CHEFMATE: RESTAURANT CLUSTERING & COOKING GUIDE APPLICATION

By

M Karan Kumar

Introduction

ChefMate is an innovative web application that elevates the dining and cooking experiences by recommending restaurants tailored to user preferences and providing real-time cooking assistance through a chatbot. Built on a foundation of machine learning, cloud computing, and an intuitive user interface, ChefMate seamlessly integrates technology to offer a personalized and efficient solution for food enthusiasts.

Objectives

- To develop a system that clusters and recommends restaurants based on user inputs such as preferred cuisines or dishes.
- To provide a chatbot assistant that guides users through cooking recipes in real time.
- To leverage cloud computing for scalable data storage, preprocessing, and application hosting.

Key Features

1. **Restaurant Recommendations**: Offers personalized suggestions based on user preferences.
2. **Dynamic Data Visualizations**: Displays interactive maps and insights such as ratings and restaurant metrics.
3. **Recipe Chatbot**: Guides users with cooking instructions and addresses recipe-related queries.
4. **Cloud Integration**: Utilizes AWS services for reliable storage, preprocessing, and deployment.

Technologies and Skills Utilized

- **Application Development**: Built with Streamlit for an interactive and user-friendly interface.
- **Machine Learning**: Developed a clustering model to recommend restaurants.
- **AWS Services**: Leveraged S3, RDS, and EC2 for data storage and application hosting.
- **Data Processing**: Employed preprocessing techniques to handle raw data.
- **Conversational AI**: Integrated a chatbot focused on recipe-related queries.

Technical Implementation

Data Management

- **Data Collection**: Used the Zomato dataset in JSON format containing restaurant and recipe details.
- **AWS S3**: Stored raw data in an S3 bucket with a structured folder organization for restaurants and recipes.
- **Preprocessing**:

    - Cleaned and transformed raw data into structured pandas DataFrames.
    - Handled missing and inconsistent values.
    - Normalized numeric data and applied one-hot encoding for categorical variables.
    - Extracted essential features like cuisine type, ratings, and geolocation for clustering.
- **Database Design**:
    - Designed and implemented relational schemas in AWS RDS to store cleaned data.
    - Created tables for entities such as restaurant details, cuisine types, and ratings for efficient querying.

Machine Learning

- **Model Selection**: Evaluated multiple clustering algorithms:
    - **K-Means**: Achieved the highest performance with a Silhouette Score of 0.912553 and a Calinski-Harabasz Score of 750248.478826.
    - **Hierarchical Clustering**: Performed moderately well.
    - **Gaussian Mixture Model (GMM)**: Underperformed with a negative Silhouette Score.
- **Model Evaluation Metrics**:

Model	Silhouette Score	Calinski Harabasz Score
K-Means	**0.912553**	**750248.478826**
Hierarchical	0.898095	586158.031909
GMM	-0.442073	62.998815

- **Model Deployment**: Stored the trained K-Means model in the application directory for real-time inference.

Application Development

- **Streamlit Interface**:
    - Developed a user-friendly application with real-time recommendations.
    - Integrated restaurant clustering to filter options by cuisine or dish type.
    - Implemented visualizations such as maps with interactive markers.
- **Chatbot Integration**:
    - Developed the "Chef Mate: Assistant" chatbot using conversational AI APIs.
    - Limited the chatbot to handle only recipe-related queries to maintain focus.
    - Linked the chatbot to recipe data stored in AWS RDS for dynamic responses.

Deployment

- Deployed the application on an **AWS EC2 instance** configured with required software (Python, Streamlit, libraries).
- Utilized AWS security groups to ensure secure HTTP access and controlled AWS service permissions.
- Optimized response times by caching frequent queries and refining preprocessing scripts.

Results

1. **Restaurant Recommendations**: Successfully implemented a clustering-based recommendation system.
2. **Interactive Visualizations**: Enabled users to explore restaurant insights through interactive maps and visual metrics.
3. **Real-Time Chatbot**: Provided users with accurate and responsive cooking guidance.
4. **Performance**: Achieved low-latency interactions and high recommendation accuracy.

Evaluation Metrics

- **Recommendation Accuracy**: Assessed through user feedback and satisfaction scores.
- **Chatbot Effectiveness**: Measured by the resolution rate of recipe queries.
- **Application Responsiveness**: Evaluated by response times for recommendations and chatbot queries.
- **User Engagement**: Monitored through interaction frequency and session duration.

Dataset Details

- **Source**: Zomato dataset in JSON format.
- **Key Variables**:
    - Restaurant ID, Name, Location, Cuisine, Ratings
    - Average Cost for Two, Price Range
    - Features (e.g., Table Booking, Online Delivery)
    - Longitude, Latitude
- **Preprocessing**:
    - Addressed missing values and inconsistencies.
    - Converted JSON data into SQL tables for integration.
    - Extracted essential features for the clustering model.

Future Enhancements

- Expand dataset to include user reviews for sentiment analysis.
- Improve chatbot capabilities with multilingual support and advanced natural language understanding.
- Integrate additional machine learning models for enhanced recommendation accuracy.
- Enable mobile-friendly application access for broader reach.

Conclusion

ChefMate successfully integrates machine learning, cloud computing, and conversational AI to provide an enhanced dining and cooking experience. By leveraging AWS services and Streamlit, the application delivers personalized restaurant recommendations and real-time recipe assistance. The project demonstrates expertise in machine learning, data preprocessing, cloud computing, and interactive application development.