Instacart: Future Basket Recommendation

I. Introduction

A. Background

Instacart is an American online grocery company that services all 50 states and some areas in Canada. They started in 2012, with headquarters in San Francisco and rapidly expanded throughout the country in just 8 years partnering with the biggest markets in the country. They most recently added Walmart to their partner roster this August 2020.

Online orders are fulfilled the same day using a personal shopper to collect, bag, and deliver groceries. Customers can shop at a variety of local retailers in their area with prices comparable to instore. Since they partner with over 250 different retailers, with top brands like Safeway, Albertsons, Target, Sprouts, Petco, and CVS, Instacart has been a great delivery option for busy shoppers all over the country. Instacart users mainly place orders on the mobile app but can also do so on the web.

B. Problem Statement

How can Instacart improve the customer experience by streamlining product selection through predicting future products based on prior purchasing behavior? Taking data from over 200,000 customers, how can we create a model that can successfully predict whether or not past items will be reordered in the future?

C. Goal

Using the datasets covering the purchasing patterns of over 200,000 Instacart customers, our goal is to predict which items will make it into future orders. This is a classification problem since for each customer and product id we need to determine whether or not the product will be reordered, ‘1,’ or not, ‘0.’

In order to do so, we will need to come up with predictor variables ‘X’ that will be able to describe the characteristic behavior and relationship between each customer and product id. Our ‘Y’ variable will be the column describing whether or not the product was reordered, it is labeled ‘reordered.’

II. Data Cleaning and Wrangling

A. Data

The original data came in 6 sets and was originally a Kaggle competition. It was split up into train, test, and prior sets where the test set was not released publicly. For purposes of this project, we will treat the given data as one whole set and split it later into train/test sets to create predictive models.

1. Aisles: This set includes all the aisles, defined by aisle ID and aisle name
2. Departments: This set includes all the departments, defined by department ID and department name

3. Products Prior: Includes all the products in each order id prior to their final order, defined by order ID, product ID, add to cart order number, and reordered column (target variable)
4. Products Train: Includes all the products in the final order for each customer within the dataset, defined by order ID, product ID, add to cart order number, and reordered column (target variable).
5. Orders: This table includes all orders in the train, test, and prior sets that were split by Instacart and defined by order ID, user ID, order number, order day of the week, order hour of the day, and days since prior order
6. Products: This table includes all products Instacart offers, defined by product ID, product name, aisle ID, department ID

B. Data Merging

In order to work with the data, we merged all datasets into one and then extracted predictor variables from the master dataset. The master dataset contains the following features:

1. Order ID
2. User ID
3. Product ID
4. Order number
5. Order day of the week
6. Order hour of the day
7. Days since prior order
8. Add to cart order
9. Aisle ID/Aisle
10. Department ID/department
11. Product name
12. Reordered (target Variable)

The merged data frame of all the orders includes:

- Order numbers ranging from 1 to 100
- Days between orders range from 5 to 30
- The largest basket contains 145 products
- There are 134 aisles and 21 departments
- Total of 33,819,106 entries and 15 columns
- On average 59% of orders are reordered

C. Feature Engineering

Since we aim to predict the probability of a customer ordering a particular product in future orders there are a few characteristics that are of concern here.

- **User**: the goal is to understand the behavior of each customer, their likelihood to reorder products, and their ordering pattern.
- **Product:** be able to describe the characteristics of a product, mainly it's reorder percentage

- **User and Product:** be able to describe a user's behavior towards particular products, which products do they reorder the most, the reorder ratio of a product by user orders after it was first ordered, and average days between each order of a product by user.

From these, we can derive predictor variables targeting reorder probability. The variables we will use to predict our target variables will be the following:

- **Days since prior order** - this gives us information about when the last time the user ordered a particular product
- **Average days between order** - this gives us information about how often a customer orders a product on average
- **The user product reorder ratio** - this gives us information on how often a customer has purchased an item after they first discovered it. We first create a data frame with user ID and product ID as our key IDs, we find the order number of the first time a product is bought by a customer, the total number of orders made by the customer, and how many times the customer bought a specific product. Then we calculate the user/product reorder ratio by the following equation:

$$\text{Up\_reorder\_ratio} = \frac{\textit{total number product is bought by customer}}{\textit{number of chances to buy after first purchase}}$$

- **Total number of orders** - this gives us information on how many total orders a customer has made with Instacart
- **Reorder ratio of each product** - this gives us the ratio of how often a particular product is reordered on average across all users. We first create a data frame of all product ID's then count total reorders and total purchases, then the percent reorder is calculated by the following equation:

$$\text{Percent\_reorder} = \frac{\textit{times a product has been reordered}}{\textit{times a product has been ordered}}$$

D. Final Dataframe:

For our final data frame, we will create a subset of our original master data frame with the following variables:

- User ID
- Product ID
- Days since prior order
- The total number of orders by user
- Reordered

We then have a data frame with the user ID and product ID as our key IDs and we calculate the average days between orders and the user/product reorder ratio and we left merge his data frame with the product data frame containing product ID and percent reorder ratio of each product.

