses. Odds are, a sedentary user (one that isn't likely to travel) has suggestions localised to a particular region/city.

This problem involves predicting the geographical location of each user based on their reviews and could play a vital role in helping recommender systems of other problems provide a more localized suggestion of businesses as opposed to the entire scope (the domain of possible cities in the Yelp dataset spans across the USA).

2. Exploratory analysis

Data in user.json has fields of ‘user_id’, ‘name’, friends’,...
Data in business.json has fields of ‘business_id’, ‘name’, ‘city’, ‘latitude’, ‘longitude’,...
Data in review.json has fields of ‘review_id’, ‘user_id’, ‘business_id’, ‘start’,...

- **Suggest additional services a business might provide to attract more customers, based on user reviews on business**

We looked at reviews on business with similar categories in the dataset and saw similar reviews on them from almost the same group of users while looking on business of a city. We also saw some reviews providing suggestions for other services to give 5 stars. We looked at the list of categories of the businesses that different users reviewed and we found that the users interest in the type of business was almost the same as is quite evident from the below list of categories of places that the two different users have reviewed.

G8YCTH5byJUaRTNshhCVuw : [['Venues & Event Spaces', 'Event Planning & Services', 'Hotels', 'Hotels & Travel'], ['Southern', 'American (Traditional)', 'Breakfast & Brunch', 'Restaurants'], ['Flowers & Gifts', 'Drugstores', 'Food', 'Shopping', 'Grocery'], ['American (Traditional)', 'Breakfast & Brunch', 'American (New)', 'Restaurants']]

oX5pvTPY8K20c0p7YRDnUQ : [['Gun/Rifle Ranges', 'Firearm Training', 'Guns & Ammo', 'Local Services', 'Shopping', 'Sporting Goods', 'Active Life', 'Education', 'Specialty Schools'], ['Shopping', 'Guns & Ammo', 'Pawn Shops'], ['Shopping', 'Guns & Ammo'], ['Food', 'Active Life', 'Farmers Market']]

This gave us a business idea that the user “G8YCTH5byJUaRTNshhCVuw” will definitely be interested in a business providing food and travel services and would be more happy if one business provided most of the services he is interested in, similarly the user “oX5pvTPY8K20c0p7YRDnUQ” would be more interested in business catering to hunting and farming weapons and tools and even happier if most of the services are provided by one business itself.
This was the idea behind finding an association rule between the categories of business, to suggest additional services to business to attract more customers.

- **Content Based Recommendation for finding similar businesses and suggesting new businesses**

The Yelp dataset is ideal to use content based recommendation system as we have enough information using text reviews and other attributes. There also information available in tips but we are not using them as there is insufficient data around them.
Yelp dataset provides reviews for businesses as well as the users who have provided them. We decided upon a threshold of businesses with reviews greater than 5 and less than 5000. The lower limit was required to ensure sufficient text data was available to create a profile for each business. The upper limit was decided to avoid issues due to higher dimensionality. For this project, we created

profiles for businesses only using the tf idf scores for the reviews corresponding to that business. We currently have following approach:

- Find a business for a particular city with review counts within limits
- Obtain all reviews for this business and the users who have reviewed them.
- Now use sklearn tf idf vectorizer and get tf idf (Term Frequency - Inverse Document Frequency) matrix for the business.
- Use SVD to reduce dimensionality of this matrix.
- Using cosine similarity on this matrix find similar businesses.

For suggesting businesses to users we follow following approaches:

- Find businesses similar to the ones the user has reviewed.
- Find cosine similarity matrix with users and business

The second approach currently proves an issue as the matrix generated is too large(memory issues). We are able to use that approach on small towns which are our target locations.

Validation:

- We are using a method of finding the Root Mean Square Error on ratings of business which are found similar by the algorithm. We are achieving a mean value of about 1. This value seems reasonable suggesting us that the restaurants similar differ by 1 ratings on average.
- Manual Validation was also done for businesses by checking if the categories of business suggested are similar. For example if we are finding restaurants similar to a Mexican restaurant it makes sense to have restaurants with Mexican cuisine to be similar.

