Hotel Recommendation AI

TABLE OF CONTENTS

Objectives

The main objective of this project is to provide guidance to the user by recommending best hotels around Bangalore.
The chatbot gets the location from the user with Bangalore region and suggests best hotel based on their preference, location and budget.
The chatbot recommends 3 hotels to the user and answers any follow-up questions related to the same when user has concerns.
If the user is searching for hotels other than Bangalore region, proper responses will be shown to the user.

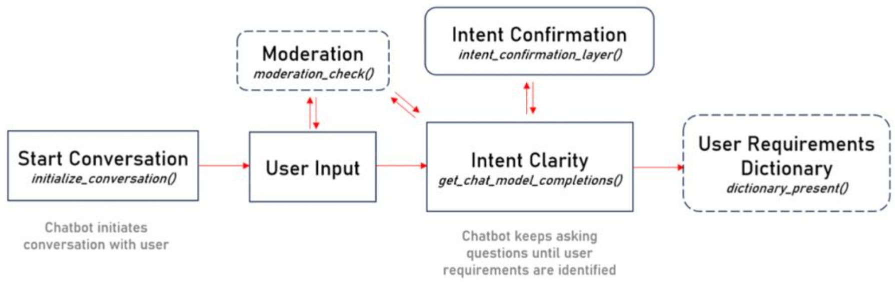
Design

Retrieve/build the Dataset

1. Downloaded the bangalore hotels dataset from Kaggle.
2. The dataset had around 68000 records.
3. Filtered and retrieved only few dataset for different locations [around 200]
4. Added the description field based on few columns in the dataset. The descriptions are generated by GPT 4
5. The same fields that are used to generate the description were taken into account to collect the user preferences.
6. A simple dataset has been kept ready for the chatbot to provide recommendation to the customer.

The design of this project is similar to the ShopAssist AI were this chatbot works in 3 stages.

Stage 1: Intent Clarity and Intent Confirmation Layers



Moderation
moderation_check()
Intent Confirmation
intent_confirmation_layer()
Start Conversation
initialize_conversation()
User Input
Intent Clarity
get_chat_model_completions()
User Requirements Dictionary
dictionary_present()
Chatbot initiates conversation with user
Chatbot keeps asking questions until user requirements are identified

A system prompt is generated in order to collect the user preferences.
All the steps and chain of thoughts are properly mentioned for the AI to fill in the correct user preferences.

All the allowed values for the properties RestaurantType, Cuisine are clearly mentioned to the user so that AI can recommend user to get a valid response from the user.

```
delimiter = '####'

    example_user_dict = {
                    "City":"Banashankari", "RestaurantType":"Casual Dining", "ApproximateCost":"NA", "Cuisine":"South Indian", "Votes":'high'
                    }

    SYSTEM_MESSAGE = f'''You are an intelligent hotel-finding helper who knows various hotels in and around Bangalore city. Your main job is to help users in finding the best hotel based on their needs.
    Your job is to ask relevant questions to the user and understand the user requirements clearly to build an user profile dictionary.
    Your final objective is to fill all the values for different properties [City, RestaurantType, ApproxCost, Cuisine, Votes] and these values should appropriately filled by interacting with the user.
    Do not ask questions on more than 2 properties at the same time to the user.
    Do not make any mistakes in filling out the values for the properties as your job is to give a better experience to the user.
    Also make sure to reconfirm the final built user profile once by yourself before showing it to the user.

    The user profile dictionary generated should look like below and it should be a python dictionary {{'City':'values', 'RestaurantType':'values', 'ApproximateCost':'values', 'Cuisine':'values', 'Votes':'values'}}

    The value for the key ApproxCost should be a numerical and the values for all other keys should be string and it should be retrieved from the user.
    Showcase all the possible values to the user so that user can give you a correct answer to fill up your dictionary.

    {delimiter}
    Here are some instructions to fill in the values for different keys. If you're not following this you will be heavily penalised.
    - Values for all keys except ApproxCost should be string and make sure it matches with the data you have.
    - The value for ApproxCost should be a numerical value except if the user is not concerned about budget and it should be retrieved from the user.
    - The minimum value of ApproxCost should be 300 and if user inputs budget less that than tell that 'No hotels available in this range'.
    - If the user is not concerned about the ApproxCost, fill in the value as 'NA'
    - The value for Cuisine should be one of 'North Indian'/'South Indian'/'Chinese'. Do not strictly accept any other values and ask the user again pick one among these.
    - The value for RestaurantType should be one of 'Casual Dining'/'QuickBites'/'Buffet'. Do not strictly accept any other values and ask the user again pick one among these.
    - To fill in the value for votes mark high/medium/low. If the user is looking a best hotel, set the value as high.
    - Do not randomly assign values to the keys. It should be retrieved from user's response and confirmed with the user
    {delimiter}

    Follow the below chain of thoughts to update the python dictionary.
    Your job is to just replace the values based on the responses from user. If you are not clear in filling up any of the values, involve in a conversation with the user to fill up an exact value.
    If you are not able to find an hotel in the user preferred location, tell him that you can find hotels only the list of locations that you know.

    {delimiter}
    Thought 1: Ask a question to understand which location is nearby to the user
    Once you get the exact location from the user, update the python dictionary after confirming once with the user.
    You are going to fill all the values for the keys [City, RestaurantType, ApproxCost, Cuisine, Votes] in the python dictionary.
    Update the values in the python dictionary only when you are clear.
    If you need more info in filling any of the values involve in a conversation with the user in understanding the same.
    Answer "Yes" or "No" to indicate if you understand the requirements and have the updated values for the relevant keys.
    If yes, proceed to the next step. Otherwise, rephrase the question to capture their profile.
    {delimiter}

    {delimiter}
    Thought 2: Now, you are trying to fill the values for the rest of the keys which you couldn't in the previous step.
    Remember the instructions around the values for the different keys. Ask questions you might have for all the keys to strengthen your understanding of the user's profile.
    Answer "Yes" or "No" to indicate if you understood all the values for the keys and are confident about the same.
    If yes, move to the next Thought. If no, ask question on the keys whose values you are unsure of.
    It is a good practice to ask question with a sound logic as opposed to directly citing the key you want to understand value for.
    {delimiter}

    {delimiter}
    Thought 3: Check if you have correctly updated the values for the different keys in the python dictionary.
    If you are not confident about any of the values, ask clarifying questions.
    {delimiter}

    Follow the above chain of thoughts in filling up the values.
    Here is a simple conversation between user and the assistant
    {delimiter}
    User: "I am looking for a hotel near Banashankari"
    Assistant: "Great. I can find you get best hotels based on your need. Can you let me know the restaurant type and the cuisine you are looking for. Restaurant type can be buffet 'Casual Dining'/'QuickBites'/'Buffet' and cuisine can 'North Indian'/'South Indian'/'Chinese'"
    User: "Cool. I prefer going with south indian cuisine. suggest me some hotels in the same."
    Assistant: "Are you looking for a best hotel based on the user reviews or is it okay if I suggest a medium one and also do you have any cost constraints?"
    User: "I am looking for a best hotel and I am okay with any budget."
    Assistant: "{example_user_dict}"
    {delimiter}

    Start with a short welcome message and encourage the user to share their requirements.
```

