**IDS 594 Machine Learning with Python**

# **Instacart Market Basket Analysis**

By:

Gaurav Adikey – 665546151

Kaustubh Sarang- 673619813

Nimita Singh-667252322

Sajal Joshi-671563590

Sharadind Peddiraju-650890429

**Introduction**

**Instacart** is an American technology company, Founded in 2012, that operates as a same-day grocery delivery and pick-up service in the [U.S.](https://en.wikipedia.org/wiki/United_States) and [Canada](https://en.wikipedia.org/wiki/Canada). It is currently valued at nearly $8 billion. Customers shop for groceries through the Instacart mobile app or Instacart.com from the company's more than 300 national, regional and local retailer partners. The order is shopped and delivered by an Instacart personal shopper

**Project Objectives:**

In this project we have used anonymized data on customer orders over time to predict which previously purchased products will be in a user’s next order. The main motivation of this project is to solve the objectives as listed below.

* **predict whether a product will be reordered or not.**
* **Predict which department a product will belong to**

**Data set:**

In 2017, the company announced its first public dataset release, which is anonymized and contains a sample of over 3 million grocery orders from more than 200,000 Instacart users.

As a part of the project, we worked on the following six datasets:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Datasets** | **Columns** | | | |
| **Aisles.csv** | aisle\_id | aisles |  |  |
| **Department** | department\_id | department |  |  |
| **Order\_products\_prior.csv** | order\_id | product\_id | add\_to\_cart\_order | reordered |
| **Order\_products\_train** | order\_id | product\_id | add\_to\_cart\_order | reordered |
| **Orders.csv** | order\_id | user\_id | eval\_set | order\_number |
|  | order\_dow | order\_hour\_of\_day | days\_since\_prior\_order |  |
| **Products.csv** | product\_id | product\_name | aisle\_id | department\_id |

**Data Cleaning**

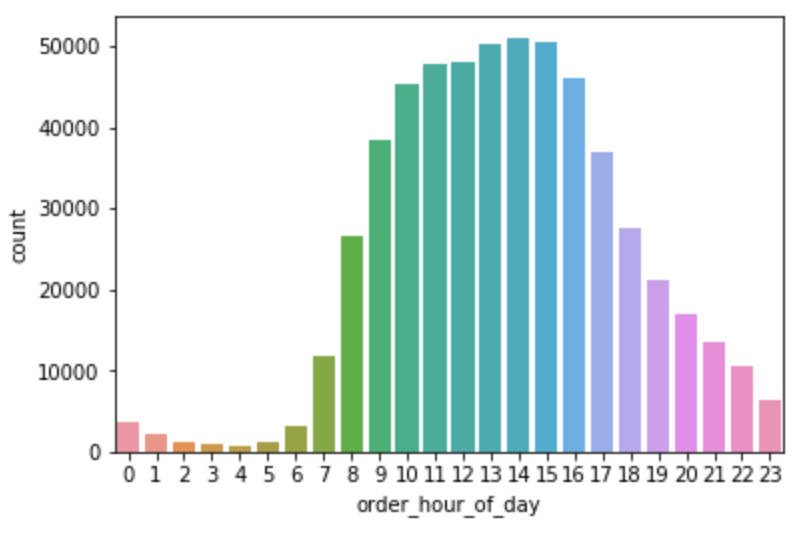
We checked the data for missing values. The data was relatively clean and only the orders.csv dataset had any missing values. We calculated the length of the entire column and then the number of missing values for each column. The percentage of missing values per columns were then calculated.

No missing values were present in the **aisles, deparments, order\_product\_prior, order\_product\_train** and **products databases**. The only missing values found were in Orders dataset. Only the orders dataset had any missing values. W observed that, only 6% of **days\_since\_prior\_order** column in the orders dataset were null. So, we decided to truncate them. This was done to ensure that there are no null values in the dataset.

