جامعة دمشق
كلية هندسة المعلوماتية
السنة الخامسة
اختصاص الذكاء الصنعي
2024/6/22

Restaurants Recommendation
Fuzzy Expert System

By:

عبد الله الزبداني

عبد الرؤوف حسحس

طوني بطرس

محمد عثمان دياربكرلي

Supervisor:

م.عليا حموي

Contents

Figure table

Introduction to Fuzzy Expert Systems for Restaurant Recommendations

Application to Restaurant Recommendations

Choosing a restaurant involves considering multiple factors that are often subjective and interdependent. The main criteria typically include:

1. **Cuisine**: The type of food served (e.g., Italian, Chinese, Mexican).
2. **Price**: The cost of dining (e.g., budget, mid-range, upscale).
3. **Location**: The geographical proximity or convenience of the restaurant.

These factors are inherently fuzzy, as individuals may have varying preferences and tolerances. For instance, what one person considers "affordable" might be "expensive" to another, and the desirability of a cuisine type can vary greatly among different users.

How the Fuzzy Expert System Works

A fuzzy expert system for restaurant recommendation comprises several key components:

1. **Fuzzification**: Converts crisp inputs (e.g., exact price, distance) into fuzzy values. For example, a cost of 2000sp might be considered partially "cheap" and partially "moderate."
2. **Rule Base**: Contains a set of fuzzy rules that capture expert knowledge. These rules describe the relationships between different fuzzy variables (e.g., "If the cuisine is Italian and the price is moderate, then the restaurant is recommended").
3. **Inference Engine**: Evaluates the fuzzy rules and combines the results to determine the degree of recommendation for each restaurant.
4. **Defuzzyfication**: Converts the fuzzy output back into a crisp value or a ranked list of recommendations.

How the system work

we have two data sources we to take our information, the user enters three different inputs, cuisine, budget (price), and the neighborhood (location) that the user is in, based on cuisine, the system gets the restaurants that serve that cuisine, then do some calculations with the other two inputs, after that, the input is ready for the fuzzy system, the result is a list of all the restaurants that serve that cuisine, with additional information like how far the restaurant, and based on the rules the system have, the user get a recommendation for each restaurant.

Datasets

we used two types of datasets one for the restaurants that contained all the restaurants we collected with their needed information, especially the cuisine, the average meal price, and restaurant location on the map (Latitude, Longitude), among other information required for display like the restaurant name, name of its location, and the other dataset is the locations of few neighborhoods in the city Damascus required for the input of the system.

Fuzzification

Cuisine

we didn't turn the cuisine into a fuzzy set, we made it more like a list of choices, however, we had an idea to turn the cuisine into a fuzzy set by finding correlations between different cuisines like Lebanese cuisine and Syrian cuisine have similarities, but that idea would be more logical if we were choosing the dish instead of the cuisine, which in this case the system would be more complex but would give better results, and more satisfaction for users because now you're not selecting a cuisine instead you're exactly choosing what you want, however by listening the cuisines you will get way more options to explore.

the cuisines chosen for this project were:

- Syrian: we all know
- Lebanese: which similar to the Syrian
- Mediterranean: which consists of seafood dishes
- Italian: in this case pizza and pasta restaurants

