# Trends and Patterns in Instacart Data

José Gutiérrez
Computer Science
University of Colorado, Boulder
Boulder, Colorado, USA
jogu0215@colorado.edu

Anna Malawista
Computer Science
University of Colorado, Boulder
Boulder, Colorado, USA
anma7757@colorado.edu

Michael Whitlock
Computer Science
University of Colorado, Boulder
Boulder, Colorado, USA
miwh5549@colorado.edu

Keaton Hoeger
Computer Science
University of Colorado, Boulder
Boulder, Colorado, USA
keho3676@colorado.edu

Aidan O'Connor
Computer Science
University of Colorado, Boulder
Boulder, Colorado, USA
aioc7481@colorado.edu

## Abstract

Instacart gives a comprehensive list of orders and the contents, time of the order, and other information useful for the classification and prediction of future orders. Instacart already provides suggestions on other items a customer might also be interested in, likely combing the customers purchases and products which are popular on the site. We had a good number of general questions when starting this project, and formulated a number of others as we went through the data preprocessing stage.

One question or goal we were seeking to accomplish was if we could accurately predict a customers next purchase. Where this might be the most likely item that a customer will purchase in their next order, or a collection of items we might suggest for them to purchase. We found that we could train a model to predict if a person would reorder 50% or more of the items they had previously purchased knowing only what items they have purchased before. We found that an support vector machine (SVM) using a rbf kernel was the most effective method; however, we also concluded that given enough time to optimize all of the parameters a multilayered perceptron could possibly outperform the SVM.

We also wanted to find out if there were noticeable trends of both order size and density over the course of a week. This type of information would be useful in ordering and preparing stock so that it would be available for any surges of purchases across the week. We found that the highest densities of orders for most departments was the greatest on Monday and Tuesday. With a couple exceptions, most notably alcohol which was ordered the most on Fridays and Saturdays.

During the preprocessing stage we started to ask if there were interesting trends in the spread of products/departments and how they are either reordered or not. We found that obviously perishable items were the most re-ordered, and non-perishables were often not reordered. We however found that there was a higher individual reorder count for perishable items like milk and eggs, where these had a smaller variety of products of which customers could purchase from. While fruits and other perishables which

had a larger variety in suppliers and varieties, had lower individual reorder counts.

We found some interesting trends during preprocessing concerning orders of Junk Food and healthier foods such as fruit and the like. We wanted to know if there were any significant trends of these two groups over a week, and if their order densities could be useful in making useful conclusions about the dataset. Healthier foods were purchased more in the general increase in orders at the start of the week, while junk foods were ordered most frequently on Saturday. These conclusions were helpful in identifying day 0 of the dataset to be Monday, in conjunction with the patterns like alcohols most frequently purchased days.

## Introduction

Instacart is an online grocery delivery service. Our group's motivation was to analyze patterns from Instacart orders. We wanted to determine if there are trends in the types or rate of products ordered depending on the time of day, week, and frequency of repurchases. Some of the questions we wanted to tackle were: "Will a product be reordered based on the order it was placed into the shopping cart or what type of item it is?"; "How do orders vary across time (day of week and time of day)?"; "Which items are most likely to be added order, based on the items already in the cart?"; "Based on purchase history, can we predict which new items a customer is most likely to purchase?"; and "Can we predict the length of time between orders?"

To address these questions began by using exploratory data analysis techniques to examine the data. We then began creating charts, graphs to give us an initial idea of the relationship between attributes. We moved on to more in depth classification methods and implementing an machine learning model. For the machine learning models there are additional orders for both training and testing our machine learning models of size 131,000 and 75,000 respectively. We intend to test multiple models in order to find the one that performs best and then optimize that model through hyperparameter tuning.

Customer acquisition and effective marketing are two of the largest obstacles companies face. Marketing consultants are estimated to make anywhere from $65-$300 an hour to help companies determine how best to sell their products [10]. Thus, there is a huge market for information on trend in customers purchases. However, there are a limited number of datasets containing data on customer purchases that are as vast, comprehensive, and publicly accessible as the Instacart dataset.

The Instacart dataset is comprised of 6 tables that are linked by ID attributes for orders, departments, etc, and constais 3.2 million data points in total for analysis. The dataset is very clean and simple to follow; there are no null or missing values. The data set provides information on what was ordered (including department and aisle of item), the day of week and time of day that the order was placed, as well as the amount of time between orders. However, no information is provided on what time of year items were purchased. A "data dictionary" was provided by Instacart and is available on GitHub [8].