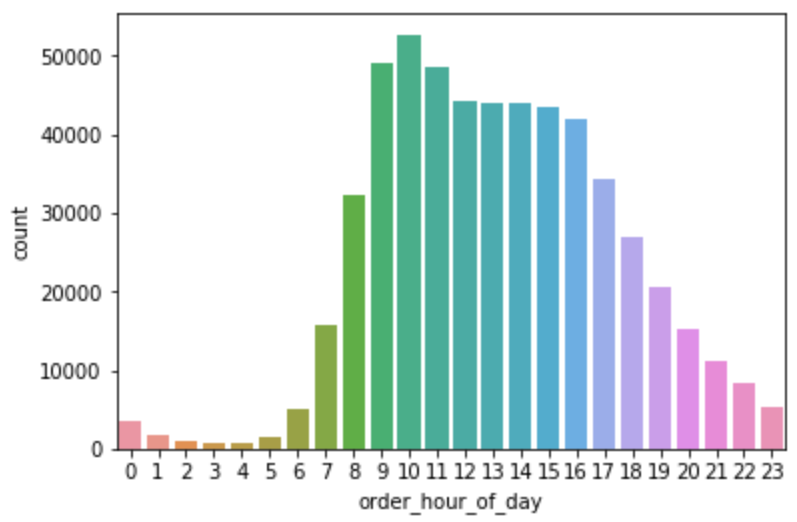
**Exploratory Data Analysis**

We were provided with six data sets. Each dataset had information regarding various facets of an order being placed. We attempted to get the check of the distribution between, prior, train and test data. We then also plotted the number of products against hour of the day for each day of the week.

We observed that the maximum number of orders is around 10 AM and 11AM followed by 3-4PM. The reason behind it that these are peak hours for the order as people are active during this time. Meanwhile orders are least at 3-4AM indicating the utilization of the app at those times.

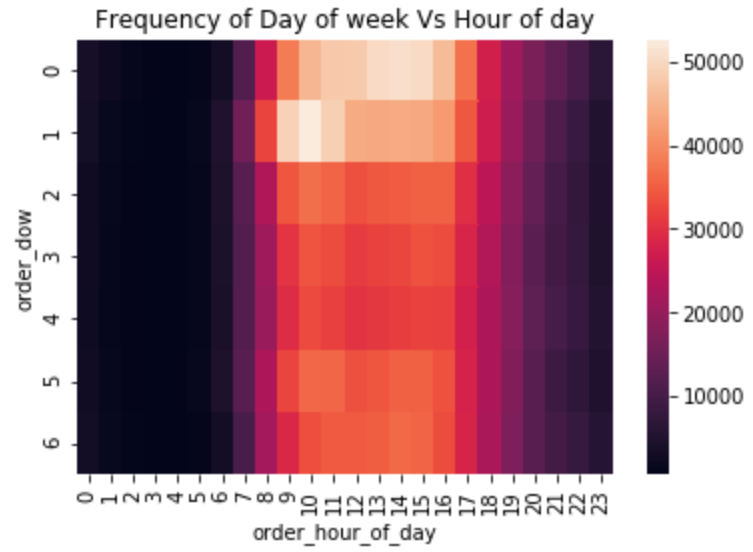


We were interested in the weekly distribution of orders as well as evident in the plot below.