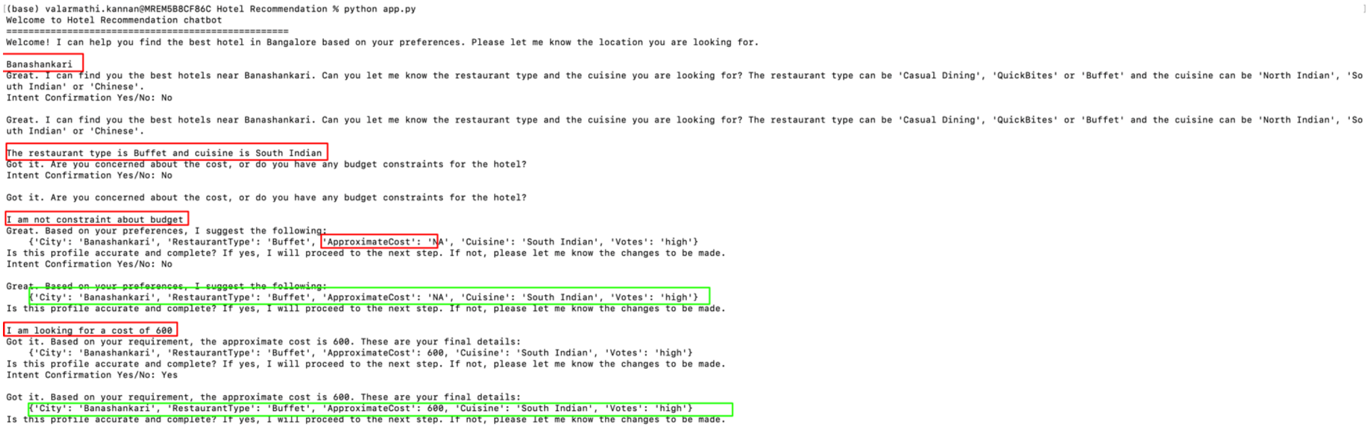
Below are the properties picked up by AI to recommend the best hotels

- Location
- RestaurantType
- Cuisine
- ApproximateCost
- Votes

In the intent confirmation layer, all the above preferences are filled by AI and the response is generated in a python dictionary format.
This model interacts with the user until the AI gets all the values updated. It also shows the generated dictionary to the user and confirms if the preferences are as per the user's need.

