Minju Lee

CptS 315

Course Project: Restaurant Recommender System

12/06/2019

Prof. Mohammed

Introduction

In today's society, we have an abundant amount of choices when it comes to choosing a place to eat and it can be difficult to decide where to eat. Currently, there are many different apps that recommend restaurants based on their location, rating, preferences of the user, and etc. Recommendation systems are widely utilized in different areas for predicting the preference or rating of a user in a product or service. As a food lover, I wanted to create a restaurant recommender system that takes account user's location then output recommendations based on the restaurant that the user already likes.

Generally, there are two main methods to the recommender system implementation, which are content-based filtering and collaborative filtering. Content-based filtering recommends based on the comparison between the descriptor of the item and a user profile. Collaborative filtering is a technique of making automatic predictions about the interests of a user by collecting preferences or rating information from many users.

The challenge of this project is to pre-filter context (location in this case) then apply collaborative filtering. We will center on geographic locations to provide related recommendations utilizing the location of the users. K-Means clustering algorithm will be used to define clusters of restaurants by its location. K-nearest neighbor algorithm will be used to implement item-based collaborative filtering. K-nearest neighbor will calculate the distance between the given restaurant and every other restaurant in its cluster then returns the top k nearest neighbor restaurants as suggestions.

The program can output a recommendation based on user location and item-item similarity.

Data Mining Task

Input data was obtained from the UCI machine learning repository as .csv files. Input data contains various restaurant information and user information including geographical data. For the first part of the filtering, the pandas data frame was used to merge geoplaces2.csv and rating_final.csvl files together after dropping unnecessary attributes. Mapbox from plotly express module was used to visualize locations of restaurants from given input data. A city with the most data was selected to implement Kmeans clustering. The city names were prepared by all upper-case letters. The determination of the optimal k-value was acquired from the elbow method evaluation. The below data shows restaurants in the same clusters based on user location.

	placeID	name	latitude	longitude	userID	rating	count	median
0	135082	la Estrella de Dimas	22.151448	-100.915099	U1088	2	9	0.0
1	135082	la Estrella de Dimas	22.151448	-100.915099	U1014	0	9	0.0
2	135082	la Estrella de Dimas	22.151448	-100.915099	U1018	2	9	0.0
3	135082	la Estrella de Dimas	22.151448	-100.915099	U1094	0	9	0.0
4	135082	la Estrella de Dimas	22.151448	-100.915099	U1115	2	9	0.0
..	...	...	...	...	...	...	...	...
93	132866	Chaires	22.141220	-100.931311	U1052	2	5	2.0
94	132866	Chaires	22.141220	-100.931311	U1008	1	5	2.0
95	132866	Chaires	22.141220	-100.931311	U1015	0	5	2.0
96	132866	Chaires	22.141220	-100.931311	U1025	2	5	2.0
97	132866	Chaires	22.141220	-100.931311	U1131	2	5	2.0

For the second part of the filtering, Item-item similarity will be calculated then K nearest neighbor approach was used to generate a recommendation to users based on a restaurant that the user likes. In this project, we will randomly pick the choice of the restaurant that the user likes. Below is an example of the output for the user number U1008.

Recommendations for Restaurant Restaurante y Pescaderia Tampico on a priority basis:

1: Potzocalli

2: KFC

3: Dominos Pizza

4: Abondance Restaurante Bar

Data mining questions that you set out to investigate in this project.

1. Top 10 most rated types of cuisine
2. Distribution of ratings
3. Top 20 restaurants from given data
4. Where are the restaurants located from given data
5. Clustering of the data by its location
6. Recommendation of the restaurants based on the user's location
7. Top 4 restaurants within the recommended restaurants from question 6 by using a collaborative filtering technique.

Key challenges to solve this task.

Learning how to use sklearn, scipy, and plotly modules. There are some tutorials online, however, I found it challenging to customize to my project. I have run into many issues to generate wanted output.

Technical Approach

In the first part of the implementation, I have used a Kmeans clustering approach using sklearn module. K-Menas is a very simple algorithm that clusters the data into K number of clusters.

Here is how the algorithm works.

