Instacart Project Final Report

Client:

Instacart is an online grocery ordering application that helps customers fill their refrigerator and pantry with their personal needs. After selecting the grocery they like to purchase, a personal shopper would do the in-store shopping and deliver them to their homes.

Problem:

Given over 30 million rows of transactional data and over 3 million rows of customers' orders data. Instacart is looking to predict which products a customer is going to repurchase given their purchase history. By doing so, Instacart would be able to improve its recommendation system to its customers and improve customer retention by reminding them if they have not purchased their groceries for the week given their regular purchase pattern.

Dataset:

The data is obtained from kaggle, https://www.kaggle.com/c/instacart-market-basket-analysis/data. The data is a relational set of files that describe Instacart customers' order over time. There are over 200,000 customers, over 3 million rows of orders data, and over 30 million rows of orders and products data.

The independent variables are those listed below and the variable that we are looking to predict is per user per product if the specific product is going to be reordered the next time. 1 for reorder and 0 for not. This can be considered as a binary classification.

There are 6 csv files which include:

1. Prior Order Data:
    a. Order Id is like a receipt number for customers
    b. Product id: products the customer purchased
    c. Add to cart order: At what order is the product added to the bag within the same order
    d. Reordered: If a customer ordered the product previously and currently reorders, then he reorders (1) else, he does not (0).

	order_id	product_id	add_to_cart_order	reordered
0	2	33120	1	1
1	2	28985	2	1
2	2	9327	3	0
3	2	45918	4	1
4	2	30035	5	0

2. Orders Data:

a. Order id: A receipt number for customers
b. User id: An id specific to each user
c. Eval set: is the data for prior, training, or testing. Prior data would be used as historical data while training data would be used as current data. These two data would be useful for predicting the model later and test if the model is a good predictor of whether one is going to reorder or not. Testing would be use for the Kaggle competition.
d. Order number: Out of the total orders, what number is the specific order for the user?
e. Order dow: What day does the user order the product? The days are given as numbers: 0-6, where 0 is for Saturday, 1 is Sunday, etc and 6 is Friday.
f. Order hour of day: What time of the day does the user order?
g. Days since prior order: How many days it has been since the user last order.

	order_id	user_id	eval_set	order_number	order_dow	order_hour_of_day	days_since_prior_order
0	2539329	1	prior	1	2	8	NaN
1	2398795	1	prior	2	3	7	15.0
2	473747	1	prior	3	3	12	21.0
3	2254736	1	prior	4	4	7	29.0
4	431534	1	prior	5	4	15	28.0

3. Products:

	product_id	product_name	aisle_id	department_id
0	1	Chocolate Sandwich Cookies	61	19
1	2	All-Seasons Salt	104	13
2	3	Robust Golden Unsweetened Oolong Tea	94	7
3	4	Smart Ones Classic Favorites Mini Rigatoni Wit...	38	1
4	5	Green Chile Anytime Sauce	5	13

4. Departments

	department_id	department
0	1	frozen
1	2	other
2	3	bakery
3	4	produce
4	5	alcohol

5. Aisle

	aisle_id	aisle
0	1	prepared soups salads
1	2	specialty cheeses
2	3	energy granola bars
3	4	instant foods
4	5	marinades meat preparation

6. Order Products Train: This dataset is similar to prior order data except that this is the current version of the data used for training the model.

	order_id	product_id	add_to_cart_order	reordered
0	1	49302	1	1
1	1	11109	2	1
2	1	10246	3	0
3	1	49683	4	0
4	1	43633	5	1

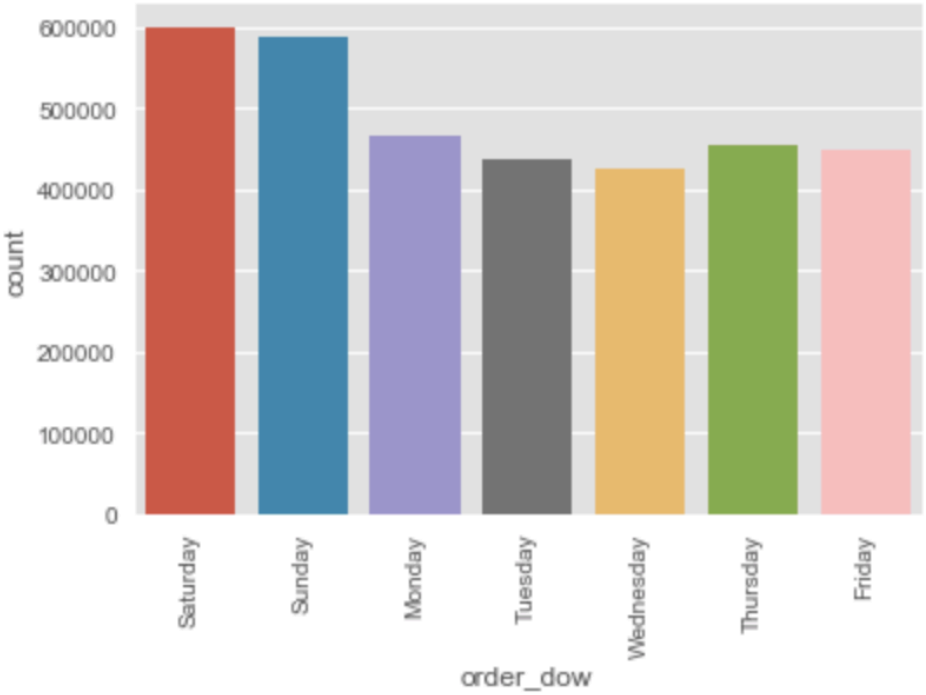
Data Wrangling:

Since there are not a lot of missing values in the data due to it being a Kaggle competition, there is not a lot of data cleaning to do.
However, there are some missing values in the days since prior order column under the orders csv file. These missing values actually mean that the customers have never ordered through Instacart previously and mean that they are first time customers.
To show this, I decided to changed these missing values to -1.