- **User Location Inference**

The Yelp dataset does not contain private information other than user_id, name, and friends, but it contains users consumption record such as review_count, average_stars, and a number of compliments, etc. Based on these it is possible to mine more private user information. To do so, we connect users with their consumption record, such as user reviews and information of business they have ever visited.
To predict users permanent/temporary locations, we need the location information from business which have been connected to users by their reviews. There are two kinds of location information in business.json. A string of city and a geographical location(latitude and longitude). We want to predict the city/geographical location of each user.

- Analysis of user data

There are 1183362 users with unique user_id, which have no direct links between them with businesses. But we could construct these links by reviews record which will connect users and business. For this collection, we consider user_id, review_count, friends for further use. When we take review_count into consideration, we noticed that there is a large variance among users. The maximum value of review_count is 11656 while the minimum value of it is 0. In this way, for users with review_count close to 0, it is impossible or highly biased to predict their locations. We prune users with lesser number of reviews than a given threshold, since the aforementioned reasons causes very unconvincing location predictions. The selection of threshold is explained later in this section.

- Analysis of business data

There are 156639 business with unique business_id. Every business has its own 'city' and ('latitude','longitude') pair with valid values except one business. The review_count of businesses is irrelevant to our location prediction for now, since we only consider the business location for aggregation.

- Analysis of review data

There are 4736897 reviews with unique review_id. We regard reviews as connections between users and business via user_id and business_id. Our expected relationships after connection of user_id and business_id should be:

[{'Ha3iJu77CxlrFm-vQRs_8g': ['San Francisco', 'San Francisco', 'Las Vegas', …….]}, ……]

- Threshold

When we set the threshold as 5, there are 204748 users with 1023740 total reviews.
When we set the threshold as 10, there are 88635 users with 886350 reviews.
We analyzed and evaluated our work based on such two thresholds.

3. Data mining analysis

1) Suggest additional services a business might provide to attract more customers, based on user reviews on business

i) Preprocessing

The algorithm was implemented on a small subset of data for all the businesses localized to Pittsburgh(city).
We found all the businesses of the city Pittsburgh and looked at all the reviews that have been given to the businesses of the city Pittsburgh. Each review is made on a business and has the "business_id" attribute in the reviews data. Each business also contains a "categories" attribute, in the business.json file of the dataset. This attribute contains a list of all services that the business operates. For all the reviews we take the categories of the business to which the review was made and add append it to a list, which is fed as input itemset list of lists for finding patterns and association rules.
We wanted to suggest only the services which would be relevant and highly rated places. For this we have only considered the reviews that have a minimum level of "star" and considered the categories of that business. Also to apply FP-Growth a Minimum Support and Minimum Confidence needed to be decided. For our development we selected a fixed value of minimum support and minimum confidence to be applied after executing the algorithms multiple times and selected our minimum support as 10 and minimum confidence to be used as 0.9.

ii) Finding Rules for the user's choice of place by applying FP-Growth

To find association rules between business categories, we created a list of category lists of business who have been rated above 3 stars and applied the FP-Growth Algorithm on this frequent itemset. Instead of reinventing the wheel in implementing the FP-Growth algorithm, the "pyfpgrowth" (https://pypi.python.org/pypi/pyfpgrowth) package was used. We have considered the Minimum Support(S) was as 10 and Minimum Confidence(c) as 0.9 . Applying the algorithm on the created

frequent item itemset we get the patterns and rules: (samples small subset of the actual patterns and association rules)

2) Content Based Recommendation for finding similar businesses and suggesting new businesses

Content based recommendation works well based upon the RMSE graph. Also we did several runs on different cities which showed reasonable results. We were not able to run with entire tf idf matrix and had to resort to SVD to reduce dimensionality. We did some runs with different values of SVD to check if it is appropriate and whether it reduces accuracy extensively. Reducing the dimensions of matrix did not affect RMSE mean values a lot. For a matrix with 100 fields we had a mean value of RMSE of 0.89 while for a matrix with 10 fields RMSE mean value of 0.97 . SVD approach reduces the suggestion accuracy but provides results faster also with less memory issues.