Step1 - Pick K random points as cluster centers called centroids (my points are latitude and longitude of restaurants)

Let's assume random centroids = c1,c2,...., ck

Step2 - Assign each input to the nearest cluster by calculation its distance to each centroid

Euclidean distance : $d(C,X) = \sqrt{(C1-X1)^2 + (C2-X2)^2 + ... + (Cn-Xn)^2} = \sqrt{\sum_{i=1}^{n}(Ci-Xi)^x}$

(where C is centroid and X is a data point in n-dimensional vector)

Step3 - Find a new cluster center by taking the average of the assigned points

Si = set of all points assigned to the ith cluster(Ci)

Ci = $\frac{1}{|Si|}\sum_{xi\varepsilon Si} xi$

Step4 - Repeat steps 2 and 3 until none of the cluster assignment changes.

For the collaborative filtering implementation, I have used a sparse matrix from the scipy module and K nearest neighbors from the sklearn module.

Step1 - create a pivot table and fill in value 0 for the empty cells.

Step2 - calculate cosine similarity between each restaurant.

Cosine similarity is a metric used to measure how similar the documents are irrespective of their size.

$$Cos\theta = \frac{\vec{a}\cdot\vec{b}}{\|\vec{a}\|\ \|\vec{b}\|} = \frac{\sum_1^n a_i b_i}{\sqrt{\sum_1^n a_i^2}\ \sqrt{\sum_1^n b_i^2}}$$

where, $\vec{a}\cdot\vec{b} = \sum_1^n a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$ is the dot product of the two vectors.

Step3 - Perform k-nn classification and fit the given data.

k-Nearest Neighbor
Classify (**X**, **Y**, *x*) // **X**: training data, **Y**: class labels of **X**, *x*: unknown sample
for $i = 1$ **to** m **do**
  Compute distance $d(\mathbf{X}_i, x)$
end for
Compute set I containing indices for the k smallest distances $d(\mathbf{X}_i, x)$.
return majority label for $\{\mathbf{Y}_i$ where $i \in I\}$

Step 4 - Predict the restaurant recommended by a given restaurant.

Evaluation Methodology

The study dataset was obtained from the National Center for Research and Technological Development in CENIDET, Mexico. There are 130 unique restaurants, 138 unique users and 1161 ratings about the restaurant. Attributes that are used in this study are ‘placeID’, ‘name’, ‘userID’, ‘rating’, ‘latitude’, and ‘longitude’. The dataset was not big enough for me to test the accuracy of k nearest neighbors classifier. I also had a hard time finding any evaluation metrics for my study.

Opposed to supervised learning where we have the ground truth to evaluate the model’s performance, clustering analysis doesn’t have a solid evaluation metric that we can use to evaluate the outcome of different clustering algorithms. Furthermore, since Kmeans requires *k* as input and doesn’t learn it from data, there is no right answer in terms of the number of clusters that we should have in any problem. Sometimes domain knowledge and intuition may help but usually, that is not the case. In the cluster-predict methodology, we can evaluate how well the models are performing based on different *K* clusters since clusters are used in the downstream modeling. Elbow method gives us an idea of what a good *k* number of clusters would be based on the sum of squared distance (SSE) between data points and their assigned clusters’ centroids. We pick *k* at the spot where SSE starts to flatten out and forming an elbow. We’ll use the geyser dataset and evaluate SSE for different values of *k* and see where the curve might form an elbow and flatten out.

Results and Discussion

Finding the top 10 most rated cuisine from the given dataset. Matplotlib module was used to visualize. The below code counts how many time each unique time of cuisine have rated then produces a bar chart. Mexican cuisine was rated the most followed by national cuisine and the 3rd was American cuisine.

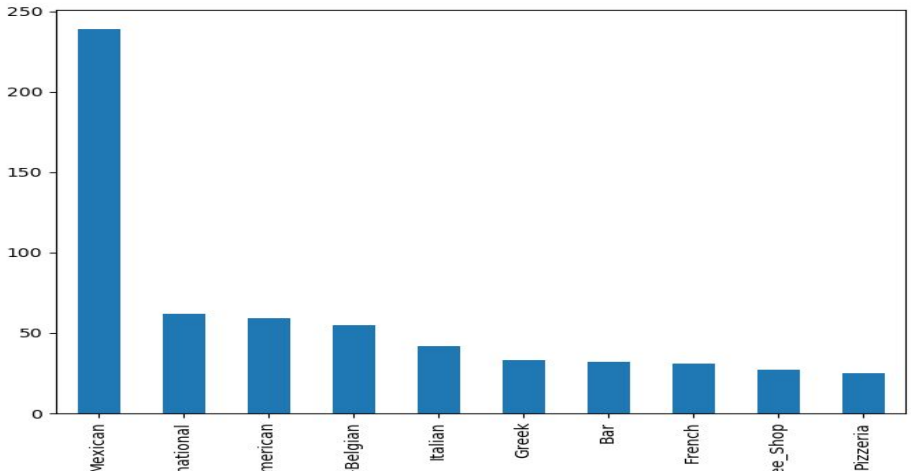

Figure1. Top 10 most rated cuisines in Mexico from input data

The median rating has approximately has a normal distribution centered at 1. Median rating was calculated from the sum of the ratings divided by rating counts for each restaurant.

