Scraping Snackpass:

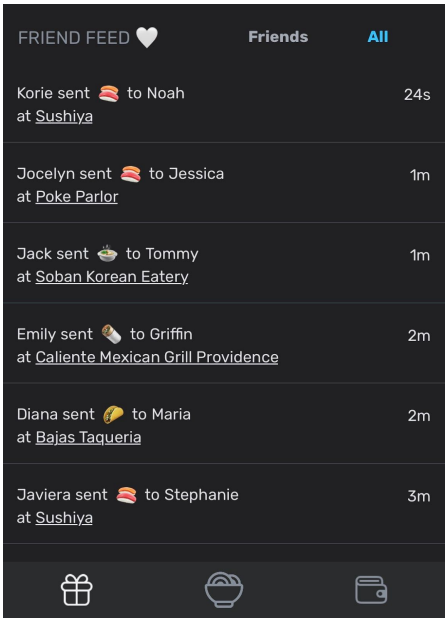
Predicting User Ordering Patterns Using XGBoost Decision Tree Regression

Josh Baum, Amy Li, Jake Sokol
Github Repository
ECON 1660 Final Project
11 December 2020

Background and Motivation

Snackpass is an app that exploded in popularity at Brown in 2018 and has been a mainstay among students ever since. Students use Snackpass to browse local restaurants and order food for pickup. What sets Snackpass apart from typical food-ordering apps and makes it so widely used on college campuses is its focus on fast pickup, rather than delivery. Snackpass was launched at Yale in 2017 and has since grown considerably throughout many campuses across the country. At Brown, Snackpass was introduced in January of 2018.

One interesting feature of the Snackpass app is its social orientation. After ordering a meal at a restaurant, users are automatically asked to send a gift to their friends via a popup modal. When accumulated, these gifts can be redeemed for rewards towards that restaurant. From personal experience and informal surveys of peers, an extremely high percentage of individuals always send gifts after each meal. Snackpass displays gift-giving activity on a feed on the app, where users can see activity among their friends and among all of the app's users. Users' gifts are displayed in this feed unless they set their gift setting to private, an option that is mostly hidden and which many users are unaware of. We hypothesize that, because of the public and frequent nature of the gifts, the "All" feed represents a nearly complete view of all ordering activity on Snackpass.



"All" Activity on Snackpass

If aggregated, this Snackpass gift activity would lend considerable insights toward student consumption habits and general restaurant performance around Brown and other colleges where Snackpass has been adopted.

Snackpass has become a significant part of Brown's campus culture. Using Snackpass data and methods learned in ECON 1660, we are able to develop interesting insights about the restaurants that we are familiar with and the student body that we are members of.

Research Questions

There are a couple questions that we were interested in answering as it pertains to Snackpass. In particular, we were most interested in answering the following: Are there cyclic consumption patterns? If there are, can we predict how users will act?

This question was interesting to us for a couple reasons. The first was that this question is the type of question that Snackpass might ask if they were planning on targeting students with advertisements or promotions to increase ordering activity on the app. The second is that this sort of predictive problem was along the lines of what we had studied throughout class.

We realized that the size and scope of this question differs considerably depending on how exactly we define, "how users will act." Snackpass might wish to predict when a user's next order might be, where a user's next order might be, who is likely to order at a given point in time, or all of the above. We ultimately decided that for the scope of our project, it would be best to predict, given a future date and time of day, whether each user will order or not. This problem narrows the scope and is more manageable to attempt to answer with our timeframe. See the Methods section below for more details on how we tackled this question.

Other questions that we were interested in answering pertained to general restaurant activity: How has restaurant activity changed over time? How has restaurant activity changed with regard to the pandemic? How have Mexican restaurants fared when an additional Mexican restaurant enters the market? Though we did not develop models to answer these questions, we were able to gain some insight by producing interesting visualizations which we will dive into in the Exploring the Dataset section below.

Creating a Dataset

The first step towards answering our research questions required creating a dataset. Fortunately, Snackpass gift data for all users is publicly available on each user's feed (if you know how and where to find it). This gift data is critical because each gift giving indicates data about when an order was placed, from what restaurant, and by whom. We were able to scrape this data by using the mitm HTTPs proxy. The proxy in effect acts as an intermediary between the Snackpass server and our personal device. Using the proxy's Python API, we were able to perform HTTPS requests to the server, the response for which was the gift data in the form of an HTTPS response. From the response, we were able to extract the gift data in JSON format and write it to a file.

From this process we were able to gather data for 1.7 million orders and 80,000 unique users from June of 2017 through October of 2020. We decided to organize this information into three tables. One table contains user information, including an ID and username for each user. Another table contains store information, including an ID and name for each store. The last table contains order information, including a user ID for the sender and receiver of the gift, the ID of the store ordered from, and a timestamp for the order creation. Throughout this project, we spent a lot of time thinking about data organization. We knew that working with millions of data points might provide challenges computationally; so, by keeping this information in separate tables, we were able to avoid excessively large and cluttered files.

Since we were also interested in examining how weather might affect cyclic consumption patterns, we searched online for a dataset containing hourly weather data from Providence in the last three years. We were able to locate a dataset (for a $10 purchase) that included useful information such as rainfall, snowfall, and temperature for every hour that Snackpass has been active. Since the dataset includes a timestamp of the hour the weather was recorded, it was easily incorporated into our model with some manipulation.