The size and depth this dataset makes it the perfect tool for trying to analyze and draw conclusions from grocery purchases. Though ordering groceries from an app is a newer, less conventional form of grocery shopping, it provides some distinct advantages when it comes to data mining. Due to the mode of purchase instacart was able to provide information about the time between purchases and maintain a history for individual users. In a standard grocery store the id of the purchaser is not recorded at each transaction, and thus this information is usually lost.

It is for these reasons that we believe that this analysis of  the Instacart dataset will  unique and informative look in the purchasing trends for groceries, and we believe that these findings could help a wide range of companies with marketing and customer acquisition.

### Related Work

Instacart posted the dataset on their website and a brief description on Medium [1, 2]. We have found that most of the prior work on this dataset is exploratory in nature. Instacart ran a competition on Kaggle in early 2016. Various analyses of this dataset were conducted as a result of this competition. One of the primary areas of research done for this competition looked at the trends related to repurchasing [3, 4].

Instacart's Medium post provides two short examples of patterns derived from the dataset; one on repurchasing and item popularity, and the other on the difference in "healthiness" of orders depending on the time of day [2]. This data (and other similar datasets) have also been used in a number of studies looking at buyer recommendation and customer targeting strategies [5, 6, 7].

### Data Set

Our dataset is from Instacart, a company which created an app that provides a grocery ordering and delivery service. This dataset is made up of 6 different CSV files [1]. The dataset is very clean and simple to follow; there are no null or missing values. According to Instacart, the data is anonymized and is a sample of over 3 million orders from more than 200,000 unique Instacart users [1]. There are prelabeled training and test sets. The data set provides information on what was ordered (including department and aisle location), the day of week and time of day that the order was placed, as well as the amount of time between orders. However, no information is provided on what time of year items were purchased, so no information can be drawn on the fluctuations in ordering across seasons. A "data dictionary" was provided by Instacart and is available on GitHub [8].

There are a few issues we have noted about this dataset. The purchase day of the week is provided as an integer from 0-6; however, thus far we have been unable to find any information on what day of the week each integer corresponds to. Others have documented this same issue, and most have concluded that based on the rate of ordering 0 represents the beginning of the weekend (Saturday) and that the days of the week are in order from that point [4,8]. This conclusion was drawn because there is a large spike in orders on day 0 and day 1, which one could attribute to the amount of free time and social gatherings people tend to have on the weekend. However, in the process of our initial research, we have discovered multiple pieces of data that have actually lead us to the conclusion that day 0 is in fact Monday. These findings are explained in the "Orders across Time" subsection of our results, which can be found on page 5 of this text.

One other issue we will face is that all data in the "days_since_prior" order attribute is capped at 30 and the number of orders for any user is capped at 99. Rather than truncating the data at these points, we have concluded that all data greater than these limits were simply assigned that max value (censored). This conclusion is based on a large spike in the data at 30 for "days_since_prior" and 99 for the number of orders per user. In most of our data visualization, we will be able to simply label this part of the data as >=30 or >=99. However, we are still assessing if we should try to account for the possibility of this censored data skewing our results through preprocessing, and if so, what the best method would be. Overall, this data set is large, clean, and fairly comprehensive; and it should provide ample opportunity for analysis and prediction.

## Main Techniques Applied

To perform out data cleaning and analysis we mainly be used Python, as the language provides a lot of libraries that are helpful in both data analysis and any machine learning we want to attempt on the datasets. NumPy and Pandas were two of the main libraries we used, and Scikit-learn is a Python library that we used for machine learning.

### Data Cleaning

As mentioned in the dataset section, our dataset is very clean with very few null or missing values. However, the days since prior order and number of orders from a single individual are "censored" at 30 and 99 respectively. After doing research on censored data, we have decided to maintain the data set as is and simply note that these data points refer to any data that is greater than or equal to that value. We will also aim to predominantly use our many uncensored data attributes whenever possible.

### Classification

We utilized rule based classification to test rules we extrapolated from our data preprocessing.

We utilized a K-Nearest Neighbors (KNN) classification algorithm to predict whether a product would be reordered based on the product_id and the order it was placed into the shopping cart. Our group trained the KNN model using the training dataset provided by Instacart.

