Humayun Raza l227548

Nawaf Ahmed i22-1939

Maheen Waseem l22-7467

Zahra Saqib l22-7453

Formal Report on the Restaurant Recommendation System

Contents

[1. Problem and Dataset Description 2](#_Toc184685507)

[Data Sources: 2](#_Toc184685508)

[Dataset Features: 2](#_Toc184685509)

[2. Data Collection and Web Scraping 3](#_Toc184685510)

[Process Overview: 3](#_Toc184685511)

[Steps Taken: 3](#_Toc184685512)

[Challenges and Solutions: 3](#_Toc184685513)

[Output: 3](#_Toc184685514)

[Random Assignment of Store IDs: 4](#_Toc184685515)

[3. Data Preprocessing and Transformation 4](#_Toc184685516)

[Handling Missing Values: 4](#_Toc184685517)

[Text Preprocessing: 4](#_Toc184685518)

[Encoding Location Data: 4](#_Toc184685519)

[Normalization: 4](#_Toc184685520)

[4. Recommendation System Methodology 5](#_Toc184685521)

[5. Sentiment Analysis of Reviews 5](#_Toc184685522)

[Metrics: 5](#_Toc184685523)

[6. Exploratory Data Analysis (EDA) 5](#_Toc184685524)

[Visualization Tools: 6](#_Toc184685525)

[7. Evaluation Metrics 6](#_Toc184685526)

[8. User Interface and Streamlit Integration 6](#_Toc184685527)

[Streamlit App Features: 6](#_Toc184685528)

[9. Expected Results 7](#_Toc184685529)

[Conclusion 7](#_Toc184685530)

Restaurant Recommendation System: Detailed Report

# 1. Problem and Dataset Description

The primary goal of this project is to develop a **Restaurant Recommendation System** that provides personalized suggestions based on user preferences and restaurant data.

## Data Sources:

1. Restaurant reviews, ratings, and other features scraped from **Google Maps**.
2. Simulated or scraped user preferences to train the recommendation models.

## Dataset Features:

The dataset was curated to include the following key features:

1. **Restaurant Name**
2. **Cuisine Type**
3. **Location** (geographical coordinates and towns)
4. **Average Rating**
5. **Number of Reviews**
6. **Restaurant Type** (e.g., fine dining, fast food)
7. **User Ratings**
8. **Popularity Score**
9. **Reviews**

From the scraped data, only **Location**, **Rating**, and **Reviews** were utilized in the recommendation process. Each restaurant was also randomly assigned a **Store ID** for consistent identification.

# 2. Data Collection and Web Scraping

Process Overview: The dataset was created by scraping restaurant data from **Google Maps** using **Selenium** and **BeautifulSoup**. The scraper focused on collecting:

* Names
* Locations (addresses and coordinates)
* Ratings
* Reviews (content and user details)

## Steps Taken:

1. **Dynamic Scrolling**: Selenium was used to scroll through the restaurant search results on Google Maps dynamically.
2. **Element Interaction**: Handled dynamic loading, click interception, and stale element references to ensure smooth extraction.
3. **Data Parsing**: BeautifulSoup parsed the loaded HTML content to extract relevant details.

## Challenges and Solutions:

* **Dynamic Content Loading**: Managed using scroll actions to load all restaurant results.
* **Duplicate Entries**: Ensured no duplicate restaurant names were stored using a tracking list.
* **Handling Errors**: Managed exceptions like stale element references and click intercepts.

## Output:

* The scraped data was stored in data.csv with columns such as:
  + Name, Phone, Address, Website, Reviews, Rating, and Location.
* For this project, only **Location**, **Rating**, and **Reviews** columns were used. A screenshot of a map

  Description automatically generated

## Random Assignment of Store IDs:

* Each restaurant in the dataset was randomly assigned a **Store ID** to act as a unique identifier in further analyses.

# 3. Data Preprocessing and Transformation

The scraped data required extensive cleaning and transformation to make it suitable for analysis and recommendations:

## Handling Missing Values:

* + Missing locations or ratings were imputed or removed.

## Text Preprocessing:

* + Reviews were cleaned by removing special characters and stopwords for sentiment analysis.