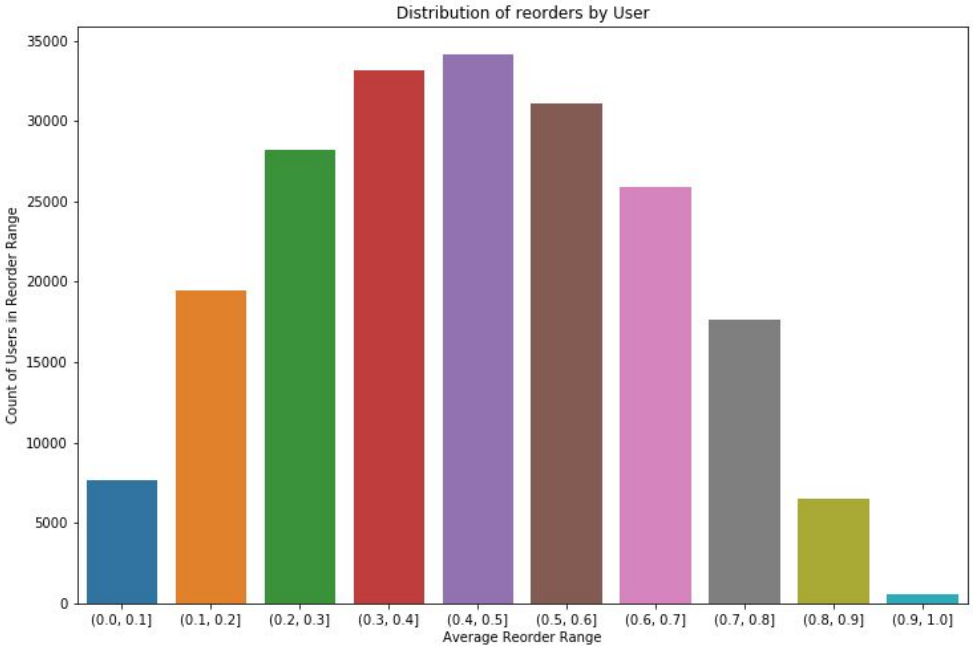
Now we have 5 features that will help us classify each product as a potential reorder or not.

III. Exploratory Data Analysis

When thinking about the features that would best serve to predict reorder probability, I considered the user, product, and the user/product relationship between reorders.

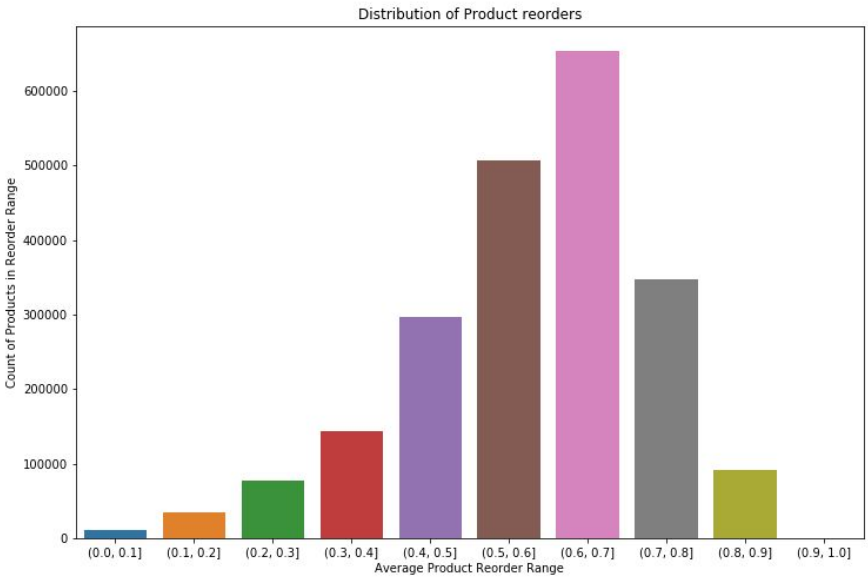
A. User Reorder Relationship

To start to understand the relationship of the user and reorder ratio, I plotted a bar graph to see the distribution of how often users are reordering items in their baskets. The figure below depicts the frequency of reorder ratio amongst users. It shows that the majority of users reorder between 30 and 60% of their orders.



