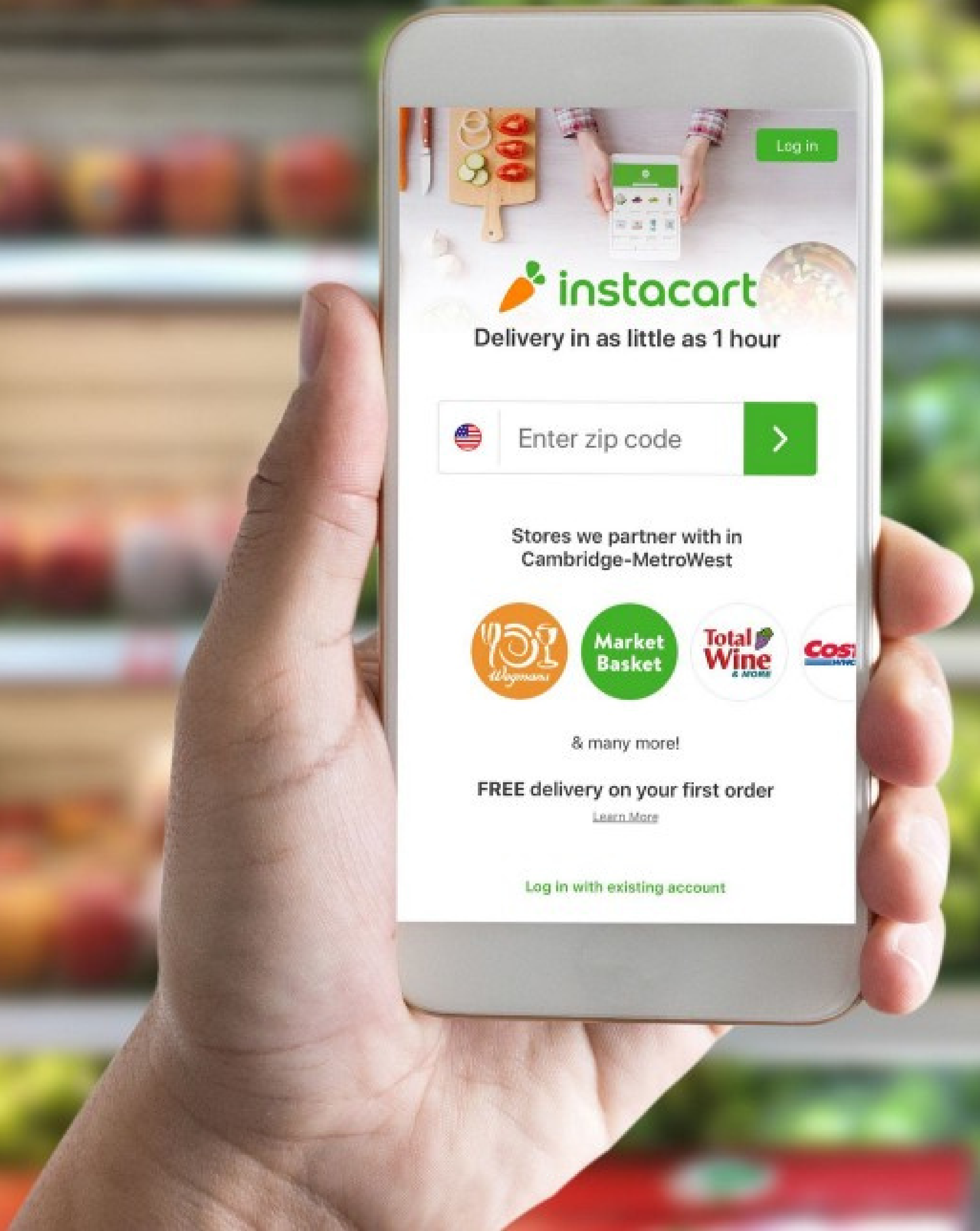

Log in
instacart
Delivery in as little as 1 hour
Enter zip code
Stores we partner with in Cambridge-MetroWest
Wegmans
Market Basket
Total Wine & MORE
Cost
& many more!
FREE delivery on your first order
Learn More
Log in with existing account

Machine Learning Project

Presented to: **Prof. Claudio Sartori**

Presented By: **Jyoti Yadav**

Academic Year: **2021-2022**

What is Instacart?

Instacart is an American company that provides grocery delivery and pick-up service. The Company operates in the U.S and Canada. Instacart offers its services via a website and mobile app. Unlike another E-commerce website providing products directly from Seller to Customer. Instacart allows users to buy products from participating vendors. And this shopping is done by a Personal Shopper. The company is expanding its platform to cover 90 millions US household in 2018.

Outline

1. Problem Overview

2. Data Source

3. Machine learning Problem

4. Performance Metrics

5. Exploratory data Analysis

6. Feature Engineering

7. Creating Training and Testing data

8. Modeling Building

9. Evaluation Metrics

10. Threshold

11. Conclusion

Problem Overview

The main objective of this competition was to predict which previously purchased products will be in the user’s next order. The problem is a little different from the general recommendation problem, in this case, we are not recommending new products to the user instead of that we are going to recommend the product which is previously bought by that user. This will also help Instacart to help local vendors with their retail business by reminding them to restock items which are bought frequently.

Data Source

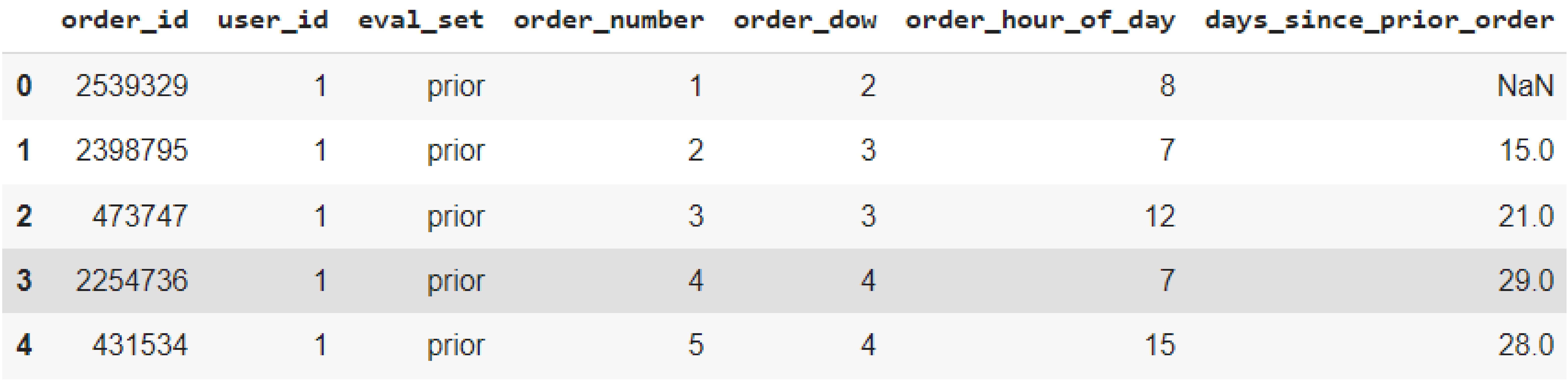
The dataset is available on the Kaggle Here.

