Instacart Order Analysis

Jainabou Barry Danfa | SI 618 Data Manipulation and Analysis | Winter 2019 | Final Project

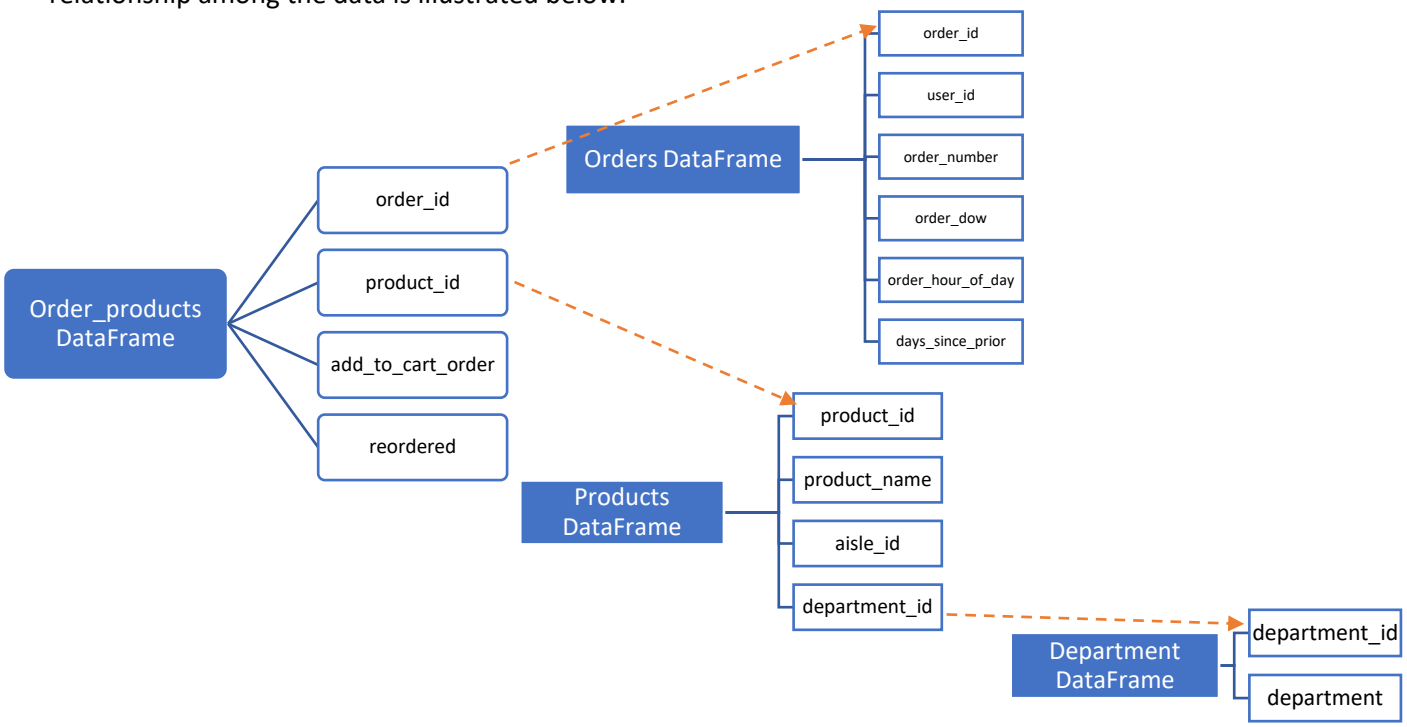
Motivation

This project looks at the Instacart Online Grocery Shopping Dataset in 2017. Instacart is an online grocery shopping company in which users select the items they want and Instacart facilitates a "shopper" for that order and then delivers it to the users address. This dataset was interesting to me due to the insights that one could gather from 3 million grocery orders. The specific questions I decided to explore in this dataset are:

1. What are the top products ordered by customers?
2. What is the typical user profile? (Number of orders, types of foods, ect)
3. What types of products are ordered around specific times?
4. Can we predict your order based on the first item?