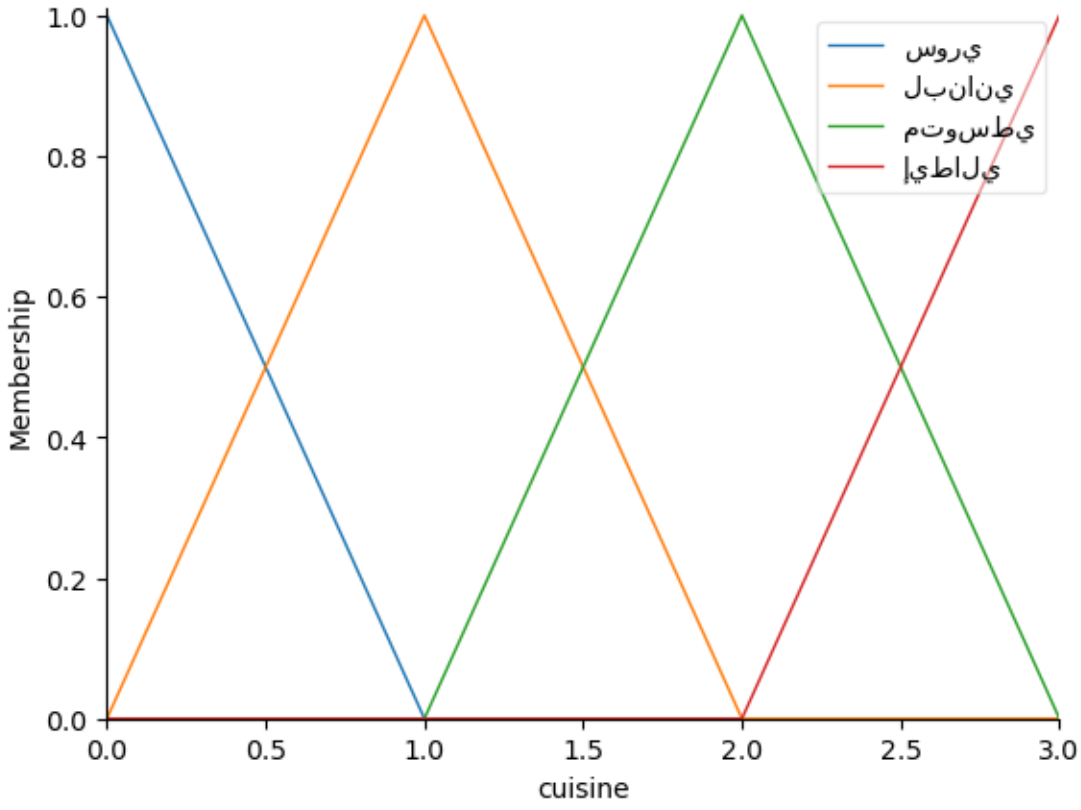


Figure 1 Cuisine listing

Price

So, we checked the average prices of a meal in each restaurant, the maximum price is 8000sp, the minimum is 4500sp, and the median is 7000sp, so we did the fuzzification on the difference between the entered price and the average meal cost at that restaurant, of course, we used the tringle membership function because at each group of fuzzy set the peak considered as the sweet spot for everyone like everyone agrees that for my budget if I paid extra 1000 or less it's still cheap, and so on

we have three fuzzy groups:

- cheap: which ranges from -5000 to 2000 and the peak at 1000 like I can pay 1000 and still be considered cheap
- moderate: which ranges from 1000 to 5000 and the peak at 3000 is still under the price of a meal so it's considered moderate
- expensive: which ranges from 4000 to 10000 now the difference is like the cost of an extra meal which is now considered expensive