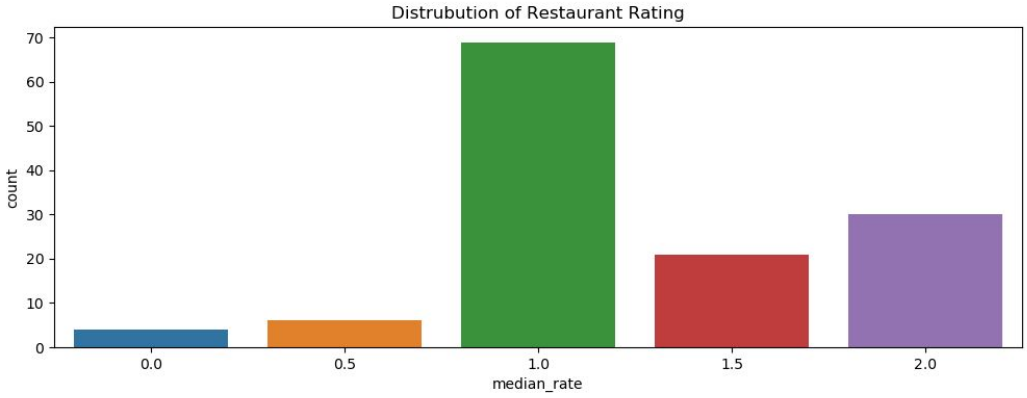

Figure2. Distribution of restaurant median rating

To calculate the popular restaurants, I have sorted the values depends on how many times the restaurant has been rated and it's median rating. The figure shows the top 20 restaurants from top to bottom in a bar graph. The most popular restaurant is Tortas Locas Hipocampo and the 2nd is Puesto de Tacos.

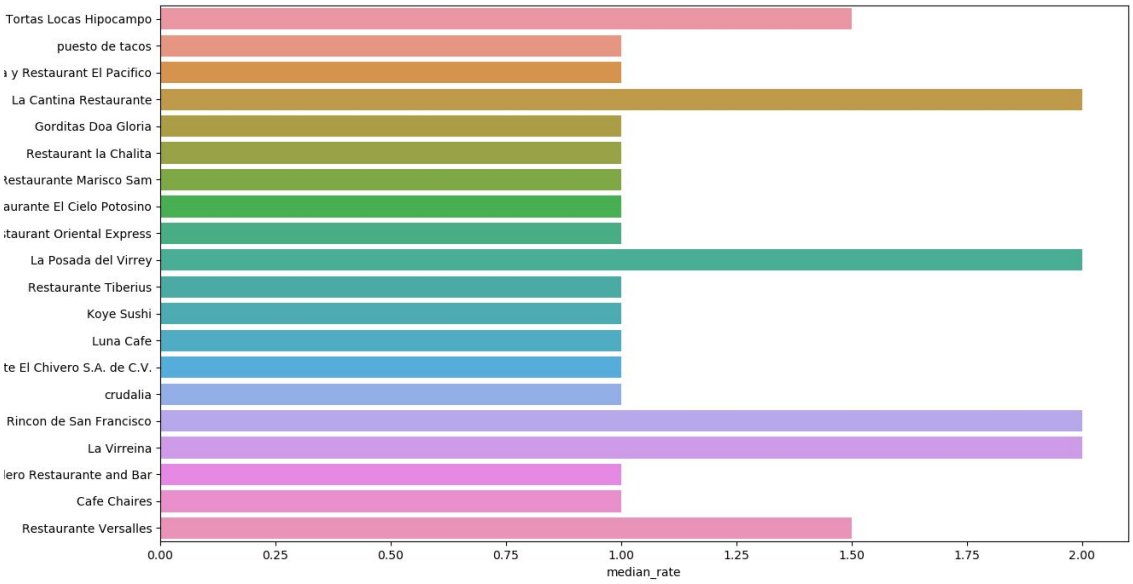

Figure3. Top 20 restaurants from input data.

Plotly express provides a scatter map box where I can illustrate geographical data. The figure below shows where the restaurants are located from out given dataset. I have used 'latitude' and 'longitude of each restaurant to plot on the map. As we can see, the dataset is distributed in 3 distinct states in Mexico which are San Luis Potosi, Tamaulipas, and Morelos.