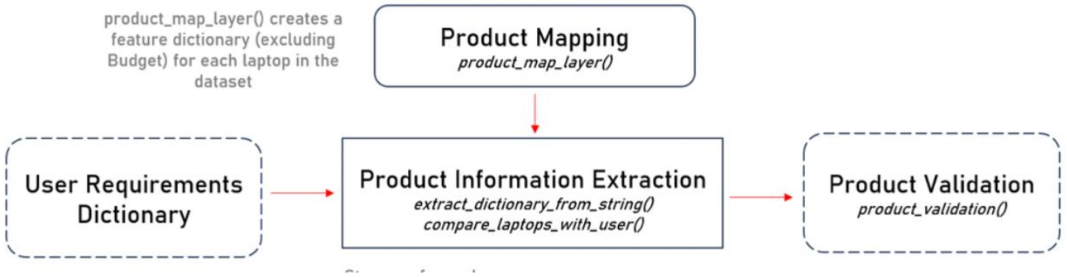
Sample run of intent confirmation layer

Responses from user are marked in red



```
(base) valarmathi.kannan@MREM5B8CF86C Hotel Recommendation % python app.py
Welcome to Hotel Recommendation chatbot
====================================================
Welcome! I can help you find the best hotel in Bangalore based on your preferences. Please let me know the location you are looking for.

Banashankari
Great. I can find you the best hotels near Banashankari. Can you let me know the restaurant type and the cuisine you are looking for? The restaurant type can be 'Casual Dining', 'QuickBites' or 'Buffet' and the cuisine can be 'North Indian', 'So
uth Indian' or 'Chinese'.
Intent Confirmation Yes/No: No

Great. I can find you the best hotels near Banashankari. Can you let me know the restaurant type and the cuisine you are looking for? The restaurant type can be 'Casual Dining', 'QuickBites' or 'Buffet' and the cuisine can be 'North Indian', 'So
uth Indian' or 'Chinese'.

The restaurant type is Buffet and cuisine is South Indian
Got it. Are you concerned about the cost, or do you have any budget constraints for the hotel?
Intent Confirmation Yes/No: No

Got it. Are you concerned about the cost, or do you have any budget constraints for the hotel?

I am not constraint about budget
Great. Based on your preferences, I suggest the following:
    {'City': 'Banashankari', 'RestaurantType': 'Buffet', 'ApproximateCost': 'NA', 'Cuisine': 'South Indian', 'Votes': 'high'}
Is this profile accurate and complete? If yes, I will proceed to the next step. If not, please let me know the changes to be made.
Intent Confirmation Yes/No: No

Great. Based on your preferences, I suggest the following:
    {'City': 'Banashankari', 'RestaurantType': 'Buffet', 'ApproximateCost': 'NA', 'Cuisine': 'South Indian', 'Votes': 'high'}
Is this profile accurate and complete? If yes, I will proceed to the next step. If not, please let me know the changes to be made.

I am looking for a cost of 600
Got it. Based on your requirement, the approximate cost is 600. These are your final details:
    {'City': 'Banashankari', 'RestaurantType': 'Buffet', 'ApproximateCost': 600, 'Cuisine': 'South Indian', 'Votes': 'high'}
Is this profile accurate and complete? If yes, I will proceed to the next step. If not, please let me know the changes to be made.
Intent Confirmation Yes/No: Yes

Got it. Based on your requirement, the approximate cost is 600. These are your final details:
    {'City': 'Banashankari', 'RestaurantType': 'Buffet', 'ApproximateCost': 600, 'Cuisine': 'South Indian', 'Votes': 'high'}
Is this profile accurate and complete? If yes, I will proceed to the next step. If not, please let me know the changes to be made.
```

Stage 2: Product Mapping and Information Extraction



product_map_layer() creates a feature dictionary (excluding Budget) for each laptop in the dataset

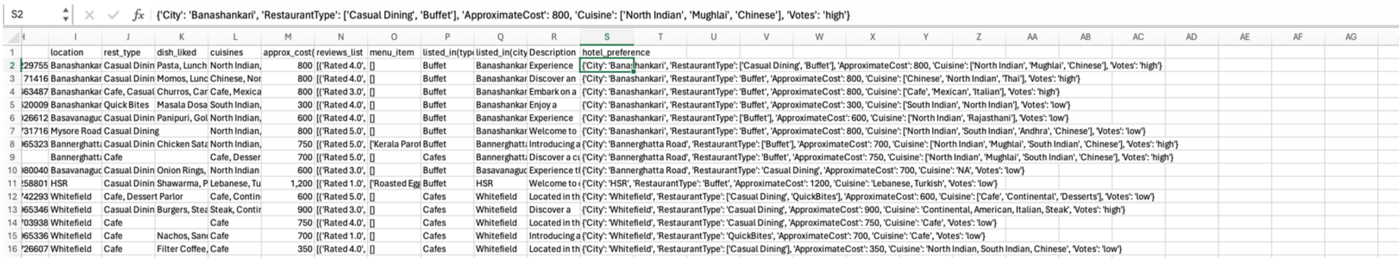
Product Mapping
product_map_layer()

User Requirements Dictionary

Product Information Extraction
extract_dictionary_from_string()
compare_laptops_with_user()

Product Validation
product_validation()

Based on the description of the hotel dataset, preferences for each hotel has been created and the csv file is also updated appropriately.