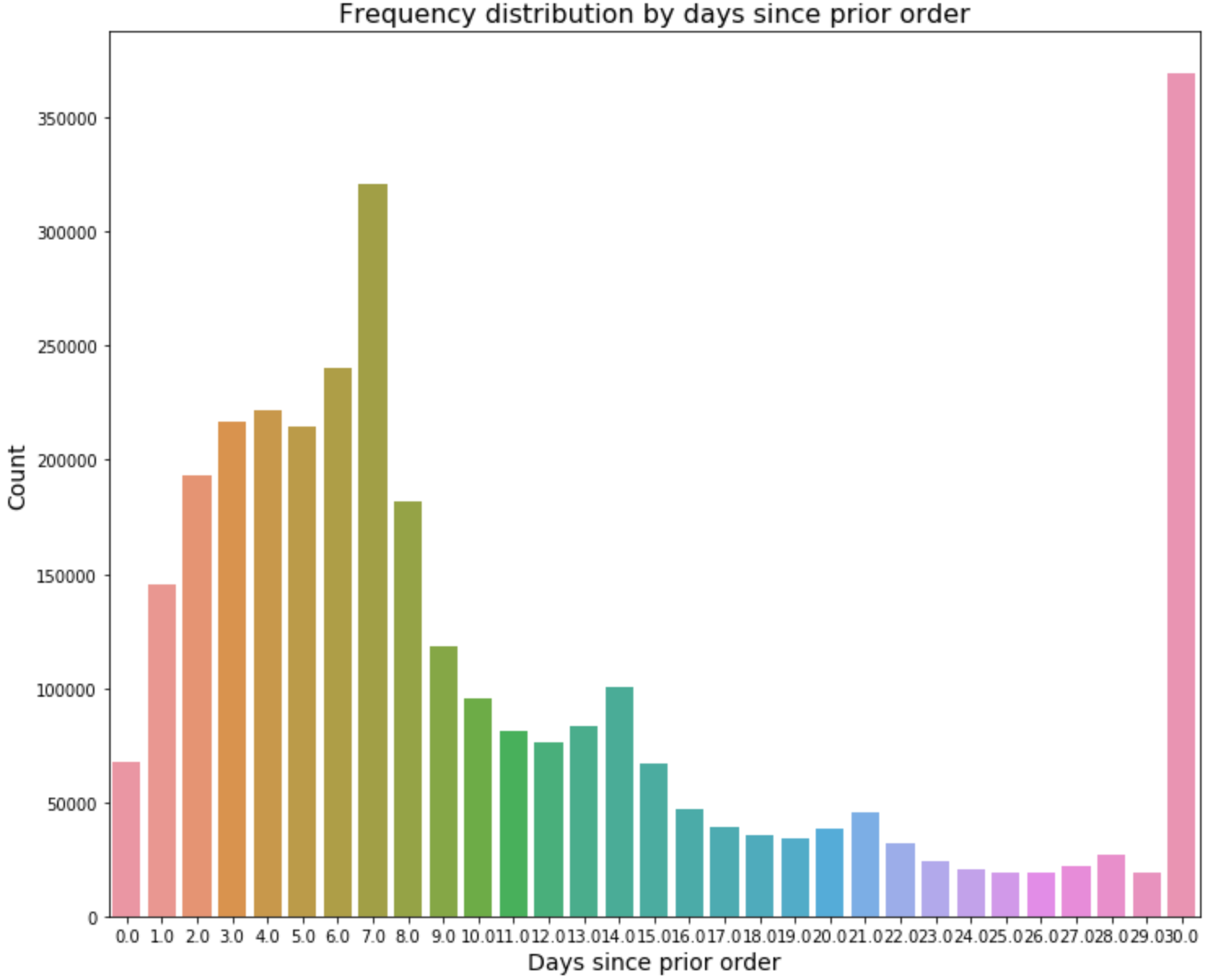


* The peak is reached at 10AM. So, most of the orders are placed in the morning on Monday from11AM. Followed by 12-3pm.

Next, we got the orders in terms of hour of the day and day of the week in a single dataset by using the GroupBy option for better visualization. After converting the newly generated data frame using the pivot feature of pandas. We were able to generate the heatmap below.