The dataset for this competition is a relational set of files describing customers' orders over time. The dataset is anonymized and contains a sample of over 3 million grocery orders from more than 200,000 Instacart users. For each user, we provide between 4 and 100 of their orders, with the sequence of products purchased in each order. We also provide the week and hour of day the order was placed, and a relative measure of time between orders.

File Descriptions

- **orders.csv -** It consists of order placed by any user. The eval_set column has three categories prior, train and test.
    - **Prior data-** Order history of every user . This data contains nearly 4–100 past orders per user.
    - **Train data-** Current order data of every user . This data contains only 1 order per user.
    - **Test data-** These are the orders which will be done in future and we have to predict the products which might be reordered.
- **Products.csv-**It consists of details of the products.
- **order_products__prior.csv -** consists of all product details for any prior order.
- **order_products__train.csv -** consists of all product details for a train order.
- **department.csv -** details of department
- **aisles.csv -** details of aisle

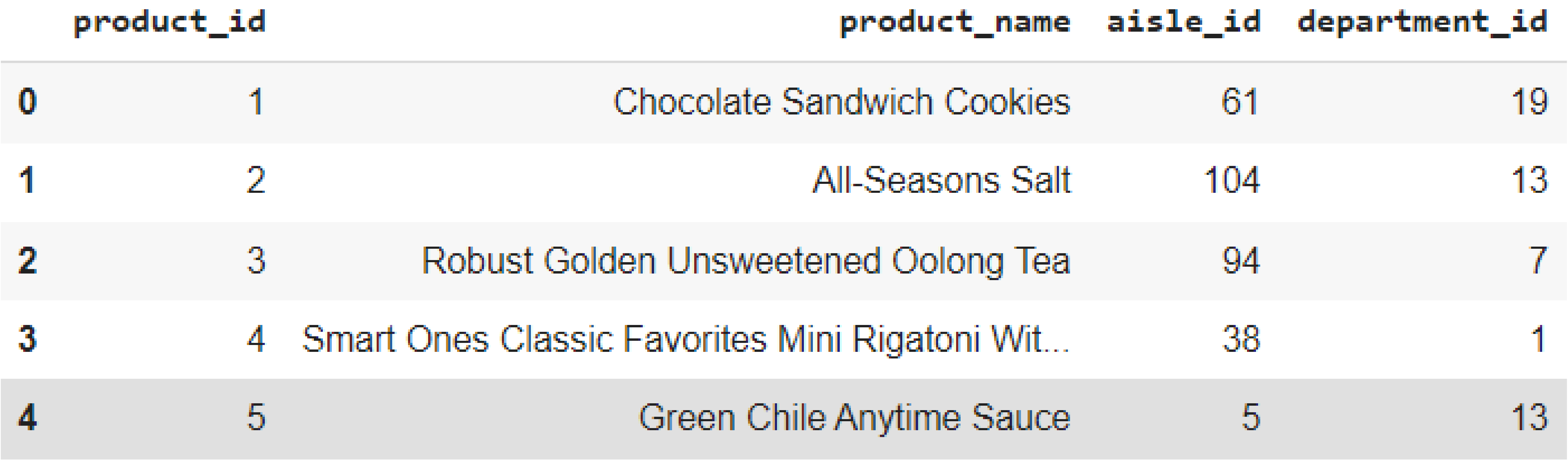
Orders.csv



	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

- Order_id : Unique for every order
- User_id : Unique for every user
- Eval_set : (prior / train / test)
- Order_number : ith order placed by user
- Order_dow : Day of week
- Order_hour_of_day : Time of day in hr
- Days_since_prior_order : difference in days between 2 orders

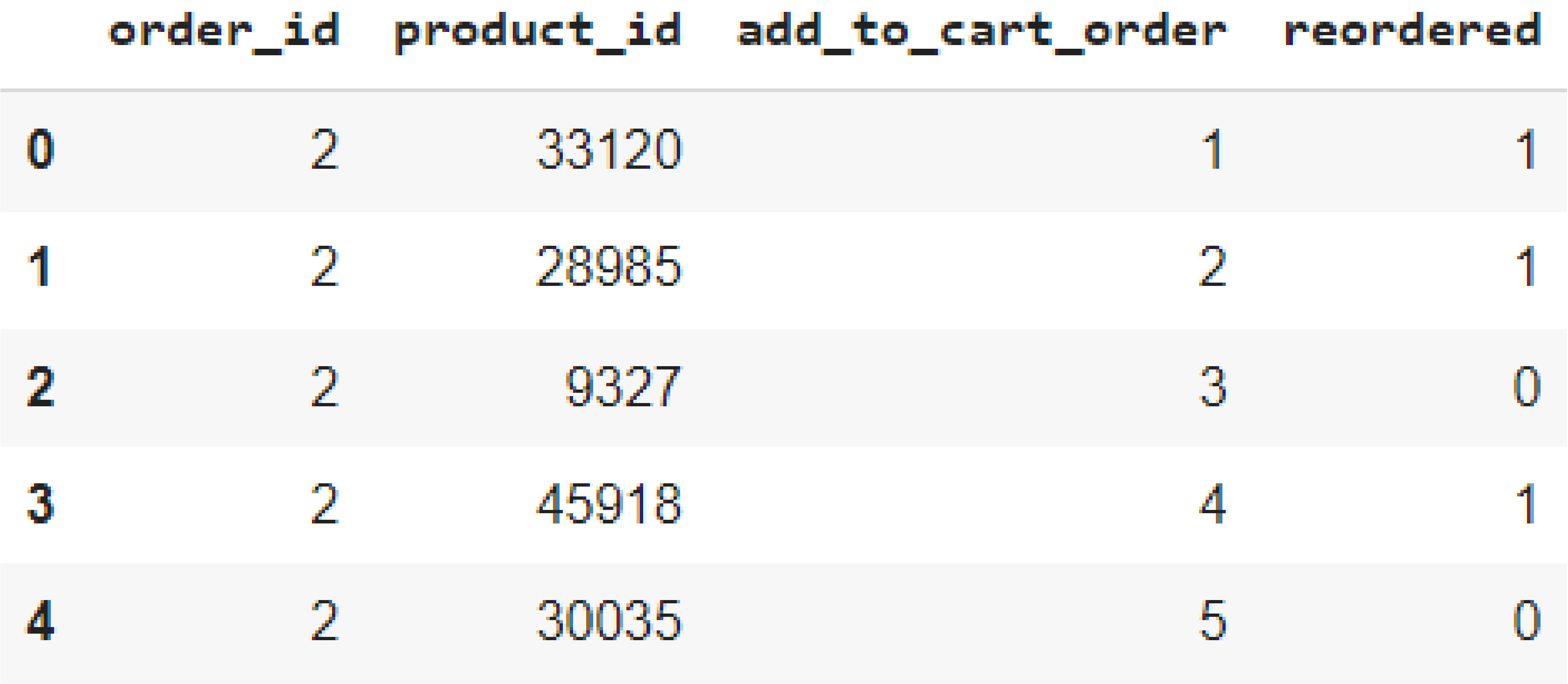
Products.csv



	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

Here for each product, we have a unique product Id along with its name and aisle id, department id it belong to.