Lastly, we found it helpful to have a subset of the store data of just Providence stores. This was useful because it allowed us to examine Brown-specific activity, which was the scope of our Brown-related research questions. To create this dataset, we only included orders from stores that we know are located in Providence.

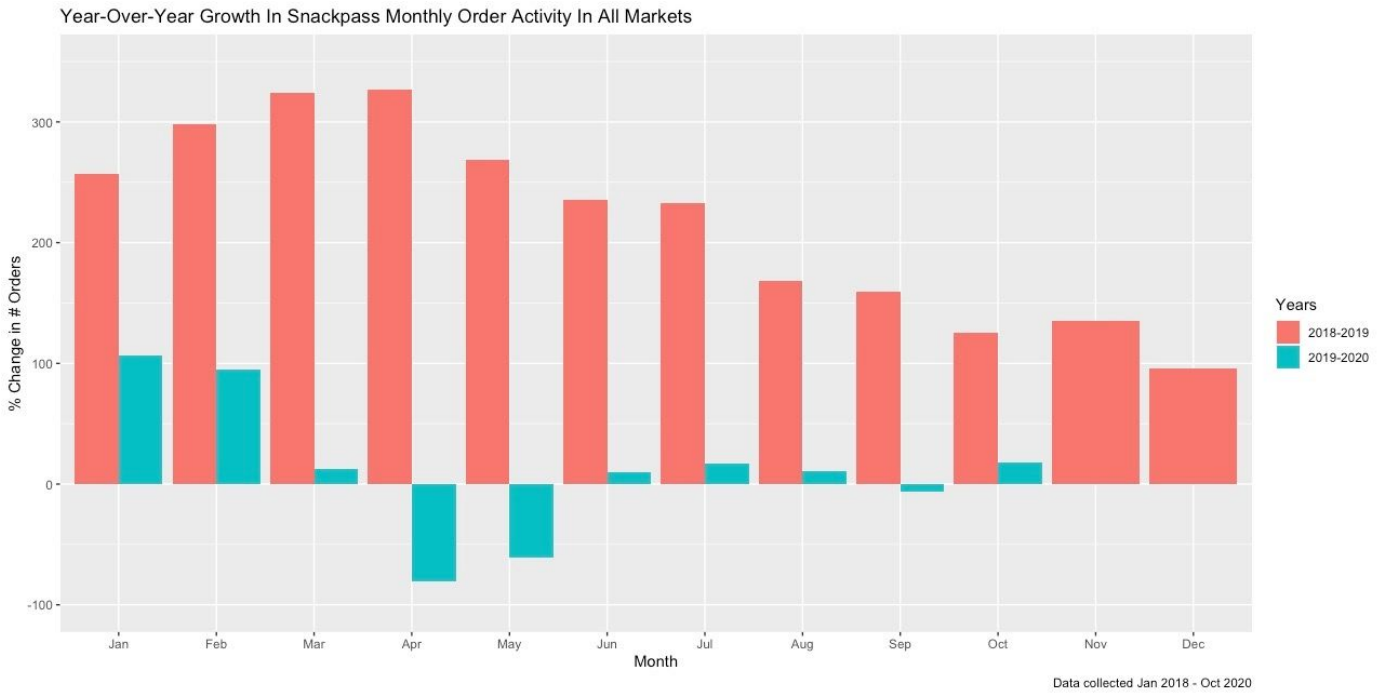
The dataset that we were able to create using these processes provided us with all the information that we needed to predict cyclic consumer behavior and also examine general Providence restaurant trends.

Exploring the Data

After creating a comprehensive dataset, we were able to delve into it and gain some insight into our general restaurant activity research questions. To reiterate, our high-level questions were the following: How has restaurant activity changed over time? How has restaurant activity changed with regard to the pandemic? How have Mexican restaurants fared when an additional Mexican restaurant enters the market?

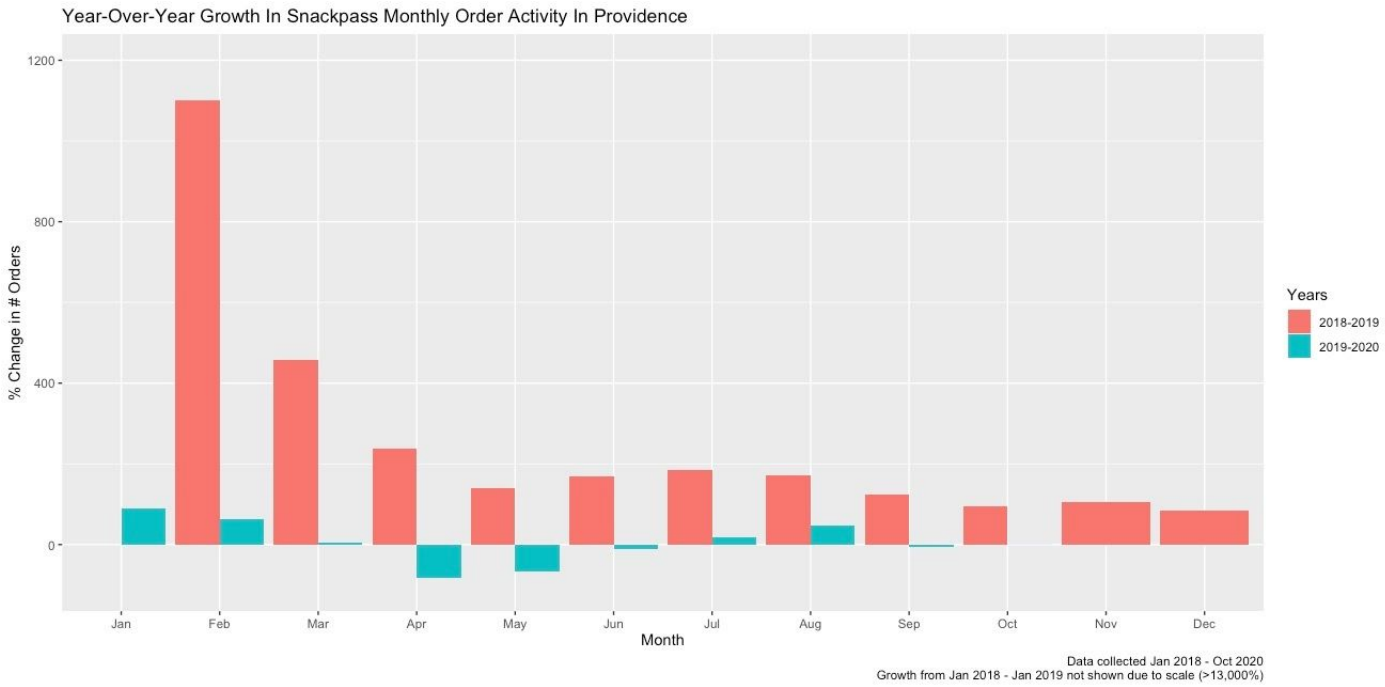
Our discoveries are discussed in the four figures below.