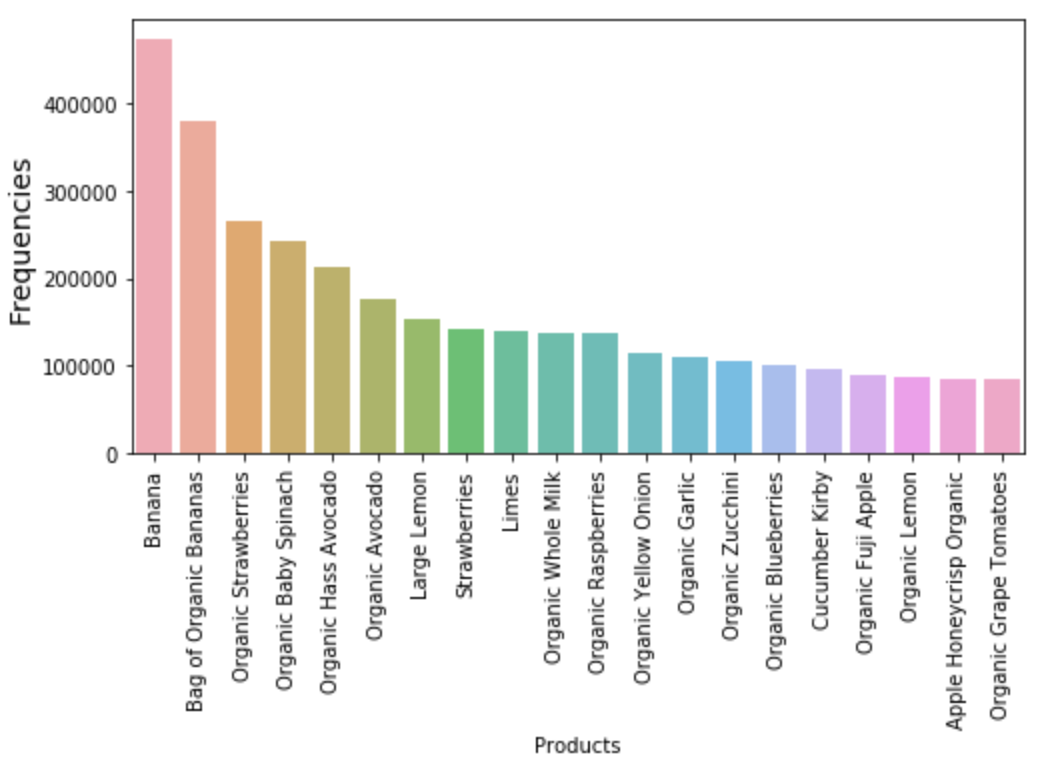


We also noted that 59% of the products are re-ordered from the prior dataset while 60% of the products are re-ordered from the train dataset. To check for seasonality the following graph was plotted.

