DATA MINING(CS451)

Project Report

B.Tech. (CSE/DS/CYS)

Semester II

L-T-P-C: 3-0-2-4

Academic Year: 2024-25

Mood based Food Recommender System

Submitted by

1	Raja Pandey	BT21GCS323	Big Data
2	Rishabh Gupta	BT21GCS020	Big Data
3	Rohit Satpute	BT21GCS039	Data Science
4	Vishrut Pradhan	BT21GCS096	Big Data

Course incharge: Dr. Sujata Bhutani

1. Objective

The aim of this project is to build a personalized food and restaurant recommendation system based on a user's mood, leveraging **data mining**, **machine learning**, and **emotion detection**. This enhances the dining experience by aligning food and restaurant choices with emotional states and user preferences.

---

2. Datasets

1. **Food Choices Dataset (`food_choices.csv`)**:
    - Contains food preferences and reasons linked to comfort foods.
    - Key Features:
        - `comfort_food`: Lists preferred comfort foods.
        - `comfort_food_reasons`: Describes emotional triggers for these food preferences.
2. **Zomato Restaurants Dataset (`zomato.csv`)**:
    - Provides detailed restaurant information from New Delhi.
    - Key Features:
        - `Longitude`, `Latitude`: Geographic location.
        - `Cuisines`: Types of food offered.
        - `Aggregate rating`, `Rating text`: Customer ratings and reviews.
3. **Pre-Trained Emotion Detection Model (`emotion_model.h5`)**:
    - Used for detecting user emotions from webcam input.
    - Predicts one of five moods: Angry, Happy, Neutral, Sad, Stressed.

---

3. Data Preprocessing

3.1 Restaurant Data (Zomato)

- **Geographic Filtering**:
    - Retained restaurants in New Delhi (`Country Code = 1` and `City = 'New Delhi'`).
    - Removed entries with invalid or missing coordinates

```python
# Preprocess restaurant data
Qodo Gen: Options | Test this function
def load_and_preprocess_restaurant_data(res_data):
    # Preprocess data
    res_data = res_data.loc[(res_data['Country Code'] == 1) & (res_data['City'] == 'New Delhi'), :]
    res_data = res_data.loc[res_data['Longitude'] != 0, :]
    res_data = res_data.loc[res_data['Latitude'] != 0, :]
    res_data = res_data.loc[res_data['Latitude'] < 29]
    res_data['Cuisines'] = res_data['Cuisines'].astype(str)

    return res_data
```

Rating Cleanup:

- Excluded unrated restaurants (`Rating text = 'Not rated'`).

Feature Engineering:

- Created a numerical mapping for `Rating text`

```python
rating_map = {
    'Not rated': -1,
    'Poor': 0,
    'Average': 2,
    'Good': 3,
    'Very Good': 4,
    'Excellent': 5
}
```

- Ensured cuisine data was stringified for uniform processing.

3.2 Food Choices Data

- **Missing Value Handling**:
    - Replaced null values in `comfort_food` and `comfort_food_reasons` with empty strings.
- **Text Processing**:
    - Removed stopwords and lemmatized text using NLTK to standardize comfort food reasons.
    - Mapped comfort foods to corresponding cuisine types.

3.3 Emotion Detection Preprocessing

- Used OpenCV for image preprocessing:
    - Converted webcam input to grayscale.

- Detected faces using Haar Cascade.
- Resized images to 48x48 pixels and normalized pixel values.

---

4. Data Mining Analysis

4.1 Clustering Analysis

- **K-Means Clustering**:
  - Applied to geographic coordinates (`Longitude`, `Latitude`) to segment restaurants into **7 clusters**.
  - Clustering helps identify densely packed restaurant areas in New Delhi for localized recommendations.
  - Implementation:

```
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=7, random_state=0).fit(res_data[['Longitude', 'Latitude']])
res_data['pos'] = kmeans.labels_
```

  - Outcome:
    - Clustered restaurants geographically to create region-based insights.
    - Example: Central Delhi clusters have higher ratings compared to peripheral areas.

