Northeastern University

Data Mining Project Report (Draft)

Recommendation Systems for Yelp Dataset

Aditya Priyadarshi, Abhay Kasturia, Xingxing Liu, Varun Nandu and Gautam Vashisht

Supervised by
Dr. Nate Derbinsky

November 22, 2017

1 Introduction and Related Work

A recommender system or a recommendation system is a subclass of information filtering systems that seeks to predict the rating or preference that a user would give to an item [2]. Recommendations systems have become very relevant today given the presence of e-commerce website like Amazon and Netflix as well as other platforms like Facebook and Youtube. These are utilized in a variety of areas such as movies, music, videos, news, books, research articles, search queries and products in case of Amazon. Two most common methods to build a recommendation system are collaborative filtering and content-based filtering. Collaborative filtering methods use user's past behaviors and behaviors of similar users to find items which a user might like. Content-based methods use the features of the items liked by the user to suggest similar items. There are also hybrid recommendation system which combine both of these techniques.

2 Dataset and Analysis

2.1 Dataset

The original dataset described in the Yelp Dataset Challenge 10 [1] has 4.7M reviews and 1M tips by 1.1M users for 156K businesses spread across 12 cities. The data is given in json format which include business.json, review.json, user.json, checkin.json and tip.json.Each business has name, address, star rating and textual reviews. Each individual review data consists of anonymized IDs for the business, user and review, star rating, review type, review text and votes on how useful, funny or cool the review is.

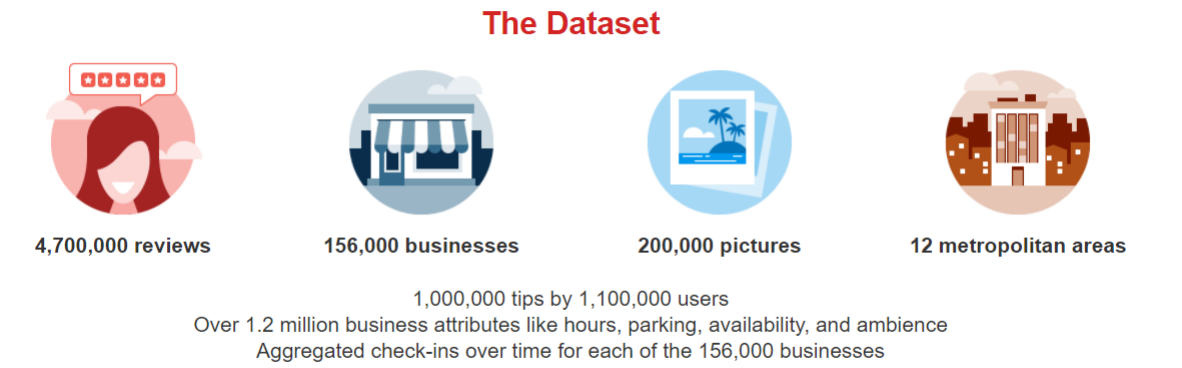

Figure 1: Dataset Details

2.2 Analysis

We did an initial analysis of the dataset. Below sections present our analysis.

2.2.1 User data

There are 1183362 total users whose reviews are present in the dataset. We plotted a histogram to understand the distribution of user reviews. Looking at the histogram, we can observe that most of the user have very few reviews and some top users have significant number of reviews. Majority of user have 25 or less reviews which is also shown by a mean of 23..72 and standard deviation of 80.5. The maximum number of reviews given by any user is 11656.