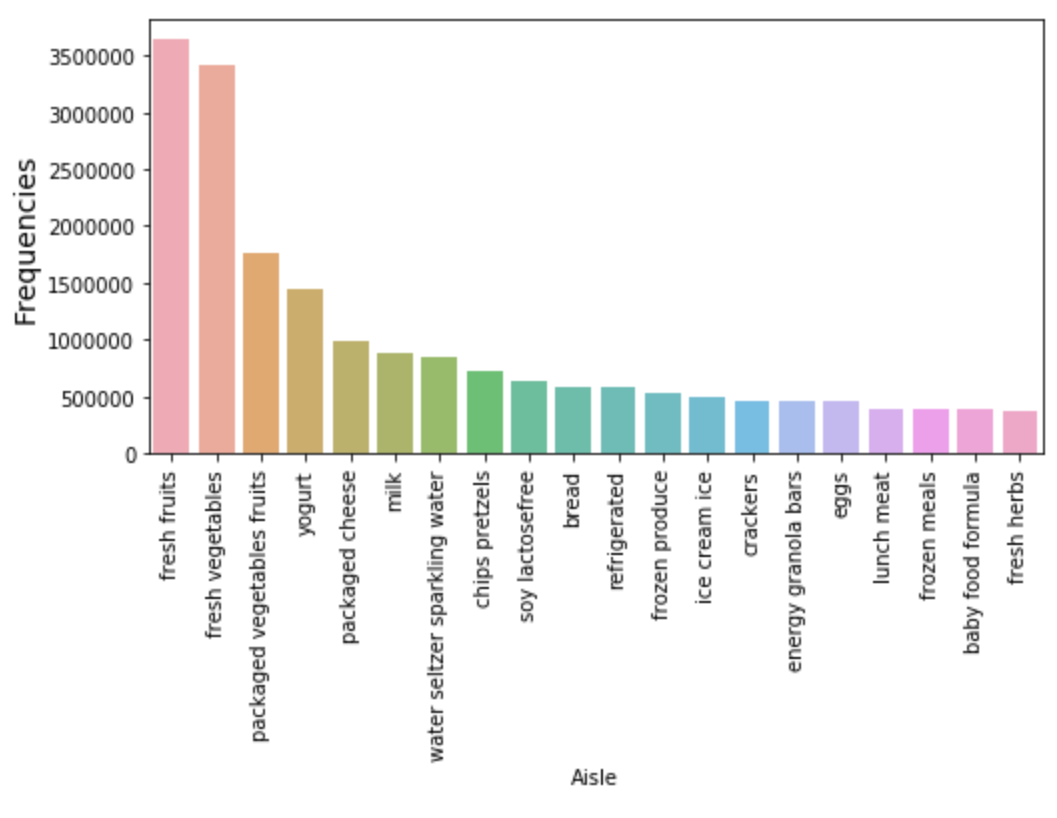


We observe that on the 7th day we have a spike. Then there is spike on 14,21 and 28 days. This indicates that every 7 days or weekly is the order frequency. Also, there’s a huge peak at the end of the month showing there's a monthly peak.

We then progressed to find the distribution of products, aisles and departments using the new merged dataset.