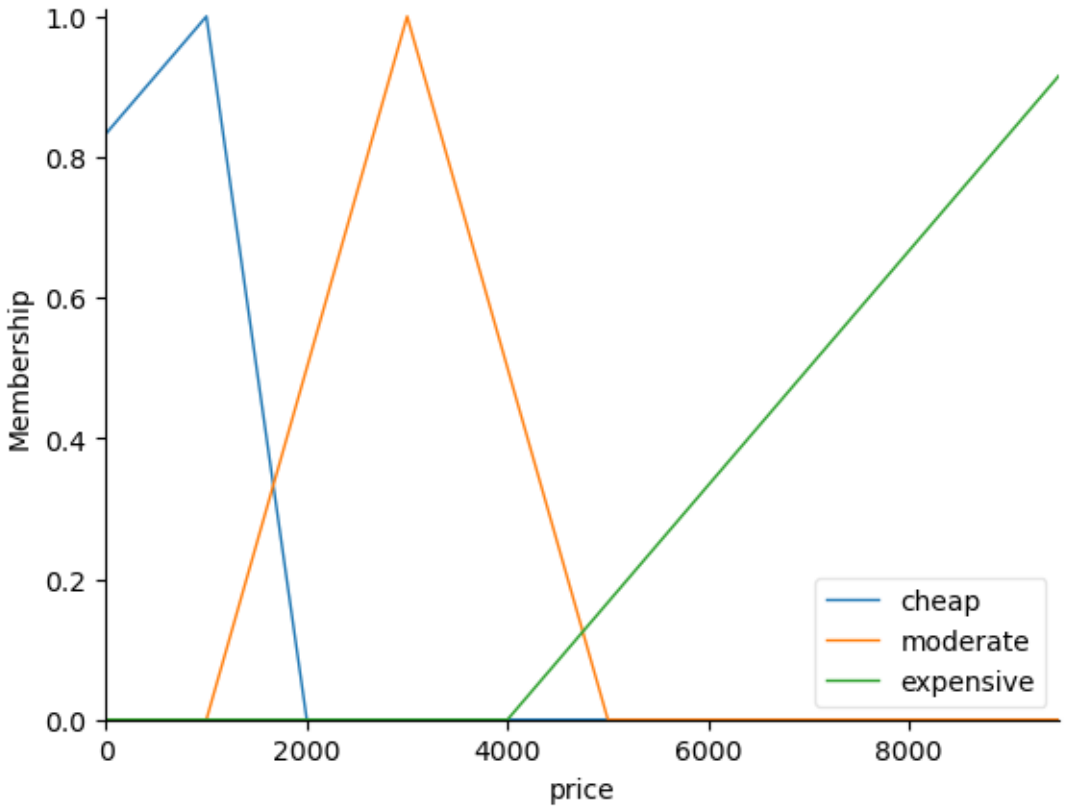


Figure 2 Price fuzzification

Location

the fuzzification done on how far the restaurant is from my location, we calculated the distances in kilometers, three fuzzy groups:

- near: ranges between 0 and 1.5 these ranges are still like a walk from the user location
- medium: ranges between 1 and 2.5 like still can walk or take a bus
- far: well, it's far so is it worth it?

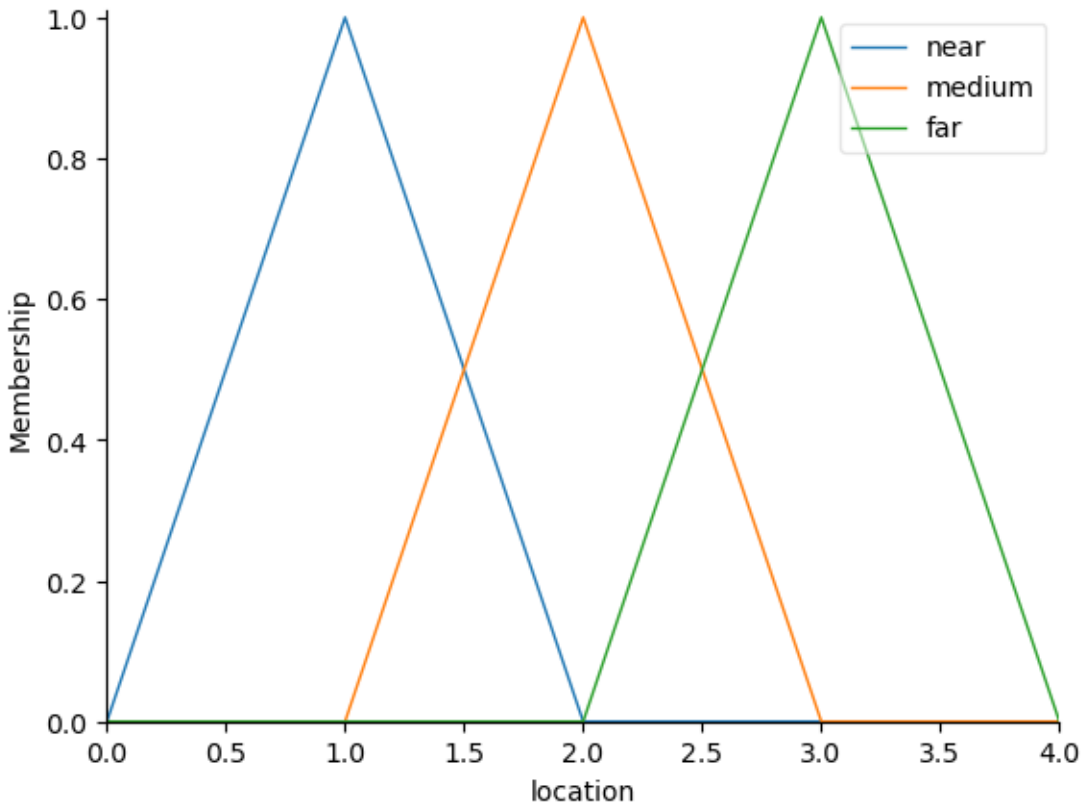


Figure 3 Location fuzzification

Recommendation

the recommendation is from 10

- not recommended: ranges from 0 to 4
- recommended: ranges from 2 to 5
- highly recommended: ranges from 6 to 10

