Press Release

Grub Guru Unveiled: Duke University Students Launch Innovative Food Decision App

Durham NC - April 21, 2024: Today we announce that Ben Matz, Brian Guo, Graham Gumbert, Ben Rubin, and Zuoyin Wang have launched the first release of “Grub Guru,” an application that helps indecisive user groups decide where to eat! Under the guidance of Dr. Laber and the rest of the STAT 561 staff, this group of Duke University students is looking to revolutionize how indecisive user groups choose where to dine.

"Grub Guru" is the brainchild of these enterprising students who recognized the ever-persisting challenge of deciding on a place to eat, especially within social settings. “How many times have you and your friends spent countless minutes, or even hours, hemming and hawing about where to grab a bite?” asks co-founder and undergraduate Statistics student Ben Rubin. “People just hate making decisions. So we do it for them.” To solve this dilemma, the team leveraged advanced algorithms and intuitive design in their app to simplify the decision-making process, even turning it into a fun and collaborative experience.

The app's features include smart recommendations. Utilizing machine learning, "Grub Guru" suggests dining options based on user preferences, location convenience, and price point. The prototype also supports real-time updates. Users can stay informed with live updates on restaurant availability, wait times, and reservation slots. With customizable filters, users can tailor their search based on the factors that are most important to them. And if a restaurant doesn’t appear in our database, users can simply upload their favorite underground spots with just a quick description.

Since its soft launch, "Grub Guru" has received overwhelmingly positive feedback from early users. A survey conducted among beta testers revealed:

- 95% of users stated they were able to make decisions faster when using "Grub Guru," with an average of 4.1 minutes of decision time saved
- 92% of users reported an increase in overall satisfaction with their dining experiences.
- 87% found the app's interface intuitive and easy to navigate.

Ben Matz, co-founder and undergraduate ECE student, expressed his enthusiasm, adding, "Whether you're planning a casual lunch with friends or organizing a team dinner, 'Grub Guru' adapts to your needs, ensuring a seamless and satisfying dining experience." In the future, the team hopes to improve the product by allowing for the more complicated input of user preferences. “Not only will users be able to input their preferred categories of food, but also their very own rankings and ratings of their favorite restaurants,” said Brian Guo, another co-founder and Computer Science Masters student, “Going forward, we see ‘Grub Guru’ becoming intertwined in the social fabric of the world of food.” The app also hopes to partner with other food rating apps such as Beli and Yelp to pool user data and improve their recommendation systems.

"Grub Guru" is now available for download on iOS and Android devices, promising to revolutionize the eating industry one decision at a time. Join the food revolution today and discover a new way to choose where to eat!

For media inquiries, please contact:
GrubGuru@gmail.com

FAQs

Q1: What is our minimal viable system?

Our minimal viable system is a program that takes in users that have restaurant preferences as well as locations. Users can then form groups with each other. Based on these groups, our application outputs a ranking of restaurant suggestions based on these factors. We utilize distance, price point, and food genre preferences in our algorithm in order to make these suggestions.

Q2: What specific restaurant features will be incorporated into our algorithm?

The aspects of restaurants that we determined were most important in generating suggestions are as follows: distance of the restaurant from those in the group, how expensive the restaurant is, and the type of food the restaurant serves (sushi, burgers, tacos, etc.). Our algorithm takes in the current distance to each restaurant, the price preference, and the food preferences of each member of the group and then outputs a recommendation of where all the members of the group should meet to eat!

Q3: How is the social aspect embedded?

In our minimal viable system, we will create a class for the users where we will randomly define preferences, price points, and locations for a group of test users. The social aspect comes from “group” settings. We have written a loop that will rank preferences for each user. Using a composite ranking of preferences, the app will output a list of restaurants. Users get the opportunity to join circles with their friends so that everyone’s opinion is represented in the food-decision process.

Q4: How will users input their own preferences between restaurants?

Our minimal viable system will not directly incorporate the preferences interface of our theoretical final app, but it will map user preferences to a preset group of categories of restaurants (e.g. fast food, tacos, burgers, etc.). We plan to create a user class where we can define new users and input randomized restaurant category preferences. This will allow us to better test that our algorithm is accounting for user inputs.

Q5: Will we use outside websites (Beli, Yelp, etc.) for a baseline understanding of restaurant popularity?