### Machine Learning

We used sci-kit learn's DictVectorizer tool to create a sparse matrix to see if we could train a model to predict if a person would reorder 50% or more of the items they had previously purchased knowing only what items they have purchased before. We then used 1000 instances of the training data in order to optimize the hyperpatemer c of an SVM model once using a linear kernel and again on a model using a rbf kernel. After finding the best c value we, trained each model on the full set to evaluate their final testing accuracy.

Our next step was to train a simple multilayered perceptron to determine if we had enough training data to allow a neural network to provide good results and see how it compared to the SVM models. Finally, we also trained a random forest classifier to see how an ensemble of decision trees could compare with the SVM and MLP models.

## Key Results

### Size of Orders

In our exploratory data analysis, we looked at rates at which different types of products are ordered, trends across time, and the amount that products are reordered. Our exploratory data analysis has shown that over 90.56% of the orders contain 20 items or less, and between 5 and 7 items is the most common size for an Instacart order (Figure 1).

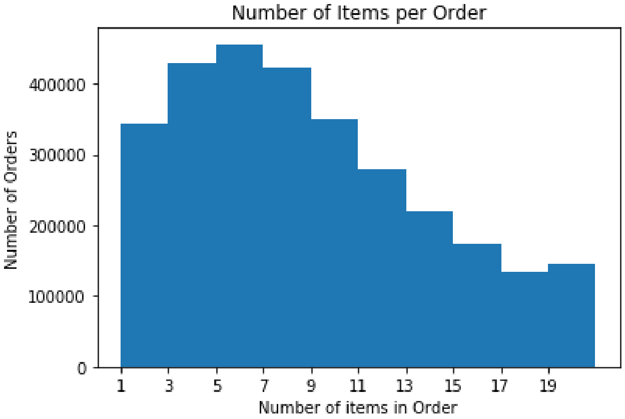


Figure 1: Histogram of number of items per order for all orders with 20 items or less

### Product Reorders by Department

We found that the most commonly ordered department by a large margin is produce making up 29.23% of the items ordered. Dairy/Eggs were second at 16.29%, with all other departments making up less than 10% of the items ordered (Figure 2). Dairy/Eggs and produce being the two highest ordered items makes sense, as they last a short time before spoiling and must be ordered frequently. The departments, bulk, other, missing, pets, alcohol, and international; none of which are known for having very perishable items,  each make up less than 1% of the total items ordered.

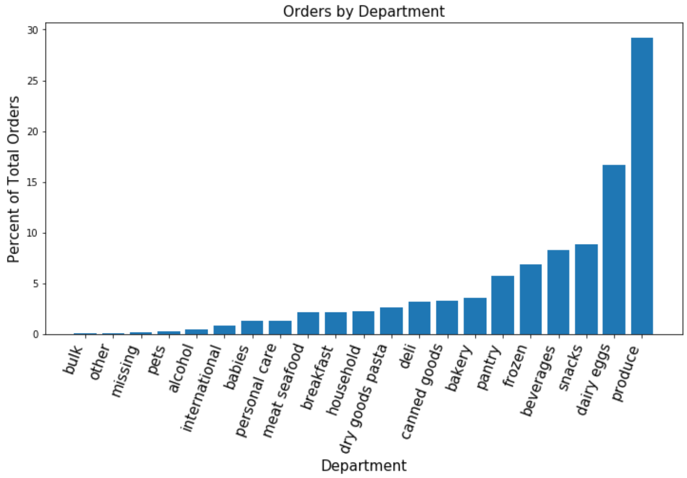


Figure 2: Bar graph of percent of total items ordered from each department

We also checked the percent of products re-ordered by department. Dairy/Eggs was the department that items were most commonly reordered from, followed by beverages and then produce (Figure 3).