orders_products_prior.csv

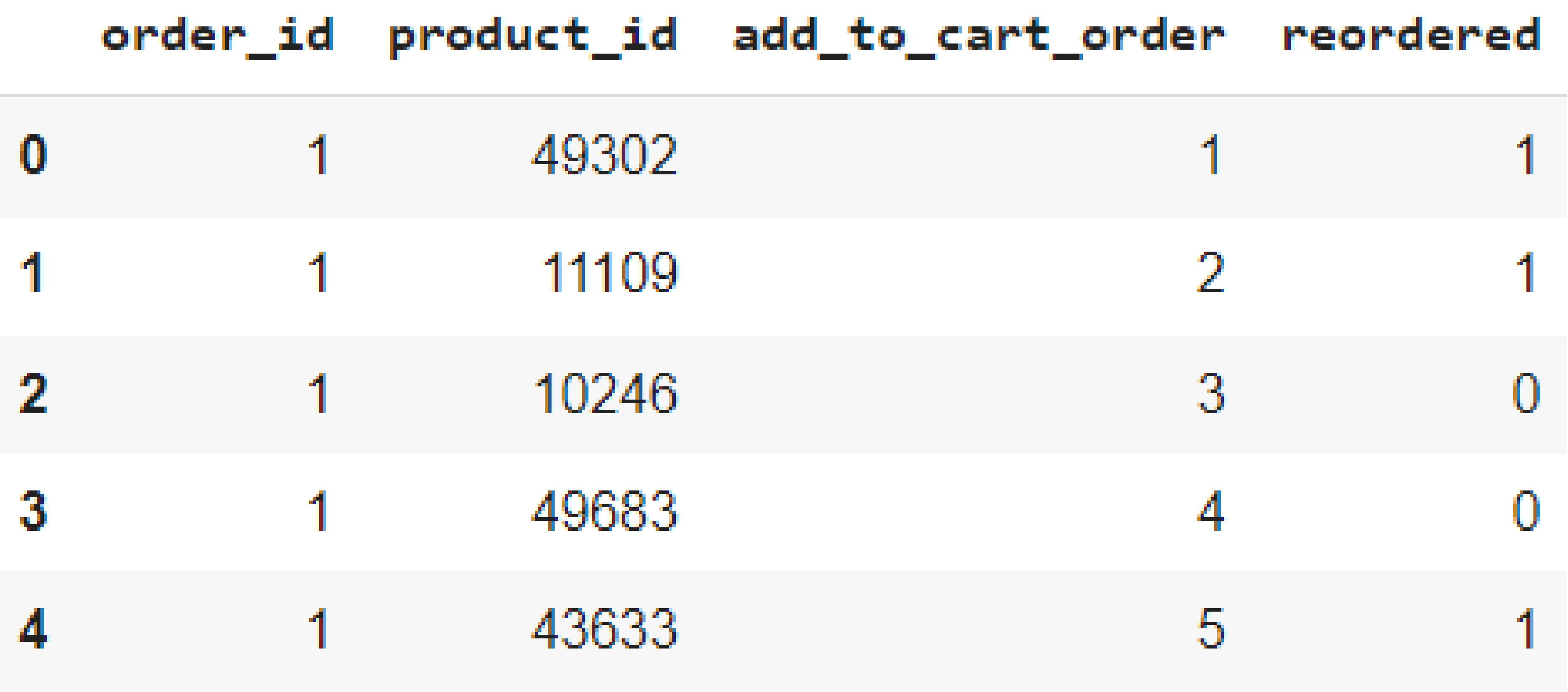


	order_id	product_id	add_to_cart_order	reordered
0	2	33120	1	1
1	2	28985	2	1
2	2	9327	3	0
3	2	45918	4	1
4	2	30035	5	0

Orders_products_prior contains the details of previous products.

- order_id : Unique order id for every order
- product_id : product ID of item
- add_to_cart_order : denotes the sequence in which products were added to cart.
- reordered : product is reordered
  Every row in this table is defining that for each order id, what is the product id (items) ordered in it and whether it is a reordered item for that user or not by reordered column. 1 stands for item is reordered and 0 for 1st time order.

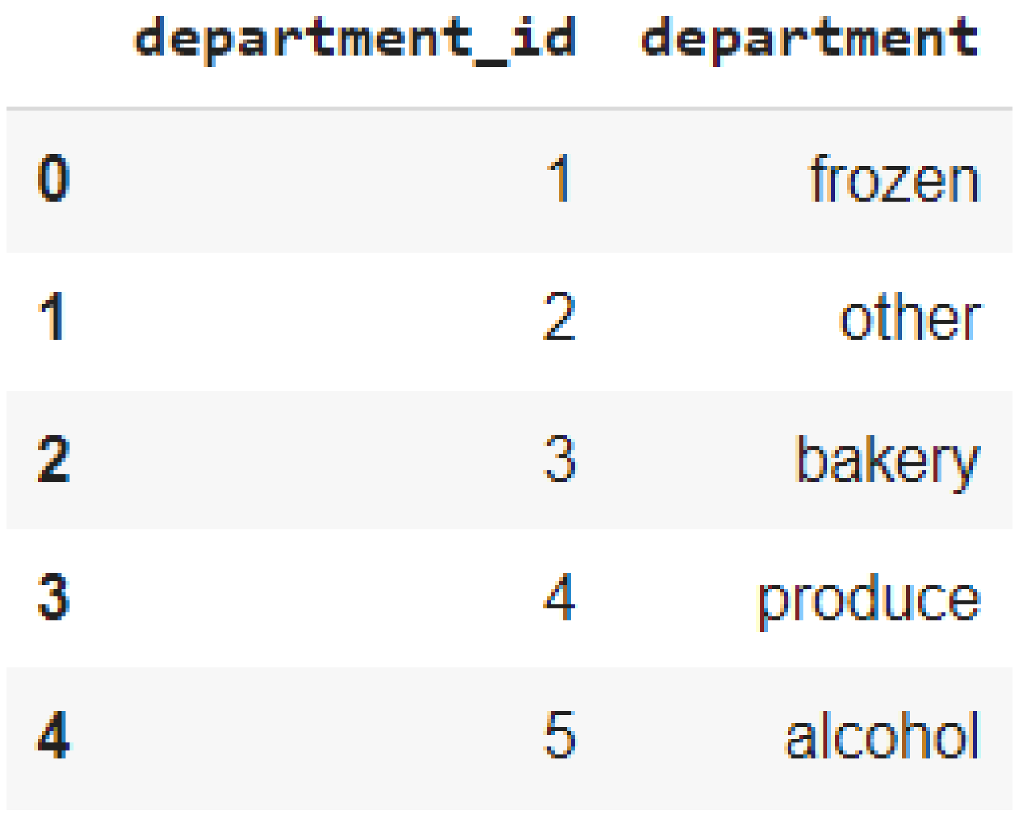
orders_products_train.csv



	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

Same as orders_products_prior table except it is for current order to train on.

