

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | |
|  | | PROJECT REPORT | | | | |  | |
|  |  | | | | | | |  |
|  | | | |  |  | | | |
|  | | | |  |  | | | |
|  | | | | 16 May 2023—Recommender System—Sir Zain UL Hassan |  | | | |
|  | | |  | | |  | | |

**Project Title:**

*RESTAURANT RECOMMENDATION SYSTEM*

**Group Members:**

* Ali Jodat (20K-0155)
* Basil Ali Khan (20K-0477)
* Muhammad Umer (20K-0225)

**Introduction:**

In today's digital age, the abundance of online restaurant information can overwhelm users when it comes to choosing a suitable place to dine. To address this challenge, we have developed a restaurant recommendation system using the K-Nearest Neighbors (KNN) algorithm and correlation similarities between users. The system aims to provide personalized recommendations based on user likings enhancing the dining experience for individuals.

**Problem Statement:**

The goal of this project is to create a restaurant recommendation system that suggests relevant dining options based on likings, restaurants similarities and popularity. The system takes into account the ratings in dataset provided by users to calculate similarities between them. By simply calculating most visited and rated restaurants percentages, KNN algorithm and correlation similarities, the system can recommend restaurants that align with a user's preferences.

**Methodology:**

Project involves following steps:

1. **Data Collection:** Restaurant data is gather from various sources such as online review platforms, social media, and user feedback. The collected data includes attributes like restaurant name, location, cuisine, ratings, and user reviews.
2. **Data Preprocessing:** The collected data is cleaned and transformed to ensure consistency and remove any irrelevant information. This involves tasks such as removing duplicates, handling missing values, and standardizing the data format.
3. **Popular Restaurants:** Calculated sum, mean and percentages to recommend most visited and rated restaurants (You may have visited web/apps and there are recommendations of popular restaurants)
4. **KNN Algorithm**: The KNN algorithm is apply to find the K nearest neighbors for a given user based on similarity scores. The neighbors' ratings are then use to make predictions for restaurants that the user has not yet rated.
5. **Similarity Correlation:** Similarity between users is calculate using the correlation coefficient. This measure captures the correlation between the rating patterns of two users. Higher correlation indicates similar preferences.
6. **Restaurant Recommendation:** Based on:
   1. Popularity (Most Visited and rated)
   2. K Nearest Neighbor (N number of Similar Restaurants)
   3. Correlation (Similar Users)

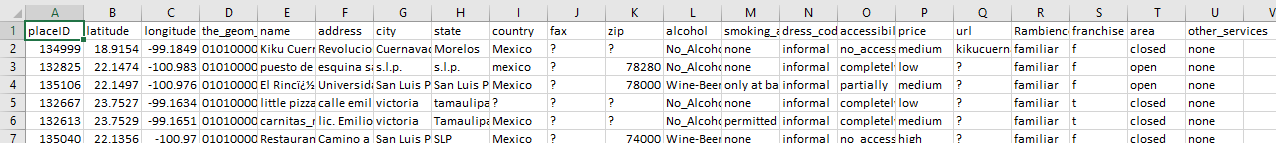
**Programming Language and Environment:**

* Python Programming Language
* Google Colaboratory

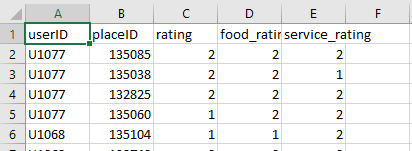
**Implementation:**

The recommendation system implemented using a programming language such as Python. The following code snippets shows implementation:

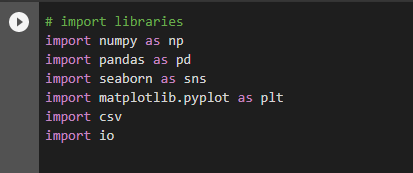
* **Datasets**
  + geoplace2.csv: Containing location of restaurants and place ID



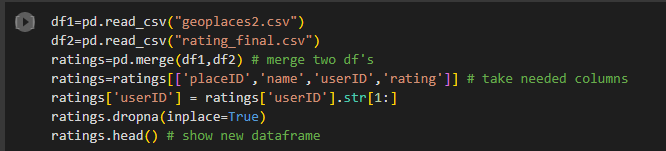
* + ratings\_final.csv: Containing ratings of restaurant done by users



