The Daily Dilemma: What's for Dinner?

A quest to create a decision support application

April 15, 2016

Introduction

After a long day of strenuous school or work, we are all faced with the same age-old question: What's for dinner? The last meal of the day is usually the one we have the most time for, and most control over. With this time and control comes the need to make a decision. Initially, it is as simple as cook or eat out? We are assuming our decision maker chooses to eat out and have laid out our problem and model from there. This decision to eat out now has new preferences that come into play. Is the decision maker inclined toward a type of cuisine? Of course we all have days where are hankering for Chinese food, or the buttery taste of gourmet French cooking. Our model assumes that a decision maker has an idea what type of food they want at the present moment, although we allow for the user to decide if this is the most important aspect of choosing their daily supper. Obviously, the decision maker has other factors that come into play for an enjoyable evening. Is the decision maker partial to hipster Lawrenceville, ritzy Shadyside, bar-filled Southside, or ethnic Squirrel Hill? Our model allows the decision maker to choose a neighborhood they would enjoy eating in. For other decision makers, price is the most important factor. Maybe they are a college student scrapping by and a cheap restaurant is exactly what they are looking for. The person dining could be a successful business man trying to impress a billion-dollar client, and will pay top dollar to wine and dine. Maybe price is not as much as a factor as ratings. Our model has ratings from 1-5 in increments of 0.5. Although in the real world rating and price are often correlated, these were selected without regard to price. After laying out our initial idea and the factors were chosen, we realized we had minimal uncertainties. Therefore, we weighted an "Additive Linear Utility" equation into our model. Essentially, we made an algorithm that takes the decision makers preferences and recommends a restaurant that fits those.

The Problem

Everyone faces the problem of what to have for dinner. This a daily issue for everyone. Our group chose to define dinner as the last meal of the day. In general, most people have more time and more control over this meal. People have control over who to eat with, how much to plan, if grocery shopping is necessary, how much cooking will be done, and other preferences. Our model centers on the choice to eat out while taking four user preferences into account. Our model weighs the decision makers' preferences and takes the guesswork out of where to eat supper. We have changed our initial proposal to reflect eating out rather than cooking because we found too many variables in ingredients, shopping, prepping and cooking. We initially did not wish to steer the decision maker toward a particular restaurant and we have since completely changed our concept to do just that. Our algorithm aims to present the user with a restaurant that will make them most happy.

The Decision Maker

Although dinner is a daily decision faced by all adults, we had in mind that the decision maker has a baseline comfort with technology. This is a desktop Java application, but could easily be ported to the web or on a mobile device. We assume that they have an idea of their own preferences and what is truly important to them for choosing a restaurant. Our model is built for Pittsburgh proper, and we assume a familiarity with the neighborhoods. We also allow for the user to rank which factor is most important. We assume that the decision maker may have a preference for a single, certain type of food. We also allow the user to change their minds and put in a different factors to re-rank the importance of their preferences. This will lead to a new set of restaurants generated. We assume that our decision maker is open to new experiences, restaurants, and parts of town. Indeed, this model will not work for decision makers that have their few favorite restaurants and don't venture beyond the tried and true. The decision maker is one that has a certain amount of trust in technology and algorithms.

Decision Options

As mentioned earlier, the initial decision of whether to cook, get take out or go out to dinner is not represented here and happened prior to our model. This was to keep the model simple as the bulk of the functionality is centered around generating a restaurant based on user preferences. We acknowledge the ultimate decision in this problem comes later after the user works through all the drop down menus, inputs and ranks their preferences. When faced with the sixty top choices that are generated, the person dining will pick one of those restaurants or opt for something else entirely. If they are not satisfied with the results, they can change the preferences in our app.

Uncertainties

Our model has no real quantitative uncertainties. Thus, these are not present in our model. However, there is the probability that the user will not select a restaurant we present to them. The application aims to take in the user's choices, and give the locations most relevant to their taste. In this way, we hope to give the person dining the most enjoyable experience possible. The better the algorithm works, the less of a chance of displeasure.