Exploratory Data Analysis:

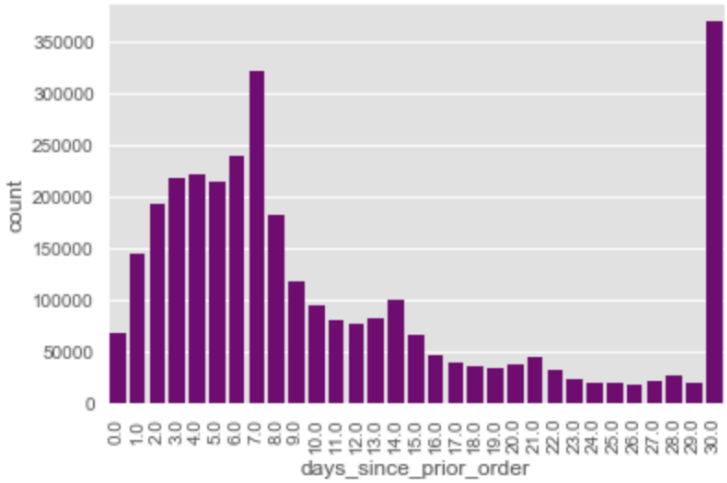
We can see that there are a lot of prior data (historical data) that we can use to analyze and train the model before using training data and testing data. It comprises of 93.9% of the whole dataset. The train data comprises 3.9% and the test data comprises of only 2.2% of the whole dataset and this would be used to get a score from the Kaggle Competition.

Which day of the week do customers usually order their groceries?



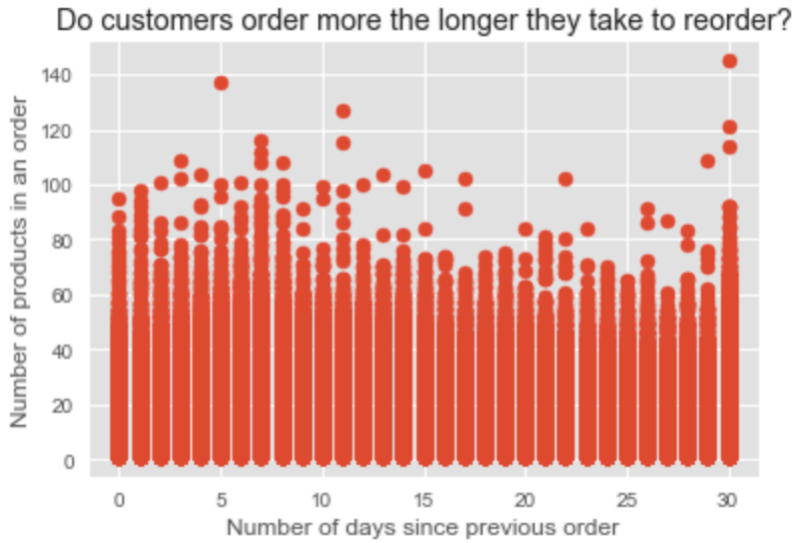
From the graph above, we can see that most orders come in on Saturday and Sunday. This makes sense because households would usually cook during the weekend and restock their fridge and pantry section over the weekend so they have something to eat during weekdays.

How long does it take one to reorder its groceries?



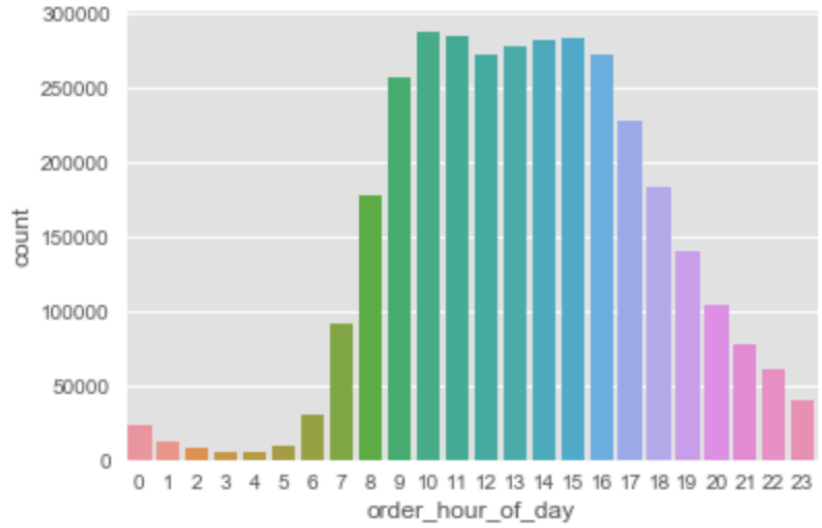
The graph above shows that from day 1 to day 7, there is increasing number of orders and afterwards, it decreases. This indicates that a lot of people re-order their grocery products every week. What is interesting is that after a month, a lot of people also re-order their grocery products. We would think that after 7 days is peak, but turns out that there are even more people re-ordering their products after 30 days.

Do customers order more products at a time as they take longer to reorder?