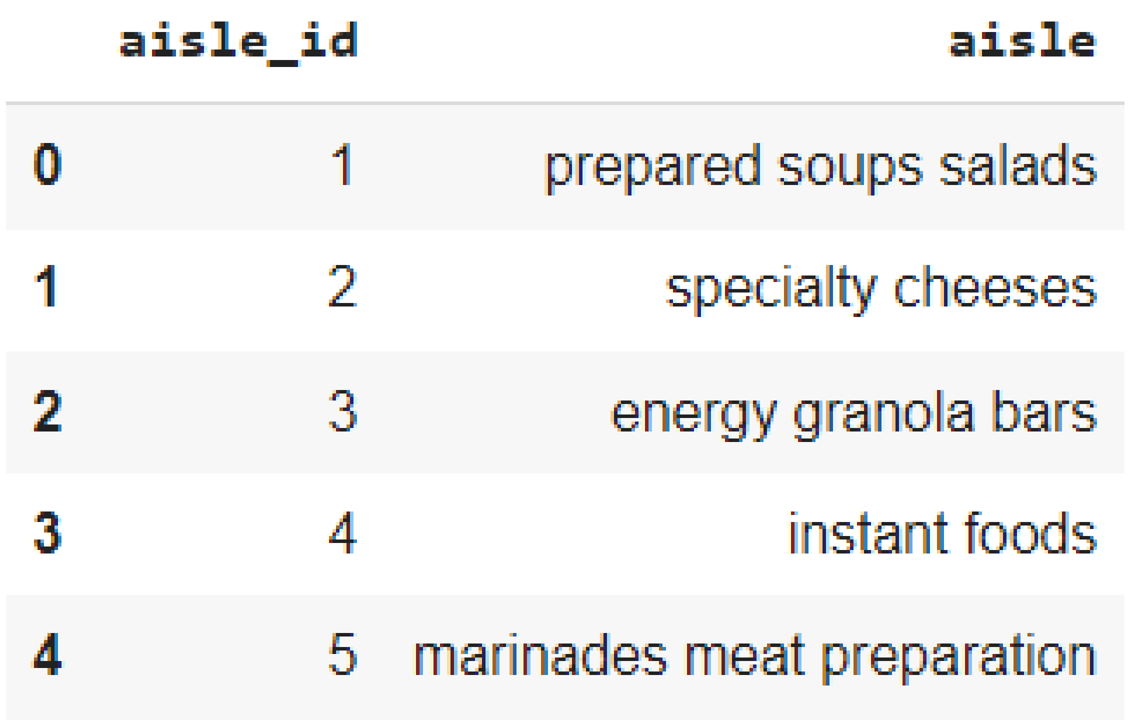
Department.csv



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

- department_id : department ID of item
- department_name : name of department

Aisle.csv



	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

- aisle_id : aisle ID of item
- aisle_name : name of aisle

Machine learning problem

We can pose it as a *classification problem* where given a customer and their previous orders, we have to predict if a product will be in his/her next order or not.

It could have also be posed as a *multi label classification problem* but there are 49688 products, and total product recommendations could be anywhere from None to N

But as we are having huge number of products we will stick to binary classification approach. So, basically we have to predict reordered column of the data as target variable.
So, for every order id we will classify each product against it as reordered or not.

Performance Metrics

The Kaggle evaluation metrics will be mean F1-Score. The F1 score can be interpreted as a harmonic mean of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. The relative contribution of precision and recall to the F1 score are equal. The formula for the F1 score is

F1 = 2 * (precision * recall) / (precision + recall)

For each orderid in the test set, we should predict a space-delimited list of productids for that order. If we wish to predict an empty order, we should submit an explicit 'None' value. You may combine 'None' with product_ids. The spelling of 'None' is case sensitive in the scoring metric. The file should have a header and look like the following:

Order_id,products
17,1,2
34,None
137,1,2,3
etc.

EDA- Exploratory data analysis

EDA is an important aspect of any machine learning or data science problem. It helps get insights in data through proper examination and is
very important because it exposes trends, patterns, and relationships that are not readily apparent. In this part of Instacart market basket analysis we will understand the problem statement and do EDA on the data to further explore the techniques to solve it.

Checking the missing/Null Values

In the orders.csv - only days_since_prior_order coloumn has null values in orders dataframe. Removing/modifying the null values is not helpful in our classification task. After analysis I came to conclusion that all nulls in ‘days_since_prior_order’ column are present because they are 1st orders for any user. We will later impute them with value 0.