User Preferences

The meat of our decision is eliciting the user's preferences to suggest a restaurant in Pittsburgh that the user would enjoy. It could be one they may never have considered prior to the use of our decision support system. We use four factors to recommend a restaurant. Often people have an idea of what kind of food they want, so we choose eight types of cuisine. These are as follows: Chinese, Thai, Indian, Italian, American, French, Mexican, and Greek. Some restaurants were excluded because they didn't match the categories (Korean, Portuguese, Japanese). If they weren't open past 7 pm, we didn't include them. Our application considers this the minimum timeframe for an early dinner. This 7pm stipulation had some unfortunate consequences, as many restaurants in Downtown and the Strip District were excluded for early closure. If something was primarily a coffee shop or juice bar it was excluded from our database. Some restaurants need more explanation. For example, any pizza place was considered Italian. If more than two types of cuisines were listed, such as Italian and American, the restaurant was classified based on the name or looking at the restaurant's website to determine what the primary cuisine was. Some restaurants were excluded if they said "Asian", but didn't specify if they were Chinese or Thai. One of the group members took time to evaluate the cuisines. If a restaurant fit into one of those labels, was listed in our database. We also ran into this problem with restaurants listed as Mediterranean. We looked at the restaurant's website to determine if Greek was the primary cuisine. The last restaurant type that needs explained is "American." Indeed, this became a catch-all term, and thus looks overrepresented in our database. Steakhouses, bars, seafood, sub shops, sandwiches, and many fast food places fall under this umbrella.

Another factor is neighborhood. We added this feature, as people are often looking for places in a new part of town. They could also live in a certain area and don't wish to travel far. We selected neighborhoods for their plethora of restaurants and notoriety in the food world. We used eight areas: Lawrenceville, Strip District, North Side, South Side, Downtown, Oakland, Shadyside and Squirrel Hill. Some places got combined, like Bloomfield and Shadyside. As these locations are within a couple miles of each other, we decided against using Google to find the closest distance. Instead we wanted to get at the preference of location by finding their chosen neighborhood.

The third factor we used was price, represented by "$". It ranged from one dollar sign for the cheapest place, to four dollar signs to represent the most expensive places. We found that this is often a huge consideration when deciding what to eat. We used Zomato (formerly Urban Spoon) to keep these as true as possible.

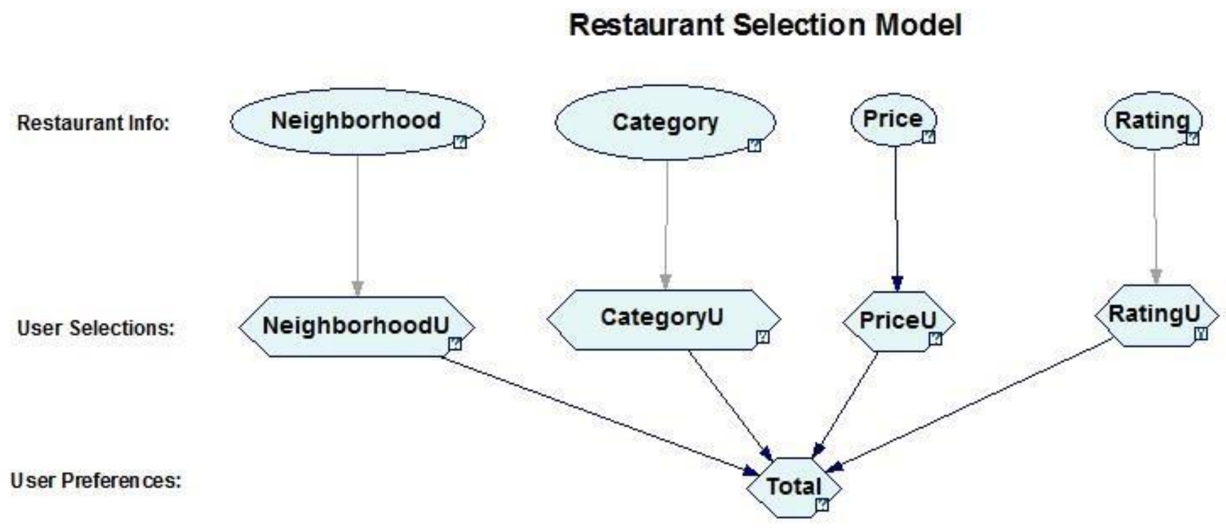
The fourth and final factor is rating. Many people turn to websites like Yelp to see what others are saying about a particular restaurant. As some people trust not only high ratings but how many people rate something, we took this out and just gave a single rating for each restaurant with no commentary. As price and rating seem to be dependent on each other, a group member arbitrarily put in the ratings and indeed tried not to be swayed by price.