4.2 Text Mining

- **Comfort Food Analysis**:
  - Processed `comfort_food_reasons` to identify relationships between moods and food preferences.
  - Example:
    - Mood: *Sad* → Comfort Foods: Chocolate, Ice Cream, Pasta.
  - Implementation

```
Qodo Gen: Options | Test this function
def search_comfort(mood):
    lemmatizer = WordNetLemmatizer()
    foodcount = {}
    for i in range(124):
        temp = [temps.strip().replace('.','').replace(',','').lower() for temps in str(food_data["comfort_food_reasons"]
        [i]).split(' ') if temps.strip() not in stop ]
        if mood in temp:
            foodtemp = [lemmatizer.lemmatize(temps.strip().replace('.','').replace(',','').lower()) for temps in str
            (food_data["comfort_food"][i]).split(',') if temps.strip() not in stop ]
            for a in foodtemp:
                if a not in foodcount.keys():
                    foodcount[a] = 1
                else:
                    foodcount[a] += 1
    sorted_food = []
    sorted_food = sorted(foodcount, key=foodcount.get, reverse=True)
    return sorted_food
```

4.3 Association Rule Mining

- **Food-Cuisine Mapping**:
    - Linked comfort foods to cuisines for restaurant filtering.
    - Example Mapping

```
food_to_cuisine_map = {
    "pizza": "pizza",
    "ice cream": "ice cream",
    "chocolate": "bakery",
    "pasta": "italian",
    "burger": "burger"
}
```

4.4 Pattern Recognition

- Identified patterns between moods and food types using frequency analysis.
- Example Insight:
    - *Stressed users* prefer savory items like Pizza or Burgers.
    - *Happy users* gravitate towards sweets like Ice Cream or Cakes.

---

5. Recommendation System

1. **Food Recommendations**:
    - Maps user moods to the top 3 comfort foods.
    - Example:
        - Mood: *Angry* → Recommendations: *Pizza, Pasta, Burger*.
2. **Restaurant Recommendations**:

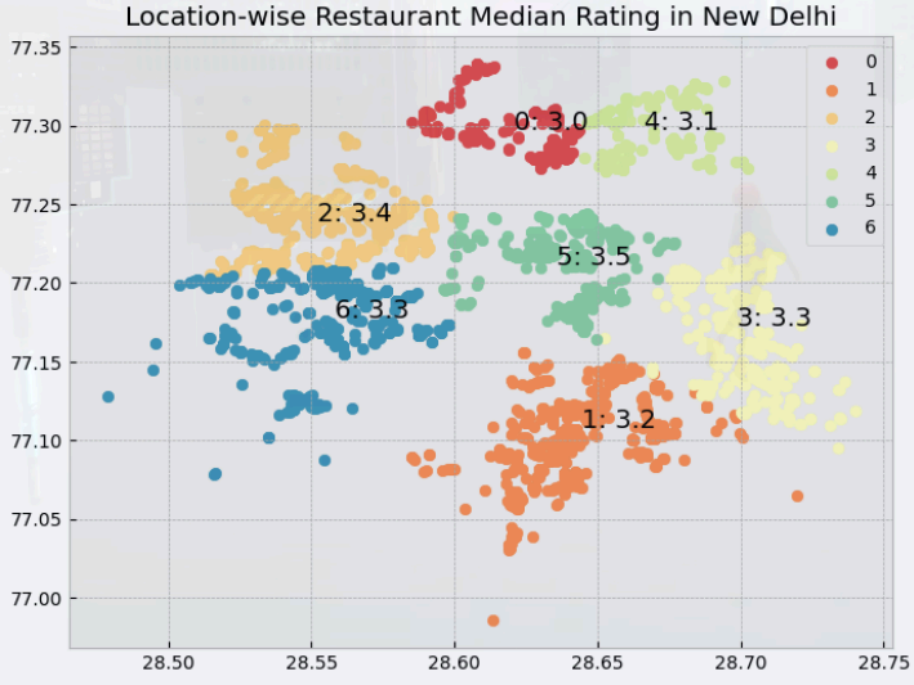
- Filters restaurants based on cuisine mappings.
- Ranks restaurants by aggregate ratings.
- Example:
  - Comfort Food: *Pizza* → Recommended Restaurants: *XYZ Pizzeria, ABC Cafe*.

3. **Interactive Map Visualization**:
   - Displays restaurant clusters and recommendations on a map for better navigation.

---

6. Results

1. **Clustering Insights**:
   - Seven geographic clusters identified areas with high restaurant densities.