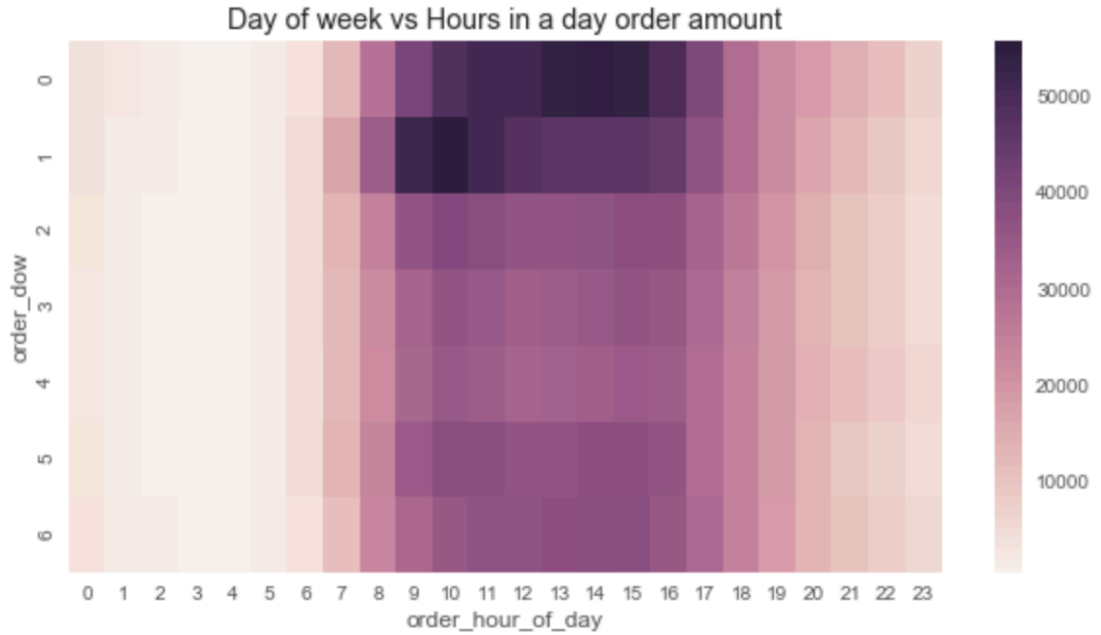
The scatterplot shows a graph of number of products each customer orders based on the period length he or she previously orders. Turns out that there is no correlation between them.

At which time of the day do customers usually order their groceries?



Given the 24 hour window that customers can order their groceries, most customers place their orders during the hours of 9am-4pm. This means that most customers place their orders during working hours on weekdays and during breakfast and lunch hours for weekends, to perhaps prepare food for lunch or dinner.

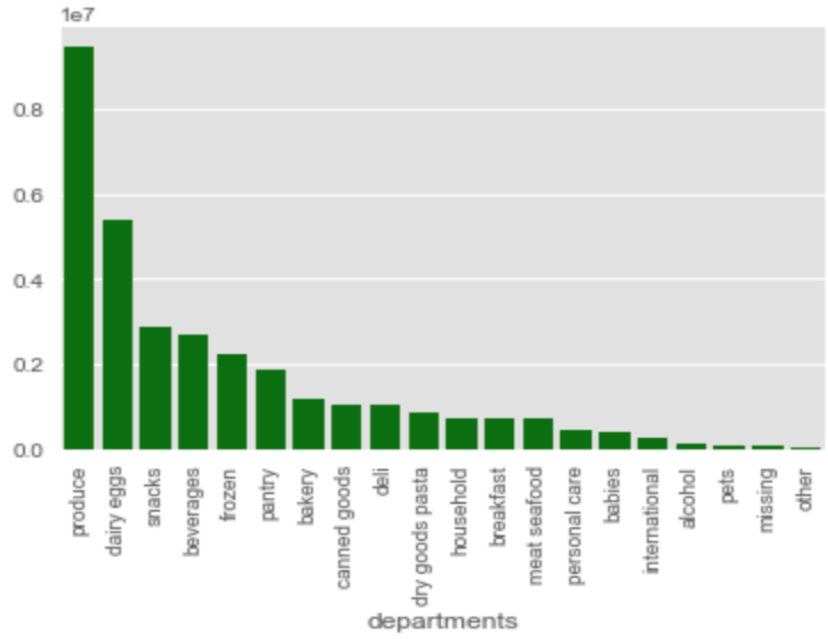
Day of week and time of day at which there are a lot of customers ordering through Instacart:



According to the heat map, as the color goes from light to dark, it indicates increasing number of customer orders. Turns out that most customers at 9am-12pm on Sundays and 12pm-5pm

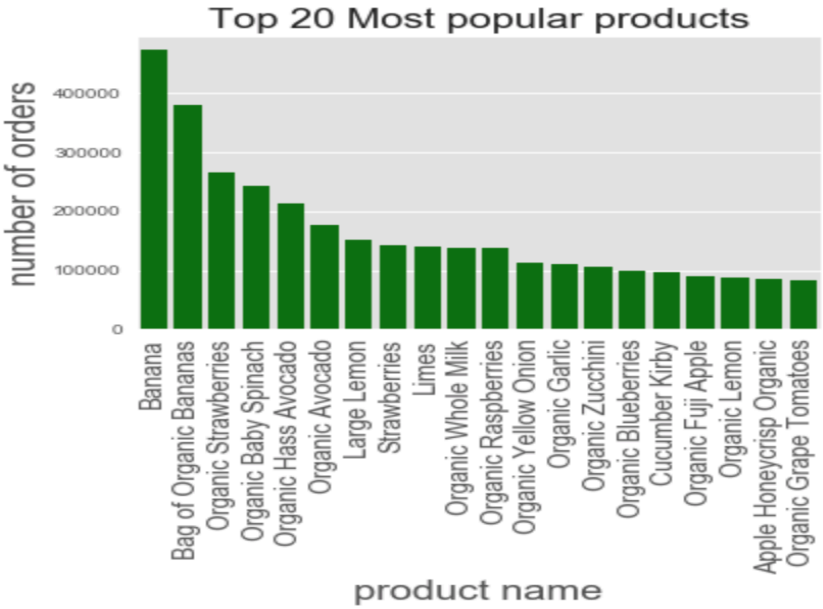
on Saturdays. However, in general, there are more orders occurring from 9am-5pm throughout all days in comparison to other times of the day.

