Title: Instacart Market Basket Analysis

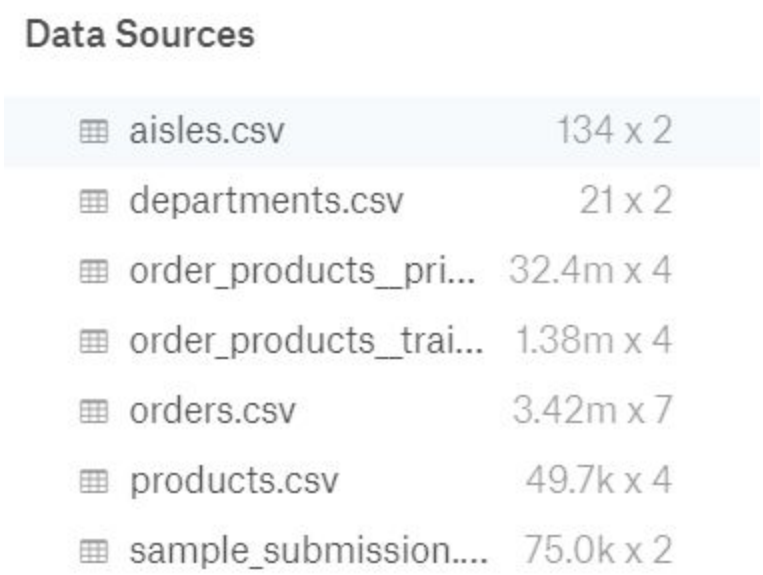
Abstract:
The purpose of this project is to determine what the customers of Instacart are going to buy in their next order. The goal is predicting what products have the highest probability of being reordered for each customer. The first phase of the project involved handling the huge data and run exploratory data analysis to understand the relations in the data and create new features. These features were created to capture more information about the customers behaviour as the given independent variables were quite few. The final phase of the project involved using apriori algorithm to find out the most frequently bought items together. Identifying the next likely cart of a customer, will help the stakeholders to target the right ads, have better product placement and improve the overall shopping experience of the customer.

Introduction:
Firstly, after setting up the data, we started off with finding the relations between and among the products and users. We looked at multiple perspectives to extract the hidden information and then ran the models to find their associations. For Example: If someone tries to buy bread, they'd likely buy butter and jam along with it. So these associations will help later on to predict the cart accurately. Converted the unsupervised question to a classification supervised learning approach by looking at if the users will be reordering the product again. Predicting the whole cart with good accuracy can be achieved by using recommendation algorithms which we haven't covered in this project.

Primary Goals:
- Using SQL and Jupyter notebook to analyze the datasets.
- Building models for the dataset to help predict the possible outcomes.
- Predicting the reorder probability of a particular product by the user.
- Minimizing the prediction errors and improving the probability of right predictions by applying appropriate feature engineering.

Data:
Summarising the data given, we have about 200K users. With maximum 100 orders data of each user. We are given the product, the hour of the day order has been placed, day of the week, days since the last order, cart

position of the item, reorders. In total we have 50K different products spread across 21 departments and 134 aisles.



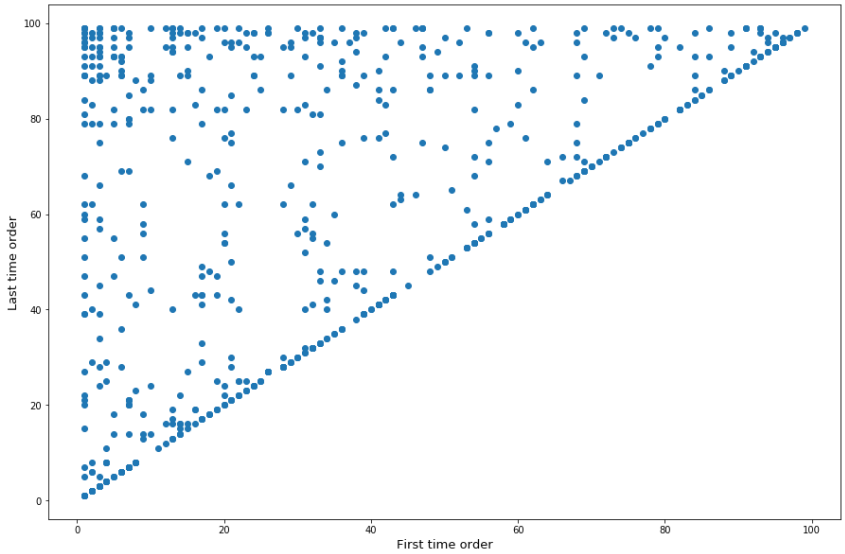
Since we have huge data in table `order_products__prior` of about 32 million, we have created 3 indexes on the table for optimal querying time. We have product_id, order_id as the primary key and have default clustered index. To analyze by slicing and dicing by user_id and product_id, we have created two separate non-unique indexes based on the querying times. For Orders table, we have order_id as the primary key and a clustered index by default. We have created an additional non-unique index on order_id for reducing querying time. The products table has 50K records and the department and aisle that it belongs to.

For some of the user-level data analysis for reduced query times, we have created subsets of tables namely - order_products__prior, Orders with names order_products__prior3 and Orders2 respectively, which have data only for users with more than 50 orders.

Analysis:
First we did the EDA and tried to see which products are most ordered, which department has the most number of orders etc. From this analysis we have observed that Bananas are the most frequently ordered and most users are preferring organic products in general. The users must be either rich or really health conscious. Since we have no features given from Instacart, most of the time went into feature engineering.