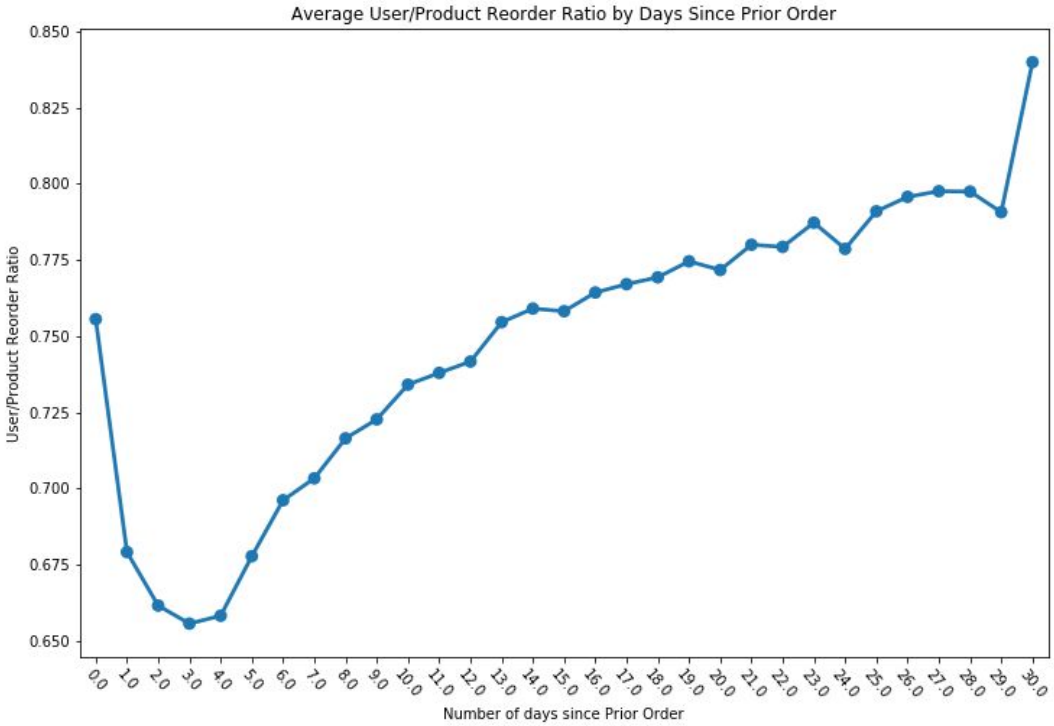
B. Product Reorder Relationship

In the figure below, we can see the frequency of each product's reorder ratio. This shows us that the products chosen in future baskets tend to be products with an overall reorder ratio of .5 - .8, with the highest peak between .6 and .7.



C. User/Product Reorder Ratio

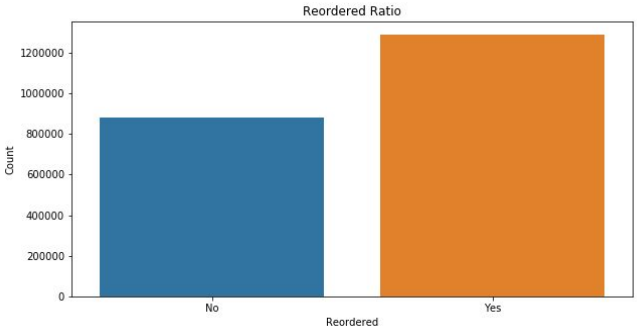
In the figure below, we can see that the likelihood of a customer reordering an item increases as the time between the last order of the product increases. The outliers between 0 and 2 days from prior order may occur from forgotten essential items in a previous basket which would make their reorder percentage high and days between prior order low. Otherwise, there is a clear trend of increasing average user/product reorder ratio as time increases.



IV. Model Selection

In the final dataset, the reorder ratio has an uneven class distribution, skewed more towards reorders than not. 60% of products in the final dataset used to predict reorders are past reorders where only 40% are new orders. Therefore, I will rely more heavily on F1 scores rather than accuracy scores as a metric for model success.

	reordered	percent
1	1286302	0.59402
0	879115	0.40598



I tested 3 different machine learning classification models: Logistic Regression, Decision Tree, and Gradient Boosting. The F1 score of each were .88, .95, and .98, respectively. I performed a Randomized Search for both decision tree and gradient boosting hyperparameters. I chose RandmizedSearchCV over a grid search since the datasets used were relatively large and wanted to optimize time as well. After performing a random search on the gradient boosting classifier, my F1 score increased from .94 to .98, which was a significant improvement.

V. Conclusion

Logistic Regression performed significantly worse than the Decision Tree and Gradient Boosting Classifiers. It turns out that the many features given in the main datasets were not very necessary in predicting reorder probability in our models. We were able to keep our features to a minimum while creating high performing predictive models.

VI. Future Work

This project made me think about the possibility of using the purchasing behavior of users to develop new products with a high probability of popularity amongst users. It would be interesting to see how to better handle and use categorical data to provide insight into food popularity and consumption. I found that it was more challenging to use the given categorical data and therefore ended up not using it. I think it would be worthwhile to explore this further to help improve predictive models in the food industry that rely heavily on categorical data to inform them about their products.

Questions:

- Explore the relationship between categorical features and reorder ratio in depth. Can we use department/aisle features to help improve predictive models for future baskets?
- Can we start to predict the order in which a user adds items to their basket?
- Can we predict which items are being ‘forgotten’ at check out to avoid customers reordering the next day?
- Where can the model be expanded to other areas in the food industry?
- Is there a way to use this model to reduce waste and supply chain management?