3) Estimation of user ratings for unvisited businesses

i) Preprocessing

The number of reviews given to a business does not necessarily represent how good/bad it is. However, a general assumption can be made that a restaurant with lesser number of reviews is not very popular.

Exploratory analysis on the number of restaurants as a function of minimum number of reviews for that restaurant is made for different cities.

Based on these multiple plots, an ideal threshold for the minimum number of users per city is identified.

ii) Estimation of ratings

The dataset provides details of tips, businesses, reviews, users and images of food. Of the data provided, businesses, reviews and users are the data that provide information pertinent to the implemented algorithm.

The reviews contain multiple features (review_id, user_id, business_id, stars, date, text, useful, funny, 'cool), of which the features relevant to this module are user_id, business_id and stars (rating).

The recommender algorithm suggests businesses local to a single city, i.e. it recommends businesses to a user based on his location.

- ➢ The data aggregation process involves extracting all businesses for this city.
- ➢ Since reviews also comprise the business to which this review belongs, a list of reviews corresponding to this city is also extracted.
- ➢ This list of reviews generated is iterated through and the following data structure is obtained.

{ user_id1: { user_business_id1: rating
user_business_id2: rating },
user_id2: { user_business_id5: rating
user_business_id6: rating }}

This is similar to a sparse representation of data, where ratings of the businesses that the user has rated is stored. This might slightly increase transitionary calculations but greatly reduces the amount of memory consumed during program runtime.

The above steps of mining data from the original dataset results in content that is void of extraneous/irrelevant data.

4. Discussion

1) Suggest additional services a business might provide to attract more customers, based on user reviews on business

Rules generated are:

Patterns : *{('Computers',): 11,*
('Computers', 'Shopping'): 11,
('Outdoor Gear',): 11,
('Outdoor Gear', 'Sporting Goods'): 11,
('Outdoor Gear', 'Shopping'): 11,
('Outdoor Gear', 'Shopping', 'Sporting Goods'): 11,
('Chocolatiers & Shops',): 11,}

Association Rules: *('Books', 'Bookstores', 'Music & Video', 'Shopping') ---> ('Mags')*
('Beauty & Spas', 'Cosmetics & Beauty Supply', 'Eyelash Service', 'Hair Removal', 'Nail Salons', 'Waxing') ---> ('Day Spas', 'Skin Care')
('Fashion', "Men's Clothing", 'Shopping', 'Sporting Goods', 'Sports Wear', "Women's Clothing") ---> ('Accessories')
('Home Services', 'Local Services', 'Self Storage', 'Shopping') ---> ('Movers')
('Active Life', 'Arts & Entertainment', 'Dance Clubs', 'Dance Studios', 'Event Planning & Services', 'Fitness & Instruction', 'Nightlife', 'Party & Event Planning', 'Venues & Event Spaces') ---> ('Performing Arts',)

Business can look at the association rules and based on their current category, check what would be the best service the business can add to its list of service to attract more customers. It is self explanatory that is a business provides services involving Beauty & Spa, Cosmetics, Eyelash Service, Hair Removal and not providing Skin Care, that it would be even attract the customers who are interested in Skin Care and increase the customer attraction to the business.

2) Content Based Recommendation for finding similar businesses and suggesting new businesses

- Scalability issues : For small towns like Matthews,NC, Tempe the code works fine, but for cities with larger counts of data (Phoenix ,Las Vegas) we were unable to handle the size of the cosine matrix for users X users or users X businesses. Here the number of users is more than 150,000.We ca use MapReduce / Spark jobs to compute co-similarity matrices.
- Merging the Content based recommendation system with collaborative system. A hybrid recommendation system which takes into account scores from both systems. Hybrid systems always can be tuned.

3) Estimation of user ratings for unvisited businesses

The gist of the prediction process consists of finding users with rating patterns very similar to the current user and extrapolating on fields that have not yet been initialized for the user.