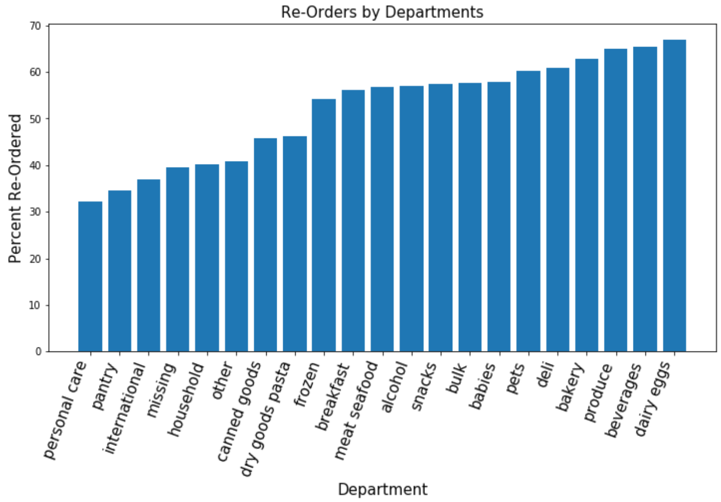
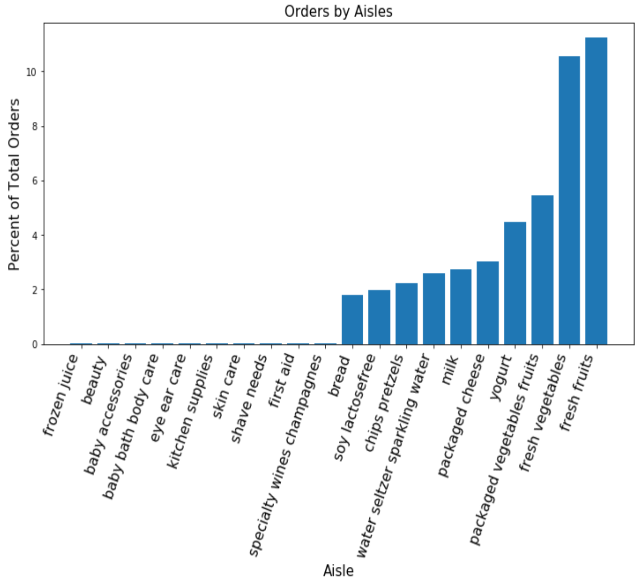


Figure 3: Bar graph of percent of items re-ordered from each department

The discrepancy between produce being the most commonly ordered from department by a large margin and being third for items being repurchased, indicates that shoppers tend to vary what produce they purchase between orders more than in the case of dairy/eggs or beverages. This makes sense given the great variety of produce and that people preferences for produce tend to change by season, as compared to dairy/eggs or beverages which stay more consistent and are may be more likely to foster brand loyalty.

Product Reorders by Aisle

We found the aisles most commonly ordered from were fresh fruits and fresh vegetables at 11.23% and 10.54% of items ordered respectively, with all other aisles making up less than 10% of the items ordered (Figure 4). This  is consistent with produce being the most commonly ordered from department.



The re-orders by aisle have milk as the most re-ordered aisle, followed by the water then fresh fruits aisle. Similar to the departments, milk and water being the most reordered from aisles, when they each make up less than 10% of the total items ordered, indicates that customers are more consistent and loyal with these purchases as compared to produce from which customers more often vary their purchases. The least commonly reordered from aisles are spices, baking supplies, and first aid; all of which tend to sell non-perishable or long shelf long items, which could last customers for extended periods of time.

Figure 4: Bar graph of percent of total items ordered from each of the top and bottom 10 most ordered from aisles

Orders across Time

As mentioned in the dataset section, others have concluded that based on large spike in orders on day 0 and day 1, 0 represents the beginning of the weekend (Saturday) and that the days of the week are in order from that point [4,8]. However, based on our initial research, we have hypothesised that day 0 is in fact Monday. The evidence that lead us to this conclusion is presented within this section.

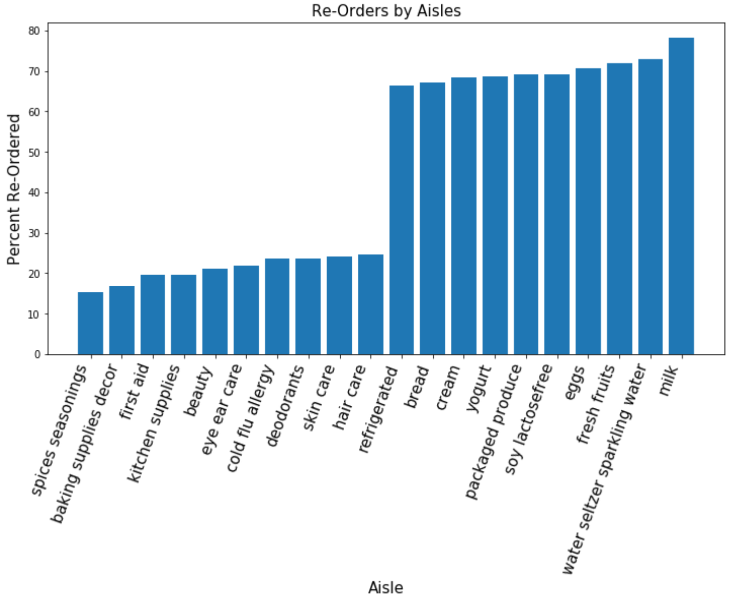


Figure 5: Bar graph of percent of items re-ordered from each of the top and bottom 10 most re-ordered from aisles