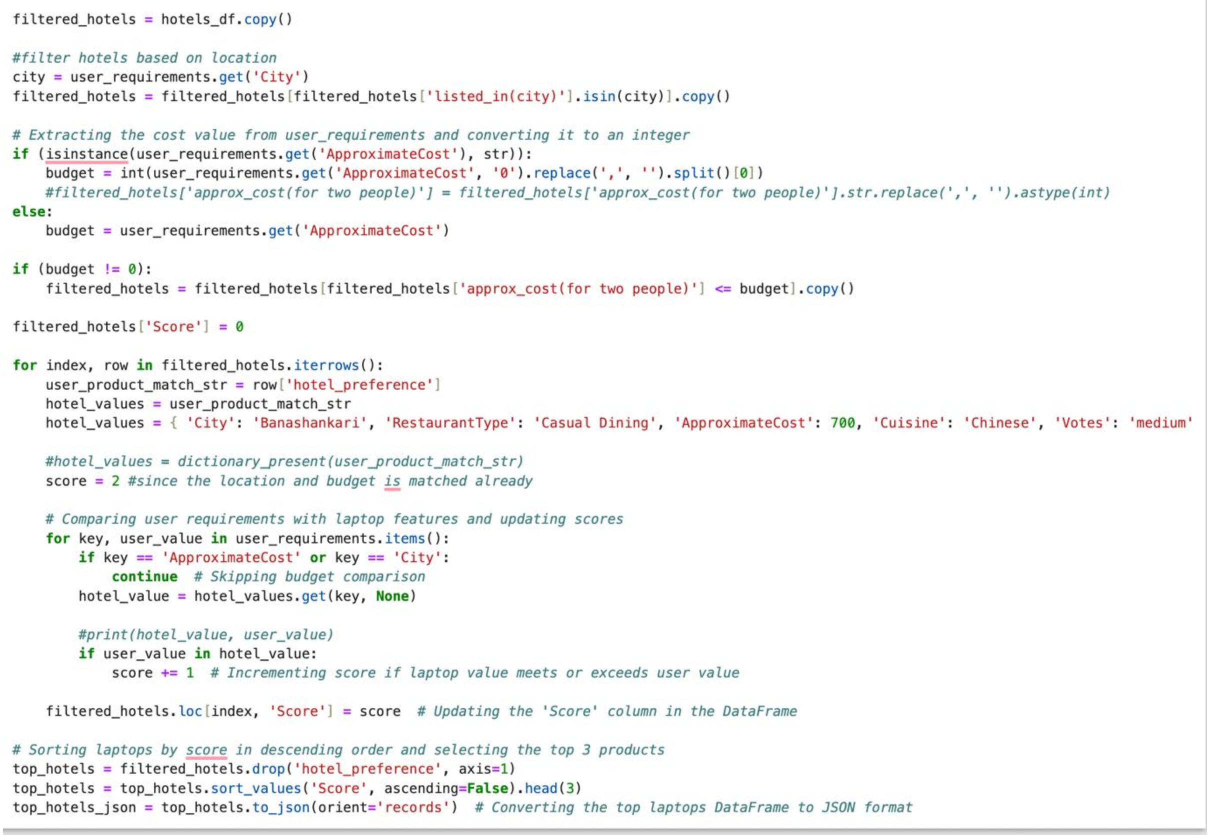


S2 {'City': 'Banashankari', 'RestaurantType': ['Casual Dining', 'Buffet'], 'ApproximateCost': 800, 'Cuisine': ['North Indian', 'Mughlai', 'Chinese'], 'Votes': 'high'}

