**Restaurant Recommender AI**

**Prepared by: Arun Kumar Singh (**[**asingh.iitr@gmail.com**](mailto:asingh.iitr@gmail.com)**) Date: 14-Mar-24**

**Goal:**

Create a chatbot that helps users find the perfect restaurant for any occasion. It can consider factors like preferences, dietary restrictions, location (lat & long) and user ratings to make tailored recommendations.

**Github repo:** <https://github.com/arunksinghbuee/restaurant-recommendation-ai>

**Data Sources:**

For recommending suitable restaurant, following data is collected as sample dataset (72 restaurants data across Delhi, Gurgaon and Noida). It has following columns.

* Restaurant Name
* Occasion
* Cuisine Preference
* Dietary Restriction
* Lat
* Long
* Rating
* Location
* Address
* Lat & long would be used to find out nearest suitable restaurant available to the user.

Example: Sample data is as below.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Restaurant Name** | **Occasion** | **Cuisine Preference** | **Dietary Restriction** | **Lat** | **Long** | **Rating** | **Location** | **Address** |
| Delhi Darbar | Family Dinner | Indian | Vegetarian | 28.6139 | 77.209 | 4.5 | Delhi | 123 Main Street - Delhi |
| Spice Route | Business Meeting | Indian | Vegetarian | 28.4595 | 77.0266 | 4.2 | Gurgaon | 456 Park Avenue - Gurgaon |
| The Grill House | Date Night | Steakhouse | Gluten-Free | 28.7041 | 77.1025 | 4.7 | Delhi | 789 Elm Street - Delhi |
| Bella Cucina | Anniversary | Italian | Dairy-Free | 28.5355 | 77.391 | 4.4 | Noida | 101 Pine Street - Noida |
| Sushi Express | Casual Dining | Japanese | Nut-Free | 28.4595 | 77.0266 | 4 | Gurgaon | 222 Oak Avenue - Gurgaon |
| La Brasserie | Brunch | French | Shellfish-Free | 28.4595 | 77.0266 | 4.3 | Gurgaon | 333 Maple Drive - Gurgaon |
| Green Garden | Lunch Meeting | Vegetarian | Soy-Free | 28.6139 | 77.209 | 4.1 | Delhi | 444 Cedar Road - Delhi |

**Possible occasions value can be:**

* **Date Night:** Restaurants suitable for a romantic evening for couples.
* **Family Dinner:** Restaurants that are family-friendly and offer a variety of dishes to suit different tastes.
* **Business Meeting:** Restaurants with a quiet and professional atmosphere, suitable for meetings and discussions.
* **Birthday Celebration:** Restaurants that offer special menus or services for birthday parties.
* **Anniversary:** Restaurants that provide a special and romantic setting for couples celebrating their anniversary.
* **Casual Dining:** Restaurants where people can enjoy a relaxed meal with friends or family.
* **Fine Dining:** High-end restaurants with gourmet food and elegant ambiance, often for special occasions.
* **Brunch:** Restaurants that offer a brunch menu, typically on weekends, combining breakfast and lunch dishes.
* **Lunch Meeting:** Restaurants suitable for business or casual meetings during lunchtime.
* **Outdoor Dining:** Restaurants with outdoor seating, perfect for enjoying the weather and atmosphere.
* **Group Gathering:** Restaurants suitable for large groups, such as family reunions or gatherings with friends.
* **Holiday Celebration:** Restaurants that offer special menus or events for holidays like Christmas, Thanksgiving, or New Year's Eve.
* **Cocktail Party:** Restaurants with a lively bar area and a selection of cocktails, suitable for cocktail parties or social gatherings.
* **Business Lunch:** Restaurants that offer a quick and convenient lunch menu, suitable for business professionals.
* **Theme Night:** Restaurants that host themed nights, such as Italian night, seafood night, or live music nights.
* **Pre-Theater Dinner:** Restaurants located near theaters and offer a quick and delicious meal before a show.
* **After-Work Drinks:** Restaurants with a happy hour or special offers on drinks, suitable for after-work gatherings.

**Possible Cousine Preferences:**

* **American:** Classic American dishes such as burgers, fries, and barbecue.
* **Italian:** Italian cuisine, including pasta, pizza, and risotto.
* **Mexican:** Mexican dishes such as tacos, burritos, and enchiladas.
* **Chinese:** Chinese cuisine, including dishes like stir-fries, noodles, and dumplings.
* **Japanese:** Japanese dishes such as sushi, sashimi, and ramen.
* **Indian:** Indian cuisine, including curries, biryanis, and tandoori dishes.
* **French:** French dishes such as croissants, coq au vin, and escargot.
* **Mediterranean:** Mediterranean cuisine, including dishes from countries like Greece, Turkey, and Lebanon.
* **Thai:** Thai cuisine, including dishes like pad Thai, green curry, and tom yum soup.
* **Spanish:** Spanish dishes such as paella, tapas, and gazpacho.
* **Korean:** Korean cuisine, including dishes like kimchi, bulgogi, and bibimbap.
* **Vegetarian:** Restaurants that specialize in vegetarian or vegan dishes.
* **Gluten-Free:** Restaurants that offer gluten-free options for those with gluten sensitivities or celiac disease.
* **Seafood:** Restaurants that specialize in seafood dishes, including fish, shrimp, and shellfish.
* **Steakhouse:** Restaurants that focus on serving high-quality steaks and other meat dishes.