The Model

We elicited the decision maker's preferences for our four factors by using an application with drop down menus, one for each of the four factors. For example, the decision maker could pick "Lawrenceville", "Indian", "$$$$" and a rating of "3", or any other combination of those factors.

Although, people do weigh these factors differently. Due to this, we asked the decision maker to pick the factor that is most and least important to them. For example, the decision maker could be in the mood for Indian food so much that the rating doesn't matter.

From this Java application, we feed these preferences into a GeNIe model with jSMILE. We built a model of this problem in GeNIe, shown below:

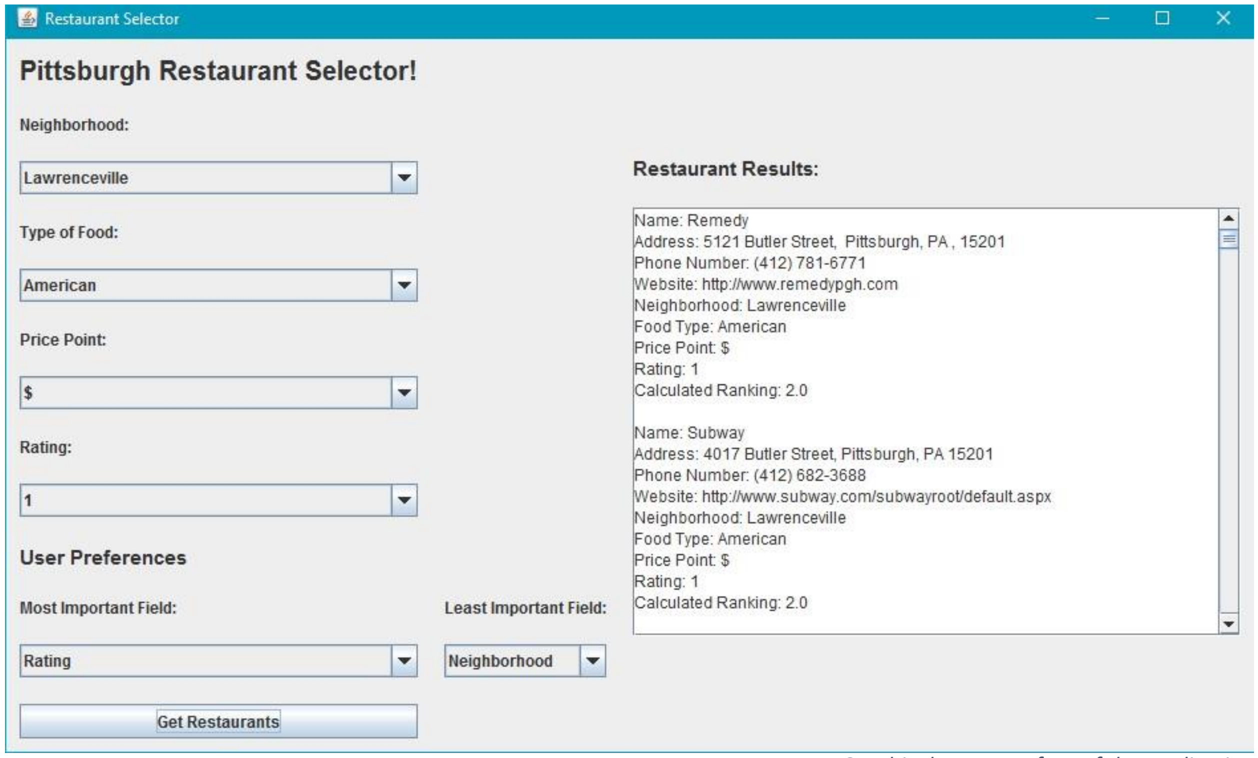


Each of the four factors (cuisine, neighborhood, rating and price) have their own node, which make up the top of the GeNIe model. These are chance nodes, and the evidence is set using tuples from the MySQL database. Each node is linked to one Utility Node with the same name. This is where the four factors that the decision maker chose get weighted. Continuing with our example, "Lawrenceville", "Indian" "$$$$" and "3" are each given a utility of 1. All other neighborhoods, cuisine, price points, and ratings are weighted as 0. Our GeNIe model reflects the factor that is most and least important, by linking all four utility nodes to Additive Linear Utility (ALU) node. The most important factor to our decision maker is weighted as 0.75. The least important factor to our decision maker is weighted in the final utility as 0.25. This means there are two neutral preferences, which are not more or less important. These are weighted as 0.5, midway between the higher and lower preference. This weighting will ensure that the most important factor is heavily weighted. Once all the evidence and definitions are set, the final node computes rankings for each and every restaurant in the database.

Our data source was stored in a MySQL database of 559 restaurants. The database not only needed to include the parameters that the user would select, but also the information to be presented to the user in the final results. The column headers were: id, restaurant_name, address, phone_number, url, food_category, neighborhood, price_point, and rating. The Java application queries the database, and stored "food_category", "neighborhood", "price_point", and "rating" in variables. These variables were then used to set the evidence in the model's chance nodes. When executed, our application returns the top sixty restaurants that, when the factors are weighted in the final utility, should generate the best eating experience.