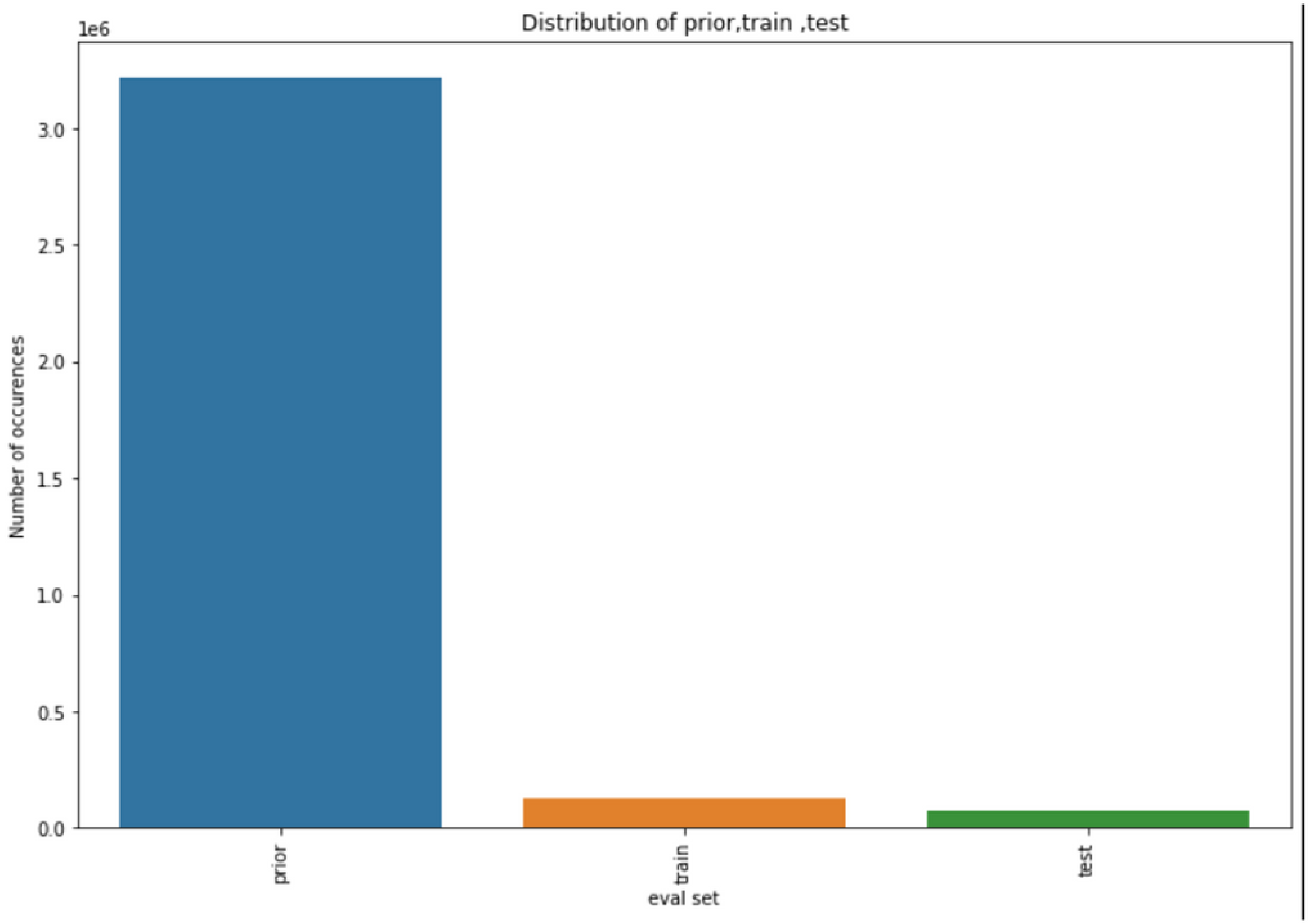
In depth EDA and problem formulation

Let's first look at the distribution of train test and prior

Prior: 3214874 data points

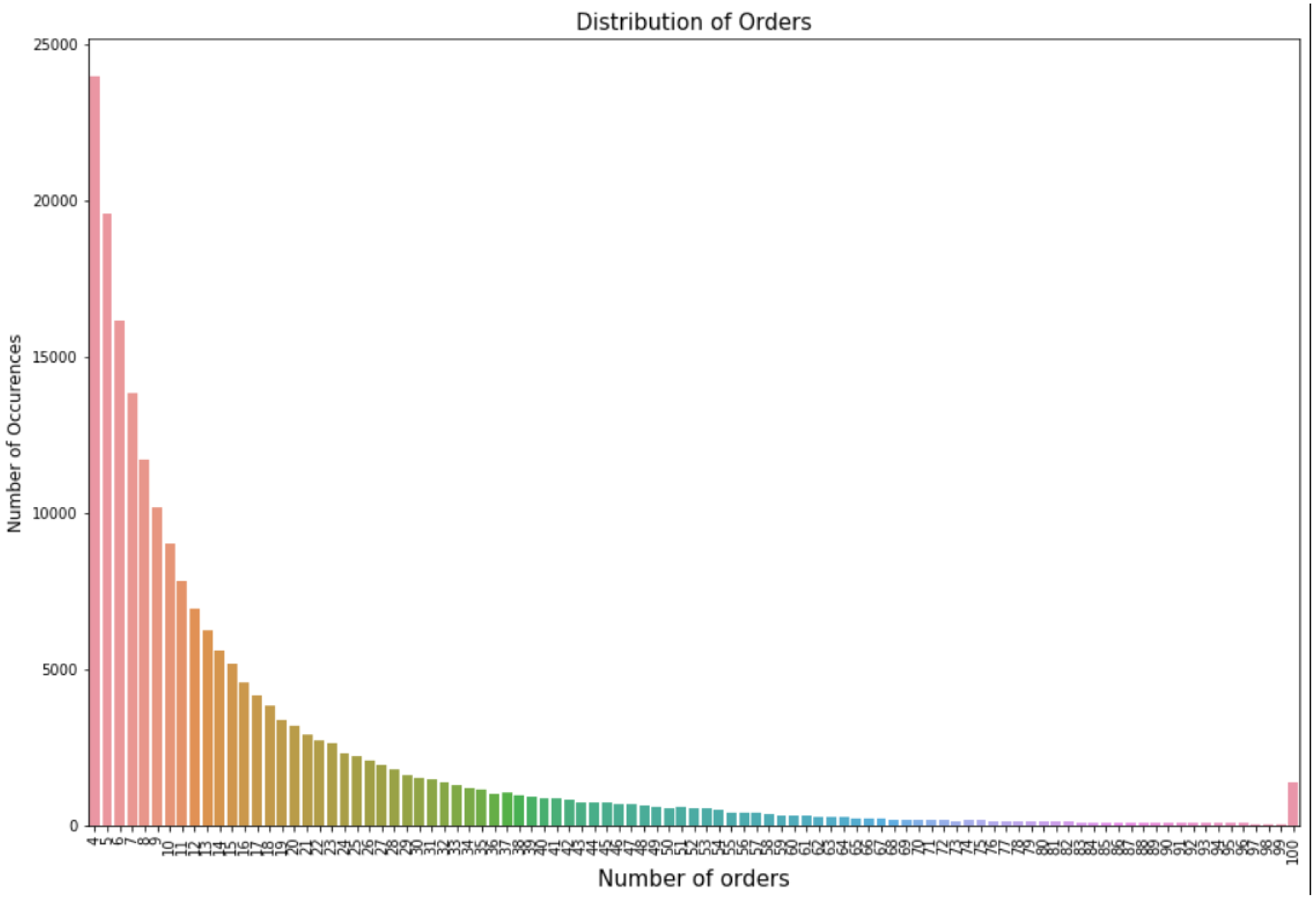
Train: 131209 data points

Test: 75000 data points



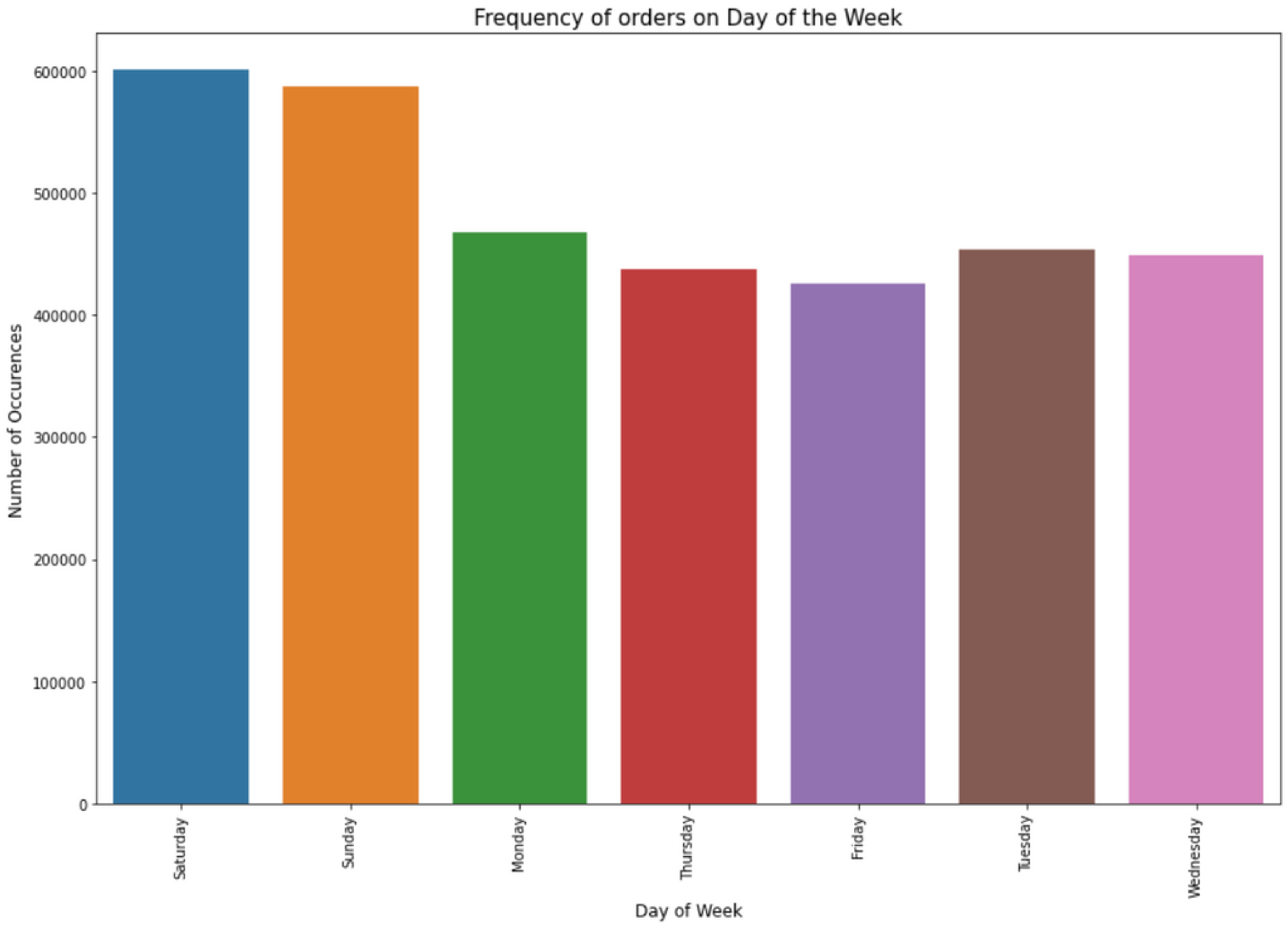
Let's check the Frequency Of Order Number Amongst The Users? Generally, how many orders a user place?