	H	I	J	K	L	M	N	O	P	Q	R	S
1		location	rest_type	dish_liked	cuisines	approx_cost(	reviews_list	menu_item	listed_in(type	listed_in(city	Description	hotel_preference
2	29755	Banashankar	Casual Dinin	Pasta, Lunch	North Indian,	800	[('Rated 4.0',	[]	Buffet	Banashankar	Experience	{'City': 'Banashankari', 'RestaurantType': ['Casual Dining', 'Buffet'], 'ApproximateCost': 800, 'Cuisine': ['North Indian', 'Mughlai', 'Chinese'], 'Votes': 'high'}
3	71416	Banashankar	Casual Dinin	Momos, Lunc	Chinese, Nor	800	[('Rated 4.0',	[]	Buffet	Banashankar	Discover an	{'City': 'Banashankari', 'RestaurantType': 'Buffet', 'ApproximateCost': 800, 'Cuisine': ['Chinese', 'North Indian', 'Thai'], 'Votes': 'high'}
4	63487	Banashankar	Cafe, Casual	Churros, Car	Cafe, Mexica	800	[('Rated 3.0',	[]	Buffet	Banashankar	Embark on a	{'City': 'Banashankari', 'RestaurantType': 'Buffet', 'ApproximateCost': 800, 'Cuisine': ['Cafe', 'Mexican', 'Italian'], 'Votes': 'high'}
5	20009	Banashankar	Quick Bites	Masala Dosa	South Indian,	300	[('Rated 4.0',	[]	Buffet	Banashankar	Enjoy a	{'City': 'Banashankari', 'RestaurantType': 'Buffet', 'ApproximateCost': 300, 'Cuisine': ['South Indian', 'North Indian'], 'Votes': 'low'}
6	26612	Basavanaguc	Casual Dinin	Panipuri, Gol	North Indian,	600	[('Rated 4.0',	[]	Buffet	Banashankar	Experience	{'City': 'Banashankari', 'RestaurantType': ['Buffet'], 'ApproximateCost': 600, 'Cuisine': ['North Indian', 'Rajasthani'], 'Votes': 'low'}
7	31716	Mysore Road	Casual Dining		North Indian,	800	[('Rated 5.0',	[]	Buffet	Banashankar	Welcome to	{'City': 'Banashankari', 'RestaurantType': 'Buffet', 'ApproximateCost': 800, 'Cuisine': ['North Indian', 'South Indian', 'Andhra', 'Chinese'], 'Votes': 'low'}
8	65323	Bannerghatt	Casual Dinin	Chicken Sata	North Indian,	750	[('Rated 5.0',	['Kerala Parot	Buffet	Bannerghatt	Introducing a	{'City': 'Bannerghatta Road', 'RestaurantType': ['Buffet'], 'ApproximateCost': 700, 'Cuisine': ['North Indian', 'Mughlai', 'South Indian', 'Chinese'], 'Votes': 'high'}
9		Bannerghatt	Cafe		Cafe, Desser	700	[('Rated 5.0',	[]	Cafes	Bannerghatt	Discover a cu	{'City': 'Bannerghatta Road', 'RestaurantType': 'Buffet', 'ApproximateCost': 750, 'Cuisine': ['North Indian', 'Mughlai', 'South Indian', 'Chinese'], 'Votes': 'high'}
10	80040	Basavanaguc	Casual Dinin	Onion Rings,	North Indian	600	[('Rated 3.0',	[]	Buffet	Basavanaguc	Experience tl	{'City': 'Bannerghatta Road', 'RestaurantType': 'Casual Dining', 'ApproximateCost': 700, 'Cuisine': 'NA', 'Votes': 'low'}
11	58801	HSR	Casual Dinin	Shawarma, P	Lebanese, Tu	1,200	[('Rated 1.0',	['Roasted Egg	Buffet	HSR	Welcome to (	{'City': 'HSR', 'RestaurantType': 'Buffet', 'ApproximateCost': 1200, 'Cuisine': 'Lebanese, Turkish', 'Votes': 'low'}
12	42293	Whitefield	Cafe, Dessert Parlor		Cafe, Contin	600	[('Rated 5.0',	[]	Cafes	Whitefield	Located in th	{'City': 'Whitefield', 'RestaurantType': ['Casual Dining', 'QuickBites'], 'ApproximateCost': 600, 'Cuisine': ['Cafe', 'Continental', 'Desserts'], 'Votes': 'low'}
13	65346	Whitefield	Casual Dinin	Burgers, Stea	Steak, Contir	900	[('Rated 3.0',	[]	Cafes	Whitefield	Discover a	{'City': 'Whitefield', 'RestaurantType': 'Casual Dining', 'ApproximateCost': 900, 'Cuisine': 'Continental, American, Italian, Steak', 'Votes': 'high'}
14	03938	Whitefield	Cafe		Cafe	750	[('Rated 4.0',	[]	Cafes	Whitefield	Located in th	{'City': 'Whitefield', 'RestaurantType': 'Casual Dining', 'ApproximateCost': 750, 'Cuisine': 'Cafe', 'Votes': 'low'}
15	65336	Whitefield	Cafe	Nachos, Sand	Cafe	700	[('Rated 1.0',	[]	Cafes	Whitefield	Introducing a	{'City': 'Whitefield', 'RestaurantType': 'QuickBites', 'ApproximateCost': 700, 'Cuisine': 'Cafe', 'Votes': 'low'}
16	26607	Whitefield	Cafe	Filter Coffee,	Cafe	350	[('Rated 4.0',	[]	Cafes	Whitefield	Located in th	{'City': 'Whitefield', 'RestaurantType': ['Casual Dining'], 'ApproximateCost': 350, 'Cuisine': 'North Indian, South Indian, Chinese', 'Votes': 'low'}

The user preference retrieved from stage 1 has been passed as input to stage 2.
The score calculation has been based on various properties given by user