We hope to soon implement a feature that will rely on Beli/Yelp rankings and data for users with unlisted preferences. While users are expected to input their preferences between local restaurant options, we also want to incorporate online rankings as a baseline. This would potentially improve our decision-making by leveraging the wisdom of the crowd in circumstances where users have not had the chance to fill out their restaurant preferences (or serve as a tiebreaker for similarly-suggested restaurants). However, we would still encourage our users to input their preferences for a more individualized experience.

Q6: How will users know the algorithm is factoring in their preferences?

In the case of the minimal viable product, the user preferences are uniformly randomized. We tested brief summary statistics of the data to ensure that categories roughly reflected preference frequencies in the user groups. In the future, we plan to expand categories to more accurately reflect distinctions in non-chain options, as well as pinpoint locations of users at the time of a request.

Q7: How will the algorithm be evaluated and improved?

In the case of the minimum viable product, our group ran some tests where we all had different preferences, decided as a group what we thought a fair ranking was for the different restaurants our algorithm could recommend, and then ran the algorithm to look at the results. If our algorithm was very far away from what we thought it should be doing, we looked into what the algorithm was calculating in practice to modify the weights of the different parameters held in the final restaurant ranking.

To get initial feedback from our algorithm, we asked members of the group to rate their satisfaction with the restaurant ranking our algorithm returns. In this way, we collected the necessary data to test future changes to our algorithm. When our user base grows large enough, we plan to do randomized controlled trials on the groups to see if the new changes to our algorithm lead to increases in user satisfaction with the rankings our algorithm outputs. If user satisfaction increases with the new changes, we will modify our original algorithm to incorporate the new changes. Otherwise, we will not include the new changes. In this way, we will ensure that user satisfaction with our main ranking algorithm always improves with new changes.

Q8: How will the app handle preference updates given new restaurants or locations?
While our minimal viable system will be drawing inputs from a set list of restaurants, we plan to eventually expand our restaurant selection to broader locations based on distance ranges. Our current system demonstrates one implementation of Grub Guru using two users in the local Durham area. We are also considering iterating towards a Google Maps API to help quantify distances. Another feature that we look to implement in the future is the ability for a user to upload a restaurant that's not already in our database.

Q9: Will we allow groups to plan for going to a restaurant sometime in the future? What if the group is going on a vacation together in a few weeks and they want to make a reservation?
While our app is best used for helping groups make decisions in the moment (as the food that you're craving will likely change from one day to the next), we plan to include a feature in our final app that will allow groups to input their future locations at a specific point in time. Our app will then output a restaurant ranking for this future date as well as reservation options at restaurants that offer them and of course allow users to make reservations with them through our app. Making a reservation in this way would automatically email all group members their reservation time as well as an ICS file to add it to their calendars.

Q10: What if a user doesn't want to share their location with us?
We will allow users to input their own location as a pin on a map. We will use the Google Maps API to do this. This will allow users to keep their exact location from us if they would like to while still accounting for how far away the restaurants are from them in the final restaurant ranking.

Q11: How will we remain profitable?
We will allow restaurants to pay us for two business opportunities. The first advantage we may give them is a boost in their ranking in the final ranking of restaurants for many different groups. We would of course keep this boost modest so as to preserve the integrity of the rankings of our app, but this boost is something we believe that many restaurants would pay for. We will also offer restaurants the opportunity to offer deals and coupons in the final ranking so as to draw more attention to themselves and make up for a potentially lower ranking. More expensive restaurants could offer deals to groups with cheaper price preferences so as to increase their standing in our rankings.

We initially considered offering our app as a subscription service that users would have to pay for on a monthly basis, but we decided against this due to the social nature of our final app. If one person in a group of friends wasn't paying for our app, their preferences wouldn't be included in the final restaurant ranking. Therefore, it would be less likely that the group would use our app to figure out where to eat.

Q12: How do we know the recommender isn't spewing out complete nonsense?
In all of our test runs, we included a restaurant called "Nothing" with a description of "blah blah blah hey now blah." This served as a control, as this fictitious restaurant consistently finished in last place as decided by our recommendation algorithm.