This plot explains, What’s the order count for each user? There is a high peak at 4,5,6,7&8 which tells us that users mostly orders 4 to 8 products in every order.



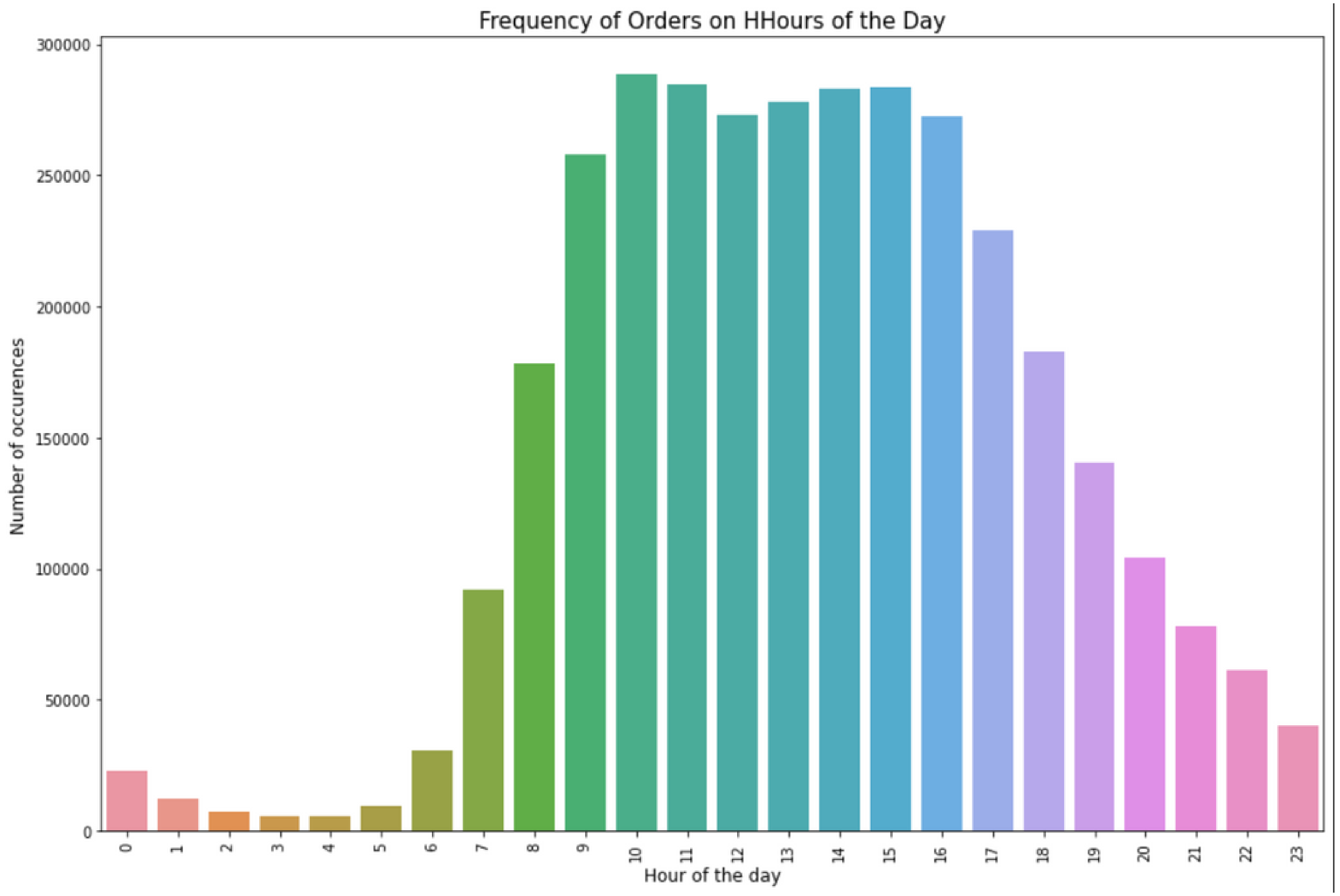
What Day Of The Week User Placed The Order?

There are 7 days in a week most of the time user placed order on that starting days of the week i.e is on Saturday and Sunday where most of the users are free on that day and have time to place the orders
Since after two days of the week frequency decreased a bit