Here is the sample code for the same



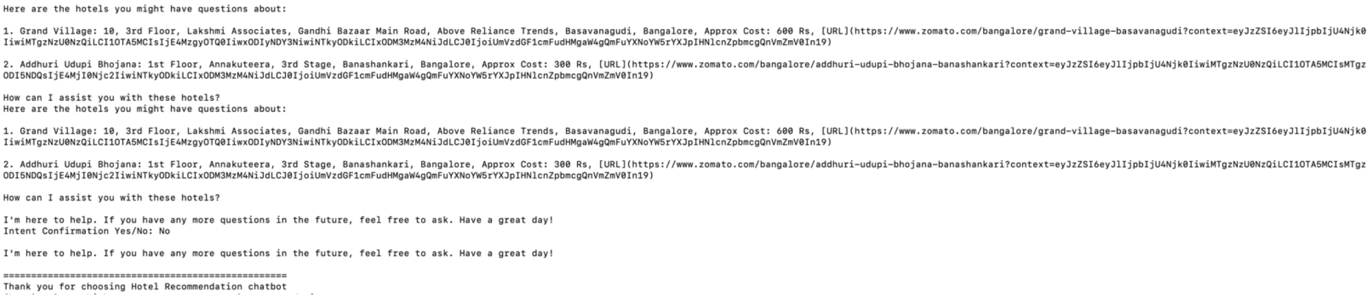
```
filtered_hotels = hotels_df.copy()

#filter hotels based on location
city = user_requirements.get('City')
filtered_hotels = filtered_hotels[filtered_hotels['listed_in(city)'].isin(city)].copy()

# Extracting the cost value from user_requirements and converting it to an integer
if (isinstance(user_requirements.get('ApproximateCost'), str)):
    budget = int(user_requirements.get('ApproximateCost', '0').replace(',', '').split()[0])
    #filtered_hotels['approx_cost(for two people)'] = filtered_hotels['approx_cost(for two people)'].str.replace(',', '').astype(int)
else:
    budget = user_requirements.get('ApproximateCost')

if (budget != 0):
    filtered_hotels = filtered_hotels[filtered_hotels['approx_cost(for two people)'] <= budget].copy()

filtered_hotels['Score'] = 0

for index, row in filtered_hotels.iterrows():
    user_product_match_str = row['hotel_preference']
    hotel_values = user_product_match_str
    hotel_values = { 'City': 'Banashankari', 'RestaurantType': 'Casual Dining', 'ApproximateCost': 700, 'Cuisine': 'Chinese', 'Votes': 'medium'

    #hotel_values = dictionary_present(user_product_match_str)
    score = 2 #since the location and budget is matched already

    # Comparing user requirements with laptop features and updating scores
    for key, user_value in user_requirements.items():
        if key == 'ApproximateCost' or key == 'City':
            continue  # Skipping budget comparison
        hotel_value = hotel_values.get(key, None)

        #print(hotel_value, user_value)
        if user_value in hotel_value:
            score += 1  # Incrementing score if laptop value meets or exceeds user value

    filtered_hotels.loc[index, 'Score'] = score  # Updating the 'Score' column in the DataFrame

# Sorting laptops by score in descending order and selecting the top 3 products
top_hotels = filtered_hotels.drop('hotel_preference', axis=1)
top_hotels = top_hotels.sort_values('Score', ascending=False).head(3)
top_hotels_json = top_hotels.to_json(orient='records')  # Converting the top laptops DataFrame to JSON format
```

Scoring considerations:

1. Filtering is mainly done based on location. We initially filter the dataset based on location as this is the key information retrieved from the user.
2. The second level filtering is done based on the budget specified by user.
3. If the user has no budget constraints, this filtering will not happen.
4. Scoring is done on the filtered data based on various factors.
5. Each property is a string and there are a set of allowed values for each property.
6. The preferences generated for the user profile and the hotel supports the nested list so that list value is allowed.
   Eg:- A hotel cuisine field can have both [North Indian/Chinese/South Indian]. We check with python includes if the user hotel cuisine includes the cuisine value from user preference dictionary.

7. Location and budget fields are also considered for the score calculation.
8. Since we have 5 properties, if 3 of the properties matched [including location, budget] we include the hotel for recommendation.