We began looking at how instacart orders varied across time by analyzing and plotting the frequency of total orders across day of the week and through time of the day. We looked at the volume of ordering throughout the week in two different ways, the first being the total number of items ordered in a day and the second being the number of different orders placed regardless of the number of items in each of those orders. Day 0 is the day on which a majority of items are ordered (19.15%), with day 1 having the second highest percentage of items ordered (17.47%) (following days: day 2 = 13.00%, day 3 = 11.85%, day 4 = 11.68%, day 5 = 12.98%, day 6 = 13.88%).

Likewise day 0 & 1 are the days on which the greatest number of orders were placed (regardless of the number of items); however, the percentage of orders placed on these two days is actually nearly equivalent (17.35% and 17.32% respectively) (following days: day 2 = 13.74%, day 3 = 12.83%, day 4 = 12.48%, day 5 = 13.25%, day 6 = 13.03%).

This difference between the number of items ordered and the number of orders placed led us to conclude that the size of orders placed on day 0 is larger than the size of orders placed on day 1. We then calculated the mean and median order size for both day 0 and day 1 to confirm this finding, and we found that on day 0 the mean order size is 11.13 items and the median is 9 items, whereas, on day 1 the mean order size is 10.18 items and the median is 8 items. All the other days of the week have a mean in the range of 9.32-9.88 items with a median of 8 items, except day 6, which has a mean of 10.74 items and a median of 9 items.

The most common hours of the day for total items ordered and number of orders placed were between 10 am and 2 pm, with approximately 8% of items being ordered each hour, and the least common hours of the day were between 12 and 6 am, with less than 1% of items being ordered each hour.

The visualization of this data was difficult to create because of the size of the dataset, so we used a sample of 10,000 items ordered to make the KDE plot. The visualization displays the distribution of the number of items ordered by day of week and hour of the day. This graph shows that on day 0 most items are ordered in the afternoon, while on day 1 the bulk of items are around 10:00 am (Figure 6).

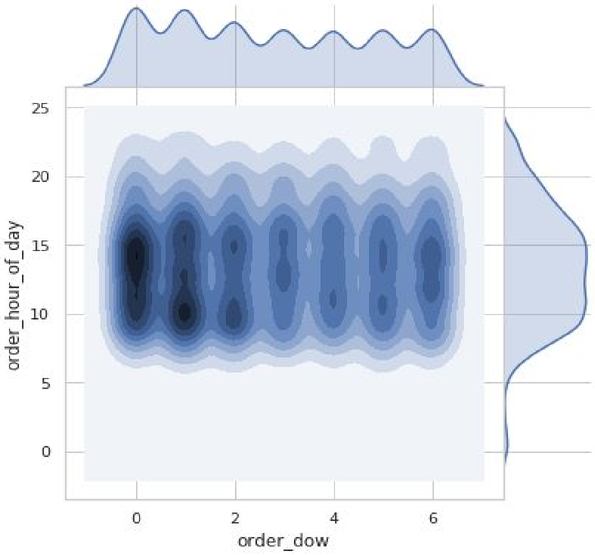


Figure 6: KDE plot of the number of items ordered by day of week and hour of the day

We then looked at how orders varied throughout the week separated by department. Every department except alcohol followed the pattern of total items ordered, with day 0 or day 1 being the day that the most items from that department were ordered. The alcohol department, on the other hand, had the most items sold on day 4 & 5 (17.01% and 17.85% of alcohol items sold respectively), and on day 0 alcohol was actually sold at the lowest rate (11.35% of alcohol items sold). So, despite the spike in orders placed on day 0 & 1 (which led others to conclude that this was the weekend), alcohol, a product that is often served at parties and other social gatherings or consumed when individuals do not need to work the next day, is still sold at a much higher rate on day 4 & 5. This was our first piece of evidence that day 0 may in fact be monday, making day 4 friday and day 5 saturday.

We thought this conclusion made sense intuitively because we typically encounter individuals beginning their count of weekdays on monday or sunday, to fit with common calendars; however, we wanted to further test this hypothesis. We next compared the most commonly products ordered from departments that would likely contain “junkfood” on day 0, which we hypothesised to be monday, to the most common products ordered on day 5, which we hypothesised to be saturday. We assert that if food we deemed to be unhealthy or “junkfood” appears to be ordered more frequently on day 5 than day 0 then this would support or hypothesis that day 0 is monday as “junkfood”, along with alcohol, is often consumed at parties and other social gatherings that tend to be more common on the weekend.