Figure 1



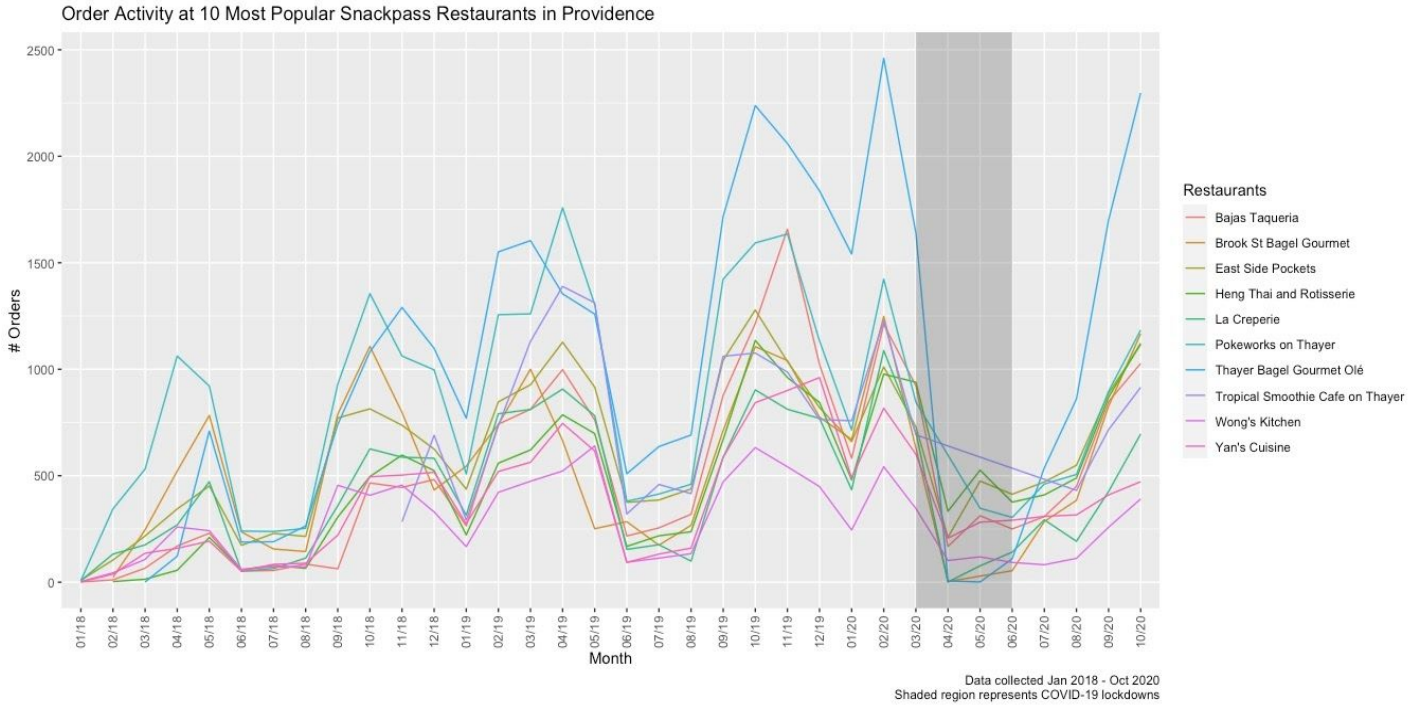
This first graph captures the year-over-year growth in Snackpass activity on a monthly basis from 2018-2019 and 2019-2020. Note that this reflects Snackpass activity in all markets (i.e. campuses), not just Brown. We can see that Snackpass grew rapidly between 2018 and 2019, which is unsurprising given that it was first introduced in June of 2017 and rapidly expanded to a number of new college campuses throughout 2018. We can also see the effect of the pandemic on Snackpass growth. In particular, growth really slowed in March through October of 2020, relative to those months in 2019. This is likely due to college campuses shutting down and students returning home. Interestingly, it looks like Snackpass growth has remained mostly stagnant to this day despite schools like Brown opening back up, which suggests that it may be a while before Snackpass nears its previous levels of growth.

