

# **Restaurant Recommendation System**

**Sharayu Mantri**

**Arvind Pillai**

## **Objective/Motivation:**

To design a system that is capable of giving restaurants suggestions based on an individual's preferences. A multi-cultural society brings an influx of a large variety of cuisines and gives everyone an opportunity to experience something they've never tried before. Therefore, a suggestion system to help in the decision process is necessary.

Moreover, an individual with special requests might find it difficult to search for restaurants which cater to them. For example, a person might be looking for a restaurant that serves gluten-free food. It becomes time consuming if such a task is done manually by the person. Some people might be looking for a restaurant for special occasions or places that cater to many people. To simplify these processes, we decided to come up with a system using sASP that is capable of suggesting restaurants.

## **Proposed Solution:**

The recommendation system takes the following factors into consideration while suggesting restaurants.

1. Event – Special occasions such as birthdays, dates, etc. are some factors which are considered. Also includes casual places if the person doesn't have a special event, this is a mandatory field which will be used for deriving other fields.
2. Budget- To rule out restaurants that are not within the price range requested, we have 3 ranges of budget that have been determined based on the average cost of dining per individual.
3. Special Request- Suppose a person is allergic to gluten, nuts, they use this field to determine restaurants which are suited to their preference. More options like vegetarian, vegan, dog-friendly, etc are included in this field.
4. Ambiance- This field is derived from size of the party and event to determine what might be the best place for this person.
5. Service- This field is derived from the size of the party and event to predict the type of service. For example, A party of 8 people for a birthday might prefer to go to a place which offers dine-in, suggesting a catering or primarily drive-thru will not be logical.

6. Time- This is hard coded into the program to determine if the restaurant is open or not. If it is closed, then the restaurant will not be suggested to the user.
7. Cuisine- The user enters the cuisine that they are searching for, suppose a user only wants to have Chinese, then restaurants which are not Chinese will not be suggested.
8. Location- It would be illogical to suggest restaurants that are not near the user first. Therefore, this is a mandatory field which helps eliminate restaurants which are far away from the user.

Sample query:

main(Event, Size, Location, Cuisine, Time, Special)

?- main(casual, 4, 75080, indian, 2000, []).

Time is represented in 24hr format with no colons or dots.

Event can be casual, formal, birthday, anniversary

Cuisine is Indian, American, Chinese, Mexican

Special can have veg, non-veg, gluten-free, dog-friendly

### **Challenges:**

1. The most difficult challenge faces were to think what the user might be looking for when they enter all the mandatory fields such as event, size, and location. Derived fields such as ambiance, service type was based on the entry of these fields.
2. Thinking about special requests and abnormalities, for example, a veg only restaurant will supersede other options, etc. Finding out a sample data that would show case these features.
3. Determining the order of restaurants displayed based on their rating, modifying sort algorithms learned in class to work for our data format and making sure that sASP can successfully implement the code.
4. Predicting which input is necessary and which is not and creating a solution which has the capability to identify which is the not part of the solution.