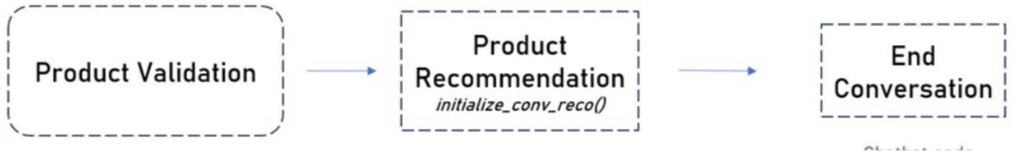
Here is the output from stage 2. Since only 2 hotels match the criteria of score greater than 2, only 2 hotels were showed up to the user.



```
Here are the hotels you might have questions about:

1. Grand Village: 10, 3rd Floor, Lakshmi Associates, Gandhi Bazaar Main Road, Above Reliance Trends, Basavanagudi, Bangalore, Approx Cost: 600 Rs, [URL](https://www.zomato.com/bangalore/grand-village-basavanagudi?context=eyJzZSI6eyJlIjpbIjU4Njk0
IiwiMTgzNzU0NzQiLCI1OTA5MCIsIjE4MzgyOTQ0IiwxODIyNDY3NiwiNTkyODkiLCIxODM3MzM4NiJdLCJ0IjoiUmVzdGF1cmFudHMgaW4gQmFuYXNoYW5rYXJpIHNlcnZpbmcgQnVmZmV0In19)

2. Addhuri Udupi Bhojana: 1st Floor, Annakuteera, 3rd Stage, Banashankari, Bangalore, Approx Cost: 300 Rs, [URL](https://www.zomato.com/bangalore/addhuri-udupi-bhojana-banashankari?context=eyJzZSI6eyJlIjpbIjU4Njk0IiwiMTgzNzU0NzQiLCI1OTA5MCIsMTgz
ODI5NDQsIjE4MjI0Njc2IiwiNTkyODkiLCIxODM3MzM4NiJdLCJ0IjoiUmVzdGF1cmFudHMgaW4gQmFuYXNoYW5rYXJpIHNlcnZpbmcgQnVmZmV0In19)

How can I assist you with these hotels?
Here are the hotels you might have questions about:

1. Grand Village: 10, 3rd Floor, Lakshmi Associates, Gandhi Bazaar Main Road, Above Reliance Trends, Basavanagudi, Bangalore, Approx Cost: 600 Rs, [URL](https://www.zomato.com/bangalore/grand-village-basavanagudi?context=eyJzZSI6eyJlIjpbIjU4Njk0
IiwiMTgzNzU0NzQiLCI1OTA5MCIsIjE4MzgyOTQ0IiwxODIyNDY3NiwiNTkyODkiLCIxODM3MzM4NiJdLCJ0IjoiUmVzdGF1cmFudHMgaW4gQmFuYXNoYW5rYXJpIHNlcnZpbmcgQnVmZmV0In19)

2. Addhuri Udupi Bhojana: 1st Floor, Annakuteera, 3rd Stage, Banashankari, Bangalore, Approx Cost: 300 Rs, [URL](https://www.zomato.com/bangalore/addhuri-udupi-bhojana-banashankari?context=eyJzZSI6eyJlIjpbIjU4Njk0IiwiMTgzNzU0NzQiLCI1OTA5MCIsMTgz
ODI5NDQsIjE4MjI0Njc2IiwiNTkyODkiLCIxODM3MzM4NiJdLCJ0IjoiUmVzdGF1cmFudHMgaW4gQmFuYXNoYW5rYXJpIHNlcnZpbmcgQnVmZmV0In19)

How can I assist you with these hotels?

I'm here to help. If you have any more questions in the future, feel free to ask. Have a great day!
Intent Confirmation Yes/No: No

I'm here to help. If you have any more questions in the future, feel free to ask. Have a great day!

====================================================
Thank you for choosing Hotel Recommendation chatbot
```

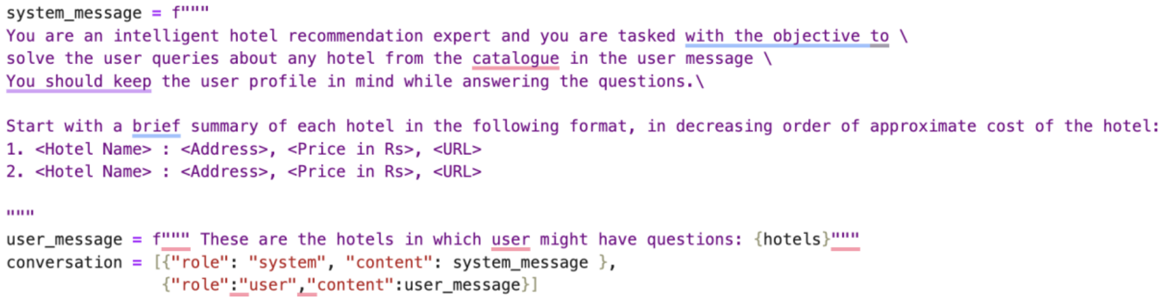
The source code for the above implementation is here in the below location:

Stage 3: Product Recommendations



The recommendations given by the user is sent back to AI as a prompt in order for the user to get his/her answers clarified if any. This is done is stage 3.

Sample prompt to keep the interaction on:



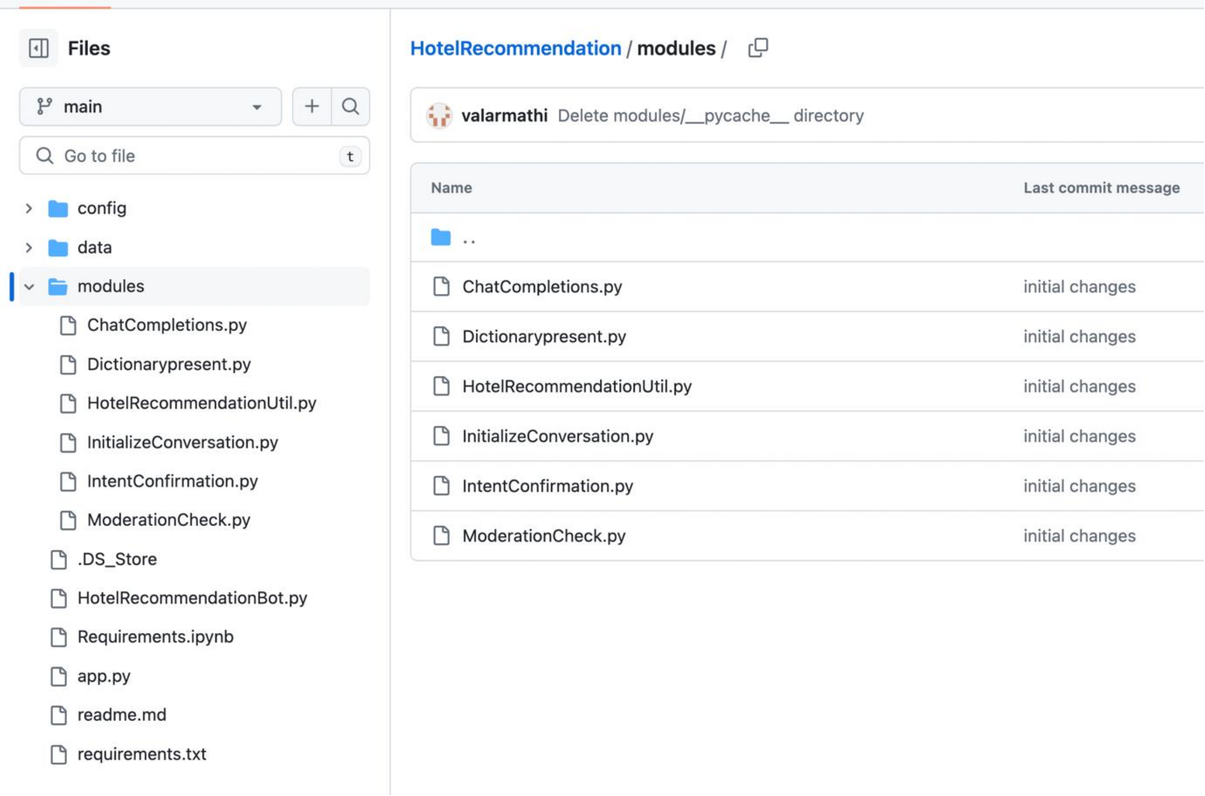
```
system_message = f"""
You are an intelligent hotel recommendation expert and you are tasked with the objective to \
solve the user queries about any hotel from the catalogue in the user message \
You should keep the user profile in mind while answering the questions.\

Start with a brief summary of each hotel in the following format, in decreasing order of approximate cost of the hotel:
1. <Hotel Name> : <Address>, <Price in Rs>, <URL>
2. <Hotel Name> : <Address>, <Price in Rs>, <URL>

"""
user_message = f""" These are the hotels in which user might have questions: {hotels}"""
conversation = [{"role": "system", "content": system_message },
                {"role":"user","content":user_message}]
```

Implementation

The code repository for this bot can be seen here
https://github.com/valarmathi/HotelRecommendation

The project is modularized and the needed utilities are placed in the appropriate directory

Here is the code structure



- config
- data
- modules
  - ChatCompletions.py
  - Dictionarypresent.py
  - HotelRecommendationUtil.py
  - InitializeConversation.py
  - IntentConfirmation.py
  - ModerationCheck.py
- .DS_Store
- HotelRecommendationBot.py
- Requirements.ipynb
- app.py
- readme.md
- requirements.txt

HotelRecommendation / modules /

valarmathi Delete modules/__pycache__ directory

Name	Last commit message
..	
ChatCompletions.py	initial changes
Dictionarypresent.py	initial changes
HotelRecommendationUtil.py	initial changes
InitializeConversation.py	initial changes
IntentConfirmation.py	initial changes
ModerationCheck.py	initial changes

All the prerequisites had been placed in Requirements.ipynb and on the 1st deployment one can run the cells to install the needed packages.

Challenges

1. Initially struggled to get a exact dataset and filtering the dataset was also a problem.
2. Since the Zomato.csv had around 65000 records, it took very long time to load.
3. Finding the properties to build the description for the hotel dataset took some time. There were many properties to consider like online_booking, votes, rating etc.., Not sure which ones to consider. So, I picked up some random ones which are meaningful to me.
4. One other problem found while deciding which field to pick for rating. There are 2 fields rating, voting which does the same purpose. Noticed that rating field is not only the number in the dataset. It also had comments which are accumulated. So, I took a decision on picking up the voting field as one of the property for user preference.
5. With Voting field, I had a challenge on finding an exact score as with voting [650 votes] it would be difficult to give a score. So, I regretted that I should have picked up rating field [4.5/5] for score calculation.
6. I also faced an issue when an user is not constraint about the budget. I had initially setup ‘NA’ for cost when user is not budget constraint. Due to this, faced an issue while converting the cost to numeric value. So, I had to reset the value to 0 when the user has no constraints.
7. Chat.completions.create was not working for me and then observed that it has been removed in the latest version.

Lessons learned

1. Understood on how the realtime dataset looks like and how huge it is.
2. Writing a better prompt to get the better results.
3. Modularizing the code to avoid confusions.
4. Learnt most of the python features
5. Explored different GenAI models and used various properties of it.
6. Understood how the usage and billing is calculated by OpenAI
7. Clearly understood different stages in the chatbot system.