Figure 2



This graph is similar to Figure 1 in that it shows year-over-year growth in Snackpass order activity, however it is limited to just restaurants in Providence. It is interesting to see that growth on Brown's campus between 2018 and 2019 outpaced that of Snackpass on a whole, which suggests that Brown has been a particularly lucrative market for Snackpass. We can likewise see the effects of the pandemic in lack of growth in March through October of 2020, relative to 2019. Similar to Figure 1, we see low growth at the present day, which suggests that growth at Brown has been in line with Snackpass growth as a whole during the pandemic.

Figure 3

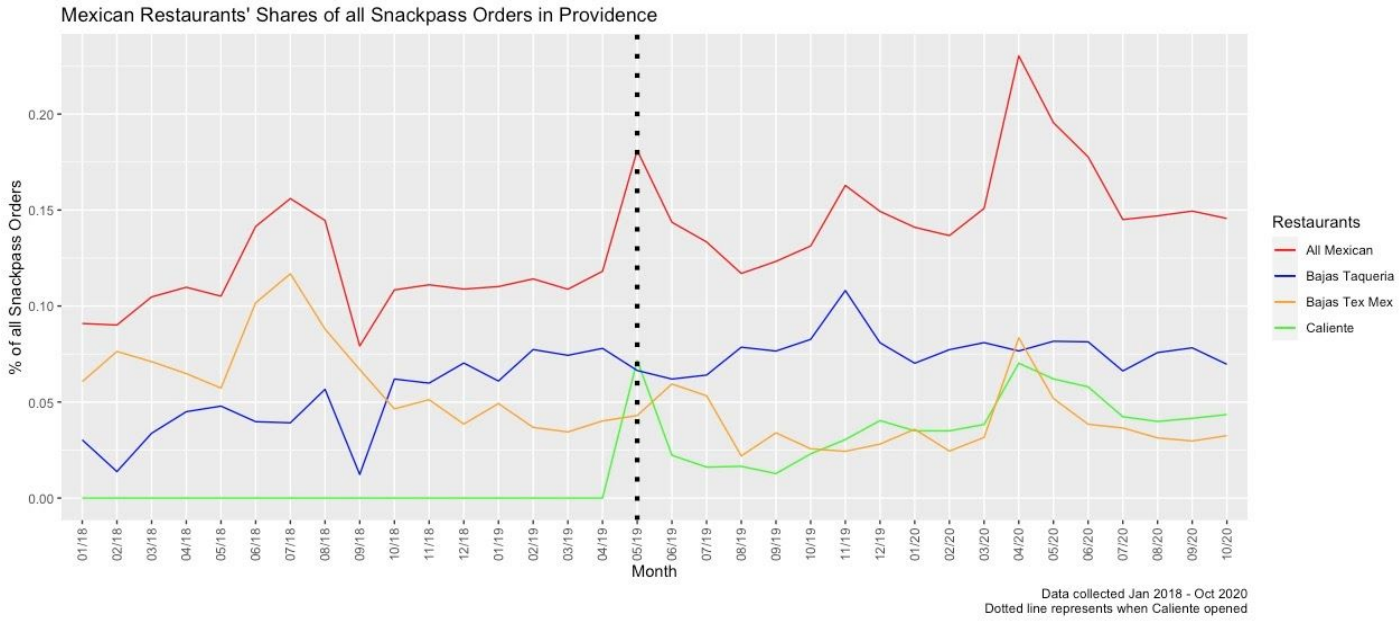


This graph, depicting aggregate order activity at the top ten most popular restaurants in Providence, is interesting for many reasons. For example, it shows the seasonality of Snackpass activity at Brown. We can clearly see dips in Snackpass activity during June-August and January, months when Brown students are mostly on recess. This suggests that Snackpass activity, and likely overall restaurant activity at Brown, is highly dependent on the student body.

We are also able to capture the effects of the pandemic in the shaded region, which represents the period of COVID-19 lockdown. During this time, students were sent home from Brown and any students that happened to stay were discouraged from going to restaurants. As we can see, there was a significant decline in activity during that time, however it appears that activity picked up again when students came back to campus in September.

One last interesting point regarding this graph is the outstanding success of Thayer Bagel Gourmet Ole (“BGO”, according to students) on Snackpass. BGO has wildly outperformed its nearest competitor on the app, especially in the last year. We hypothesize that this is because students might order breakfast as a meal more consistently. Also, BGO lends itself well to a fast-pickup service; it has limited seating, its food is more compatible for on the go, and it was already pretty popular.

Figure 4



This last graph really satisfied a curiosity that we had about Mexican restaurants at Brown. As students, we witnessed a fourth restaurant opening at Brown (Chipotle is not listed because it is not on Snackpass), and were wondering how the market for Mexican restaurants at Brown were affected (given that it seemed we already had more than enough Mexican food supply). From the dotted line on the graph we can see that the additional Mexican restaurant, Caliente, opened in May of 2019, and interestingly, aggregate Mexican restaurant activity continued to grow. Furthermore, the existing Mexican restaurants appear to have continued with similar levels of activity. This suggests that there were not significant competitive effects with the addition of Caliente and that Brown students consume more Mexican food when provided with more Mexican food.

Methods

After our initial exploration of the data, we set our sights on answering our main question of predicting cyclical consumer behavior.