Analysis

We wanted to keep the functionality and results of the application simple. We kept the type of food to one primary category. This works for a smaller database of just Pittsburgh restaurants, but would limit results on a global scale such as Yelp. The same applies for multiple neighborhoods, price ranges, and ratings.



Neighborhood: Lawrenceville
Type of Food: American
Price Point: $
Rating: 1

User Preferences
Most Important Field: Rating
Least Important Field: Neighborhood

Restaurant Results:

```
Name: Remedy
Address: 5121 Butler Street,  Pittsburgh, PA , 15201
Phone Number: (412) 781-6771
Website: http://www.remedypgh.com
Neighborhood: Lawrenceville
Food Type: American
Price Point: $
Rating: 1
Calculated Ranking: 2.0

Name: Subway
Address: 4017 Butler Street, Pittsburgh, PA 15201
Phone Number: (412) 682-3688
Website: http://www.subway.com/subwayroot/default.aspx
Neighborhood: Lawrenceville
Food Type: American
Price Point: $
Rating: 1
Calculated Ranking: 2.0
```

Graphical User Interface of the Application

Recommendations

If we were to make this application more robust, we would include a way to select a broader spectrum of choices for each category. It would also be useful to create a login page that allows the decision makers preferences to be stored. We might also allow for some uncertainties such as number of guests, if the establishment is child friendly, wait time, or if the restaurant takes reservations. The ultimate uncertainty is whether or not the decision maker decides they are risky enough to try a new restaurant. We would also like to port our application to a server in order to host it on the internet. This way anyone would be able to access the helpful functionality of the restaurant selector.

Conclusion

The restaurant industry is one of the largest in the world. Every evening millions of people leave their homes and descend upon eateries to fill their stomachs. There are countless restaurants in every city, and the selection process can be overwhelming. We wanted to create a smarter way to make sure you have a good dining experience. Using our application’s algorithm, the chances of enjoying a meal significant increase. We take in various parameters and user preference to tailor a set of results to each specific decision maker. The final list of ranked results is a group of dining locations tailored just for you!