The detailed step by step implementation is as follows:

- Let the user for which prediction must be made be up and the user from dataset currently being considered be ui .
- The similarity between the user (up) and all other users (ui) in the dataset is generated using the following steps.
- The sparse data obtained for the current user and test user is converted into a vector with length equal to that of number of businesses. The jth index of the vector corresponds to the jth business in this city.
- The angle between the two vectors (cosine distance) is measured subject to the constraint that, the non-zero fields from (up) is used and the same fields are correspondingly used for (ui)
- A map of {user_ids:distance} is generated for this user.
- The similarities are arranged in decreasing order (since cosine value of 1 between two vectors implies they have the same direction).
- The highest k among these are used for the prediction step.
- Let vp,vi be vectors corresponding to up and ui respectively and let vij represent the jth value of vector vi. The prediction step is as follows:

$$v_{pj} = \sum v_{ij} * sim(u_i, u_p) \; / \sum_{where\ v_{ij} \neq 0} sim(u_i, u_p)$$

- Although this step appears to be tedious for each element, vector addition is easily implemented in python.

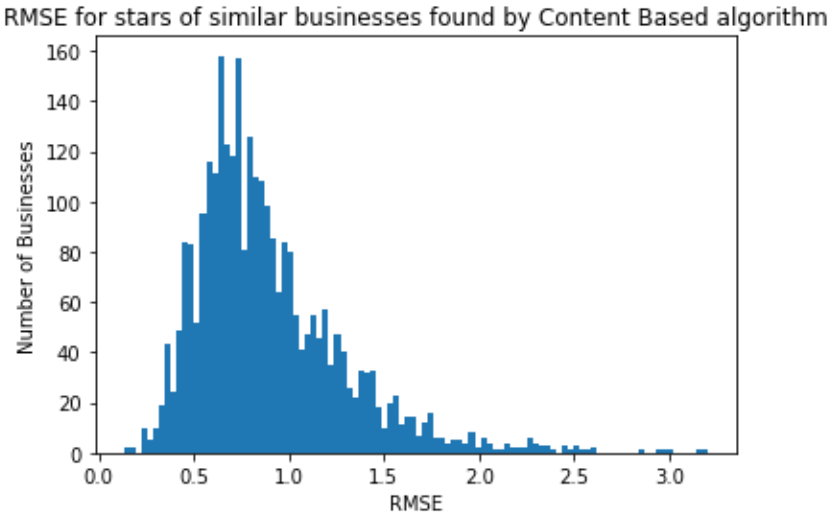
The debugging of this implementation and generating the mean squared error between predicted results and already assigned values generated the following plot and the generated prediction in ratings is used to recommend restaurants (in decreasing order of predicted ratings).