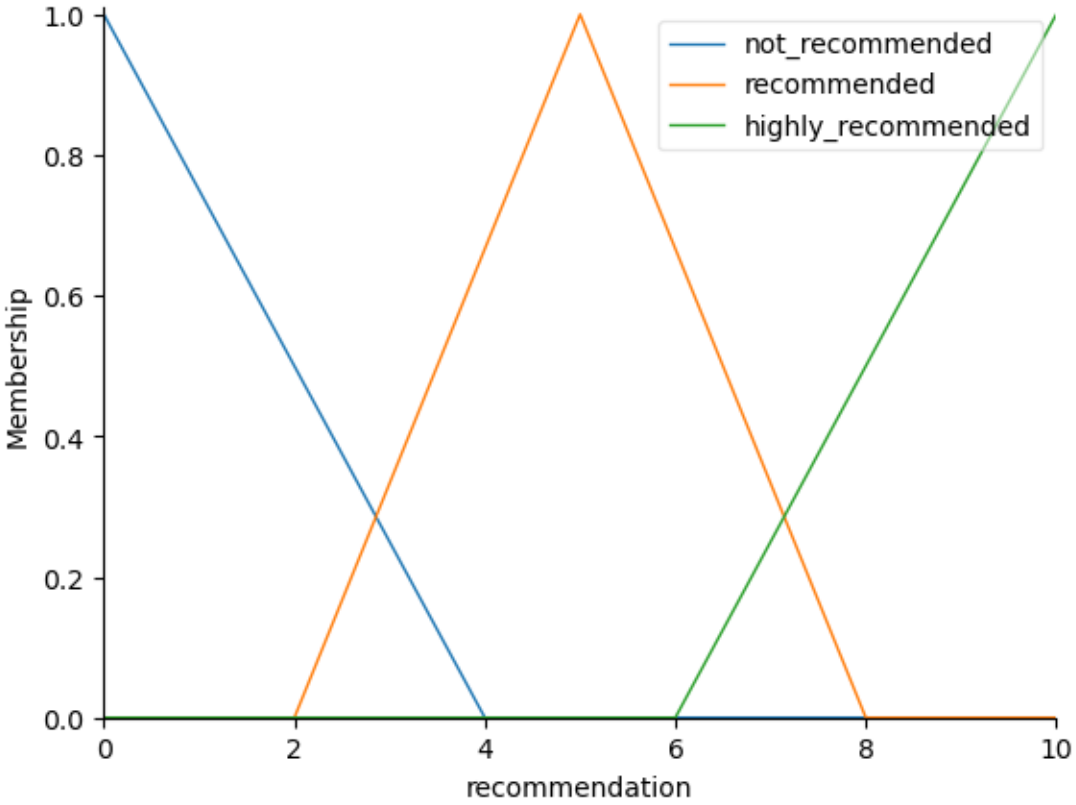


Figure 4 Recommendation fuzzification

Rules

our system has 36 rules, and the rules depend on the cuisine and what's common here in Damascus for example, if you want Syrian cuisine, you want a close cheap, or moderate restaurant and it's okay if it's a bit far(medium), but if it gets so far, it's not that much of recommendation, specifically if it's expensive.

1+ = highly recommend, 1 = recommended, 0 = not recommended.

Syrian	cheap	moderate	expensive
near	1+	1	1
medium	1	1	0
far	0	0	0

Lebanese	cheap	moderate	expensive
near	1+	1	1
medium	1	1	0
far	0	0	0

Mediterranean	cheap	moderate	expensive
near	1+	1+	1
medium	1+	1	1
far	1	1	1

Italian	cheap	moderate	expensive
near	1+	1	1
medium	1+	1	0
far	1	0	0

Defuzzification

Defuzzification is a process used in fuzzy logic systems to convert a fuzzy set into a single crisp output. In fuzzy logic, inputs and outputs are typically represented by fuzzy sets that express degrees of membership across a range of values. Defuzzification takes these fuzzy sets, which result from fuzzy inference processes, and produces a precise, actionable output.