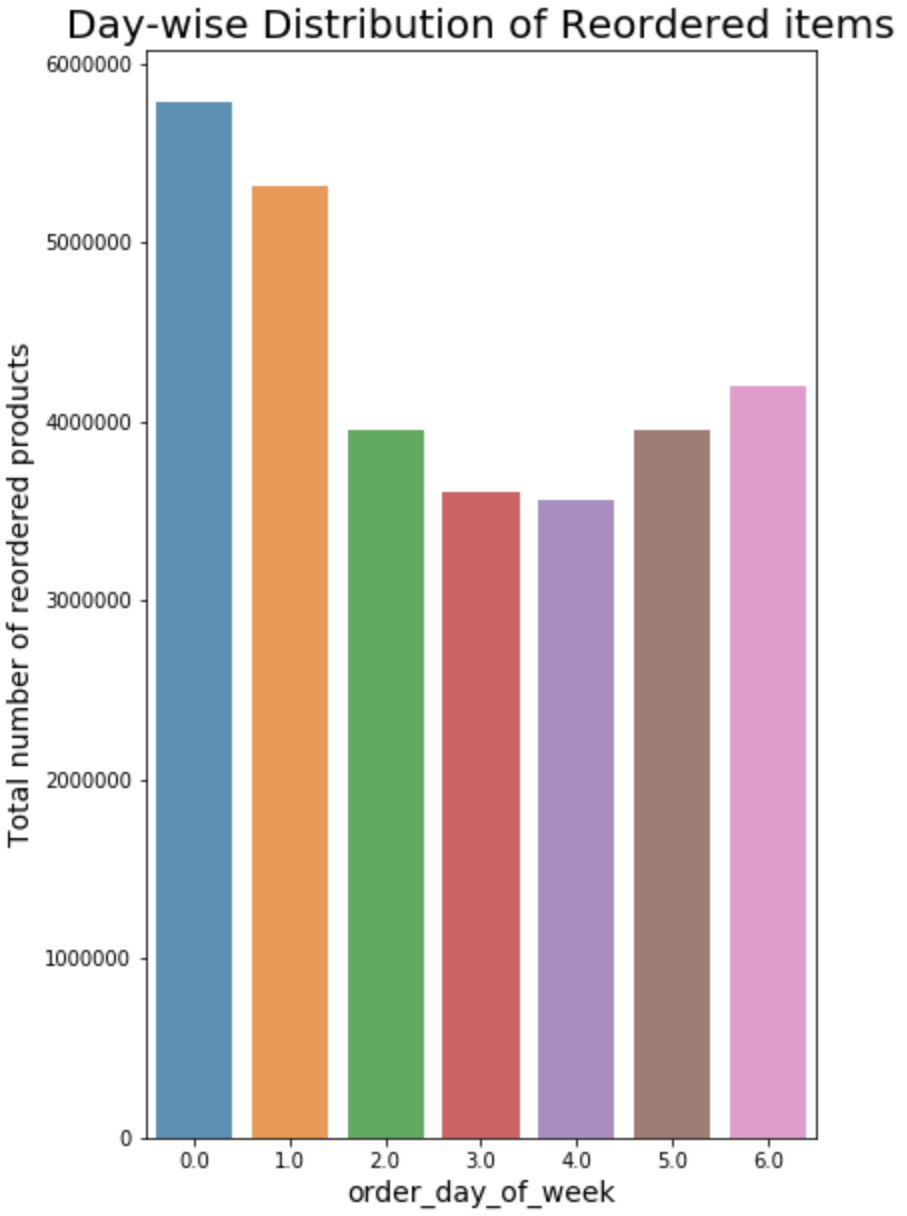


**TOP 20 ordered products on Instacart**



**TOP 20 ordered aisles on Instacart**

We were also interested in knowing when items were being reordered.



**Feature Engineering and Selection:**

**Feature engineering** is the process of transforming raw data into **features** that better represent the underlying problem to the predictive models, resulting in improved model accuracy on unseen data. **Feature engineering** turn your inputs into things the algorithm can understand.

We created many features that we felt would help us in our model building. These features were created after understanding the problem statement and business model of Instacart. Feature selection was also an important part of building the model. We ensured that only relevant variables were considered before building a model. As stated earlier, we are trying to solve for two problem statements. We used features as listed below.

* **predict whether a product will be reordered or not.**

**Features Used:**

|  |  |  |
| --- | --- | --- |
| Order\_id | Order\_number | Average\_days\_between\_orders |
| Nb\_orders (Number of orders) | Average\_basket | Total items |
| Aisle | Department | Product |
| Order\_hour\_of\_day | Order\_dow(day of week) | Days\_since\_prior\_order |
| User\_id | Days\_since\_ratio |  |

* **Predict which department a product will belong to**

**Features Used**

|  |  |  |
| --- | --- | --- |
| Order\_id | Order\_number | Average\_days\_between\_orders |
| Nb\_orders(Number of orders) | Average\_basket | Orders |
| Reorders | Reordered rate | Total items |
| User\_id | Order\_hour\_of\_day | Order\_dow(day of week) |
| Days\_since\_prior\_order | Days\_since\_ratio |  |

**Model Building**

The features created in the previous section were used to build models. This feature engineering helped us in achieving good accurate results in predicting whether a product would be reordered or not and for predicting the department to which a product will belong.