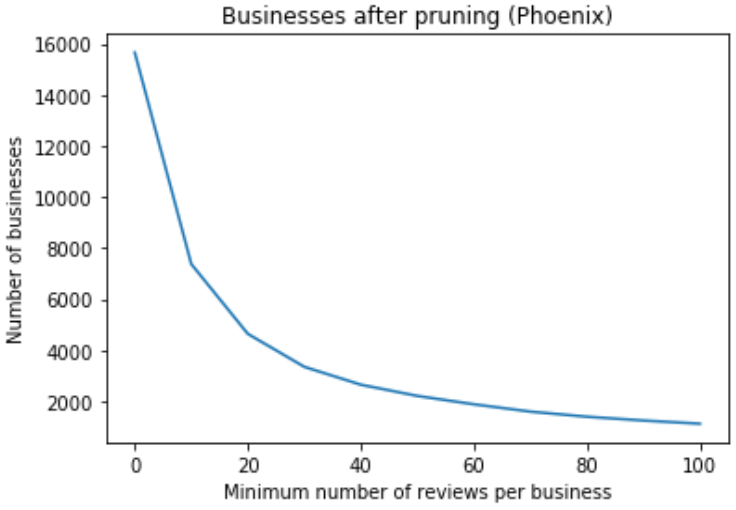
4) User Location prediction

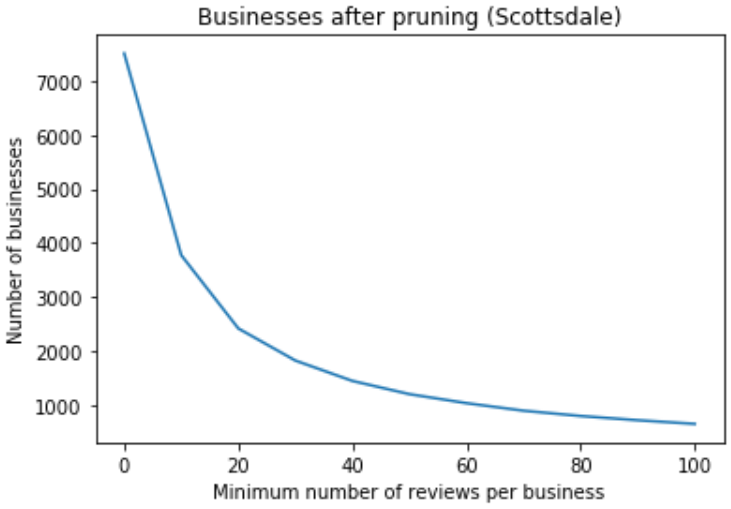
Firstly, we parse the business.json to extract the city and geographical information of each business. We could see most cities have less than 1000 business in our dataset and there are several cities with huge amount of business even as large as 10000. We could predict that the distribution of users predicted cities should be such biased as well.
Then we consider a user_id with no less than 10 reviews and its corresponding business_id from review.json. And we add business location/geographical information to users through business_id. We simply consider the most common cities among locations of business respectively to be the location/cities users live in.
This coincides with the distribution of businesses, the 10 cities with most business are almost the same in this predicted list, i.e. a higher number of businesses in a single location, suggests a higher user density. Mostly consistent with the distribution of business.
Another case we have to handle with is if users have relocated recently, we cannot consider all their consumption record. Thus, to avoid the case if users like traveling as well, we extract users' 15 most recent reviews and analyse based on that.
Since there is users network information in dataset and it seems like users should have a relative large number of local friends in their network, we take one more step to consider user network as well. Based on prediction we have for now, we apply another prediction by user network. Similar to

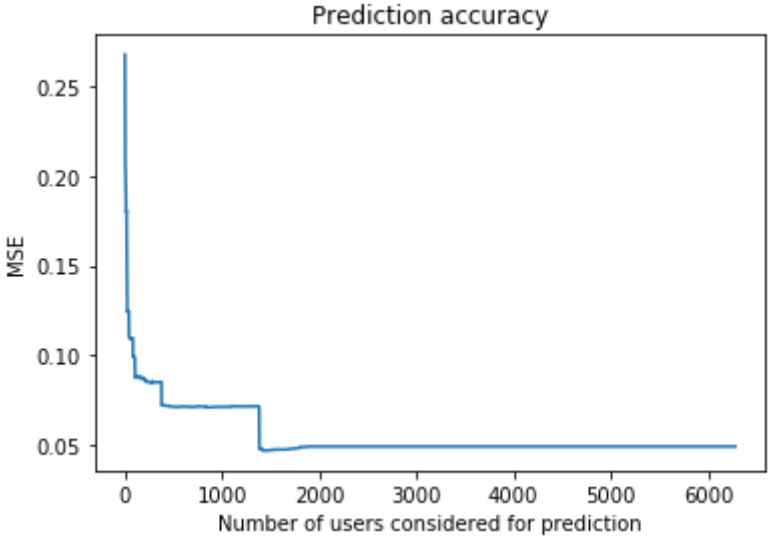
what we did for user-business, friends' cities contribute to the prediction of users respectively. The top 10 predicted cities with most users by network is:
Equivalent to our first prediction, i.e. most friends in users network are local friends and live in same cities as users.