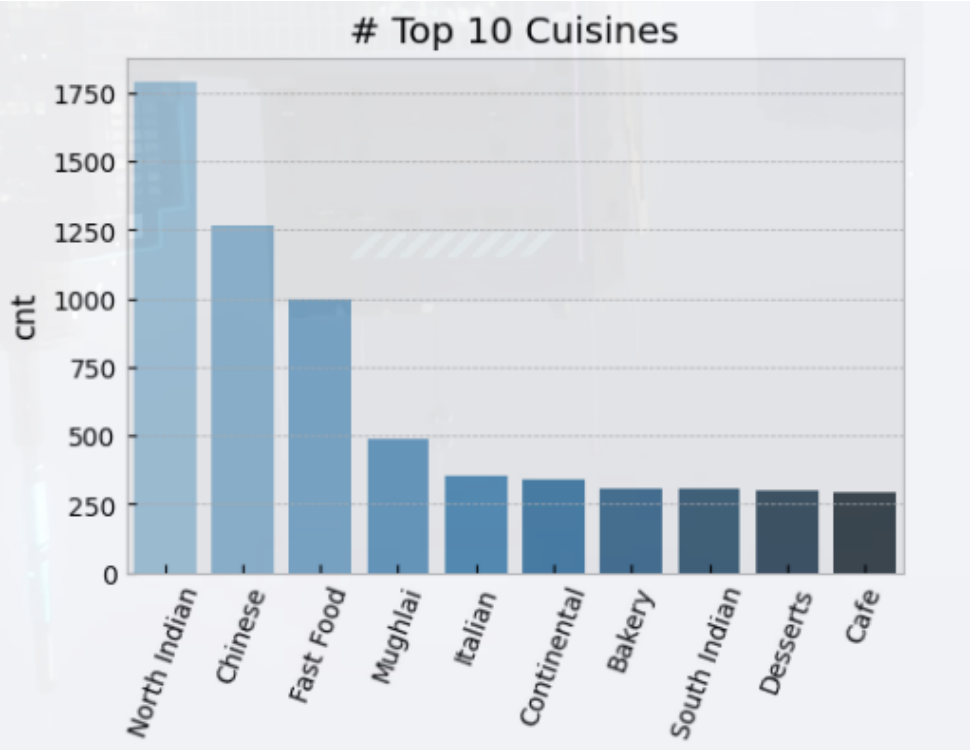


[img1: Location wise Restaurant median rating in New Delhi]

- Central Delhi clusters had higher-rated restaurants compared to peripheral clusters.

2. Cuisine Popularity:

- Top cuisines in New Delhi: North Indian, Chinese, Fast Food, Italian, Mughlai.



[img 2: Top 10 Cuisines Served]

3. Mood-Food Associations:

- Mapped common comfort foods for moods like Sad, Happy, and Stressed.

4. Restaurant Rankings:

- Provided ranked lists of restaurants for each recommended comfort food.