We looked at two departments snacks and frozen. We chose these departments because the snacks department includes items like chips and cookies and the frozen department includes items like ice cream, all of which are considered common types of “junkfood”. For the snacks department we found that the top 5 most commonly purchased items on day 0 and day 5 were as follows:

| Day 0 | Day 5 |
|---|---|
| Lightly Salted Baked Snap Pea Crisps | Lightly Salted Baked Snap Pea Crisps |
| Trail Mix | Sea Salt Pita Chips |
| Original Veggie Straws | Organic Tortilla Chips |
| Pretzel Crisps Original Deli Style Pretzel Crackers | Sea Salt & Vinegar Potato Chips |
| Sea Salt Pita Chips | Chocolate Chip Cookies |

In the case of the frozen department, we found that the top 9 most ordered items on Day 0 and Day 5 are the same (with the only the order swapped for a couple items) and primarily consist of frozen produce. However, the following 5 items in the frozen aisle do differ (items 10-14), and those are as follows:

| Day 0 | Day 5 |
|---|---|
| Gluten Free Whole Grain | Chocolate Ice Cream |

| | |
|---|---|
| Bread | |
| Berry Medley | Gluten Free Whole Grain Bread |
| Organic Brown Rice | Frozen Broccoli Florets |
| Frozen Broccoli Florets | Vanilla Ice Cream |
| Organic Cheese Frozen Pizza | Organic Cheese Frozen Pizza |

As can be seen from these lists, potato chips, cookies, and ice cream are all ordered ranked higher in the top ordered products on day 5 than day 0, and none of these items actually appear at all in the day 0 top 5 differing products for these two departments of interest. Thus, we concluded that day 0 likely represents monday, and that it we will operate under this assumption throughout our final analysis.

Rule Based Classification

For rule based classification, we started with a very simple rule. Our rule was that if a product is from the alcohol department, then it is ordered on a friday or saturday. The coverage for this rule was only 0.47% and the accuracy is 34.86%. This is compared to 23.53% of total items being ordered on the weekend, if the rule had no weight. On the other hand, we found that if a order is placed on saturday (coverage of 13.25%; as opposed to coverage of items ordered on saturday, which, as is stated earlier the paper is 12.98%) that only a small portion of orders actually contained alcohol (accuracy or percent of orders on saturday with alcohol 3.37%; though as stated earlier this day still makes up 17.85% of alcohol items ordered in general.

We then tried to make a more accurate model to predict alcohol purchases. If we look at the rule if alcohol is ordered then it is ordered in the afternoon or evening (when "order_hour_of_day" >= 12) then the accuracy is 69.17% and a coverage of 0.33%. We then decided to look into specific categories of alcohol, and we found that the greatest discrepancy in orderings applies to the "beers coolers" aisle. Items ordered from this aisle have only a 0.15% coverage. However when we look at the rule if an item from the beer item is ordered then it is ordered on friday or saturday we find an accuracy of 39.27%, as opposed to 31.77% for wine and 23.53% for total items.

We next decided to use rule based classification to try to determine which factors contribute to an item being reordered. We looked at the order that items are placed in one's cart as being an indication of potential reorders, and we found that the rule, if an item is the first item placed in a cart then that item will be reordered has a 9.91% coverage and a 67.75% accuracy (this is the peak reorder rate based on when an item was added to the cart). This is as opposed to a reorder rate of 58.97 for all items. For items ordered 11th it is more likely that the item will be not be reordered (<50% chance of reorder), and the lowest rate of reordering based on order number is for items ordered 42nd or later with a reorder rate of only 40.69% (not reordered 59.31% of the time).

As discussed earlier, dairy and eggs is the department with the highest rate of reorders, so we next tested the rule, if an item is the first added to the cart and is from the dairy/eggs department then it will be reordered. We found that this rule has a coverage of 2.01% but an accuracy of 76.13%. The milk aisle, likewise is the most reordered from aisle, so we next tested the rule, if an item is the first added to the cart and is from the milk aisle then it will be reordered. And, though only having a coverage of 0.58, the accuracy for this rule is 84.07%.

KNN classifier

We created a KNN classifier model using Sci-Kit Learn to predict whether an item would be reordered based on the product ID and the order an item was placed into the shopping cart.