Data Source

The dataset I used was from Instacart published dataset for orders in 2017. The link to the data is: https://www.instacart.com/datasets/grocery-shopping-2017. The format was in 4 separate csv files with order_id, product_id, and user_id as unique identifiers linking the separate tables. There was a total of 3,421,083 orders and 49,688 products in our data. The section (Data input and processing in the code workbook) details the columns and size of all tables used and how they were joined for the analysis. The relationship among the data is illustrated below:



Methods and Results

This section will explain the methods and results for each question listed in the motivation section.

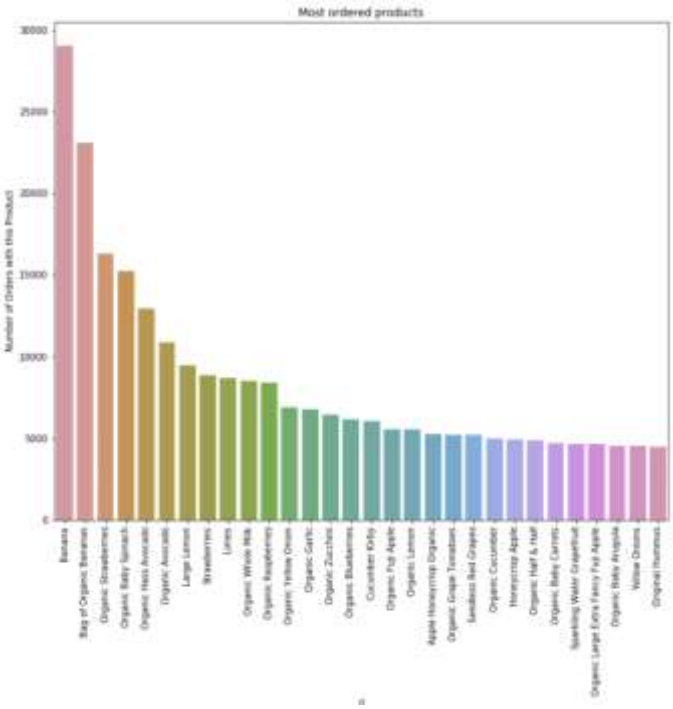
What are the top products ordered by customers?

Methods

The data was manipulated by joining on the orders, products, department, and individual order-product selection table. I then took a sample of 2 million order items to make my sample size more manageable to run. I then changed the product name column to a list and used the python method Counter to return a dictionary of the unique product_name and counts. I then selected the top 30 products and placed that into a dataframe with the name and count to plot the top 30 ordered products. Then, I wanted to get the most used descriptor in the words and joined the words and made a wordCloud of those words.

Results

When looking at the descriptors of the top 100,000 ordered products, we see trends emerge of the types of products that are being ordered such as Organic, natural, gluten-free, original. This gives us insight to the types of shoppers and foods Instacart users are buying.



In the "most ordered products" chart we can see that the most ordered products are bananas and organic bananas. Most of the top products ordered are produce. This surprised me because I thought that people would be hesitant to have a personal shopper pick out produce for you and have them choose boxed and/or canned products easier. I would be interested to see how the tops products ordered has changed since they opened for business to now. I was also curious to see what departments to the top products come from to see what kids of products to customers usually order. I converted my dataframe to a csv and made this visualization in tableau. What we see in this visualization is produce, dairy and eggs, snacks, beverages, frozen, and pantry are the departments that most products come from. This was not that surprising to me after seeing the top products ordered chart.