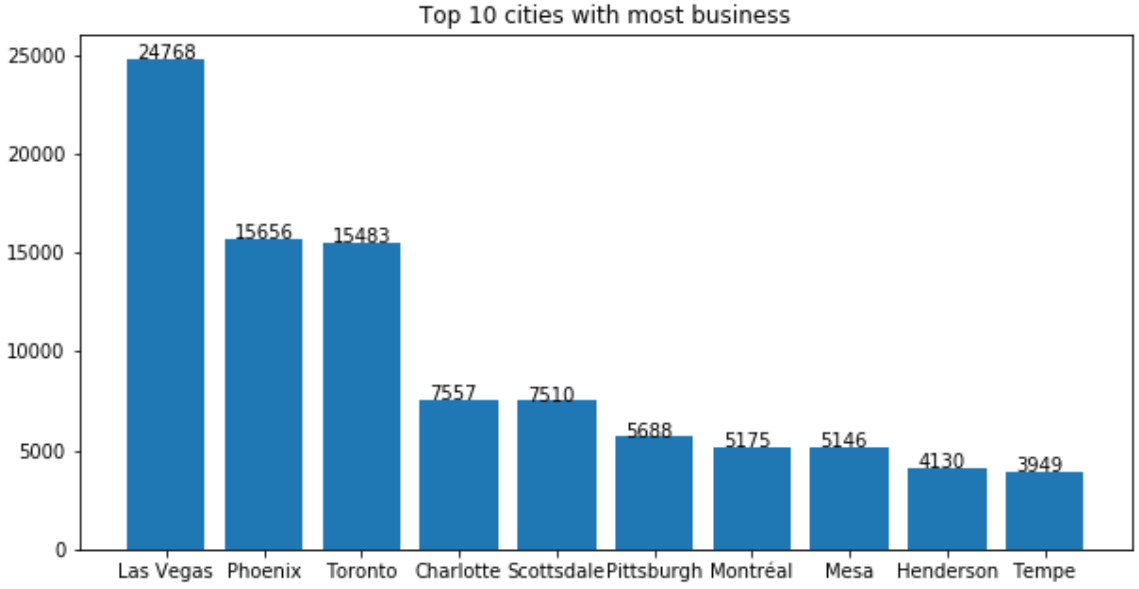
5. Appendix

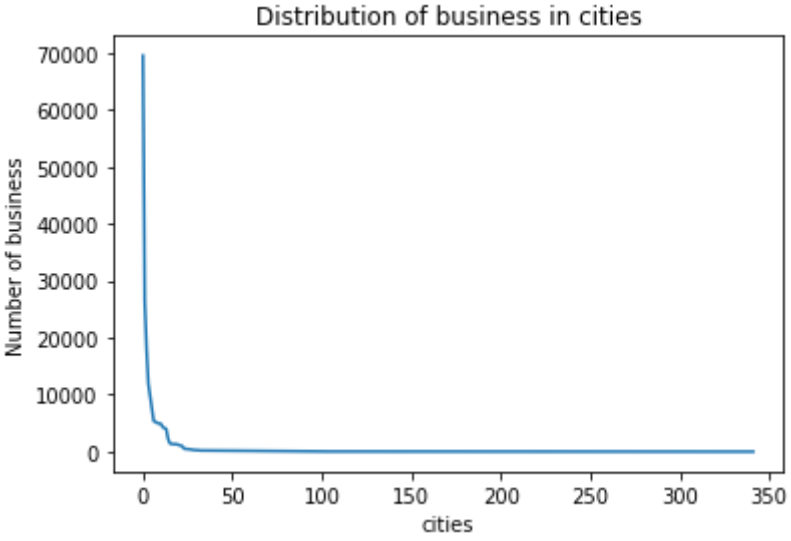


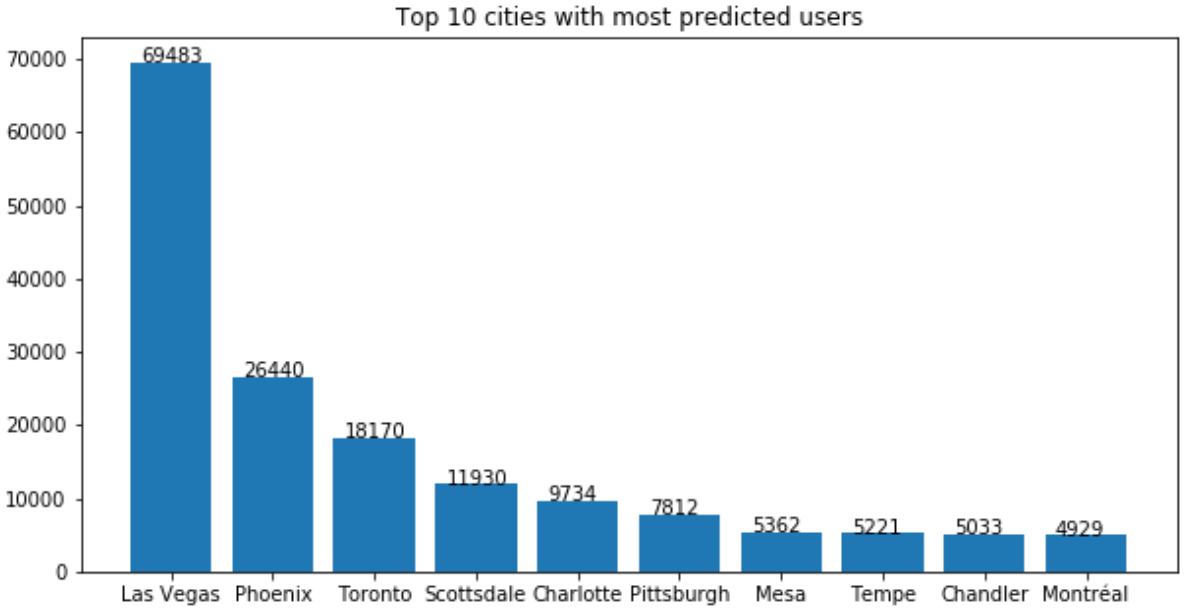


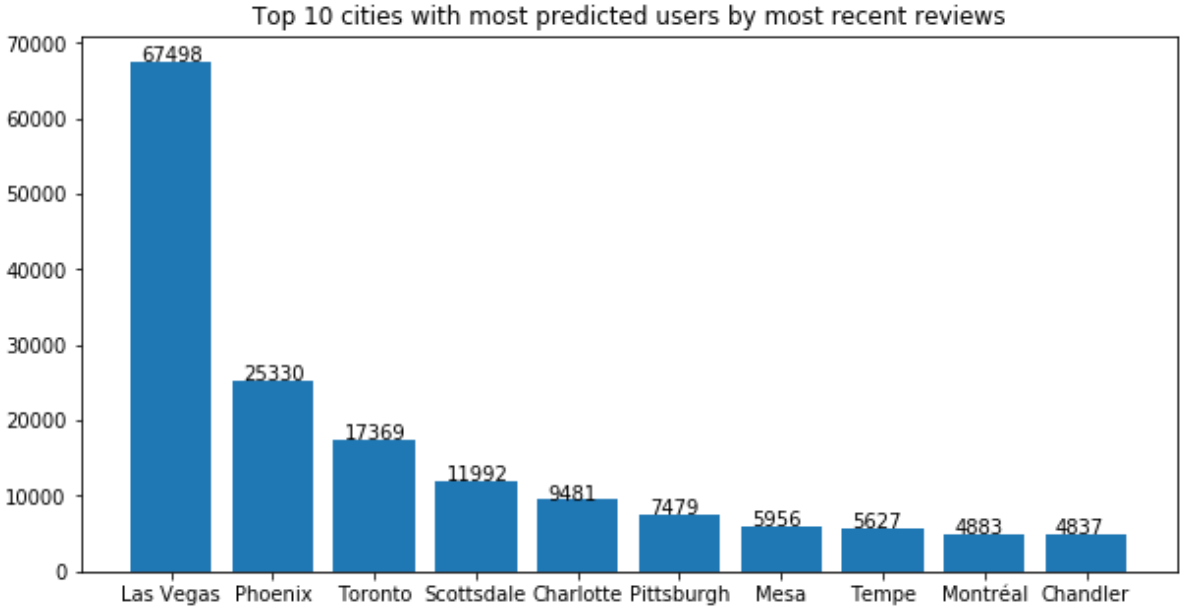


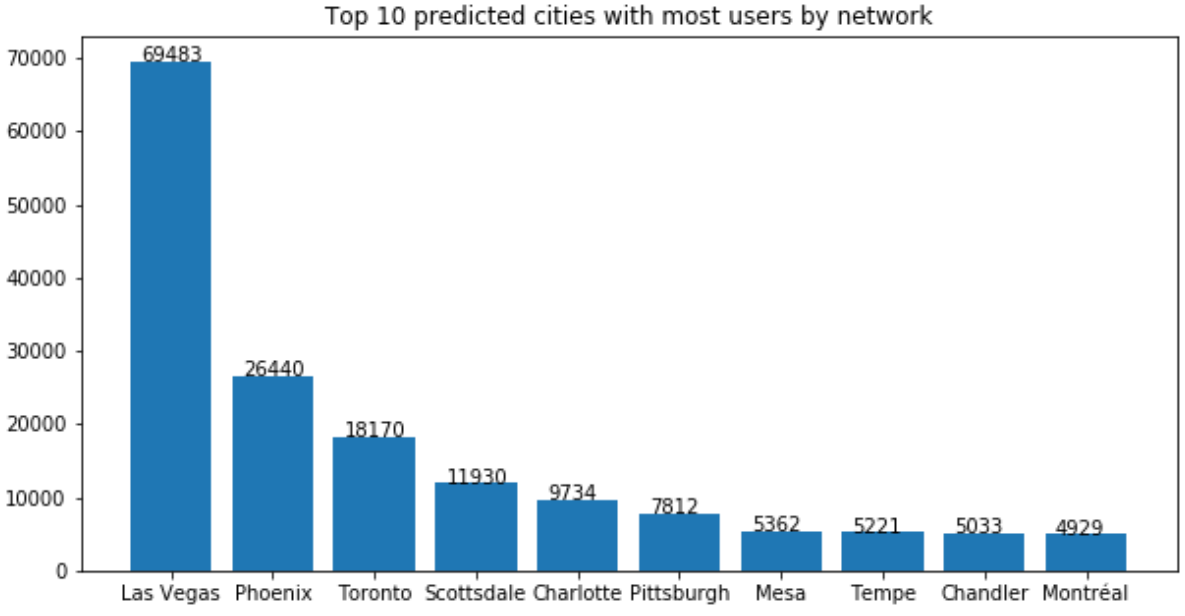

Prediction accuracy
MSE
0.25
0.20
0.15
0.10
0.05
0
1000
2000
3000
4000
5000
6000
Number of users considered for prediction


Top 10 cities with most business
25000
20000
15000
10000
5000
0
24768
15656
15483
7557
7510
5688
5175
5146
4130
3949
Las Vegas
Phoenix
Toronto
Charlotte
Scottsdale
Pittsburgh
Montréal
Mesa
Henderson
Tempe


Distribution of business in cities
Number of business
70000
60000
50000
40000
30000
20000
10000
0
0
50
100
150
200
250
300
350
cities


Top 10 cities with most predicted users
70000
60000
50000
40000
30000
20000
10000
0
69483
26440
18170
11930
9734
7812
5362
5221
5033
4929
Las Vegas
Phoenix
Toronto
Scottsdale
Charlotte
Pittsburgh
Mesa
Tempe
Chandler
Montréal


Top 10 cities with most predicted users by most recent reviews
70000
60000
50000
40000
30000
20000
10000
0
67498
25330
17369
11992
9481
7479
5956
5627
4883
4837
Las Vegas
Phoenix
Toronto
Scottsdale
Charlotte
Pittsburgh
Mesa
Tempe
Montréal
Chandler


Top 10 predicted cities with most users by network
70000
60000
50000
40000
30000
20000
10000
0
69483
26440
18170
11930
9734
7812
5362
5221
5033
4929
Las Vegas
Phoenix
Toronto
Scottsdale
Charlotte
Pittsburgh
Mesa
Tempe
Chandler
Montréal