Top 20 products' departments:



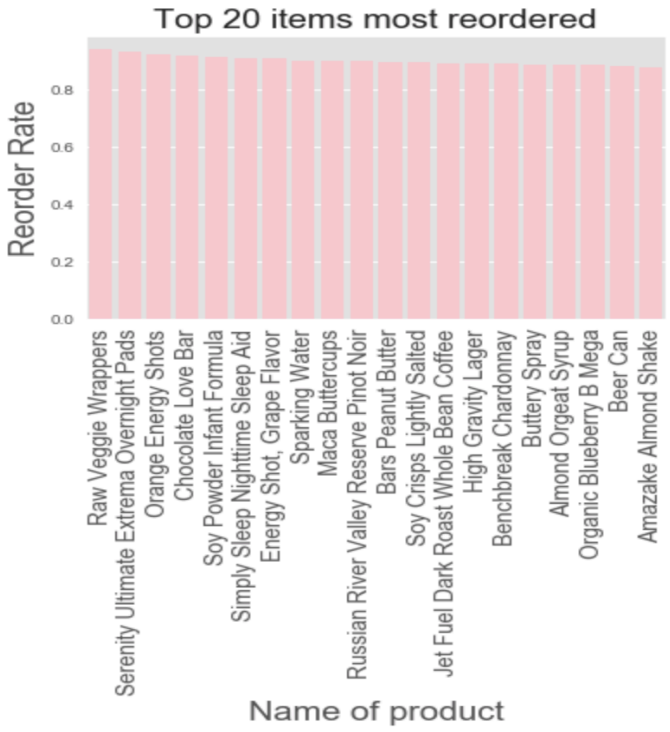
Produce, dairy eggs and beverages make it into the top 20 list and this is no surprise because customers would definitely order them a lot as they are part of humans' daily essentials.

Top 20 Most popular products ordered by customers:



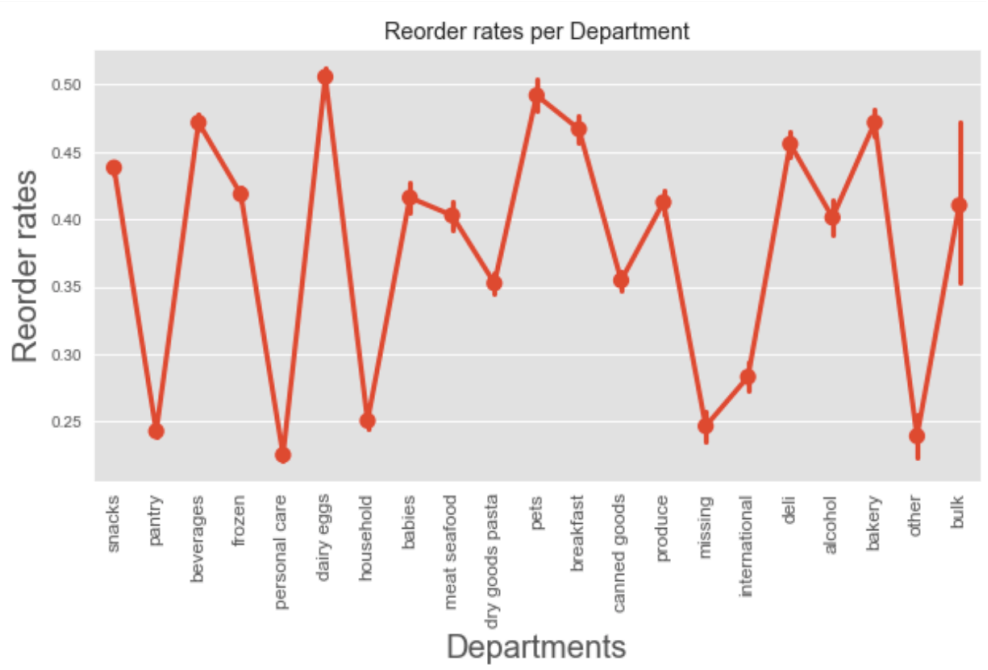
The top 20 products ordered by customers comprise of only fruits, vegetables, and dairy products. Banana and strawberries topped the list, followed by Spinach and Avocado which are vegetable products. However, which products have high reorder rate?

Which products have high reorder rate?