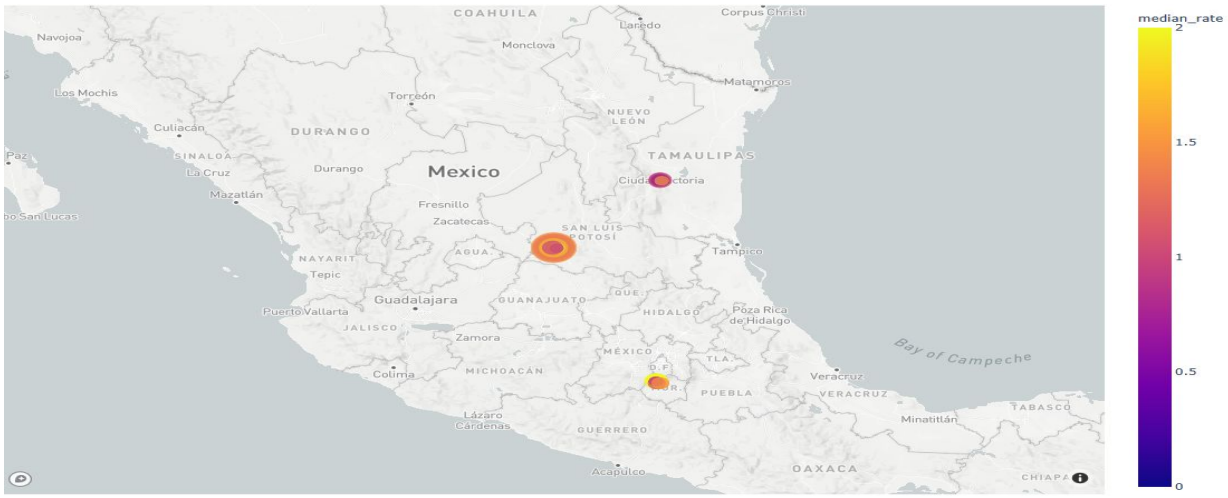

Figure4. Scatter mapbox to show locations of the restaurants from input dataset.

I have narrow it down to San Luis Potosi as my dataset to be able to perform K-means clustering. It is because out given dataset already has 3 distinct clusters based on its location and the user will most likely want a recommendation within their city. The first map shows all the restaurants in San Luis Potosi. The size of the circle represents the counts of the rating and the color represents their median rates. Blue being 0 and yellow being 2. The second map shows 5 distinct clusters that are based on their locations. The color in the 2nd map represents each cluster. I have chosen 5 clusters using the elbow method from sklearn.