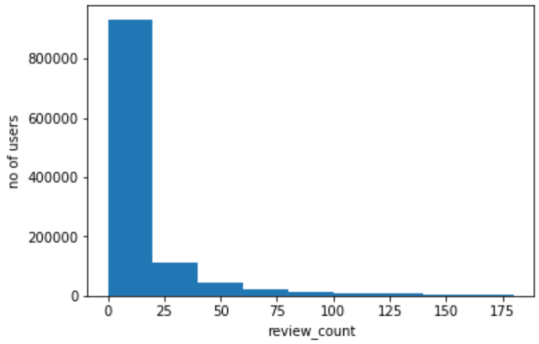

Figure 2: Review count per user

In addition to number of reviews, we also looked at distribution of star ratings given by a user. Looking at the histogram, we can observe that more users give higher rating which is shown by a median of 3.89 star rating. Mean and standard deviation for the same are 3.71 and 1.10 respectively. In order to group the reviews as positive, average and negative reviews, we have used the following method. We assume that if the rating lies in the range of (mean standard deviation, mean) which is 2.6 to 3.7, we will categorize it as average. Reviews lower than 2.6 will be considered as a negative review and anything greater than 3.7 will be considered as positive reviews with two extremes being 0 and 1.

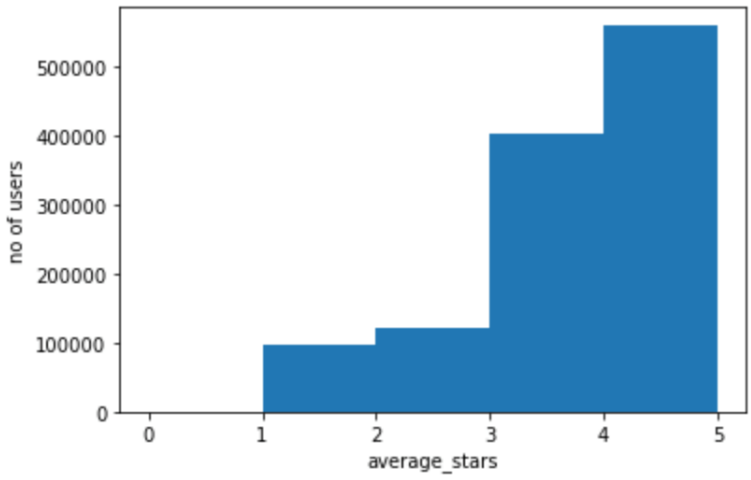

Figure 3: Rating per user

We also did some analysis to see the user growth on yelp. User growth has started declining

after an increase in users joining from 2005 to 2014 .

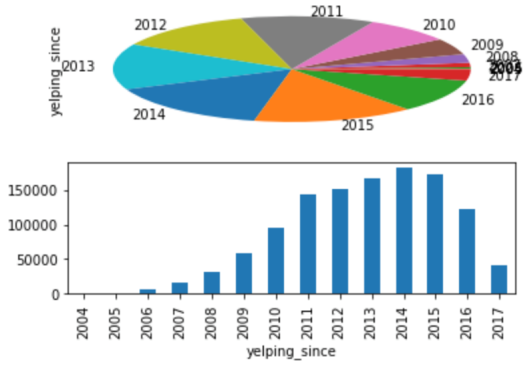

Figure 4: User Growth

2.2.2 Business Data

There are total 156639 business in the dataset. We grouped business according to city and business category to determine popular cities and categories. Below pie-charts give idea about popular cities and categories.

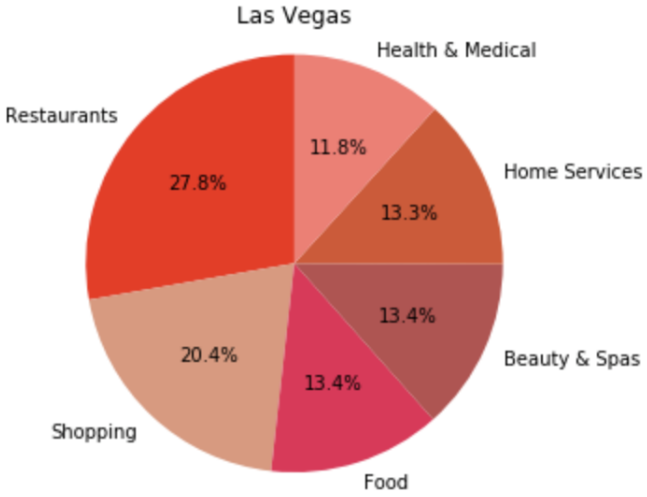

Figure 5: Top cities

We did more analysis into sub-categories of our most popular category i.e. resturants to map its distribution.