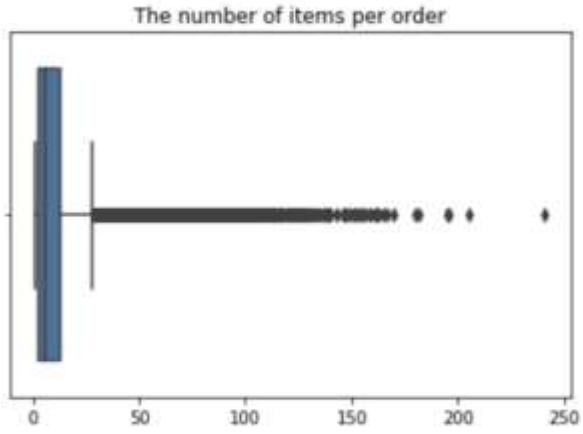
What is the typical user profile? (Number of orders, types of foods, ect.)

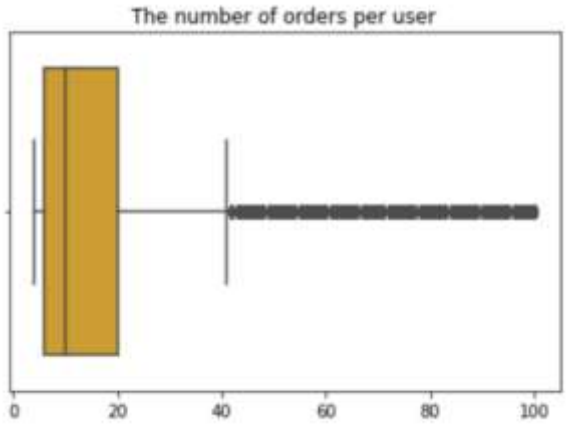
Methods

The data was manipulated by joining on the orders, products, department, and individual order-product selection table. I then took a sample of 2 million order items to make my sample size more manageable to run. The data was then grouped by user_id to obtain information per user. The sample included 19,095 users and their respective orders, ect. Using the .size() function on the pandas dictionary, I was able to gain information on the users number of orders and number of items per user.

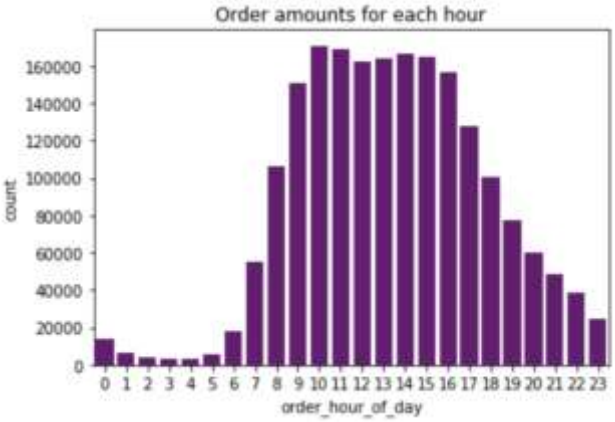
Results

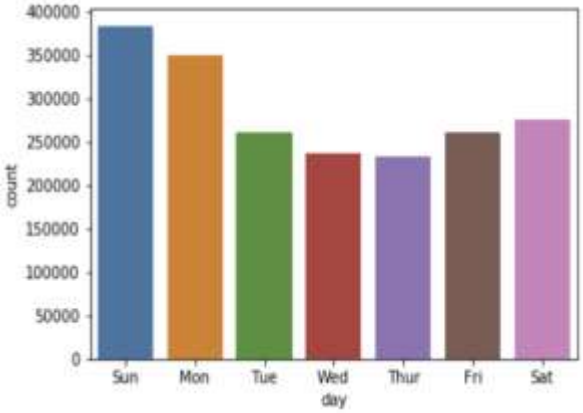
Out of the 190,095 users, the average number of items in each order is 10. While the average number of





orders each person in our dataset is 12. When looking at the times that users make orders, they are mostly concentrated in the early afternoon, peaking at noon. This distribution did not surprise me much





because Instacart has a 2-hr delivery window and to get things for dinner in time, you must order around that time. When looking at the days that people order the most, we see that people normally order on Sundays and Mondays and dips to its lowest point on Wednesday. This did not surprise me much since that's when people normally plan for their week.