---

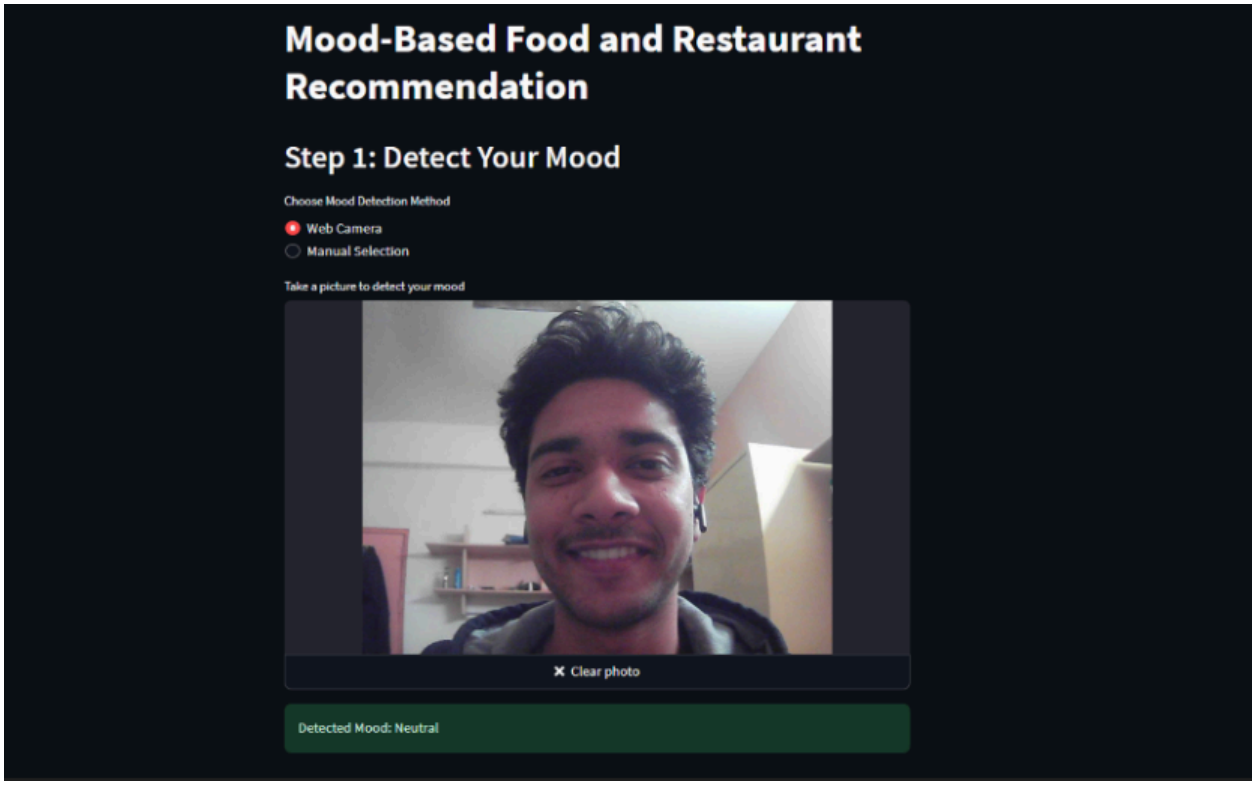
7. Visualization

1. **Geographic Clustering**:
   - Scatter plot of restaurant clusters by location.
2. **Cuisine Distribution**:
   - Bar plot showing the top 10 cuisines by popularity.
3. **Interactive Map**:

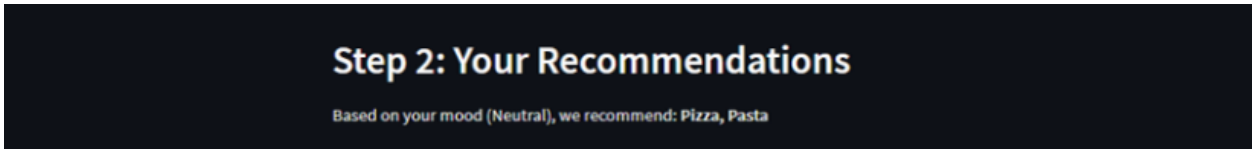
- Map visualization of recommended restaurants for user-selected foods.