2.2.3 Checkin Data

Finally, we did analysis on use checkin data to find out popular timing in the top cities shown in our earlier analysis.

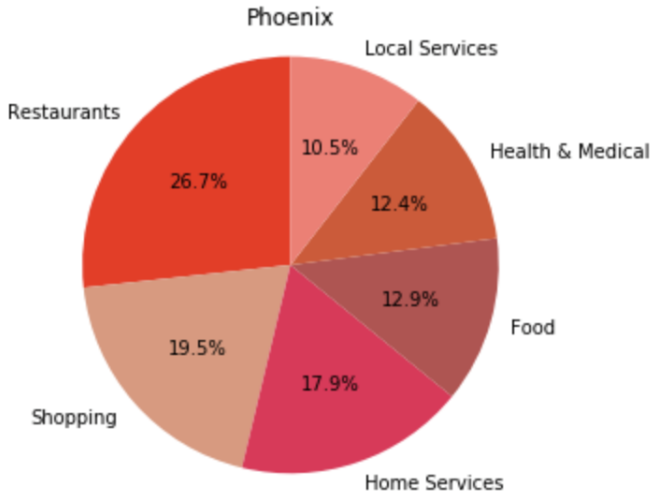

Figure 6: Top Business Categories

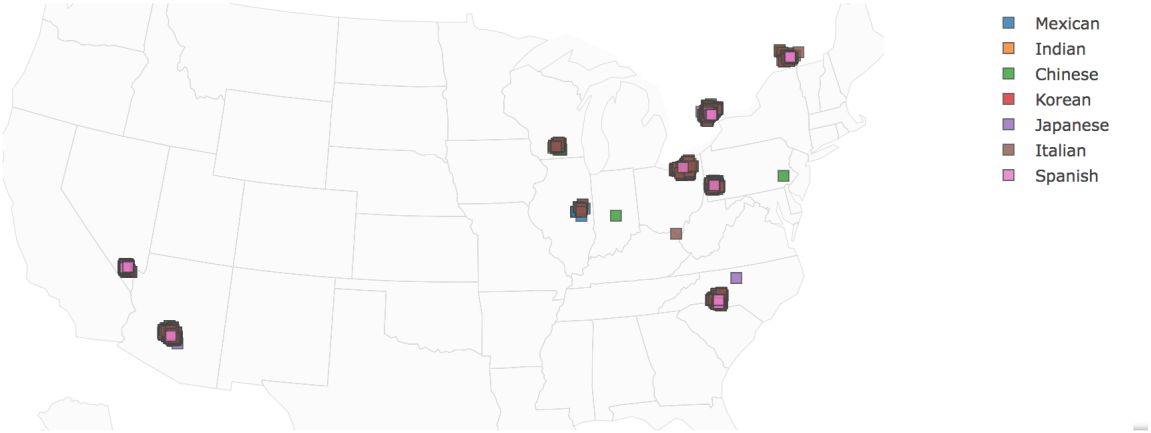

Figure 7: Resturant Sub-categories

3 Methods

3.1 Clustering Based Approach

3.2 Collaborative Filtering

Collaborative Filtering is the method to implement recommedation system. It is the way to recommend item to user 'u1' by collaborating the choice of item of other users similar to user 'u1'. We create a n X m matrix, where n is the number of users and m is the number of business, with each cell representing the rating given by a user to that particular business. Each row in matrix represent the vector of star rating given by user for all the businesses.