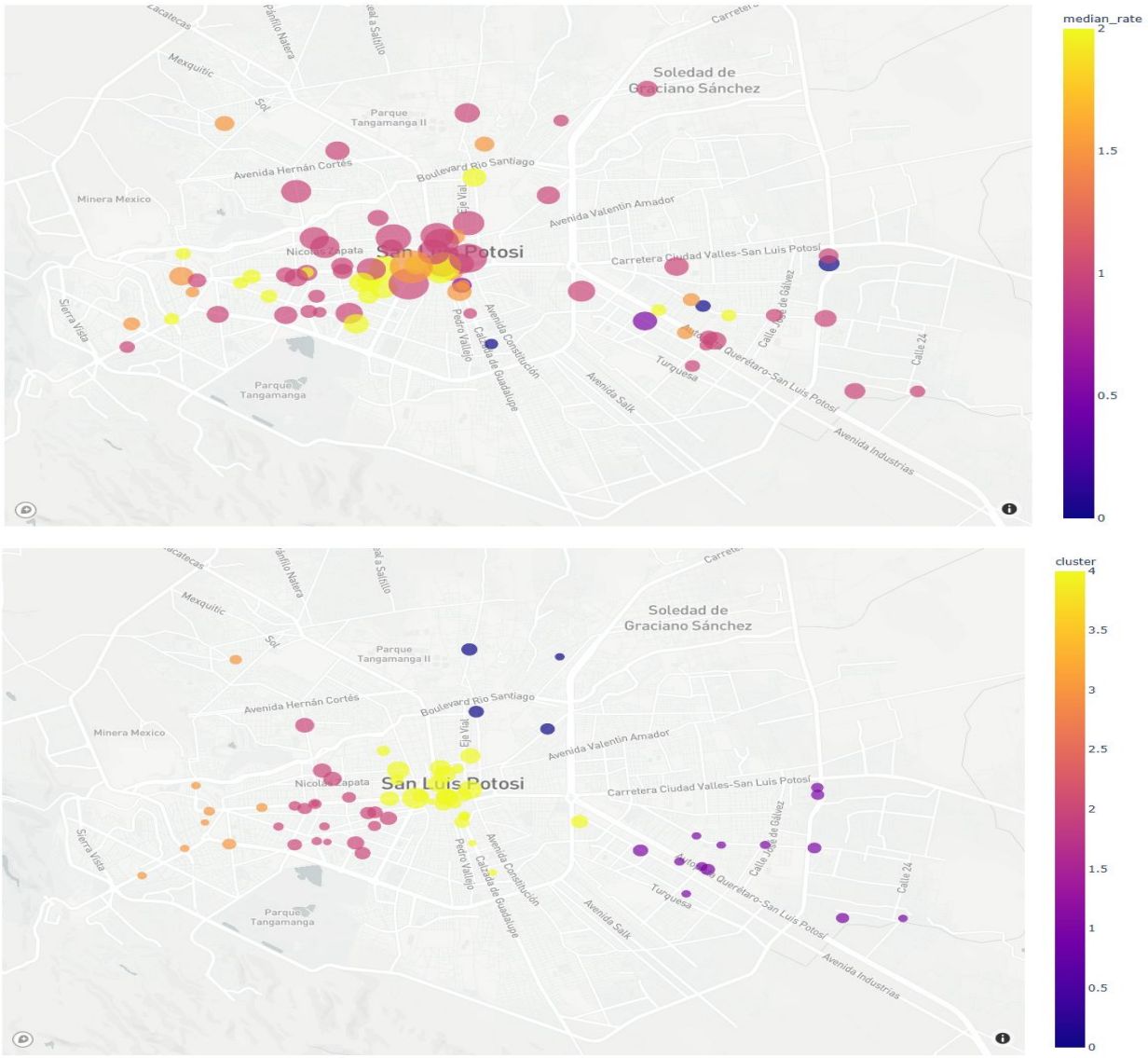

Figure5. Kmeans clustering by its location before and after

Elbow method will result in optimal k and as we can see from the chart below the graph flattens out at around 5 for dataset in San Luis Potosi.

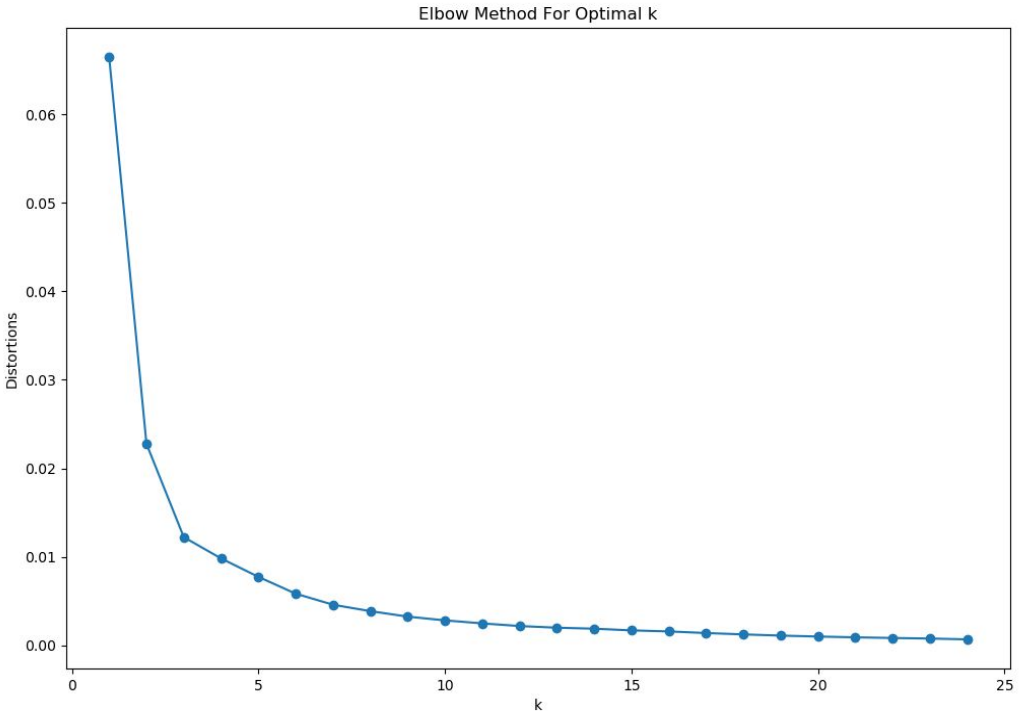

Figure6. Elbow Method for Optimal k

The below graph shows recommendations based on the user (U1008)'s location.