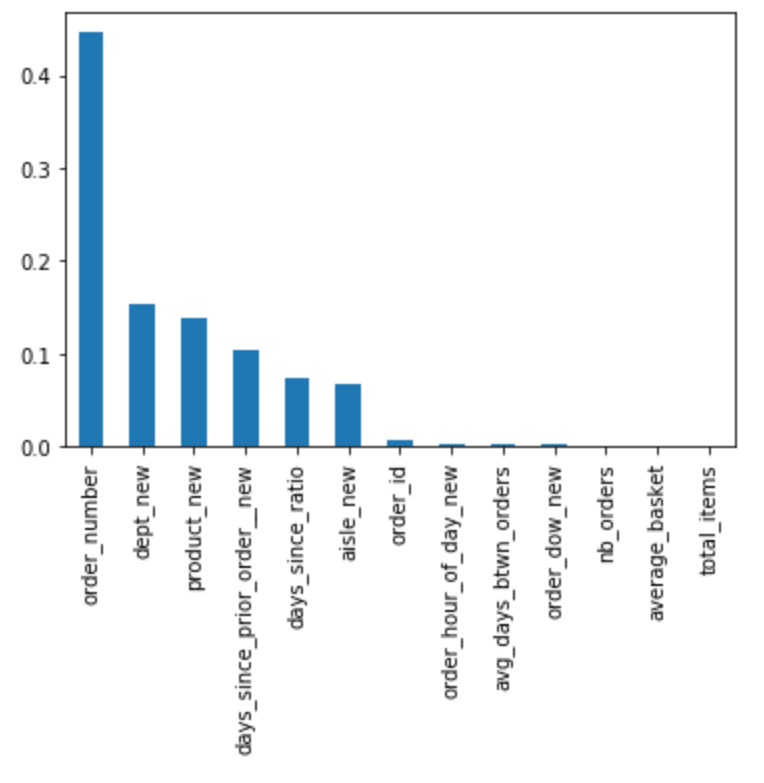
We split the data into train and test and then ran the models on them.

**Predict whether a product will be reordered or not.**

We used various models to predict if a product will be reordered or not, such as, Logistic regression, Random Forest, Adaboost Classfier and Gradient Boosting. Logistic regression was the worst performing model while Gradient Boosting was the best performing model with an Accuracy of 67.07% on the test data. It also had the highest AUC of 0.66. Listed below are the metrics for each model on the training as well as the test data.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model Selection** | **Test Data** | | | |
| **Accuracy** | **Precision** | **Recall** | **AUC** |
| **Logistic Regression** | **59.70%** | **85.63%** | **61.83%** | **0.56** |
| **Random Forest** | **59.76%** | **85.63%** | **61.83%** | **0.56** |
| **Adaboost Classifier** | **65.57%** | **80.02%** | **68.06%** | **0.65** |
| **Gradient Boosting** | **67.55%** | **82.87%** | **69.08%** | **0.66** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model Selection** | **Test Data** | | | |
| **Accuracy** | **Precision** | **Recall** | **AUC** |
| **Logistic Regression** | **59.71%** | **85.69%** | **61.79%** | **0.55** |
| **Random Forest** | **66.40%** | **82.06%** | **68.37%** | **0.65** |
| **Adaboost Classifier** | **65.65%** | **80.13%** | **68.12%** | **0.63** |
| **Gradient Boosting** | **67.06%** | **82.52%** | **68.70%** | **0.65** |

Looking at the Variable Importance for predicting whether a product will be reoredered or not, the most important features we observed are order\_number, department, product, days since prior order and aisle

* **Predict which department a product will belong to:**

The objective here is to not only only say if an object belongs to one of the 21 categories, but to also provide the probability that it belongs to these classes. We believe that the log loss score is best suited for measuring the effectiveness of the model. Log Loss score quantifies the accuracy of the classifier by penalizing false classifications. So if we minimize the log loss, it will result increase in accuracy of the classifier.

We used Random forest, Adaboost classifier, Gradient Boosting on the dataset. Our best performing model was the Random Forest Classifier with the Lowest Log loss score of 2.342. Listed below are the metrics for each model on the training as well as the test data

.

|  |  |  |
| --- | --- | --- |
| **Model Selection** | **Train Data** | **Test Data** |
| **Log Loss Score** | **Log Loss Score** |
| **Random Forest** | **2.34** | **2.33** |
| **Adaboost Classifier** | **2.28** | **2.34** |
| **Gradient Boosting** | **2.95** | **2.97** |

**Team Members Contribution:**

Data Cleaning: Sajjal Joshi, Kaustubh Sarang.

EDA: Nimita Singh, Gaurav Adikey, Kaustubh Sarang

Model Creation: Gaurav Adikey, Nimita Singh, Sharadind Peddiraju