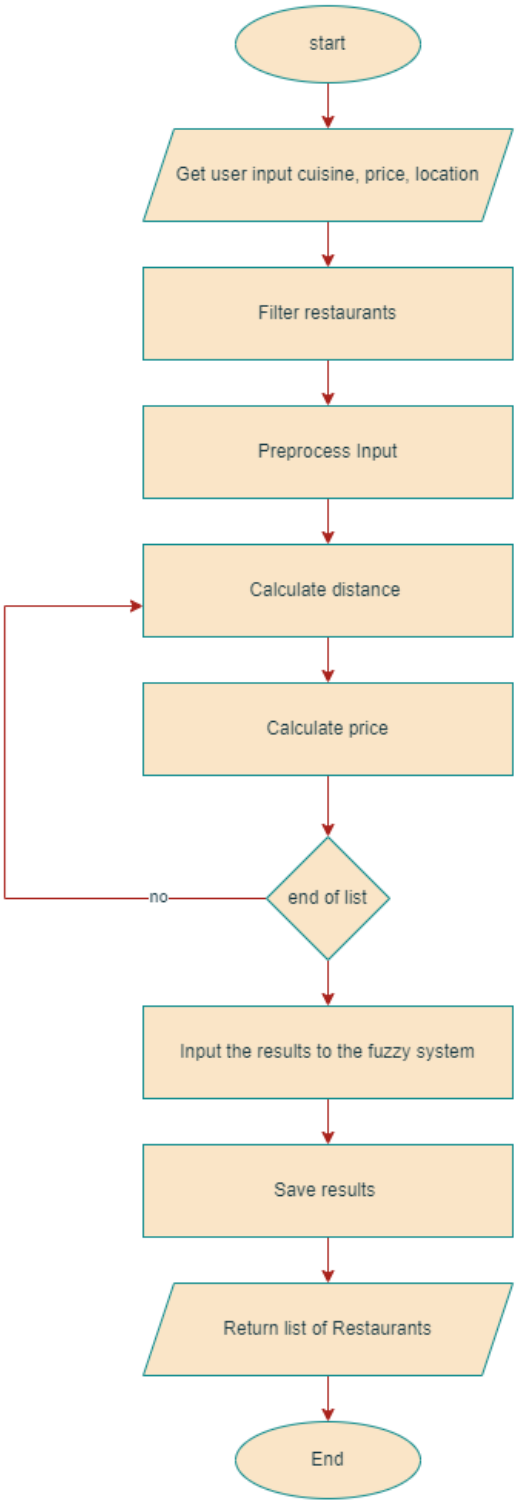
There are several methods for defuzzification, with the most common being:

Centroid Method (Center of Gravity or Center of Area)

This method calculates the center of the area under the curve of the membership function. It is the most commonly used defuzzification method due to its accuracy.

Formula: $\text{Ccentroid} = \frac{\int \mu(x)dx}{\int x\mu(x)dx}$

Pipeline


start
Get user input cuisine, price, location
Filter restaurants
Preprocess Input
Calculate distance
Calculate price
end of list
no
Input the results to the fuzzy system
Save results
Return list of Restaurants
End

Get user input:

The user inserts three inputs cuisine, budget (price), and neighborhood (location).

Cuisines are Syrian, Lebanese, Mediterranean, and Italian.

Budget (Prices) ranges from 1000 to 10000.

neighborhood (location) Old City (Ancient Damascus), Al-Midan, Abu Rummaneh, Malki, Mezzeh, Kafar Souseh, Baramkeh, Qanawat, Bab Touma, Rukn al-Din, Mazraa, Salhiyah, Jobar, Yarmouk, and Bab Sharqi.

Filter restaurants:

Get all the restaurants from the database that match the selected cuisine.

Preprocess input:

Give the number of the cuisine, Syrian = 0, Lebanese = 1, Mediterranean = 2, and Italian = 3.

Calculate distance

For every restaurant that matches the chosen cuisine, get the location of the restaurant and user location calculate the distance, and put the distance in a new column.

Calculate price

For every restaurant that matches the chosen cuisine, get the average meal price subtract the budget entered by the user, and put the result in a new column.

Input the result to the system

Input the cuisine the calculated distance and the calculated price.

Save results

Save the results in new columns, recommendations, and distances, …

Return list of restaurants

And save them to a JSON file.