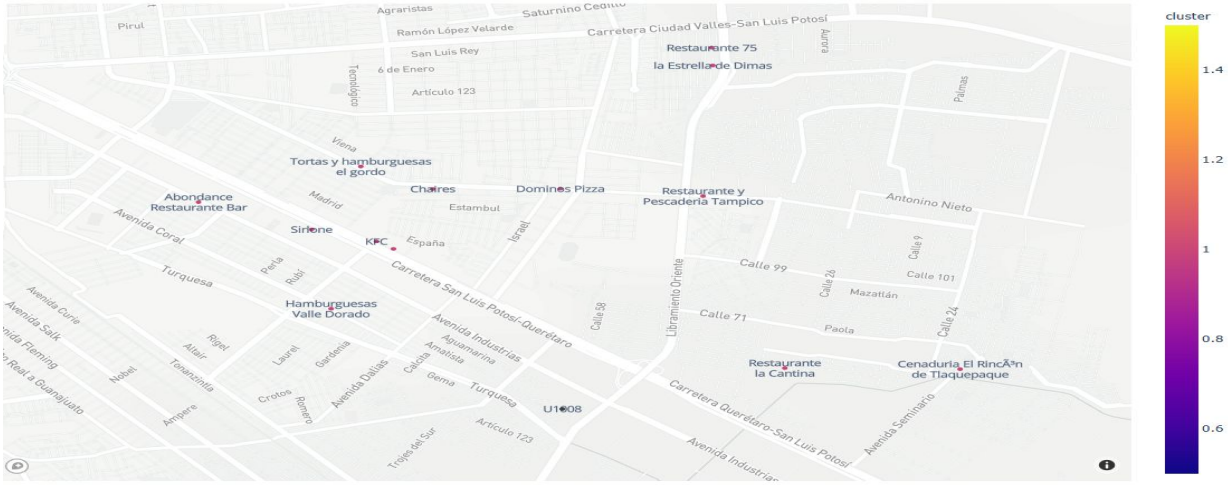

Figure7. Restaurant recommendation for user ‘U1008’

After filtering out the recommendation based on the user’s location, I have implemented k-nn algorithm provided by sklearn to apply item-item collaborative filtering. So if a user ‘U1008’ likes a restaurant called ‘Hamburguesas Valle Dorado’ the system will recommend similar restaurants that the user might also like using cosine similarity metrics.

Recommendations for Restaurant Hamburguesas Valle Dorado on priority basis:

1: Sirlone

2: Tortas y hamburguesas el gordo

3: Cenaduria El RincÃ³n de Tlaquepaque

4: KFC

Overall, my system works, however, I do not know the accuracy of the recommendation and also it can vary from individuals. I think pre-filtering the data based on the user’s preference can improve the recommender system. I have chosen the location as the factor that I used to pre-filter the data for each user.

Lessons Learned

I have learned that having more data helps in collaborative filtering technique to provide better accuracy. Having proper data that are well fitted to the purpose of the study is important to achieve better performance. Evaluating unsupervised learning is a lot more tricky and there isn’t a real solution to it.

K-means algorithm iteratively improves the cluster quality by assigning and reassigning data to different groups until an equilibrium is reached. We should be aware that even strong relationships can change at any time for no obvious fault, just like in k-means.

I would like to further incorporate other aspects of user preference, such as their price range, dress_preference, ambiance, and etc. There needs to be a deeper understanding of evaluation metrics and machine learning algorithms. Overall, I have grown my knowledge about machine learning and data mining through this course, however, I think it is just a tip of the iceberg of what is out there.

Acknowledgments

https://plotly.github.io/plotly.py-docs/generated/plotly.express.scatter_mapbox.html

https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

https://scikit-learn.org/stable/modules/neighbors.html

https://www.geeksforgeeks.org/python-data-analysis-using-pandas/

Ganarapu. (2019). Restaurant Recommendation System: https://medium.com/@sumith.gannarapu/restaurant-recommendation-system-b52911d1ed0b

https://www.machinelearningplus.com/nlp/cosine-similarity/
https://www.researchgate.net/figure/Pseudocode-for-KNN-classification_fig7_260397165

https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a