I want to be able to...	So that ...	Info needed from DB	ML algorithm/statistic
#1 Customer receive a recommendation of a restaurant based on own past experience	I don't have to write the full name of a restaurant I have easier access to restaurants I went to before (autocomplete functionality)	- names of restaurants that were typed before by the user - info about what the user currently typed	Insert words into a trie, each node has a frequency (the frequency of the corresponding node). Additional parameters may be added.
#2 Customer Receive a recommendation of a restaurant based on other users past experience	I can try a new kind of food based on the popularity of that type of food	- names of the restaurants that were typed before by other users - info about what the user currently typed	Popularity may be a function with parameters such as: the number of times it was ordered overall, the number of times it was ordered but this time counting only once per person. Clusterization based on popularity. / Or the idea from #7
#3 Customer Receive a recommendation of a kind of food based on the current time (Breakfast,Lunch,Dinner)	I can eat appropriate food for every period of the day	- table(s) containing info about the user, the food ordered and the time of the day /hour when the food was ordered	Make three clusters corresponding to the three periods of the day (it works with any ML algo where the number of clusters is given, for example Kmeans / Hierarchical).
#4 Customer Receive a recommendation of a kind of food based on a chosen set of ingredients	I can eat similar food types	- table(s) containing info about each food type ever ordered and the ingredients present in that type of food - each food has a certain score (a binary number, where a bit of 0 corresponds to a missing ingredient and a bit of 1 corresponds to a present ingredient)	I may work with a supervised ML algorithm, such as KNeighboursClassifier (training on 0.4/0.3 of the data, testing on the rest) Comparison of two food scores is based on the Hamming Distance (Euclidean distance behaves worse is two foods are quite similar)

#5 Customer Receive a recommendation of a kind of food based on a set of ingredients that I don't want to eat	I can avoid certain food types	- table(s) containing info about each food type ever ordered and the ingredients present in that type of food - each food has a certain score (a binary number, where a bit of 0 corresponds to a missing ingredient and a bit of 1 corresponds to a present ingredient)	I recommend a supervised ML algorithm, such as KNeighboursClassifier (training on 0.4/0.3 of the data, testing on the rest) Comparison of two food scores is based on the Hamming Distance (Euclidean distance behaves worse is two foods are quite similar)
#6 Customer Receive a recommendation of a kind of food based on a budget (different ranges of budgets)	I can find food that fits my budget	- Table(s) containing info about each food type ever ordered and the prices of it in different restaurants - The price range that the user might be disposed to pay	In a simple scenario, a food can be recommended if the mean of its prices fits the range given by the user.
#7 Receive suggestions of restaurants that have similarities to the restaurants i currently like		- List of liked restaurants - Where he eats - What he eats - When he eats	Statistics/rating system for each user, and make clusters of users to recommend to each other(within the same cluster). Hierarchical might work. Distance may be calculated considering the kinds of food i like, ingredients i like, when i eat. Hamming distance could be used to calculate the similarity between the foods I like/ordered in the past and the food a restaurant offers, or between restaurants (look at a restaurant and recommend another similar restaurant to the clients of the first restaurant)

#8 Recommend kinds of food based on the budget of the user (different ranges of budgets)		- Budget of the user - Price of food - What he ate in the past - Ingredients contained in foods - Categories of foods(Pizza, Soups..)	basic statistics
#9 Get suggestions of new items to add on the menu based on what my current clients eat at other restaurants		- Who are my clients - Where they eat - What they eat - Are these types of food appropriate to my style?	When a user likes a kind of food, share the likes with all the ingredients of the food, and recommend kinds of food containing as many “good” ingredients.
#10 See statistics about the period of the day when the customers eat		- When they eat	-basic statistics
#11 Receive recommendations about what kind of food my customers want to eat		- What they eat - What food I have - Reservation list - (Past)orders list	Based on reservations, recommend kinds of food to have in stock (based on what the people with reservations ordered in the past)
#12 See statistics about what kind of food my customers want to eat		- What they eat - What food I have	-basic statistics

Statistics/rating system for each user, and make clusters of users to recommend to each other(within the same cluster). Hierarchical might work. Distance may be calculated considering the kinds of food i like, ingredients i like, when i eat.

Based on this we could make more recommendations regarding food to try, best time to eat, new restaurants