Recommender System for Yelp Users

Team Members: Hardik Manoj Katariya 1001030751

Objective: To design a recommender system that would help predict what rating a user would give to a given item (business in our case) or recommend a particular item (business) to the user.

Overview: Yelp data set provides us with reviews and ratings given by people for various businesses. We can perform analysis on this information and can perform matching between different users. We can also compute relationships between different businesses and can develop recommendations based on them.

Data Mining Tasks:

1. I implemented user-based collaborative filtering in order to predict what rating a user might give to a particular test business.
2. I also performed item-based (business-based) collaborative filtering in order to recommend other items (businesses) to the user which he/she might be interested in.

Design of methods:

Since I mainly wanted to perform user based collaborative filtering and item based collaborative filtering, the main data of my interest was of user, businesses and the ratings that the user gives to different businesses.
For user based collaborative filtering, I first extracted {user_id, business_id, rating} from the reviews file and then created a json file which contained user_id as key and the value consisted of a key value pair where keys were the businesses the user had given ratings to and the values were the corresponding rating. Then on this data I carried out user-based collaborative filtering.

For item based (business based) collaborative filtering, I initially planned to perform the same thing as for user based collaborative filtering with the key being the business_id and value being key value pairs of users that had rated the business and the corresponding rating. But based on results and further analysis, I realised that it was better to find out similarity between businesses based on its attributes like categories,neighbourhoods,hours,city,stars,attributs etc rather than the ratings users had given them because for most of the businesses there were very few users in common so similarity based on the ratings would not be very accurate.

Implementation of methods:

In order to learn about how to build recommender systems, I first completed an online course 'Introduction to Recommender Systems' offered by courser. I also referred a paper from where I learnt about a good way to use the similarity in order to perform user based collaborative filtering and item based collaborative filtering. I have uploaded the paper on github ('sigir06_similarityfusion.pdf'). The paper also proposed a way to fuse both user based and item based collaborative filtering. I was not able to accomplish the fusion of user based and item based collaborative filtering due to lack of time. I also referred a book 'Programming Collective Intelligence'

(online free pdf at http://edc.tversu.ru/elib/inf/0251.pdf). The 2nd chapter of the book dealt with 'Making Recommendations'. I followed that chapter.

For user based collaborative filtering, I first found similarity between users using 'Pearson Correlation' similarity. After finding the similarity, I used the below formula to predict the rating $\widehat{x}_{k,m}$ for test item m by test user k :

$$\widehat{x}_{k,m} = \overline{u}_k + \frac{\sum_{\mathbf{u}_a \in \mathcal{S}_{\mathbf{u}}(\mathbf{u}_k)} s_{\mathbf{u}}(\mathbf{u}_k, \mathbf{u}_a)(x_{a,m} - \overline{u}_a)}{\sum_{\mathbf{u}_a \in \mathcal{S}_{\mathbf{u}}(\mathbf{u}_k)} s_{\mathbf{u}}(\mathbf{u}_k, \mathbf{u}_a)}$$

[1]

Where $\overline{u}_k$ and $\overline{u}_a$ [1] denote the average rating made by users k and a. Sometimes, the rating for item m by the user a are missing. In such cases, I replaced $x_{a,m}$ [1] by the average score for business m. The paper proposed of using either 0 or average rating of user a, but that lowered the prediction. So I experimented by using the average score for business m and was getting comparatively reasonable results.

For item based collaborative filtering, I first found similarity between different businesses. I wanted to use item based collaborative filtering for recommending items(businesses) to users, so I found similar businesses for each business the test user had rated and put them into a list and then used a random function to select 10 businesses out of all the similar businesses to be recommended to the user.

Results:

Based on the above implementation I was able to predict ratings for different users for different businesses that they had not previously rated and I was also able to recommend users with businesses that might be of interest to them.
I was not able to determine how to evaluate the recommendations. I evaluated the results for some users by manually removing some of the known rating by the user for a business and then predicted the rating for that business by the user. The resulting prediction was not exactly same at the rating what the user had actually given but it was close to the actual ratings.

Presentation:

[Note: Please go through the readme.txt document uploaded on git hub before running the web app locally (only if there is some hosting problem and the app is not behaving as expected on the hosted server). This is because we the python programs read some json files and since the size of the json files was large, it was not possible to upload them on git hub]

In order to demonstrate the working of the system I created a web application in python using flask. I created a simple website where we can login for any user from the list of users that we got from yelp and then specify a business_id for which the we want to find the predicted rating. After we

specify the business_id, the system provides us with a score that the user might give to the provided business_id. I also wanted to demonstrate the recommendation but I was getting some error in flask and was sine this is the 1st time I am working with flask and as I was falling short of time, I was not able to resolve it. The working of the recommendations part can be tested by manually running the 'Recommendations.py' program.

Instructions:

- When logging in, use any of the key value pair from 'authentication.json' as username and password.
- When providing business_id, use any of the business_id from 'yelp_academic_dataset_business.json' file.

URL of the project website:

I have hosted the project on pythonanywhere.com. The link is
http://hardik9107.pythonanywhere.com/login

References:

The sources that I referred when creating the project are as follows:

[1] http://siplab.tudelft.nl/sites/default/files/sigir06_similarityfusion.pdf

[2] https://www.coursera.org/learn/recommender-systems

[3] http://edc.tversu.ru/elib/inf/0251.pdf