Specifying the Task

Our first step in approaching this prediction question involved taking a step back and determining what exactly we wanted to answer: given what inputs, what outputs? As alluded to in the Research Questions section, we realized that our scope would have to be narrow to accommodate our timeframe. We also wanted to make a model that was practical - something that could hypothetically be utilized by Snackpass. With this as our guiding principle, we specified our task to predict: Given some time period X, what is the likelihood that a user orders in period X?

Our next step was to determine what exactly a “time period” should be. Orders need to fit into some unit of time broader than their current representation as a timestamp down to the millisecond. For example, one could imagine hourly units; i.e. what is the likelihood that a user will order from 11am-12pm? We believed that this type of an approach would lead to poor results because ordering behavior is not frequent enough to merit such granular predictions.

Instead, we decided that we would think about orders as belonging to a date and meal--breakfast, lunch, or dinner. This makes more sense to us because it splits the day into manageable buckets of time and is more aligned with how students think about ordering food. That is, students think about ordering for a meal rather than for an hour of the day.

With this in mind, we have specified our question to be, Given some meal and some date, what is the likelihood that a user will order? This question was granular enough to be interesting and general enough to be accomplishable.

The next step was to organize the data in a way that would allow us to answer this question.

Organizing the Data

To organize and structure the data to answer our question, we used a structure that we refer to as a “tranche”. A tranche represents a date and a meal for a single user. For example, ‘10-20-2020 lunch’ represents one tranche. Each order is mapped to a tranche based on the timestamp of an order. Each user has a list of tranches, one tranche for every possible date and meal since their first order on Snackpass.

Each tranche contains two things: the truth value for the question we are asking, and predictive features pertaining to the tranche. The truth value in our case is the number of orders that occurred at that tranche. The truth value was critical to include because it allows our model to be trained and tested. It is the answer to our question: Given some meal and date (i.e. tranche),

what is the likelihood that a user will order? The predictive features are features that were engineered to provide more information about Snackpass activity at the time of that tranche. We will discuss the specifics of the features in the Feature Engineering section below.

After building every traunch for every user, our data was primed to approach the predictive problem.

The Model

After organizing the data in a way that was useful, we had to decide on the method of answering the prediction problem. We ultimately decided on using a decision tree algorithm, XGBoost (extreme gradient boost) regression. We decided to use XGBoost regression because we were familiar with decision trees from class, and it provided an interesting opportunity to expand our studies from class to a more complex algorithm. XGBoost allows us to predict the likelihood that a user will place an order in a tranche, discussed more in the Modeling and Execution section below.

One fundamental question that we had to grapple with to solve the problem using a decision tree algorithm is how to handle tranches during times when school is not in session. These include summer break, winter break, spring break, and Thanksgiving. This posed a problem to us because, for various reasons, Snackpass activity is significantly different when school is not in session. Most importantly, during the breaks, most students are not using Snackpass at all because they are not on Brown's campus. Incorporating the semester breaks into the scope of the model would have added a significant level of complexity to the prediction problem, and ultimately we decided that this would be too difficult.

With this in mind, we modified our approach in two ways. First, we removed every tranche during a break, and we limited our analysis to Providence restaurants. This focused the data on Brown students' behavior during the school year. Next, we limited our analysis to orders after January 1st, 2018. We did this because the final months of 2017 after the initial launch of Snackpass has far sparser activity, and doesn't reflect the reality of using Snackpass at Brown today. The removed tranches, those outside of school semesters and ones before January of 2018, represented only ~2% of all Brown orders. From this point forward, our data only pertains to Providence Snackpass orders placed on or after January 1, 2018 and during a school semester.

Following these modifications, we were left with ~7,000 unique Providence users and ~6 million tranches total. At this point, we had a comprehensive representation of our data in a form that was useful to the question we wanted to answer, and we were ready to build out the predictive features.

Feature Engineering

Feature engineering was an important part of the project because, when using a decision tree algorithm, the quality of the features determines the quality of the predictions. We really focused on engineering features that we believed were most indicative of Brown student Snackpass activity in Providence. After much deliberation we decided to use the following features:

Note: XGBoost only accepts numeric (rather than categorical) values, so we employed one-hot encoding to adjust our previously categorical variables (namely, meal and day of week). Thus, all feature values below are numbers.

Feature	**Meaning and Significance**
"Monday"	1 if the tranche is on a Monday, 0 otherwise. This allows us to capture the day of week in our model. We thought this was useful because, from personal experience, student behavior on Snackpass is largely dependent on student schedules, which are mostly determined by their classes. At Brown, students have classes on Mon/Wed/Fri or Tues/Thur. Thus, Snackpass behavior might be heavily dependent on the day of the week. Additionally, behavior on weekends may differ from weekdays.
"Tuesday"	1 if the tranche is on a Tuesday, 0 otherwise. See "Monday".
"Wednesday"	1 if the tranche is on a Wednesday, 0 otherwise. See "Monday".
"Thursday"	1 if the tranche is on a Thursday, 0 otherwise. See "Monday".
"Friday"	1 if the tranche is on a Friday, 0 otherwise. See "Monday".
"Saturday"	1 if the tranche is on a Saturday, 0 otherwise. See "Monday".
"Sunday"	1 if the tranche is on a Sunday, 0 otherwise. See "Monday".
"Breakfast"	1 if the tranche represents breakfast hours, 0 otherwise. We