While doing this, we face two issues 1. As expected, there are a lot of missing values in matrix(for items which user has not given rating). We treat missing values as the average computed after following the second step - which will always be zero. 2. There is an

	city	time
0	Chandler	[(2:00, 34479), (19:00, 36988), (1:00, 39656)]
1	Charlotte	[(22:00, 64241), (17:00, 64472), (23:00, 72857)]
2	Cleveland	[(0:00, 21725), (22:00, 23961), (23:00, 25273)]
3	Edinburgh	[(12:00, 7720), (17:00, 7822), (18:00, 8324)]
4	Glendale	[(0:00, 20438), (2:00, 20472), (1:00, 23372)]
5	Henderson	[(20:00, 51439), (2:00, 53762), (1:00, 54136)]
6	Las Vegas	[(20:00, 441484), (1:00, 447715), (2:00, 481180)]
7	Madison	[(1:00, 15968), (23:00, 18773), (0:00, 19747)]
8	Mesa	[(2:00, 30860), (19:00, 31840), (1:00, 34917)]
9	Montréal	[(22:00, 17128), (0:00, 17196), (23:00, 18541)]
10	Phoenix	[(2:00, 170967), (19:00, 173996), (1:00, 188606)]
11	Pittsburgh	[(16:00, 36995), (22:00, 41485), (23:00, 43827)]
12	Scottsdale	[(0:00, 116834), (1:00, 128978), (19:00, 142432)]
13	Tempe	[(2:00, 49823), (19:00, 53595), (1:00, 54233)]
14	Toronto	[(0:00, 67390), (22:00, 67789), (23:00, 75615)]

Figure 8: Checkin Times

business_id user_id	-MhfebM0QIsKt87iDN-FNw	-cYOKJ5kbVZqzSYQlzZcqA	-lC6gIVhl7vY6W_dnw08YA	-pV9kWNoA9vyHfM_auYecA	03SYJLErY8XpNfY-qiDZcw	0nyM
---1lKK3aKOuomHnwAkAow	NaN	NaN	NaN	NaN	NaN	
--Nnm_506G_p8MxAOQna5w	NaN	NaN	NaN	NaN	NaN	
--P-Qvza7AED8gnDrZkMgA	NaN	NaN	NaN	NaN	NaN	
--ZNfWKj1VyVEIRx6-g1fg	NaN	NaN	NaN	NaN	NaN	
-00MbjbaOlSrcuV7jOVRlg	NaN	NaN	NaN	NaN	NaN	

5 rows × 227 columns

Figure 9: UserVsBusiness - Stars Rating Value

issue of handling the rating given by soft users and hard users i.e some users may rate the business they like with a 3 star ratings and there are some users who may rate the business they don't like with a 3 star ratings. To normalize these ratings for each user we use the centered cosine similarity. We normalize the ratings by subtracting the row mean for each user.[8]

To recommend new businesses to a target user 'target', we find the cosine similarity between all other users and 'target'. Top 'k' businesses rated with positive average star rating, by users of having cosine similarity greater than 1, will be recommended to user 'target'.[9]

3.3 Collaborative Deep Learing

We are using Colloborative Deep Learning [7] (CDL) approach suggested by Hao Wang and team. We looked at various deep learning approaches towards building recommendation systems and we choose this work as it was generally applicable compared to other techniques

which either target music or videos recommendations. CDL is a heirarchical Bayesian model. Stacked Denoising autencoders [10] are used for feature learning and cleaning the noise from the input. In below sections, I will be brief about the input parameters and neural network arcitecture.

3.3.1 Notation and Problem Formulation

As explained in [7], collection of J items (business) is represented by J-by-S matric X_c, where row j is the bag-of-words vector $X_{c,j*}$ for item j based on a vocabulary size S. Assuming I users, an I-by-J rating matrix R. Given part of the ratingsin R and the content information Xc, the problem is to predict the other ratings in R. Here, X_c is the clean matrix after using SDAE and X_o is the noise-corrupted matrix.