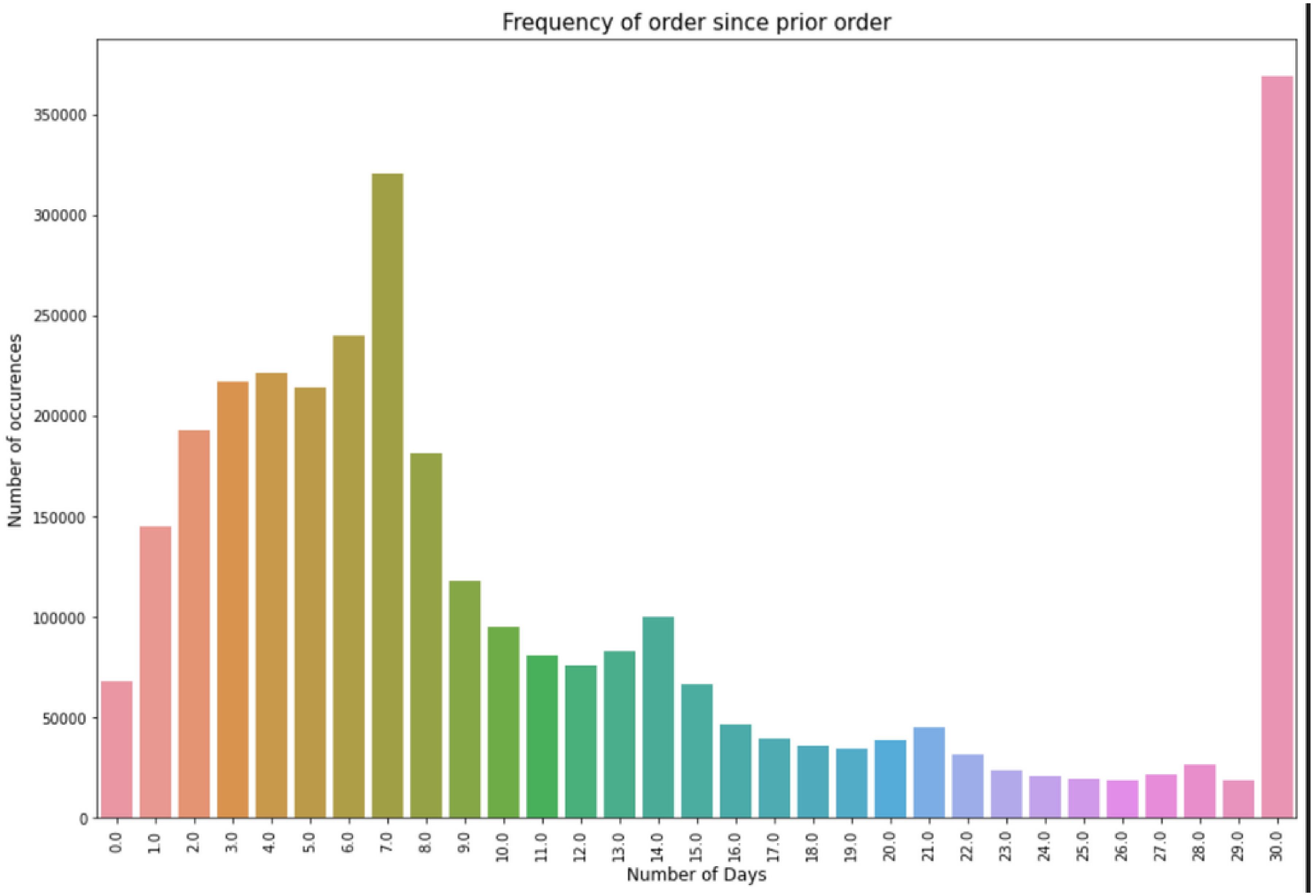
In which hour user place the order

This plot tells us what’s the user’s preferred time for placing an order?. There are high peaks in the morning hours and it slowly gets decreases. Between 10 am to 15 pm there is a high order ratio amongst users



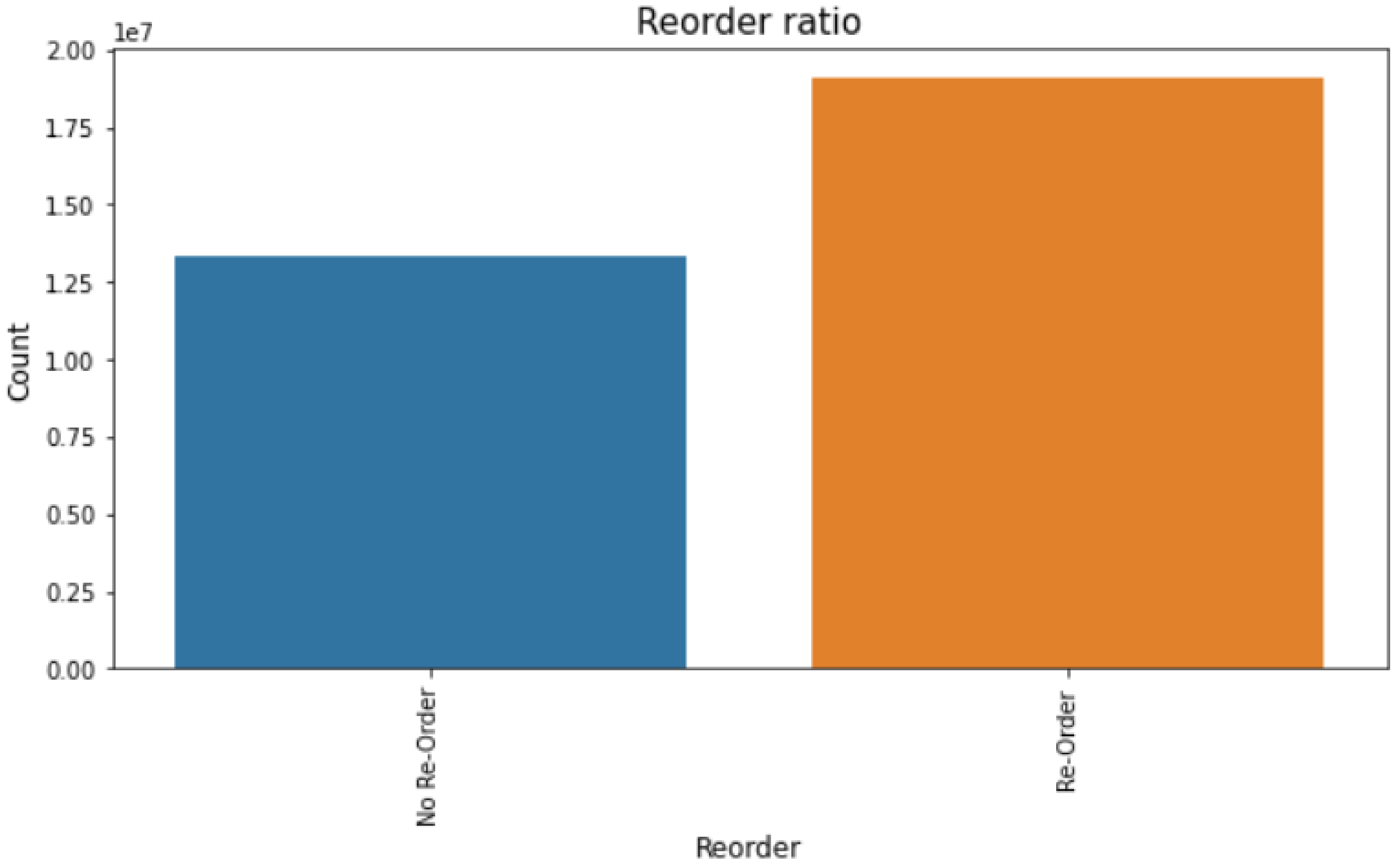
How Many Days The User Takes To Place an Order?

After placing an order, how many days a user takes to order again?. Usually, there is a gap of 7 days after each order. If you see carefully there is a high peak on the 7th day and after that on the 14th day and then at end of the month. The user takes a week to order once again.