	define breakfast hours as 7am-11am. This allows us to capture the meal in our model. This was important because behavior across meals can be correlated. For example, it might be the case that people have a morning routine, which affects their Snackpass behavior specifically at breakfast. NOTE: Because mealtimes for breakfast, lunch, and dinner only span from 7am - 9pm, we simply ignore any Snackpass orders not in this timeframe. This is ~8% of all Brown orders.
"Lunch"	1 if the tranche represents lunch hours, 0 otherwise. We define lunch hours as 11am-3pm. See "Breakfast".
"Dinner"	1 if the tranche represents dinner hours, 0 otherwise. We define dinner hours as 3pm-9pm. See "Breakfast".
"# Orders Previous Meal In Snackpass"	The average number of orders per active user in the previous tranche across all users. We define an active user as someone who has used the app at least once prior to the given tranche. This acts as an indicator for random shocks that may impact certain days. For example, something like Spring Weekend or the Brown dining halls being closed may systematically increase Snackpass orders.
"# Orders Past 24 Hrs" Note: +	The number of orders in the past 24 hours from the starting hour of the tranche. This allows us to capture the behavior immediately before the tranche of interest. We made sure to specify that orders from the tranche itself were not included in this metric, because we wanted to capture past behavior only.
"# Orders Past 3 Days" Note: +	The number of orders in the past 3 days from the starting hour of the tranche. This allows us to capture behavior in the days leading up to the tranche of interest. This represents similar information to "# orders past 24 hours" but with a greater cushion for immediacy.
"# Orders Past 7 Days"	The number of orders in the past week from the starting hour of the tranche.

Note: +	This is similar to the above two features. Weekly behavior is slightly more consistent than the past one or three days, so we include it as another recency metric.
"# Orders Past 30 Days" Note: +	The number of orders in the past 30 days from the starting hour of the tranche. This, too, is similar to the previous features. We include it to have a wider metric for ordering frequency.
"% Orders Same Meal In Sem." Note: *	The percentage of orders that are in the same meal as the tranche within the current semester. This allows us to capture consistency in ordering habits. If users have only used Snackpass for a certain meal, that can be highly informative to their future behavior.
"% Orders Same Day of Week In Sem." Note: *	The percentage of orders that are on the same day of the week as the tranche within the current semester. This also allows us to capture consistency in ordering habits. If users have only used Snackpass on certain days, that can be highly informative to their future behavior.
"Feels Like Temp." Note: -	The temperature (in Fahrenheit) at the starting hour of the tranche. This allows us to capture how temperature affects order behavior. We predicted that higher temps would be correlated with more orders, while lower temps would be correlated with fewer orders. This could be due to a variety of reasons; people may order less in the colder months as a whole or people might not want to go outside to pick up their order when it is cold.
"Rain Amt." Note: -	The rainfall at the starting hour of the tranche. This allows us to capture how rainfall affects order behavior. We predicted that higher rainfall would be correlated with fewer orders; intuitively, students would be less inclined to go outside to pick up their Snackpass orders if it is raining.
"Snow Amt." Note: -	The snowfall at the starting hour of the tranche. This allows us to capture how snowfall affects order behavior. We predicted that higher snowfall would be correlated with fewer orders for the same reason provided for rainfall; no one wants to venture out into the snow even to pick up their food.

“Avg. # Orders / Week In Sem” Note: !	The average number of orders per week in the semester. This allows us to capture student behavior on the semester level. We believe that behavior over the course of a semester is consistent, and students who use the app frequently will continue to do so and vice versa. We average this on the “per week” level to standardize the metric across different weeks of the semester.
“Avg. # Orders / Week All-time” Note: !	The average number of orders per week for all-time. This allows us to capture the user’s Snackpass activity as a whole. While it is possible that each user’s Snackpass activity has changed over time (as discussed above), it is also possible that users have used the app in a mostly similar manner since first downloading it. This figure would capture any lasting behavior that is consistent from the user’s first download.
“Avg. # Orders Same Day Of Week & Meal In Sem” Note: !	The average number of orders per week in the semester where the order was placed from the same day of week and meal as the tranche. What this is meant to capture is similar behavior for the same day of week and meal in the current semester. As discussed before, we predict that student behavior for each meal of the week might differ by semester due to class schedules. For example, a student might get lunch with friends right after their class every Wednesday. This feature is similar to “Avg # Orders / Week in Sem” except it also provides some level of granularity from the actual day of week and meal.

+ For all features denoted with a “+”, i.e. those counting the number of orders in the past X days, we had to consider an additional question: How do we handle the case when the past X days include days that are not in the school semester? User behavior varies drastically when users are in semester versus not. Notably, when students are not in semester they are unlikely to be using Snackpass at all because they are unlikely to be in Providence. To handle this, we created an intermediate feature (i.e. not used in the final model) that stored which semester the tranche belonged in. The semester was represented by an integer 1-7 for every possible semester since the beginning of the app, or -1 if the tranche was not during a school semester. Thus, for every tranche we were able to determine whether it was from the same semester as the tranche in question. Finally, when counting the number of orders in the past X days we assigned a null value for the feature if any of the tranches in the past X days were not in the same semester. We found it sufficient to use a null value because there were relatively few cases where this was the case. We used the Brown academic calendar to accurately mark the start and end of each semester.