3.3.2 Stacked Denoising Autoencoders

SDAE [10] is a feedforward neural network for learning representations (encoding) of the input data by learning to predict the clean input itself in the output as shown in figure

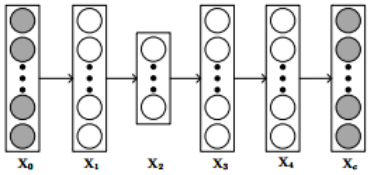

Figure 10: Stacked Denoising Autoencoders

3.3.3 Collaborative Deep Learing Model

Below figures show the model in detail, where W is weight matrix. The part inside the dashed rectangle represents an SDAE. An example SDAE with L=2 is shown. Here, λ_w, λ_n, λ_u, λ_s and λ_v are hyperparameters. The middle layer $X_{L/2}$ serves as a bridge between the ratings and content information.

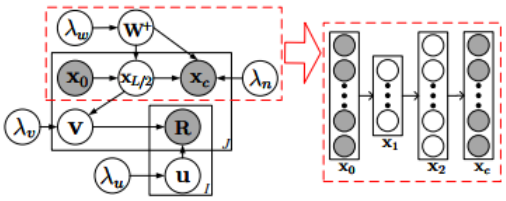

Figure 11: Graphical model of CDL

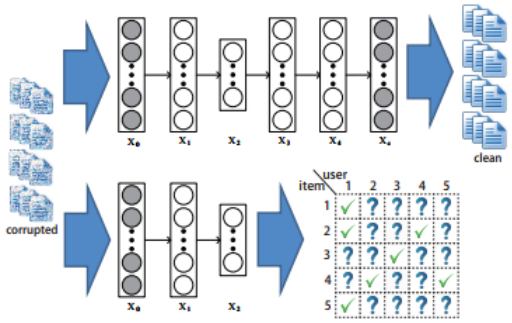

Figure 12: NN representation for degenerated CDL

4 Experiments and Results

4.1 User-User Collaborative Filtering

For the initial setup, we have worked on 100,000 rows of reviews.json file. We have created a matrix of 78276 users and 4224 businesses.We randomly choose one user to find the set of similar users of count 200. Based on positive average star ratings given by 200 users, 531 set of businesses are recommended to users.

5 Future work

1. Modify original implementation[11] of paper [7] to anaylize for our dataset.
2. All the algorithms are run on a small subset of data. Look into distributive processing framework(Spark), to perform the same on all the data or leverage the Google Cloud Platform.
3. Finalize evaluation criteria. The general approach that we plan to take would be:
   (a) Divide training and test data based on the timeline of reviews - to get past activity and future activity.
   (b) Define users in future activity as target users. Use the past activity of the target users to make recommendations.
   (c) Evaluate the recommendation as correct, if the recommendation shows up in the users future activity.
   (d) Get the total accuracy based on the same and evaluate our model!
4. Handle the problem of cold start - where a new business or a user is added. The general approach here would be to recommend the most popular business to the new user. Have to search for a way to tackle new businesses.

References

[1] Yelp Dataset Challenge `https://www.yelp.com/dataset_challenge`

[2] Recommendation System Wiki https://en.wikipedia.org/wiki/Recommender_system

[3] Collaborative filtering https://en.wikipedia.org/wiki/Collaborative_filtering

[4] Collaborative filtering Techniques https://www.youtube.com/watch?v=h9gpufJFF-0

[5] Movie Recommenation System https://beckernick.github.io/matrix-factorization-recommender/

[6] Paul Covington, Jay Adams, Emre Sargin *Deep Neural Networks for YouTube Recommendations*, ACM 2016.

[7] Hao Wang, Naiyan Wang, Dit-Yan Yeung*Collaborative Deep Learning for Recommender Systems*, ACM 2015

[8] Lecture on CF - Stanford https://www.youtube.com/watch?v=h9gpufJFF-0&t=854s

[9] Recommendation System Netflix https://www.youtube.com/watch?v=bLhq63ygoU8&t=5598s

[10] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. *Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion.* JMLR, 11:33713408, 2010

[11] CDL Implementation https://github.com/hexiangnan/neural_collaborative_filtering