## Restaurant Recommendation Engine Documentation

### Project Title:

**Restaurant Recommendation Engine using Location-Based Clustering**

### Objective:

To build a recommendation system that predicts which restaurants (vendors) a customer is most likely to order from, based on their delivery location and historical order behavior.

### Dataset Overview:

The following datasets were used:

| File Name | Description |
| --- | --- |
| orders.csv | Order history of customers including vendor and location info |
| test\_locations.csv | Delivery locations for test customers |
| vendors.csv | Information about each vendor |
| SampleSubmission.csv | Format reference for final submission |

### Approach: Location-Based Clustering + Nearest Vendors

This approach groups customer locations using KMeans clustering and recommends vendors based on popularity within each cluster.

### Step-by-Step Process:

#### Step 1: Load and Explore Data

* Loaded test\_locations.csv, vendors.csv, and orders.csv.
* Inspected structure, missing values, and coordinate validity.

#### Step 2: Cluster Test Locations

* Applied KMeans clustering on latitude and longitude from test\_locations.
* Assigned a cluster ID to each location.

#### Step 3: Rank Vendors by Cluster Popularity

* Merged orders.csv with test\_locations on customer\_id and LOCATION\_NUMBER.
* Counted number of orders per vendor\_id within each cluster.
* Ranked vendors in each cluster by order frequency.

#### Step 4: Generate Recommendations

* For every test location, identified top 5 vendors from its assigned cluster.
* Constructed combinations of customer\_id, location\_number, and vendor\_id.

#### Step 5: Format Final Output

* Formatted recommendation rows as: CID X LOC\_NUM X VENDOR
* Saved final result to final\_submission.csv.

### Output Files:

| File Name | Purpose |
| --- | --- |
| cluster\_based\_recommendations.csv | Raw recommendation data with customer-location-vendor triplets |
| final\_submission.csv | Final formatted submission file |

### Tools and Libraries:

* Python 3.13
* Pandas (data handling)
* Scikit-learn (KMeans clustering)

### Possible Improvements:

* Use vendor ratings to improve vendor ranking.
* Include delivery distance as a factor in scoring.
* Weight recent orders more heavily using time-decay.
* Add fallback logic for customers with sparse data.

### Author:

Built by jayanth reddy based on the specification from VariableDefinitions.pdf.