Example

Cuisine = 0 #Syrian

Price = 3000 #Different

Location = 1 #1km distance

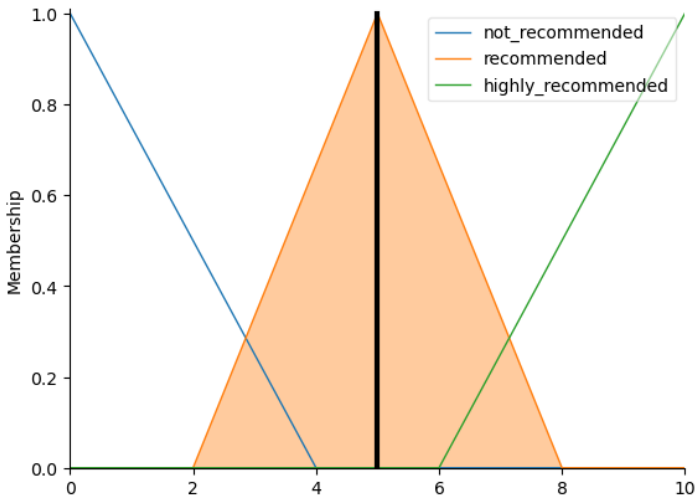
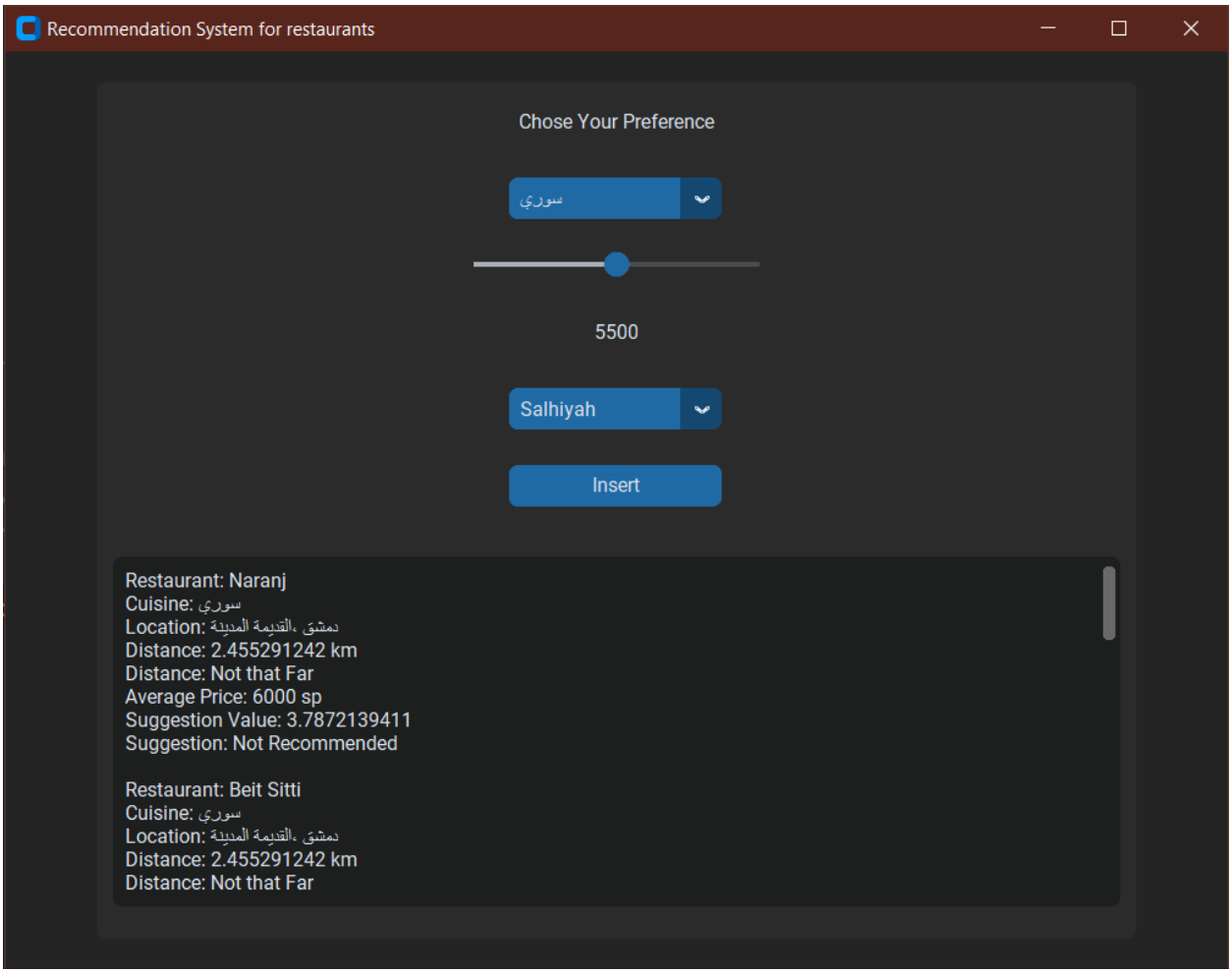


Figure 5 output

GUI



We have dropdown list for cuisines, slider for budget, and dropdown list for neighborhoods (Locations),

When the user hit the insert buttons the system gives a list of all restaurants with selected cuisine, with their recommendation and the distance, among other info like name.