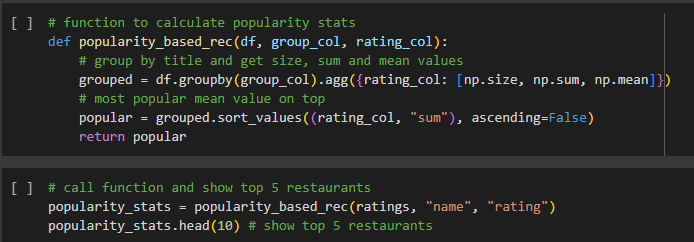
* **Libraries**



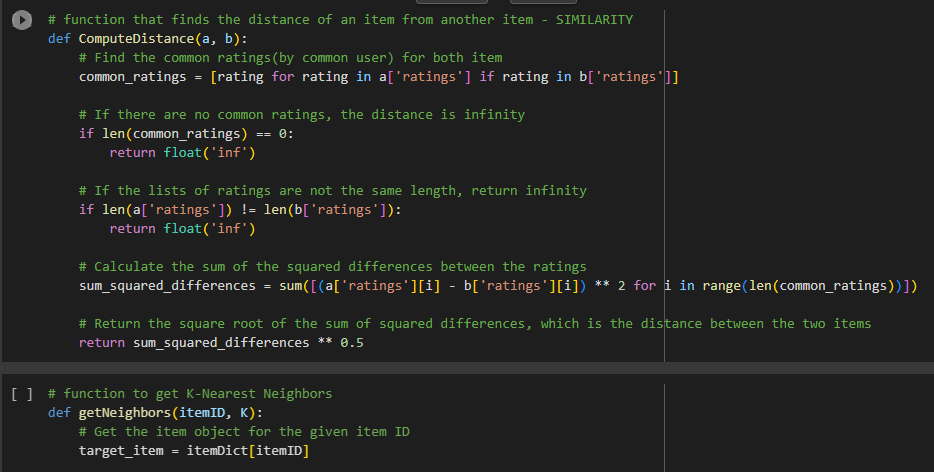
* **Preprocessing**



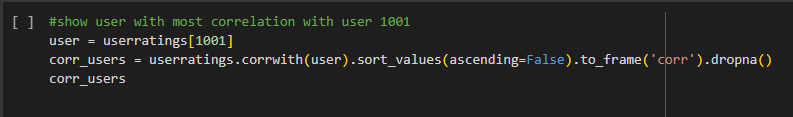
* **Popularity Based Recommendation**

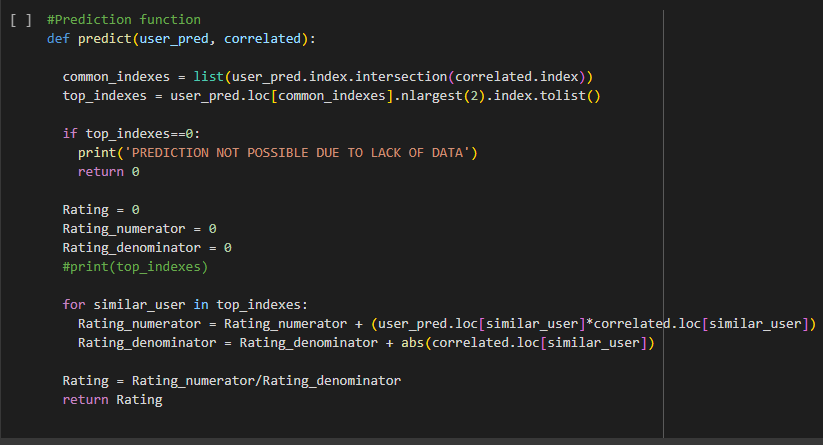


* **KNN Algorithm**



* **Correlation Similarity**





**Conclusion:**

In this project, we have developed a restaurant recommendation system using the KNN algorithm and correlation similarities between users. The system leverages user ratings to suggest personalized restaurant recommendations. By employing this system, users can easily discover new dining options that align with their preferences and enhance their overall dining experience. Further enhancements can be made by incorporating additional features like location-based recommendations, user feedback analysis, and real-time data updates.

**Future Directions:**

To build more friendly graphical interfaces, the next goal for a further project is to improve the performance of System and build an application for this and recommend based on geographical places. Few suggestions that might improve your user based recommender system. One is to normalize utility matrix before taking the dot product. This will help to account for any scale differences between users (some users may tend to rate items higher or lower than others do). Might also use a technique called singular value decomposition (SVD) to decompose your utility matrix into three matrices. This can help to reduce the noise in your data and improve the accuracy of your predictions