The KNN classifier was not very successful in predicting whether a product would be reordered. We initially ran the model on the same dataset that was used to train the model and returned the following results:

| | |
|---|---|
| KNN = 1 | 67.6% accuracy rate |
| KNN = 2 | 61.2% accuracy rate |
| KNN = 3 | 68.1% accuracy rate |
| KNN = 4 | 64.8% accuracy rate |

Given that the results with KNN=1 were nearly as good as with KNN=3, we ran the model with KNN set to 1 against the murch larger prior dataset. This classification model had an accuracy rate of 56.8% when used on the full dataset.

Since the initial results were not satisfactory we added two parameters (aisle_id and department_id) to the KNN model and re-ran the classifier. Our hypothesis was that these additional factors would lead to a more accurate model, but the results did not change much; the accuracy only improved from 56.78% to 56.92%

Apriori Itemset Prediction

We used a file called apyori.py, which had the functionality to run a slightly optimized version of the apriori we learned in class. We then would take a subset of data from the orders.csv file, and found all the products in the order_products__prior.csv file and compiled those in a list of lists. This list of lists contained what a customers cart was for the order, and served as the comparison for the frequent itemsets. We could also set the minimum support value, minimum confidence value, and minimum length of the ending frequent itemsets. Even with the optimizations on the dataset this process was still slow, and took time in the minutes to give results even for processing the first thousands of orders in the orders.csv file. This meant that we couldn't reasonably do the entire span of data, but the results for this small subset of the total data still returned a fair amount of useful results. The rules generated by the apriori algorithm can be used to check what items would be frequently bought with a given item you input, we well as the confidence value for this rule. Using these rules you could cross reference them with a given item in a customers cart, and using the confidence values of items frequently bought with it, suggest they also buy the most likely items. Had we enough storage space and runtime to process larger parts of the dataset, this could be a useful predictor/suggestion method for what a customer might want to order. However as it is there isn't too much we can do with it on the dataset other than view the frequency rules it generates on small parts.

Machine Learning

We wanted to see if we could train a model to predict if a person would reorder 50% or more of the items they had previously purchased knowing only what items they have purchased before. For features we created a sparse matrix through the use of sci-kit learn's DictVectorizer tool. By creating a list of all of the items a user has ordered and creating a Counter of this list of items, the feature matrix becomes a 1-hot encoding in which the product ID of an item they purchased corresponds to one and all other products correspond to zero. We restricted the dataset to look at 50,000 users which equated to 7,882,635 of the orders in order to be able to train models more quickly. These 50,000 users were split into a training set of 40,000 and testing set of 10,000. We found the percentage of orders that were re-orders for each user and if the value was .5 or higher we changed the value to a 1 and if it was under it became a 0. So the models were tasked with a binary classification task.

To begin we evaluated a linear support vector machine (SVM) using only 1000 training instances to look for the best hyperparameter c to use. We used values of c = [.0001, .001, .01, .1, 1, 10, 100, 1000]. C is the parameter that controls how strong a penalty is associated with the error term, so if it becomes too high the classifier can begin to overfit the training data. And for the linear SVM we saw the highest training accuracy of .7638 with a c value of .001, as with any values of c higher the training accuracy began approaching 100% to the detriment of the testing accuracy.

Then we did the same process for a SVM using an rbf kernel instead of linear. The rbf kernel didn't suffer from overfitting nearly as quickly and gave the highest testing accuracy of .8287 with a c value of 100. The testing accuracy was only negligibly smaller at c=1000 when the training accuracy jumps up to .993, so the rbf kernel is much less prone to overfitting on this data set compared to the linear. However, with the smaller c values its training accuracy was under 60% while the linear kernel's accuracy didn't vary more than 5% regardless of c.

After finding the optimal hyper parameters for both kernels we trained them on the full training set. The testing accuracy of the rbf kernel was .8853 and the linear kernel had an accuracy of .7638. So the rbf kernel was clearly superior and ended up achieving a fairly high accuracy rating considering the limited amount of info supplied to it. The lower accuracy of the linear kernel would indicate that the data is likely not completely linearly separable.

