**Chef Mate: Restaurant Clustering & Cooking Guide Application**

**Project Overview and Objective**

The **Chef Mate** project is a web-based application designed to solve key challenges in the food and beverage domain. Its primary objective is to provide personalized restaurant recommendations and interactive recipe guidance to enhance the user's dining and cooking experience.

**Problem Statement**

The problem revolves around enabling users to discover restaurants that match their preferences (e.g., cuisine, location, cost) while also assisting them in preparing meals through a chatbot. The project tackles the challenges of structuring unorganized data, efficient data preprocessing, and creating a user-friendly application with cloud support.

**Step-by-Step Approach**

1. **Unstructured to Structured Data**:
   * Transform raw JSON restaurant data into a structured format suitable for analysis and model training.
2. **Data Cleaning and Preprocessing**:
   * Handle missing values, remove duplicates, normalize data, and select relevant features for clustering and visualization.
3. **Machine Learning Model Building**:
   * Use clustering techniques (K-Means) to group similar restaurants and evaluate the optimal number of clusters.
4. **Streamlit Application**:
   * Develop an interactive web application for restaurant recommendations and recipe guidance.
5. **AWS Cloud Integration**:
   * Utilize AWS services for data storage, database management, and application hosting:
     + **S3**: Store and retrieve raw and cleaned datasets.
     + **RDS**: Maintain structured restaurant data for queries.
     + **EC2**: Host the Streamlit application for real-time access.

**Dataset**

* **Source**: Zomato dataset in JSON format.
* **Variables**:
  + Restaurant ID, Name, Location, Cuisine, Ratings, Average Cost for Two, Price Range, Features (e.g., Table Booking, Online Delivery), Longitude, Latitude.

**Data Preprocessing**

1. **Handling Missing Values**:
   * Filled missing values with default placeholders or dropped incomplete records.
2. **Normalization**:
   * Flattened nested JSON structures into tabular data using **pandas.json\_normalize ().**
3. **Feature Selection**:
   * Selected relevant columns for clustering, such as ratings, average cost, price range, and cuisine.
4. **Data Cleaning**:
   * Removed duplicates and inconsistent entries.
   * Renamed and reorganized columns for clarity.

**Machine Learning Model**

**Clustering**

* **Algorithm**: K-Means Clustering
* **Number of Clusters**: Determined using the **Silhouette Score** and **Elbow Method**.
* **Features Used**:
  + Ratings, Average Cost for Two, Price Range, Cuisine (encoded).

**Implementation Steps:**

1. Pre-processed data to scale numeric features using StandardScaler.
2. Applied K-Means clustering to group restaurants.
3. Visualized clusters using scatter plots and centroids.

**Streamlit Application**

**Features**

1. **Restaurant Recommendations**:
   * Sidebar filters for user preferences (e.g., location, price range).
   * User input for cuisines
   * Displays clustered restaurants with relevant details and ratings.
2. **Chatbot Integration**:
   * Provides cooking assistance using OpenAI GPT API.
   * Example Query: "How to cook butter chicken?"
3. **Dynamic Column Selection**:
   * Allows users to choose which restaurant attributes to display.

**Code Highlights**

* **Frontend**: Developed entirely in Streamlit.
* **Backend**:
  + Clustering logic implemented in Python using scikit-learn.
  + Data retrieved from AWS S3 and pre-processed dynamically.

**AWS Cloud Integration**

**AWS Services Used**

1. **S3**:
   * Stored raw JSON datasets and cleaned CSV files.
   * Used boto3 to upload and download files.
2. **RDS**:
   * Structured database storage for restaurant details, enabling efficient querying.
3. **EC2**:
   * Hosted the Streamlit application for real-time accessibility.

**Deployment Steps**

1. **Prepare EC2 Instance**:
   * Installed Python, Streamlit, and required libraries.
2. **Host Application**:
   * Ran Streamlit app on the EC2 instance and made it publicly accessible.

**Requirements**

* **Programming Languages**:
  + Python (for model building and application logic).
* **Libraries**:
  + pandas, scikit-learn, streamlit, boto3, openai.
* **Cloud Services**:
  + AWS S3, RDS, and EC2.
* **Dataset**:
  + Zomato restaurant dataset in JSON format.
* **Environment**:
  + AWS CLI and Python virtual environment.

**Conclusion**

The **Chef Mate** project successfully combines machine learning, cloud computing, and interactive web development to deliver a seamless restaurant recommendation and cooking assistance experience. By following a structured approach—from preprocessing raw data to deploying the application on AWS—this project addresses key challenges in the food and beverage domain and provides a robust platform for future enhancements.