---

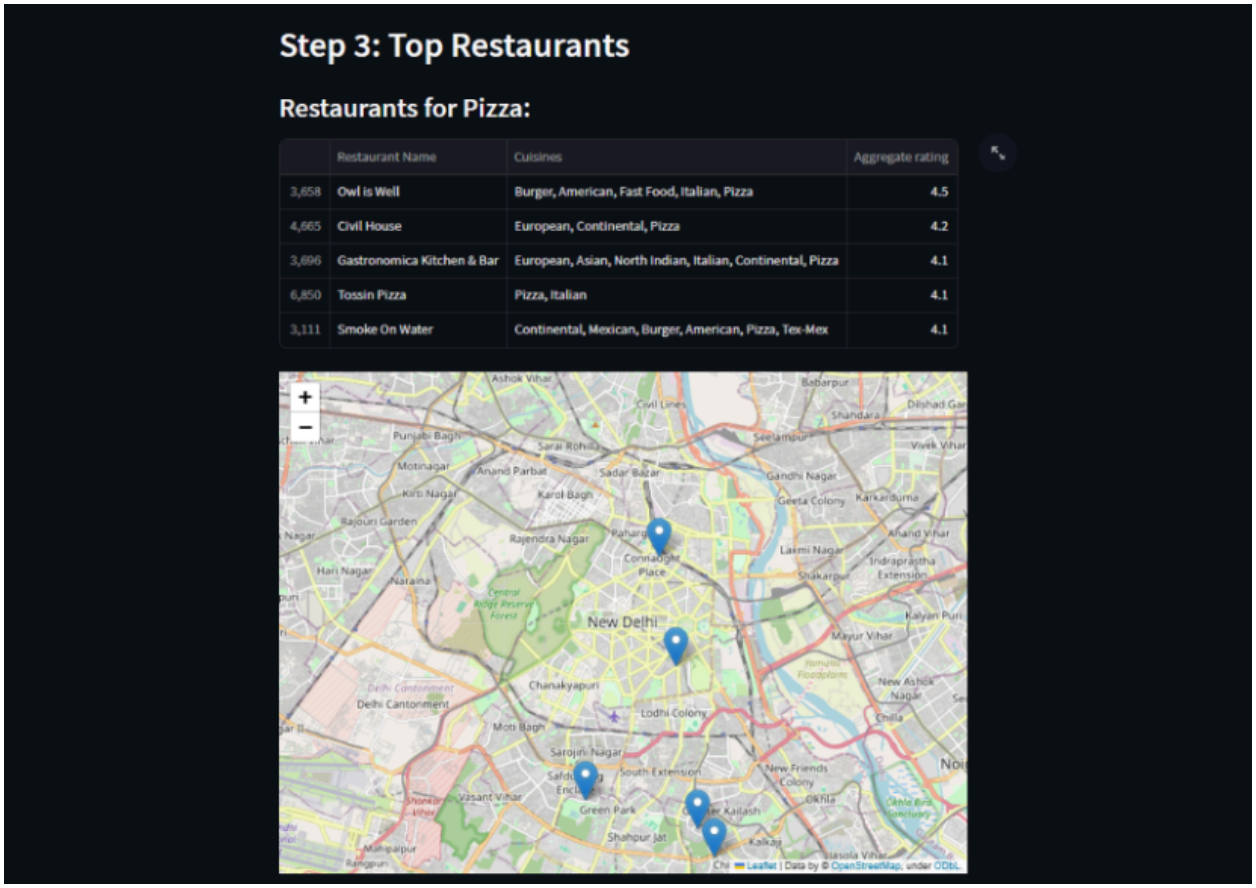
8. Outputs



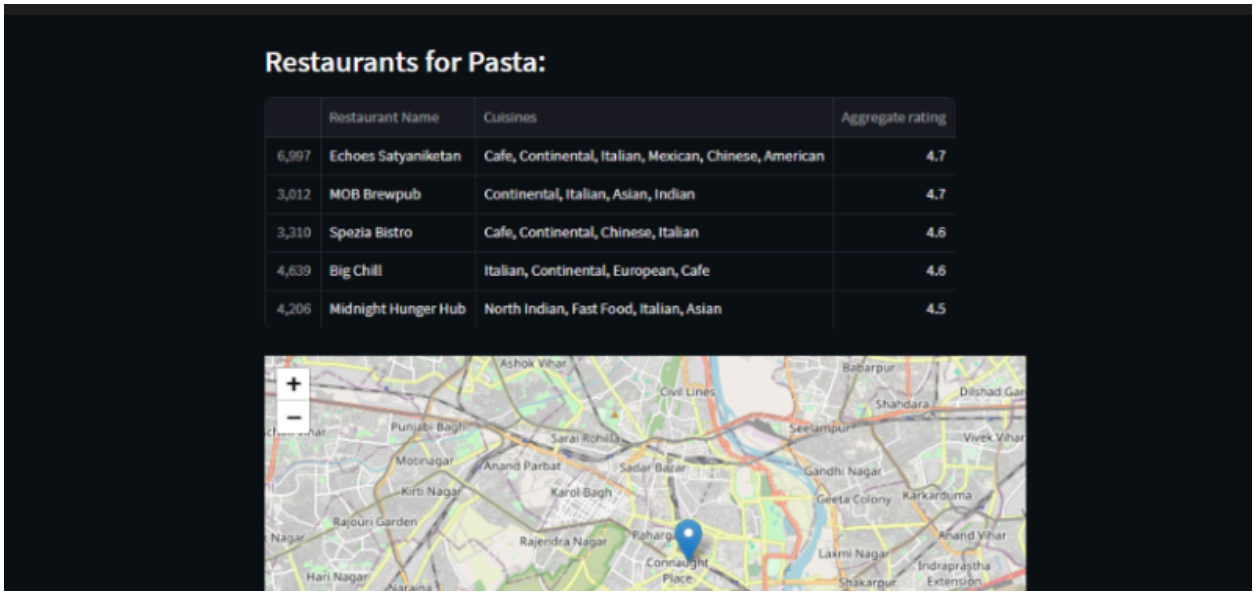
[img 3: Real-time emotion detection interface using webcam, showing the detected mood of the user]



[img 4: Personalized food recommendations based on the detected emotional state]



[img 5: Curated list of top restaurants with their cuisines and ratings based on recommended foods]



[img 6: Interactive map visualization showing the locations of recommended restaurants in New Delhi]

9. Future Scope

1. Expand restaurant database to include more cities.
2. Integrate dietary preferences for personalized recommendations.
3. Enhance emotion detection accuracy with advanced deep learning models.
4. Implement user feedback for continuous improvement of recommendations.