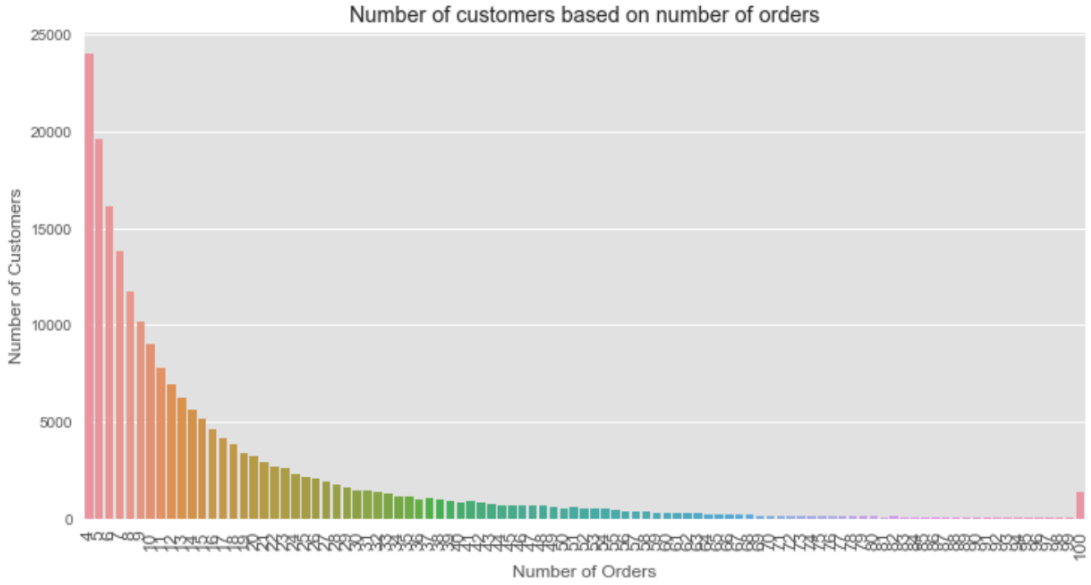
The reorder rate is defined as the mean of reorders per product. So if a product is reordered 5 times out of 10, it would have a reorder rate of 0.5. We can see that fruits, vegetables, and dairy products do not make it to the list but rather pads, chocolate, sleep aids, water, and peanut butter made it to the list. This is pretty surprising because this shows that perhaps people who purchase produce products do not order it on a streak basis.

Which Departments are often reordered?



We can see that departments that have high reorder rates in comparison to its peers include dairy eggs, beverages, pets, breakfast, and produce. This indicates that departments that have a lot of orders tend to get a lot of reorders as well from customers. This makes sense because these products are part of human essentials.

How many orders have customers placed with Instacart?



The bar graph above shows how the number of customers changes given the number of orders they have placed with Instacart in the past. The graph shows a decreasing trend where most

customers have only placed 4 orders with Instacart. There are quite a number who have placed 100 orders with Instacart in the past though.

Do customers tend to reorder products they place first in the cart?

Scatterplot above shows that products that are added to cart earlier tend to have a pretty high range of reorder rate ranging from 0 to 1, meaning that products added earlier to cart do not necessarily mean that they have high reorder rate. However, we know for sure that products added to cart last, especially if one has over 30 in his or her carts, tend to have very low reorder rate, close to 0.

Feature Extraction and Feature Selection:

There are several features that I include here:

1. User Features: This evaluates what features are important to know about users' behaviors.

user_id	average_days_between_orders	Total_orders	total_items	All_products	total_distinct_items	average_basket
1	19.000000	11	59	{17122, 196, 26405, 14084, 46149, 26088, 13032...	18	5.363636
2	16.285714	15	195	{45066, 2573, 18961, 1559, 32792, 23, 22559, 1...	102	13.000000

- Average Days between orders: On average, how many days does it take a customer to reorder his or her groceries?

- Total orders: How many orders has the customer place so far?
- Total items: Number of Items the customer has placed in the pat.
- All products: Product ids that have been purchased by the customers in the past
- Total Distinct Items: How many distinct items has the customer purchased in the past?
- Average Basket: On average, how many items does the customer put into his shopping basket per order?

2. Product Features: Evaluates features on why the product gets reordered.

	product_id	total_order	Number_of_times_reordered	reorder_rate	average_position_in_cart	product_name	aisle_id	department_id	department
0	1	1852	1136	0.613391	5.801836	Chocolate Sandwich Cookies	61	19	snacks
1	2	90	12	0.133333	9.888889	All-Seasons Salt	104	13	pantry
2	3	277	203	0.732852	6.415162	Robust Golden Unsweetened Oolong Tea	94	7	beverages
3	4	329	147	0.446809	9.507599	Smart Ones Classic Favorites Mini Rigatoni Wit...	38	1	frozen
4	5	15	9	0.600000	6.466667	Green Chile Anytime Sauce	5	13	pantry

- Product Id
- Total order: Total number of orders placed on the product
- Number of times reordered: How many times has the product been reordered?
- Reorder Rate: Number of times reordered/ Total order
- Average Position in Cart: On average, when is the product added to the cart? At the beginning or at last?
- Product name
- Aisle id
- Department id
- Department: Under which section is the product in?

3. User-Product Features: Combines each user behavior along with products

user_product_id	Number_of_orders	Last_order_id	average_position_in_cart
100196	10	2550362	1.400000
110258	9	2550362	3.333333

- User product id: An id for each user for a specific product
- Number of orders: How many times has the user order that specific product?
- Last order id: The most recent order id of the specific product.
- Average position in cart: For that specific user, does the user usually add the product to cart early or before checking out?