## Encoding Location Data:

* + Coordinates were linked to town names using an external file town\_coordinates.csv.

## Normalization:

* + Ratings and review counts were normalized to scale their influence.

# 4. Recommendation System Methodology

Three filtering techniques were explored to build the recommendation system:

1. **Content-Based Filtering**:
   * Recommended restaurants based on their features (e.g., rating, cuisine type).
2. **Collaborative Filtering**:
   * Leveraged user ratings and preferences to identify similar users and recommend restaurants they liked.
3. **Hybrid Approach**:
   * Combined the strengths of content-based and collaborative filtering for better accuracy.

# 5. Sentiment Analysis of Reviews

The **VADER sentiment analysis** model was used to analyze restaurant reviews, providing insights into user satisfaction.  
Steps:

1. **Extracted Review Sentiment**:
   * Each review was classified as Positive, Neutral, or Negative.
2. **Aggregated Sentiments**:
   * A restaurant’s overall sentiment was calculated by averaging its review sentiments.

## Metrics:

* **Sentiment Scores**: Ranged from -1 (most negative) to +1 (most positive).

# 6. Exploratory Data Analysis (EDA)

Key insights from the dataset:

1. **Cuisine Trends**:
   * Identified the most popular cuisines in Lahore based on the number of restaurants.
2. **Correlation Analysis**:
   * Explored relationships between average ratings and the number of reviews.
3. **Clustering**:
   * Restaurants were grouped by rating and price range using clustering techniques.

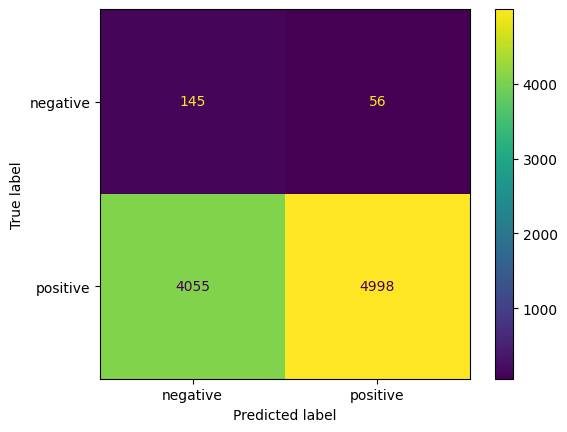
## Visualization Tools:

* Bar charts and scatter plots were used for visualizing trends.

# 7. Evaluation Metrics

For both classification and regression tasks, appropriate evaluation metrics were employed:

1. **Classification Tasks**:
   * Accuracy, Precision, Recall, and F1 Score were calculated for predicting user preferences.



1. **Regression Tasks**:
   * A screenshot of a computer screen

     Description automatically generatedRMSE, MAE, and R² were used for evaluating rating predictions.

# 

# 8. User Interface and Streamlit Integration

## Streamlit App Features:

1. **Restaurant Selection**:
   * Displays a list of all restaurants based on user-selected cuisine types or towns.

A screenshot of a computer

Description automatically generated

1. **Details and Reviews**:
   * A screenshot of a computer

     Description automatically generatedFor each selected restaurant, the app shows up to 5 reviews, with a "See More" option to display up to 20 reviews.
2. **Google Maps Integration**:
   * Displays restaurants' locations on an interactive map using coordinates from town\_coordinates.csv.

A map of a city

Description automatically generated

# 9. Expected Results

The system effectively provides personalized restaurant recommendations. It is evaluated based on:

1. **Recommendation Accuracy**: Improved through hybrid filtering.
2. **User Engagement**: Enhanced by displaying reviews and location details.
3. **Scalability**: The modular design allows for integration of additional features like price ranges and dining types.

# Conclusion

The restaurant recommendation system was successfully implemented using real-world data scraped from Google Maps.  
Key Outcomes:

1. The scraped dataset contained rich information, including reviews and ratings, to enable effective recommendations.
2. Data preprocessing and sentiment analysis contributed to building a robust hybrid recommendation model.
3. The Streamlit interface and map-based visualization enriched the user experience, making the system practical and user-friendly.