**Possible dietary restrictions:**

* **Vegetarian:** Dishes that do not contain meat or animal-derived ingredients. This can include lacto-vegetarian (dairy is allowed), ovo-vegetarian (eggs are allowed), or vegan (no animal products at all) options.
* **Vegan:** Dishes that do not contain any animal products, including meat, dairy, eggs, and honey.
* **Gluten-Free:** Dishes that do not contain gluten, a protein found in wheat, barley, and rye. This is important for people with celiac disease or gluten sensitivity.
* **Dairy-Free:** Dishes that do not contain dairy products, suitable for people who are lactose intolerant or have a dairy allergy.
* **Nut-Free:** Dishes that do not contain nuts or nut-derived ingredients, important for people with nut allergies.
* **Shellfish-Free:** Dishes that do not contain shellfish, important for people with shellfish allergies.
* **Soy-Free:** Dishes that do not contain soy or soy-derived ingredients, suitable for people with soy allergies or sensitivities.
* **Low-Carb:** Dishes that are low in carbohydrates, suitable for people following a low-carb or ketogenic diet.
* **Paleo:** Dishes that adhere to the paleo diet, which focuses on foods that would have been available to our ancestors, such as meat, fish, vegetables, and fruits, while excluding grains, legumes, dairy, and processed foods.
* **Allergen-Free:** Some restaurants may offer dishes that are free from common allergens, such as gluten, dairy, nuts, and soy, to accommodate customers with multiple allergies.

**Design System:**

**A diagram of a company

Description automatically generated**

1. LLM model starts conversation with user and try to deduce values of
   1. Occasion
   2. Cousion Preference
   3. Dietary Restriciton
   4. Location

By asking multiple questions.

1. Intent clarity layer tries to get user inputs multiple times, until it deduces values of above values. Values of above variables can only be out of above mentioned dataset.
2. Intent confirmation layer works as a LLM model which make sures that all required fields are filled with correct values.
3. Since we have extracted user profile data from user inputs, for location, it uses Geo Location API to get lat, long of the address of the user.
4. Data is loaded into panda data frames and data is filtered based on user’s occasion, cousion preference and dietary restrictions (Optional)
5. Since we have got filtered data which fulfills user profile, now its time to find out the nearest restaurant. For this, a new column is formed on filtered dataset, which denotes **geo-distance** of each hotel from user’s location using mathematical formula as below:



1. Filtered data is sorted using geo-distance column, order by ascending and returning nearest 3 restaurants.
2. Chatbot uses LLM to nudge user to select best suitable option for them.

**Key Design decisions:**

As shown in the image, the chatbot contains the following layers:

* Intent Clarity Layer
* Intent Confirmation Layer
* Restaurant Extraction Layer
* Restaurant Recommendation Layer

**Major functions used by chatbot:**

1. initialize\_conversation(): This initializes the variable conversation with the system message.
2. get\_chat\_model\_completions(): This takes the ongoing conversation as the input and returns the response by the assistant
3. moderation\_check(): This checks if the user's or the assistant's message is inappropriate. If any of these is inappropriate, it ends the conversation.
4. intent\_confirmation\_layer(): This function takes the assistant's response and evaluates if the chatbot has captured the user's profile clearly. Specifically, this checks if the following properties for the user has been captured or not

Occasion, Cousion Preference, Dietary Restriction, Location

1. dictionary\_present(): This function checks if the final understanding of user's profile is returned by the chatbot as a python dictionary or not. If there is a dictionary, it extracts the information as a Python dictionary.
2. Compare\_restaurants\_with\_user(): This function compares the user's profile with the different restaurants and come back with the top 3 recommendations.
3. initialize\_conv\_reco(): Initializes the recommendations conversation
4. **initialize\_conversation():** This initializes the variable conversation with the system message. Using prompt engineering and chain of thought reasoning, the function will enable the chatbot to keep asking questions until the user requirements have been captured in a dictionary. It also includes Few Shot Prompting(sample conversation between the user and assistant) to align the model about user and assistant responses at each step.



1. **moderation\_check():** This checks if the user's or the assistant's message is inappropriate.

****

1. **intent\_confirmation\_layer():** This function takes the assistant's response and evaluates if the chatbot has captured the user's profile clearly. Specifically, this checks if the following properties for the user has been captured or not.
   1. Occasion
   2. Cousion Preference
   3. Dietary Restriciton
   4. Location



1. **dictionary\_present()**: This function checks if the final understanding of user's profile is returned by the chatbot is a Python dictionary or not. This is important as it'll be used later on for finding the right laptops using dictionary matching.



1. **extract\_dictionary\_from\_string():** This function takes in the output of the previous layer and extracts the user requirements dictionary



1. compare\_laptops\_with\_user(): This function compares the user's profile with the different laptops and come back with the top recommendations. It will perform the following steps:

* It will take the user requirements dictionary as input
* Filter the restaurants based on their Occasion and Cuisine Preferences. If Dietary Restrictions data is available in user profile, then the same is also compared.
* A new column is formed on filtered dataset, which denotes **geo-distance** of each hotel from user’s location using mathematical formula using Haversine formula.
* Sort the restaurants based on their geo-distance in ascending order and rating in descending order.
* Return the top 3 restaurants as a JSON-formatted string.



1. Finally, in restaurant recommendation layer, it takes the output from the compare\_restaurants\_with\_user function in the previous layer and provides the recommendations to the user. It has the following steps.

* Initialize the conversation for recommendation.
* Generate the recommendations and display in a presentable format.
* Ask questions basis the recommendations.



1. dialogue\_mgmt\_system(): This contains the logic of how the different layers would interact with each other. This will be the function that is called to initiate the chatbot.



**Sample Example:**

**A screenshot of a chat

Description automatically generated**

**A screenshot of a chat

Description automatically generated**

**A screenshot of a chat

Description automatically generated**