- For all features denoted with a “-”, i.e. those related to weather, we had to consider an additional question: How do we integrate the weather dataset into the tranche features? In particular, since tranches represent a date and mealtime, which spans for multiple hours, we had to decide from which hour to pull the corresponding weather data from our weather dataset. We decided that the starting hour of the tranche was sufficient for our purposes. While an argument could be made that for dinner, which is 3pm-9pm, weather could vary considerably in the six hour time span, we decided that data from the starting hour of the tranche was close enough.

✱ For all features denoted with a “*”, i.e. those that take percentages, we made a conscious decision to use a percentage. We could not simply use absolute counts because that would penalize tranches that were earlier in the semester. Using percentages is a measure to treat all tranches equally.

! For all features denoted with a “!”, i.e. those that take averages, we had to make the same considerations as those explained above (the features denoted with “*”). Again, to avoid punishing tranches closer to the start of the semester for having fewer orders, we took an average for certain metrics rather than an absolute count. This achieved the same purpose as taking a fraction by avoiding favoring tranches toward the end of the semester where the absolute number of orders was inherently higher.

Modeling and Execution

XGBoost (extreme gradient boost) is a relatively new and extremely popular machine learning algorithm. We will not delve into the details of how the algorithm works or the advantages it offers over traditional ML tree-based algorithms, but that information can be found here and here. After evaluating a wide selection of ML algorithms, we decided that XGBoost was most beneficial to us because of its efficiency with large amounts of data and its ease of tuning a range of hyperparameters such as learning rate, tree depth, and more.

XGBoost has two different frameworks for prediction: classification and regression. Our first instinct was to employ the classification framework. However, we quickly realized that this may be suboptimal due to the makeup of our data. Our primary concern was the dramatic imbalance between tranches with and without orders in our data. Of the ~6 million tranches, only 2.9% actually contain an order. Because this percentage is so small, we were concerned that this classification algorithm would over-classify tranches as non-orders to achieve minimal loss.

An algorithm that classifies everything as non-orders may be highly accurate in our case (when only 2.9% of tranches contain orders), but in practice is mostly useless. We prefer a model that has a lower overall level of accuracy but can correctly predict a larger share of instances in which a tranche did in fact have an order. If we were to analogize this to advertising, it would be more useful to predict that an order will happen than to predict that an order will not happen.

More formally, we used three main model evaluation criteria for the predictions:

1. $P(\textit{correct prediction})$
2. $P(\textit{was an order} \mid \textit{predicted an order})$
3. $P(\textit{predicted an order} \mid \textit{was an order})$

In words:

1. The overall accuracy of the model. This is how models are traditionally evaluated.
2. The probability that the tranche contained an order given that we predicted there would be an order.
3. The probability that we predicted there was an order given that the tranche contained an order.

Although we remain cognizant of the overall model accuracy, we were particularly interested to see how we could boost evaluation criteria #2 and #3.

With these ideas in mind, we chose XGBoost's regression framework instead. Because our order 'truth' values are almost entirely 0s and 1s (a user rarely orders more than once in a tranche), the regression output will be a value between 0 and 1. We can treat this value as the probability of an order in the tranche.

To prevent over-classification of non-orders, we can then manually adjust for a bias towards non-orders. We transform the order probabilities to order classifications using a probability cutoff of our choice. For example, we can classify any tranche with an order probability greater than 0.4 to have an order. We monitor our three evaluation criteria as we adjust the probability cutoff.

We ran the model with industry-standard hyperparameter values. We used a learning rate of 0.1, used a maximum tree depth of 5, sampled a random 80% of predictors for each tree, and sampled a random 80% of the training data for each tree. Additionally, we used an 80/20 train test data split. We do not believe that we overfit our model because its prediction performance on the testing and training data sets are nearly identical.

From this setup, we were able to produce meaningful results, discussed below.

Results

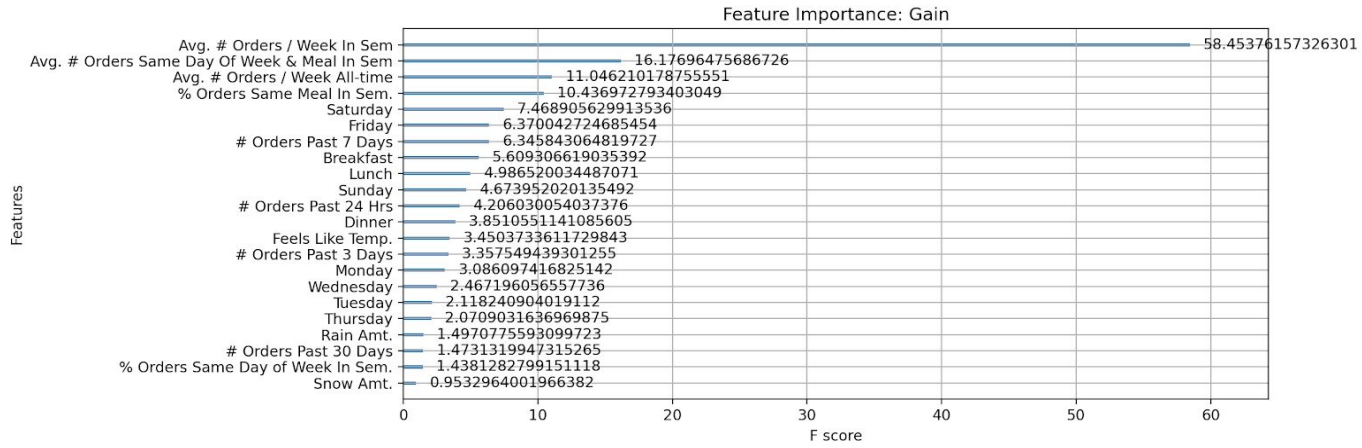
We ultimately decided that a 0.15 probability cutoff best suited our data and the tradeoff we were willing to make between a higher overall accuracy and higher percentages for evaluation criteria #2 and #3. Our evaluation criteria stabilized around the following values when we measured the model predictions against the true values in the testing samples:

1. $P(\textit{correct prediction})$ = 94%
2. $P(\textit{was an order} \mid \textit{predicted an order})$ = 25%
3. $P(\textit{predicted an order} \mid \textit{was an order})$ = 40%

We believe that these results demonstrate that we can meaningfully predict Snackpass user behavior. When we predicted a user would order, we were correct 25% of the time. When a user ordered, we had predicted they would 40% of the time. We believe that with additional work to the model we could still improve upon these numbers.

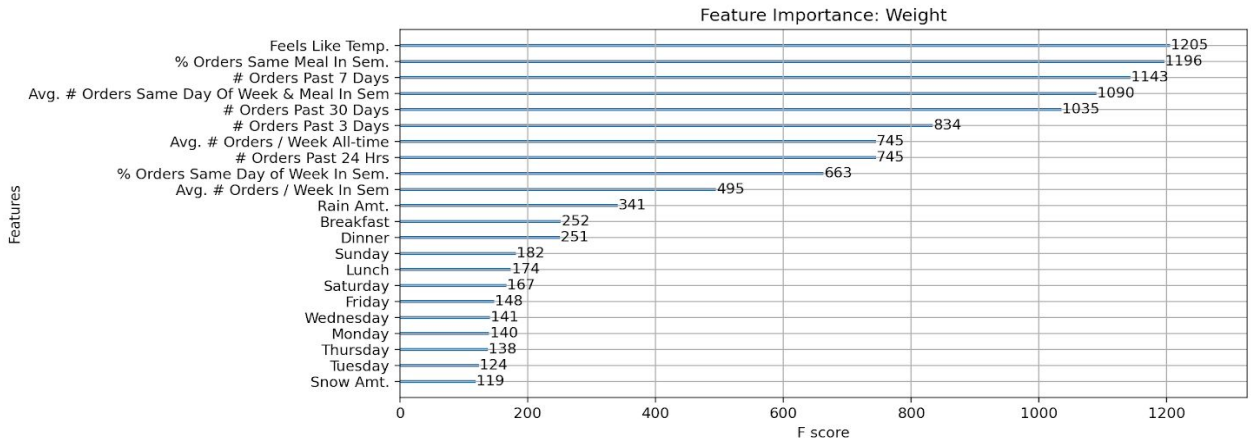
Beyond the high-level accuracy percentages, we were also interested to see which feature predictors were most impactful to the model.

Figure 5



In Figure 5, we can see the relative ‘gain’ among all of the predictors. Gain is defined as “the average gain across all splits the feature is used in.” This is the most relevant measure of how important a feature is to the predictions in the model. As we can see, the features that performed strongest were measures of ordering regularity, averaged over some period of time, particularly within the semester of interest. What this suggests is that if someone has used Snackpass consistently this semester, they will likely continue to do so. This is unsurprising to us given how we observe members in the Brown community using (or not using) Snackpass.

Figure 6



In Figure 6, we can see the relative ‘weight’ among all of the predictors. Weight is defined as the “the number of times a feature is used to split the data across all trees.” Interestingly, there is a moderate correlation between weight and gain, but it is not perfect. For example, the feature with the largest weight is ‘Feels Like Temp,’ but this has relatively low gain.

Overall, however, the gain and weight of the features aligns with our expectations. The biggest surprise was likely how little the “Past X Days” features seemed to matter. We spent a lot of time considering and implementing features, so it was very rewarding to see how they stack up against each other.

Future Work

With more time and guidance, there are a few noteworthy changes we would have liked to make.

First, and most importantly, we wanted to write a custom objective (loss) function for XGBoost. By default, the XGBoost regression framework uses Root Mean Squared Error (RMSE) as its objective function. This function treats all ‘loss’ as equal. Mathematically, misclassification of orders and non-orders are equally significant. We want to change this. More formally, we want to implement an asymmetric loss function because we believe that misclassifying an order as a non-order is worse than misclassifying a non-order as an order. This would likely reduce the need to use a probability cutoff as low as 0.15. Perhaps we could even switch to the XGBoost classification framework with the custom objective function.

Second, we want to optimize our XGBoost hyperparameters. We would like to perform a grid search to optimize our parameters. We believe that this could provide marginal improvement to the model.

Third, we currently do not consider when students graduate or go abroad. They simply continue existing in the model with tranches despite no longer being on campus to place orders. We would likely retroactively remove these students. We have not thought deeply about this problem, but we would likely remove tranches for a student who has 0 orders in a semester. Although it is possible this removes some users still in Providence who never use the application, this is probably still a smart decision.

Conclusion

This project was especially rewarding because it had multiple distinct, interesting, and challenging steps. From learning how to use the mitm proxy, to delving into the modeling complexities of dealing with millions of data points, to feature engineering, and to XGBoost, we pushed ourselves each step of the way.

In the end, we believe that we produced a robust model with significant results. With a fairly high degree of certainty, we can predict user behavior on Snackpass. We believe that we can do even better with more time and expertise guiding us to really take the project to the next level.