Additionally, we were curious if we had enough training data to allow a neural network to provide good results. Since often simpler models can outperform a neural network without a large amount of data. So we trained a simple multilayered perceptron with mostly default parameters with the exception of an alpha value of 1 instead of the default .001. We used this higher alpha as it accounts for how strong the L2 penalty is (controls regularization), so with a smaller data set we wanted each data point to be able to correct more. Even without the more extensive hyperparameter tuning the SVM models received the MLP model managed to get an accuracy of .842. This indicates that we had an adequate amount of data to properly train a neural network, and with more computational power to allow for more extensive hyperparameter tuning this would likely be the most successful model.

We also trained an ensemble model in the form of a random forest. First, we optimized for the maximum depth that the trees could go and tested out values of [5, 10, 15, 20, 25, 30, 35, 40] with a static number of trees in the forest of 100. Every time the depth increased the accuracy of the model increased, however the increase between 35 and 40 only yielded a .006 improvement, so we capped it out at 40 to mitigate vs overfitting. Then using the depth of 40 we tested out varying values of trees in the forest testing the values [50, 100, 120, 150, 180, 200, 225, 250]. At this point regardless of the number of trees used the testing accuracy stayed at almost the exact same level never varing more than .001 from the .804 accuracy we originally had from getting to a max depth of 40. This is interesting since even with more hyperparameter tuning the random forest was not able to outperform the MLP. It seems that decision trees aren't the best suited for determining a person that is likely to reorder based solely on the sparse matrix of the items they have already ordered.

Overall, we found that a SVM using a rbf kernel was the most effective method when using a c value of 100. However, based on the success of the MLP without any hyperparameter tuning it is very likely that it would be able to outperform the SVM given enough time to optimize all of the parameters. Additionally, if other features were allowed to be used such as how often a user spent between orders was used accuracy could increase even more. But it is useful to know that Instacart could fairly reasonably correctly predict if a customer will frequently reorder with only the knowledge of past items ordered and no other info about the time of purchase or what was bought together.

## Applications

Ideally, our research could be used to help companies determine how best to target and advertise to customers, what products to stock in the highest quantity and possibly determine where the market appears to be untapped.

One very interesting area for possible future research would be to compare these findings to similar measures for standard grocery stores (purchased from in person) to determine what, if any, differences are seen when grocery purchases are provided through a digital, delivery service. The potential application for this research are quite broad, and trends that are found could have implications or not only Instacart but also any other grocery and food delivery service, as well as any companies selling products through these services.

## References

[1] Instacart. 2017. The Instacart Online Grocery Shopping Dataset 2017. Retrieved March 10, 2019 from https://www.instacart.com/datasets/grocery-shopping-2017

[2] Stanley, Jeremy. 2017. 3 Million Instacart Orders, Open Sourced. (May 2017) Retrieved March 10, 2019 from https://tech.instacart.com/3-million-instacart-orders-open-sourced-d40d29ead6f2

[3] Annie George. 2017. Instacart Market Basket Analysis - Reorder Analysis. (Sept. 2017) Retrieved March 10, 2019 from https://nycdatascience.com/blog/student-works/capstone/instacart/

[4] Philipp Spachtholz. 2017. Exploratory Analysis - Instacart. Retrieved March 10, 2019 from https://www.kaggle.com/philippsp/exploratory-analysis-instacart

[5] Wan, Mengting, Di Wang, Jie Liu, Paul Bennett, and Julian McAuley. 2018. Representing and Recommending Shopping Baskets with Complementarity, Compatibility and Loyalty. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, (Oct. 2018), 1133-1142. DOI: https://doi.org/10.1145/3269206.3271786

[6] Bhade, Kalyani, Vedanti Gulalkari, Nidhi Harwani, and Sudhir N. Dhage. 2018. A Systematic Approach to Customer Segmentation and Buyer Targeting for Profit Maximization. In 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), (Jul. 2018) 1-6. DOI: https://doi.org/10.1109/ICCCNT.2018.8494019

[7] Asha, K. N., and R. Rajkumar. 2017. Pre-processing of user behaviour for e-commerce. In 2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon), (Aug, 2017) 715-719. DOI: https://doi.org/10.1109/ICCCNT.2018.8494019

[8] Stanley, Jeremy. 2017. data_description. (May 2017) Retrieved March 10, 2019 from https://gist.github.com/jeremystan/c3b39d947d9b88b3ccff3147dbcf6c6b

[9] Our Group's GitHub repository: https://github.com/MichaelWhitlock/CSCI4502Group1.git

[10] Daniel Herndon. 2015. How Much Should I Pay a Marketing Consultant?. (Dec. 2017) Retrieved May 4, 2019 fromhttps://milesherndon.com/blog/how-much-should-i-pay-a-marketing-consultant