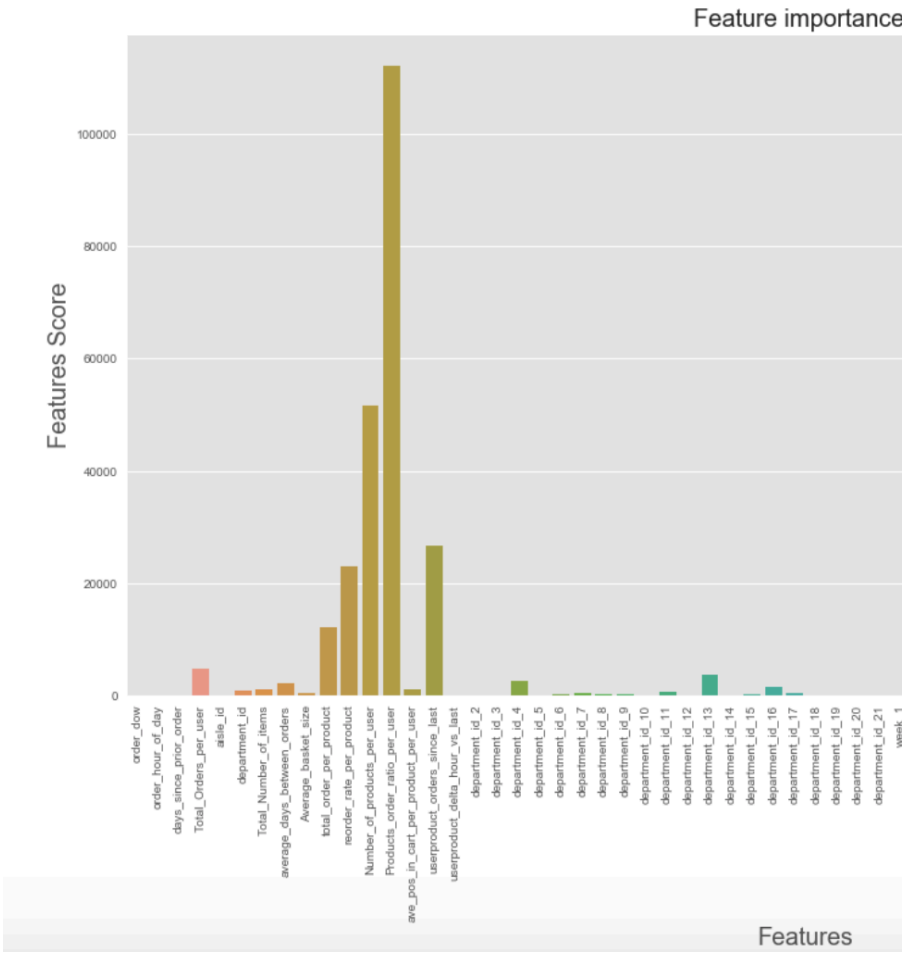
Combining all the features above and modifying some of the features, the independent variables that I included for prediction later on include:

- Categorical Features:

    - Order Day of week: a column for each day that indicates 1 if the person orders on any day of the week.
    - Order hour of day: a column for each hour that indicates 1 if the person orders on any hour of the day
    - Aisle Id: a column for each aisle that indicates 1 if the person orders on any aisle
    - Department Id
- Numerical Features:
    - Days since prior order: How many days does it take the user to reorder based on previous order?
    - Total orders per user: Number of orders a specific user has placed
    - Total number of items the user orders: In total, how many items has the user order?
    - Average days between orders: Average number of days one takes to reorder
    - Total orders per product: How many orders for the specific product has the user placed historically?
    - Reorder rate per product: What is the probability of the product getting reordered on a general level based on all customers?
    - Number of products per user: How many distinct products have the user placed in the past?
    - Products order ratio per user: what is the order ratio of the product based on a specific user?
    - Average position in cart per product per user: Does the user usually place the specific product first or last in the cart?
    - User product orders since last: how many days it has been since the specific user order a specific product?
    - User product delta hour vs last: Assuming that one orders a product at say 9am previously, and this time at 11am, the user product delta hour vs last would be 2.

Using SelectKBest, and a filter of only taking features with scores of above 1000, I narrowed down the features to only 11. The 11 features are shown below. Departments 4,13, and 16 refer to Produce, Pantry, and Dairy Eggs specifically.

	Index	score	feature
0	3	4912.037353	Total_Orders_per_user
1	5	1010.069853	department_id
2	6	1158.752484	Total_Number_of_items
3	7	2154.156615	average_days_between_orders
4	9	12261.521310	total_order_per_product
5	10	22999.493189	reorder_rate_per_product
6	11	51498.274751	Number_of_products_per_user
7	12	112109.882968	Products_order_ratio_per_user
8	13	1055.116142	ave_pos_in_cart_per_product_per_user
9	14	26641.345428	userproduct_orders_since_last
10	18	2652.095895	department_id_4
11	27	3644.610300	department_id_13
12	30	1563.346423	department_id_16



Statistical Inference:

There are two statistical testing that I did.

1. Ho: There is relationship between day of the week and the hour at which one orders his or her grocery
   Ha: There is significant relationship between day of the week and the hour at which one orders his or her grocery

   Using chi-square test, I got a value of 13722 and a p value of 0. At 0.05 significance level, this indicates we should reject the null hypothesis in favor of the alternative. This means that there is significant relationship between day of week and hour at

which one orders his or her grocery.

```
chi2, p, dof, expected = stats.chi2_contingency(week_vs_hours)
```

```
chi2
```

```
13722.50153097765
```

2. Ho: There is no correlation between the number of days it takes one to reorder and the number of item one buys
   Ha: There is significant correlation between the number of days it takes one to reorder and the number of item one buys

   Using Pearson-correlation test, we could see that the r-value is only 0.05. This means that the two variables are weakly correlated. Thus we could say that there is almost no correlation between the number of days one reorders and the number of items one buys.

```
stats.pearsonr(x = days_vs_orders['days_since_prior_order'],y =days_vs_orders['No_Of_orders'])
```

```
(0.05938871618766222, 0.0)
```

Machine Learning Techniques:

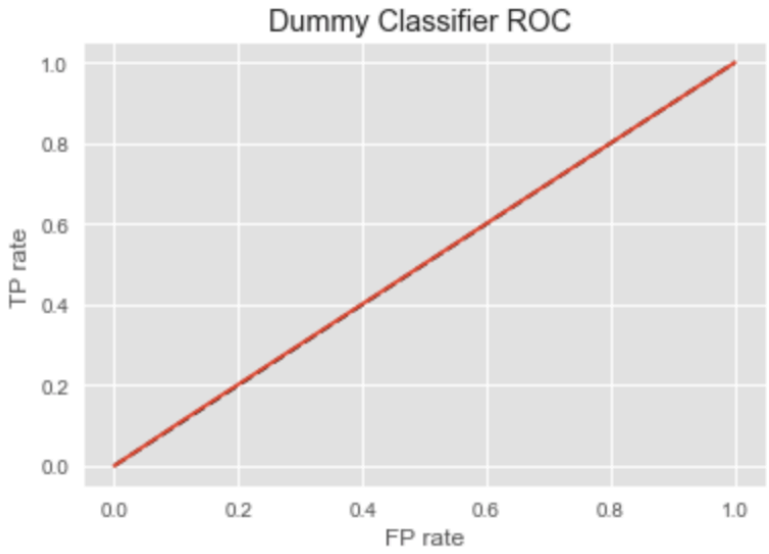
I used several machine learning techniques to evaluate which model would be most accurate in predicting which products each user is going to reorder. The score metrics used to measure the performance of a model here is F-1 score.

Dummy Classifier: A classifier that makes prediction using simple rules. Here I used stratified as its strategy which means that given 100 rows, where 90 is 0 and 10 is 1, the output the model would give would be 90% 0 and 10% 1. Accuracy would be less than 90% on most cases. This is useful as a baseline for comparison for other models to see how they are performing.

The Dummy classifier shows an accuracy score of 78%, and an F1 score of 0.12. This means that the baseline for other models is to have an F-1 score of 0.12.

The ROC curve also shows that the AUC lies on a 45-degree angle, which simply means that it is no better than random guessing.

```
accuracy score: 0.787872387858
Confusion Matrix:
[[131764  17804]
 [ 18354   2532]]
F1 Score:
0.122847023434
```

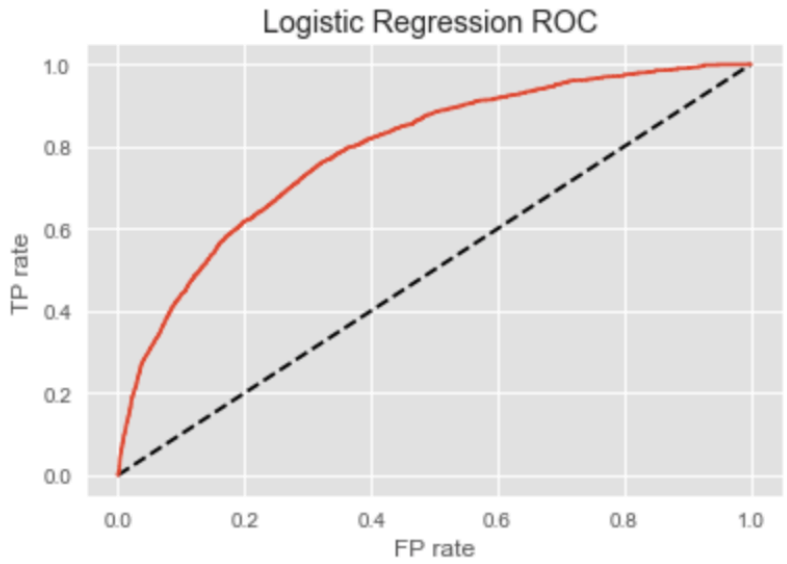



```
0.5010966871384569
```

Logistic Regression: This algorithm makes sense for this problem because it gives an outcome of 0 or 1 and takes in a lot of variables as input.

The threshold below indicates the level at which an observation should be considered as a 1 or a 0. T For instance, if an observation has a probability of 0.7, it would be a 1 since it is greater than the threshold T(any values below 0.7).
Turns out that a threshold of 0.2 gives the best f1 score, which is 0.41. The logistic model has shown that it passes the baseline model and is a pretty good model for this problem.

```
treshold: 0.1 | auc: 0.70657659241 | accuracy: 0.614136365236 | f1_score: 0.344913447939
treshold: 0.12 | auc: 0.717843380509 | accuracy: 0.673090687224 | f1_score: 0.368112490786
treshold: 0.14 | auc: 0.714677053789 | accuracy: 0.72117404109 | f1_score: 0.382934524351
treshold: 0.16 | auc: 0.708122497264 | accuracy: 0.76522698206 | f1_score: 0.397663987477
treshold: 0.18 | auc: 0.702613654684 | accuracy: 0.803307637251 | f1_score: 0.414937614519
treshold: 0.2 | auc: 0.682493415951 | accuracy: 0.828035716381 | f1_score: 0.41102716605
treshold: 0.22 | auc: 0.667280676709 | accuracy: 0.846122707593 | f1_score: 0.406677675481
```




```
roc auc score for cross validated: 0.608597699611
```

Decision Tree: The decision tree is also another appropriate model for this case because we could divide the data into several parts and figure out if an observation falls as a 1 or a 0.

The model produces the best f1 score at a threshold of 0.22 in this case. The model seems to perform slightly better than Logistic Regression in predicting the data.

