Report: Recommending the Restaurant

Problem Set :

Food is an integral part of the culture of a place. When we visit a place it might be difficult to select a restaurant as we expect different things like ambiance, budget etc. This project uses FourSquare Api to collect restaurants near CP ,Delhi, India. Is used Folium to plot on a map, geolocator to get latitudes and longitude of a neighbourhood. It uses sklearn to make a model which takes reviews and likes of a place and can recommend a place based on your input of string.

Data acquisition:

FourSquare API is used to collect venue_id, user_id, venue latitudes, venue longitudes, likes and reviews.
Location of Neighbourhood is collected by using geolocator.
CSV files are used to store the dataframe values.

Cleaning of data:

NLTK is used to remove emoji, pronunciations, common nouns and leaves the review with adjectives and adverbs which can be tokenised to get a meaningful information.
DataFrame is used to store data is organised in an organised way.

Feature Selection:

After cleaning the data there were 77 Restaurants with one review each. This gives a total of 77 reviews and 77 like count. After using NLTP to extract features, there were 128 features with 15 user in uers_df and 128 features with 68 different restaurants and other various food places.
There were several entries in 15 * 68 cells in matrix created by fusing the user_df data frame and venue_df frame which were NaN as all users have not used all the restaurants in the city. All those cells were dropped.

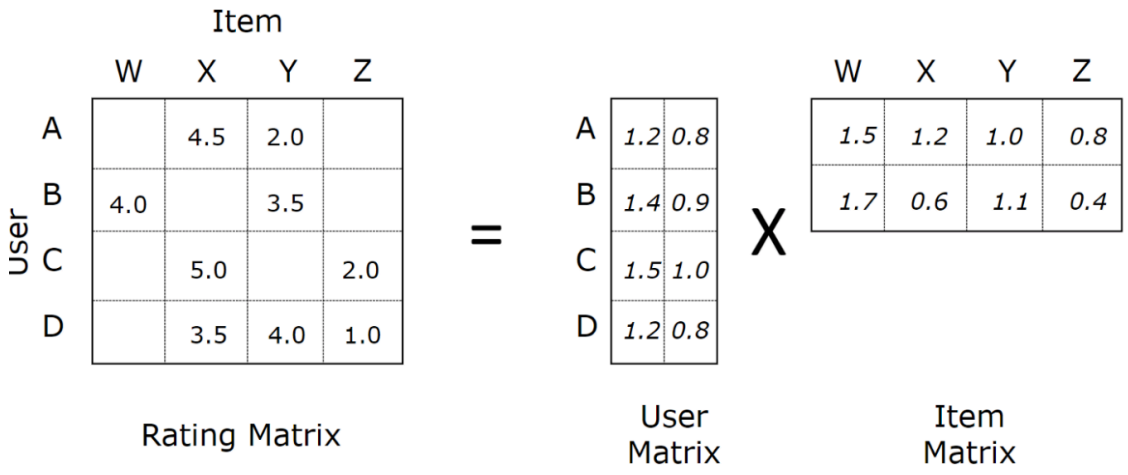
Calculations

User_df is created by keeping rows as user_id and columns as features extracted from reviews from all the restaurants visited.
Venue_df is created by keeping rows as venue_id and columns as features extracted from reviews from various users.

These are then clubbed together in a matrix which looks like this :

Venue Id User Id	4b59921bf964a520fe8c28e3	4b5c78e9f964a5206f3129e3	4b7421c5f964a5204ac92de3	4b79606ef964a52089f62ee3	4b9f365df964a5202418
1a21ab1c498e411929349f40	87.0	56.0	35.0	11.0	
4b7812db70c603bb373b92b4	NaN	NaN	NaN	NaN	
4e8ff234d22dccc37e935193	NaN	NaN	NaN	NaN	
4f0b05fee4b071c57819891f	NaN	NaN	NaN	NaN	
502a3170e4b022b4b5336955	NaN	NaN	NaN	NaN	
5140a733e4b0a19ca1f642b5	NaN	NaN	NaN	NaN	
5190fefa498e2fe606ae3fee	NaN	NaN	NaN	NaN	
536fa01911d2228b33b8d208	NaN	NaN	NaN	NaN	
53f2364b498edb0047f65edd	NaN	NaN	NaN	NaN	
53f9a2af498e6618bcace91e	NaN	NaN	NaN	NaN	
55265092498ea4ac7a14f3b1	NaN	NaN	NaN	NaN	
556e01f5498e040d55f4a0bf	NaN	NaN	NaN	NaN	
5724dfbe498ee18fe7397fd3	NaN	NaN	NaN	NaN	
57bb1b07498eec95a752cdc7	NaN	NaN	NaN	NaN	
5933888cd4cc987ec10bc230	NaN	NaN	NaN	NaN	

Rating Matrix is created as shown below (Image source : https://medium.com/@connectwithghosh/simple-matrix-factorization-example-on-the-movielens-dataset-using-pyspark-9b7e3f567536)



Predict the most relevant restaurant based on the user search i.e. simply the inner product of the feature vector of plain text and feature vectors of business Id. Out of all, top 3 records to be fetched.

Likes given by user and reviews are consistent i.e. one with the bad reviews will have bad rating in the rating matrix and more negative words like ‘bad’, ‘tasteless’, ‘not upto the mark’. Recommendations look like this on a map.

```
words = 'I want to eat best food available here'
```