Technical Details/Preliminary Results

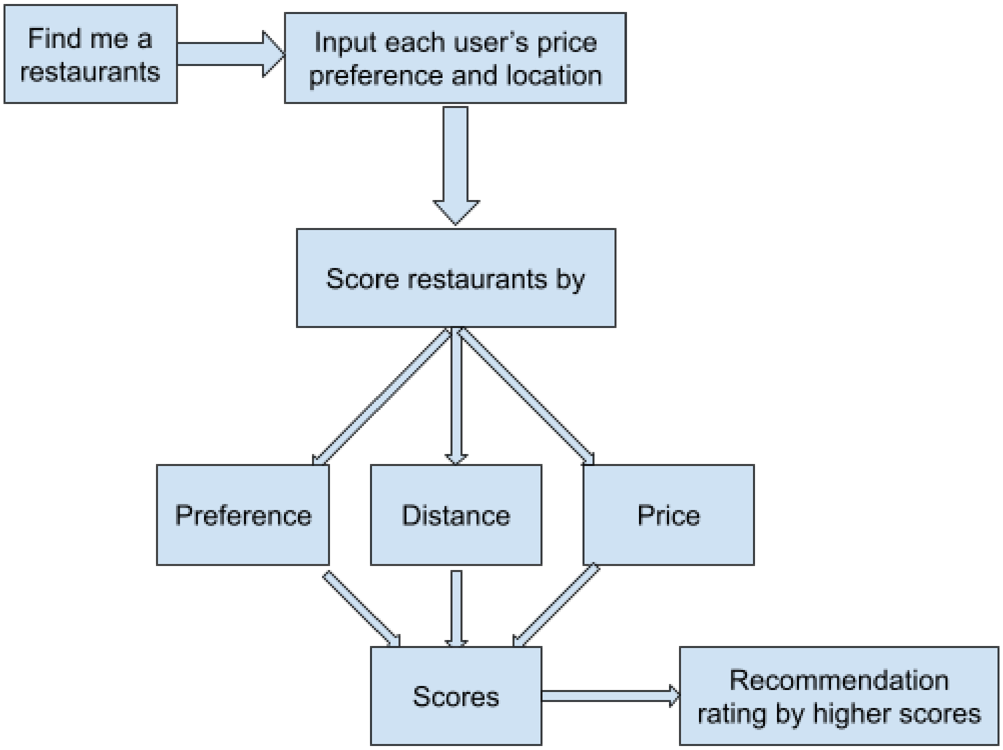
Psuedocode

```
1. Users populate the GrubGuru App with their dining preferences(categories supported are
   genre, price, and location)
2. Restaurants are pulled from a database such as Yelp
3. function makeRecommendations(restaurants, users)

    a. Initialization
        i. Individual users add themselves to a group
        ii. This group queries GrubGuru for a restaurant

    b. Location
        i. System calculates distance between restaurants and each users
        ii. Goal is to find a spot that minimizes the total distance
            1. High-distance locations will be penalized
    c. Price
        i. System determines the difference in price point between restaurants and
           the groups' desires.
        ii. The goal is to minimize this distance
            1. Generally, penalization is heavier for restaurants that are above the
               desired price
    d. Genre
        i. System has a neural network that calculates the similarity between the
           users' category preferences and the description of the restaurant
        ii. Goal is to maximize similarity

    e. Recommendation
        i. These three categories are weighted and tabulated into a single score
        ii. The top matching restaurants are outputted from the system
```


Find me a restaurants
Input each user's price preference and location
Score restaurants by
Preference
Distance
Price
Scores
Recommendation rating by higher scores

Acknowledgments:

We thank Brian Guo, Graham Gumbert, and Ben Rubin for their core contributions to the codebase. We thank Ben Matz, Zuoyin Wang, and Ben Rubin for their contributions to the nontechnical report. We also thank Zuoyin for taking the brunt of HW 7/8 so the rest of us could focus more on the project. We thank Ben Matz for organizing the group and serving as the "project manager" of sorts. Finally, we thank Ben Rubin for coming up with the original idea for this project.

We also extend our heartfelt gratitude to everyone who supported us in completing this project. Special thanks to our supervisor, Dr. Eric Laber, for his invaluable guidance and insights. We also appreciate the support from the staff in the Department of Statistics at Duke University, and our lab mates for their helpful discussions. We are grateful for the financial support from Duke University, and we thank our families for their endless love and encouragement. Finally, we thank all the participants and respondents, whose contributions were crucial to our research.

Code In Separate File