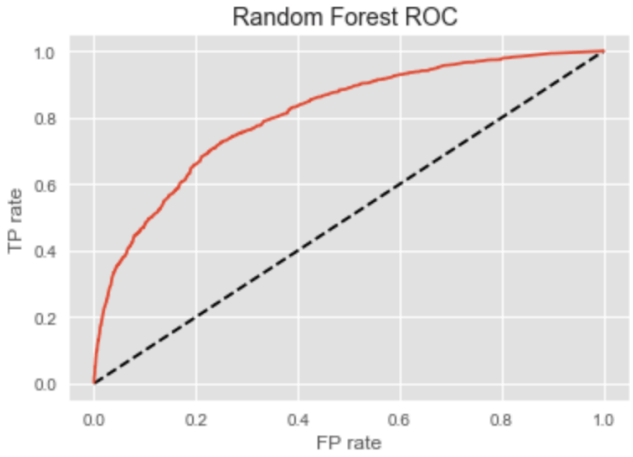
```
treshold: 0.1 | auc: 0.714013487063 | accuracy: 0.644537529187 | f1_score: 0.357203479737
treshold: 0.12 | auc: 0.710074008644 | accuracy: 0.739771433935 | f1_score: 0.387122625216
treshold: 0.14 | auc: 0.71691875647 | accuracy: 0.813122601992 | f1_score: 0.435992775948
treshold: 0.16 | auc: 0.71691875647 | accuracy: 0.813122601992 | f1_score: 0.435992775948
treshold: 0.18 | auc: 0.705054409039 | accuracy: 0.831338660284 | f1_score: 0.438638626911
treshold: 0.2 | auc: 0.705054409039 | accuracy: 0.831338660284 | f1_score: 0.438638626911
treshold: 0.22 | auc: 0.687350595282 | accuracy: 0.851596325108 | f1_score: 0.436865538736
treshold: 0.24 | auc: 0.687350595282 | accuracy: 0.851596325108 | f1_score: 0.436865538736
treshold: 0.26 | auc: 0.679133189807 | accuracy: 0.857163809591 | f1_score: 0.432007465298
treshold: 0.28 | auc: 0.654041044125 | accuracy: 0.868069977824 | f1_score: 0.407680556287
treshold: 0.3 | auc: 0.6407295422 | accuracy: 0.876770272332 | f1_score: 0.394825549569
```


```
roc auc score for cross validated: 0.705054409039
```

Random Forest Classifier: Random Forest classifier is similar to Decision Tree except that it is a combination of many trees.

A threshold of 0.26 seems to give the best f1 score, which is 0.44. This model is doing only slightly better than the Decision Tree algorithm and takes a lot more time to produce the result.

```
treshold: 0.1 | auc: 0.728182297412 | accuracy: 0.69767872946 | f1_score: 0.381003685158
treshold: 0.14 | auc: 0.735269694578 | accuracy: 0.772638666316 | f1_score: 0.422169202012
treshold: 0.16 | auc: 0.718071916408 | accuracy: 0.801615680834 | f1_score: 0.425933936636
treshold: 0.18 | auc: 0.709681313524 | accuracy: 0.817593101964 | f1_score: 0.42958918454
treshold: 0.2 | auc: 0.695805905208 | accuracy: 0.835980291325 | f1_score: 0.429943056652
treshold: 0.22 | auc: 0.689158680739 | accuracy: 0.848734722189 | f1_score: 0.433805590994
treshold: 0.24 | auc: 0.684703627702 | accuracy: 0.855162763998 | f1_score: 0.434620232482
treshold: 0.26 | auc: 0.68253159594 | accuracy: 0.860813461216 | f1_score: 0.437631369138
treshold: 0.28 | auc: 0.674125460219 | accuracy: 0.865573201942 | f1_score: 0.431594649946
```




```
roc auc score for cross validated: 0.68253159594
```

Conclusion:

The top features:

1. Out of all orders, how many times does the user include the product
2. How many products the user have bought in the past to indicate if he oftens buy products from instacart
3. How long has it been since the product is last purchased by the user
4. How often does the user reorder the product after buying it right previously?
5. Out of total orders, what percentage of it does the product comprise of?
6. How many orders have the users placed?
7. Department id 13: Pantry Department. Turns out that pantry department are often reordered
8. Average basket size per user
9. Department id 4: Produce Department. Produce department products are often reordered as well, which is not a surprise.

10. Average number of days the user usually takes to reorder.
11. Department id 16: Dairy Eggs Department. Dairy Eggs also tend to get reordered a lot, which is not a surprise.

Model Discussion

Using F-1 score to determine which model is most suitable, I found out that:

- Dummy Classifier: The Dummy Classifier indicates an F-1 score of 0.12, which makes sense that it does not do well as it acts as a baseline for other model to compare against.
- Logistic Regression: The Logistic Regression indicates an F-1 score of 0.4 and an AUC of 0.65, meaning that the Logistic Regression is a better model in comparison to the dummy classifier model.
- Decision Tree: The Decision Tree indicates an F-1 score of 0.43 and an AUC of 0.68. This means that this model is an even better performer in predicting the test samples in comparison to the Logistic Regression.
- Random Forest: The Random Forest indicates an F-1 score of 0.44 and an AUC score of 0.68. This is not much different from that of the Decision Tree in predicting the test samples.

Overall, the best model is the Decision Tree since it produces roughly the same result as Random Forest and takes a lot less time to process than the Random Forest.

Recommendations:

To increase customer retention, I would focus on customers that order a lot of produce, pantry, and dairy products and remind them (email reminders) based on their purchase pattern (once a week, once every 3 days, etc) so that they would keep on purchasing their products from Instacart.

For potential product cross-selling or up-selling it would be useful as well to analyze similar customers' behavior and their basket composition. Perhaps, there are items that they normally buy together. The app should show a list of recommended products when the customer is shopping to improve product sales. The list of recommended products would be based on other similar customers as well as the specific customers' historical orders, such as what items does the user usually buy together.