Mainly we have tried to slice and dice the data from 3 main perspectives:
Overall Item perspective**:**
How users generally order the product (like overall re-order probability) ?

User and product interaction**:**
Generally, how frequently (no of days for ordering an item) does the user reorder the product, when is the first time that user ordered the product and last time user has ordered the product. This is the most important parameter for judging whether the user will reorder the product. For example user might have been buying diapers for their children and after some time the children may be past that age of diapers and user might not buy diapers again. In cases like this, we might have high frequency, but we cannot predict that diapers will be in his next order.
For a random user, in his first 100 orders we tried to capture the data in the following graph:



Here X-axis represents the first time the user has made an order, Y-axis represents the last time user has made an order. The data points in the left top represent the products that user has been persisting since the beginning, these are the products that will be likely in his next order.

The items in upper middle and right side ones are the items that are products that user has started to order recently. The items on the straight line are the items that user has ordered only once or twice. The data points in the middle left are the items that are ordered by user only once are twice.

The other important observation that we have made is that, we cannot just go by the frequency of order of the user, each item generally has its own reorder frequency by a user, which depends on the shelf life of the product.

Day of the week and hour of the day**:**
We tried to slice the data by hour and day and tried to get the item reorder probability of the user, however, since the amount of this data very less at this level, we realised that it is better to group the data by the range of hours like morning, afternoon, evening and night time. But due to time constraint, we couldn't do this.

Loyal users vs fickle users**:**
Another, interesting behaviour we observed is that, some users are loyal to the products and brand, some users they stick with the product but buy different brands. Following is the snippets of such users

Loyal user**:**

user_id	product_name	product_id	reorderedCnt
50	Natural Artesian Bottled Water	6182	52
50	Reduced Fat Milk	5612	33
50	Organic Leek	23165	22
50	Organic Tomato Cluster	41950	18
50	Organic Russet Potato	19678	17
50	Bag of Organic Bananas	13176	16
50	Brown Eggs	6341	16
50	Organic Hass Avocado	47209	16
50	Organic Zucchini	45007	15
50	100% Premium Select Not From Concentrate Pure Prune Juice	47018	12
50	Natural Mint Gum	20367	11
50	Organic D'Anjou Pears	22825	10
50	Organic Granny Smith Apple	39877	10

Fickle user:

user_id	product_name	product_id	reorderedCnt
27	Pure Coconut Water	2966	29
27	Natural Artisan Water	1194	22
27	Natural Artesian Water	14233	19
27	Total 2% Lowfat Greek Strained Yogurt with Peach	33787	17
27	Total 2% Lowfat Greek Strained Yogurt With Blueberry	4957	16
27	Total 2% with Strawberry Lowfat Greek Strained Yogurt	33754	15
27	Total 0% Nonfat Greek Yogurt	46676	14
27	Hibiscus Organic Raw Kombucha	9604	12
27	Synergy Organic Kombucha Gingerberry	48559	11
27	Organic & Raw Strawberry Serenity Kombucha	6287	10
27	Half & Half	27086	10
27	Coconut Chia Bar	23291	10
27	Cinnamon Raisin English Muffins	24721	9

As we can see, user 50 and 27 both user same product Natural water, but user 27 has tried different brands (Although the product name is same, observe that they have different product ids, they have not given us the full name), from this we can see that we need to run the algorithms at users level.

Model:

We tried to build an apriori model to find out the most frequently bought items together. But in this case it is difficult to find the rules with high support and confidence as we have high number of orders, so we set the thresholds very low and found out the association rules.
Few of the association rules that we found out are:
{"Cage Free Brown Eggs-Large, Grade A"} -> {Banana} (conf: 0.261, supp: 0.001, lift: 1.755, conv: 1.152)
{Organic Avocado} -> {"Clementines, Bag"} (conf: 0.022, supp: 0.001, lift: 1.743, conv: 1.009)
{Organic Baby Spinach} -> {"Clementines, Bag"} (conf: 0.016, supp: 0.001, lift: 1.318, conv: 1.004)

Although, the original problem statement is unsupervised learning, we rephrased the statement "to predict if the user will reorder the product given product id, day of the week, hour of the day".We used the features from analysis: item overall reorder probability, user reorder probability(this no of orders of the product by total orders), no of days for reorder, last order of the product, day of the week, hour of the day and build XGBoost model for user level and tested it for 2 users, we got around 87% accuracy.

Conclusion:

Since the problem is unsupervised learning, and no explicit features given to us, the more time we spend, the more features and observations we can find out.

We felt that instead of using the MySQL, it's better that we use SQL server Analysis service (SSAS OLAP) type of tools for better analysis of data and insights. SSAS kind of tools provide Multi dimensional data expressions (MDX) pre aggregate functions and formulas. For example in this problem when we figured out that each item has its own average number of days for reorder by user, we wanted to extract out this average for user in SQL which is difficult as the data provided has only the number of days since last order by the user, but not the number days since reorder of the particular item. We had to write this particular logic in Python. But for such cases in SSAS we can define an custom aggregators and utilise it very effectively.

Future directions:

- Need to solve the original problem of predicting the entire cart of user instead of predicting if the user will reorder the given product
- Recommendation algorithms can prove really helpful in predicting the whole cart
- The more time we spend on the data, the more relations we discover and the better features we can extract