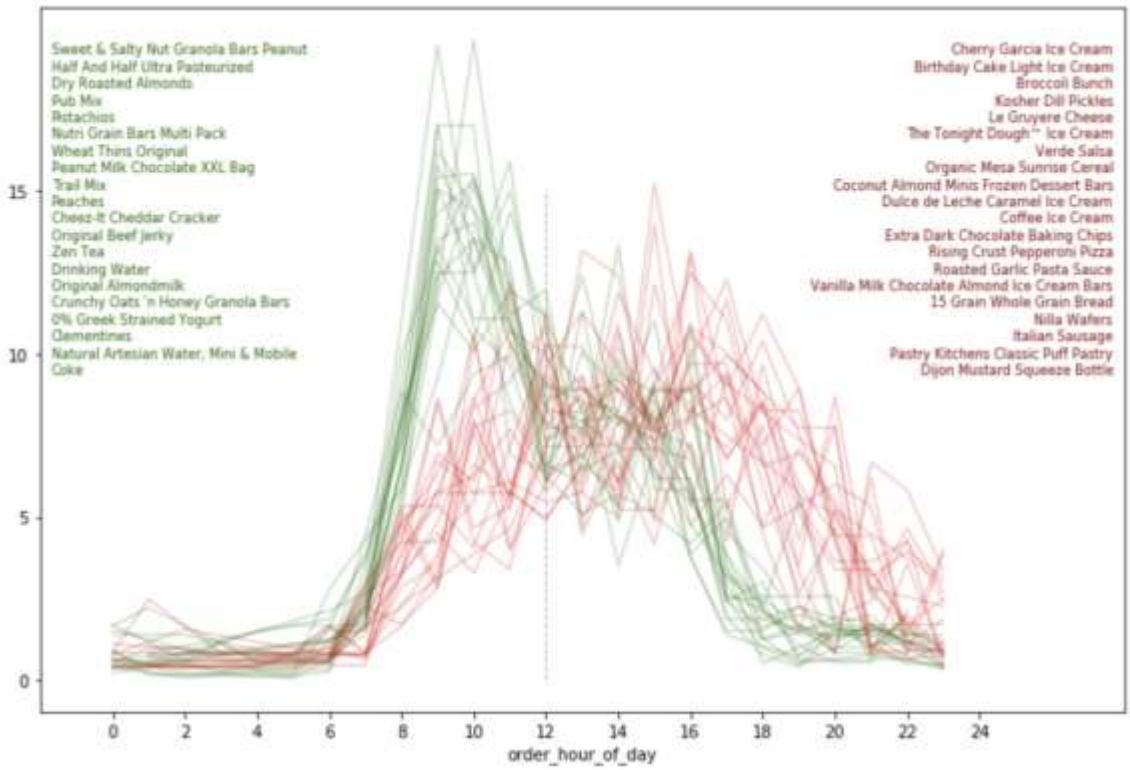
What types of products are ordered around specific times?

Methods

I took inspiration from the visualization by Jeremy Stanley https://tech.instacart.com/3-million-instacart-orders-open-sourced-d40d29ead6f2. I wanted to visualize what products were bought in the morning and at night. To do this, I first tried to groupby the order hour of the day and product name, then find the highest products for each hour. However, I kept on getting snacks since they have higher quantity in that breakdown. Then I decided to create a metric for the percent the product makes up in that hour and also calculate the mean hour the product was ordered, to plot that on the x-axis continuously instead or bucketing them. I then separated the dataframe into products who are highest in the morning and those highest in the afternoon. I then grouped by the product_id on both dataframes and graphed the both lines on one graph. Then I listed the top 20 products for each morning and afternoon groups. This analysis required a lot of google and stack overflow support=)

Results

As we can see in the graph below, healthier foods are ordered earlier in the day (granola, wheat thins, water, zen tea) compared to later in the day where we see varieties of ice cream dominating the list.



These results do not surprise me much, but I was surprised to see the overlap around noon till 3pm. I would be interested to see how these items get reordered at the same time or do they converge in the middle over time.

Can we predict your order based on the first item?

Methods

To build an order predictor, my initial implementation was building a recommendation system with affinity scores of each product compared to the other products. This implementation did not work because building a numpy array of each product x product was causing a memoryerror since it was too large. The computational power to find the scores will take too long.

Secondly, I ended up developing a recommendation system based on k-nearest Neighbors(kNN) algorithm. I decided this algorithm was sufficient for the task of recommending products because it finds clusters of similar users based on the reorder rate. This was completed by merging the tables to have one dataframe with the reorder rate, product_name, and user_id. This dataframe was then reduced to only the most common 5000 products in our dataset. This allows us to not have outlier or very seldomly ordered products to increase the size of our dataset. Then a pivot table was developed with the values as the reorder amount, index as product_name and columns as the user_id. The matrix was then converted to a sparse row matrix to reduce the size and manageability of our matrix. The kNN model was initialized using parameters highlighted in the code snippet above. Using a random selected index, we were able to obtain the closest 5 neighbors to the indexed product.

```
: from scipy.sparse import csr_matrix
  from sklearn.neighbors import NearestNeighbors

: #condense matrix into csr
  order_knn_matrix=csr_matrix(order_knn.values)

: #initialize the model
  model_knn=NearestNeighbors(metric='cosine',algorithm='brute')

: #fit the model to our data
  model_knn.fit(order_knn_matrix)

: NearestNeighbors(algorithm='brute', leaf_size=30, metric='cosine',
                   metric_params=None, n_jobs=None, n_neighbors=5, p=2, radius=1.0)
```

kNN model initialization parameters

Results

After training our model using the kNN algorithm, we found the top 5 nearest neighbors to the input product name. In the several examples below, we can see that the model does not perform the best since the top 5 products related to milk come up as hot dogs, bagels, chicken tikka masala, garlic sauce, and feta. Now, these products may be bought together and not necessarily similar so that could be the case also. Also, many of the similarity scores were very small (to the power of 0e-7. I would use a log-scale to see if the that allows the differences to be highlighted easier. Also, maybe kNN algorithm is not the best algorithm to for recommender systems.

```
Recommendations for 0% Fat Free Organic Milk :

1 : Uncured Beef Hot Dog
2 : Everything Inside Bagels
3 : Chicken Tikka Masala with Cumin Infused Basmati Rice Frozen Meal
4 : Chili Garlic Sauce
5 : Traditional Feta Cheese Chunk
Recommendations for Plain Non-Fat Greek Yogurt :

1 : Spinach Souffle
2 : Double Chocolate Chip Protein Bar
3 : Sparkling Kiwi Strawberry Soda
4 : Deviled Eggs
5 : Original Bran Cereal
Recommendations for Organic Dark Sweet Cherries :

1 : Organic Plain French Style Yogurt
2 : Organic Zucchini Squash
3 : Fire Roasted Crushed Tomatoes
4 : Everything Inside Bagels
5 : Stringles Organic Colby Jack Cheese
```

Predictive output from kNN product

Sources

https://gist.github.com/jeremystan/c3b39d947d9b88b3ccff3147dbcf6c6b

https://towardsdatascience.com/predictive-customer-analytics-part-iv-ab15843c8c63

https://towardsdatascience.com/how-did-we-build-book-recommender-systems-in-an-hour-part-2-k-nearest-neighbors-and-matrix-c04b3c2ef55c

https://medium.com/datadriveninvestor/how-to-build-a-recommendation-system-for-purchase-data-step-by-step-d6d7a78800b6

https://tech.instacart.com/3-million-instacart-orders-open-sourced-d40d29ead6f2

SI 618 In-Class Notebooks and Homeworks

Everything on https://stackoverflow.com/