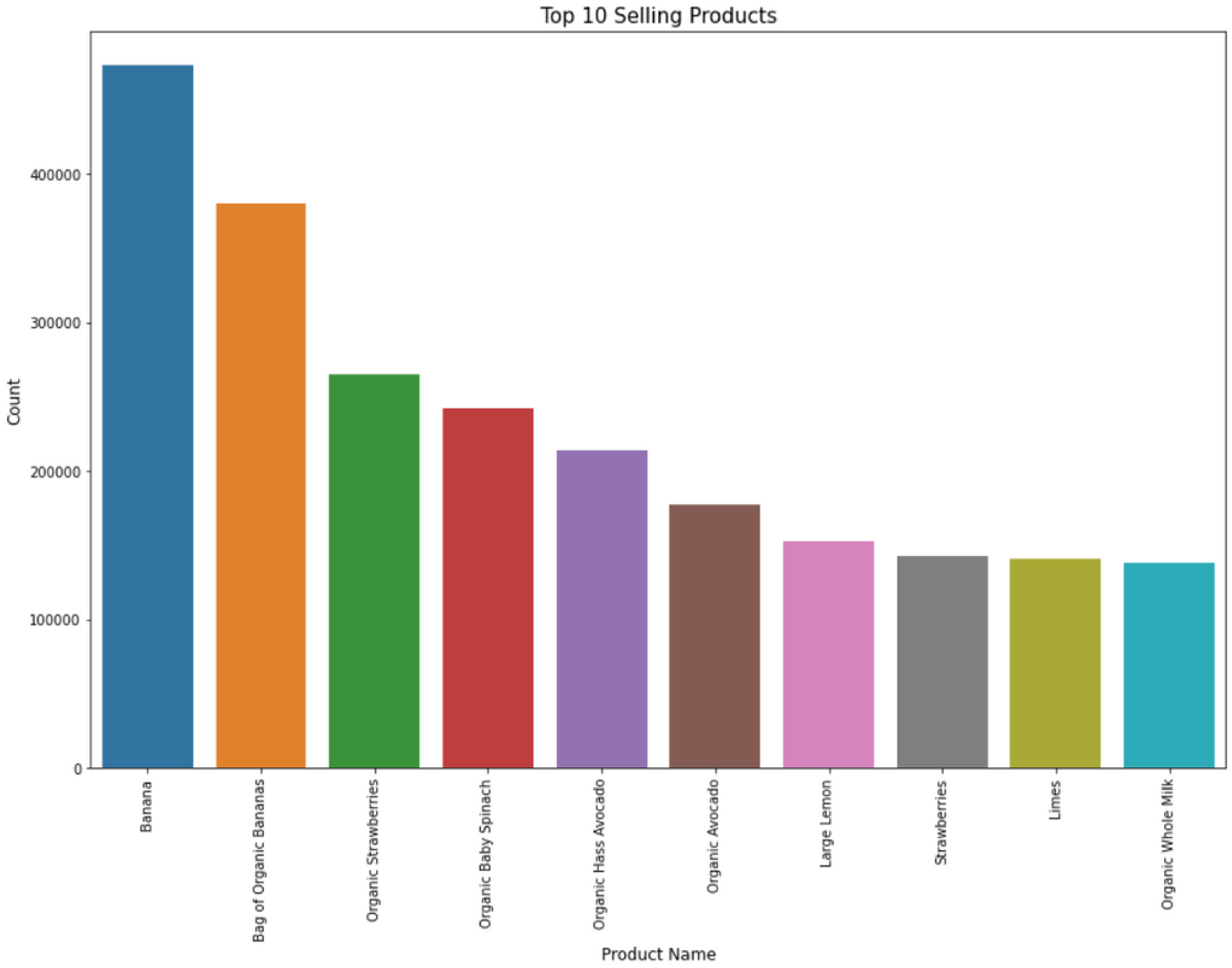
How many times user have reorder the same item?

It was observed most of the time user reorder the same item.



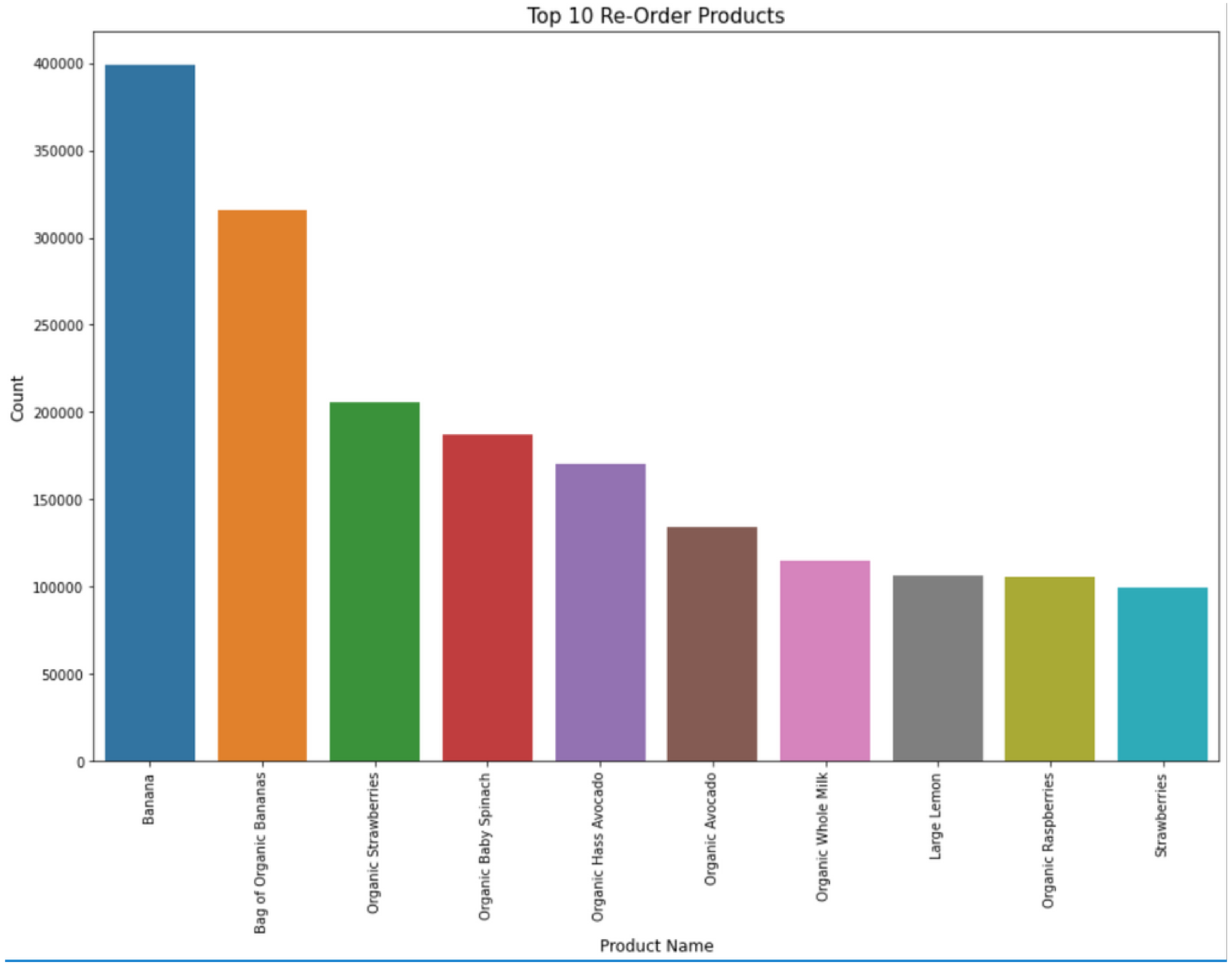
Top 10 selling and reordered items and also which item user added to the cart first

These are the top 10 selling products. We can clearly observe from the bar plot that the top most selling product is Banana and Organic Banana Chips.



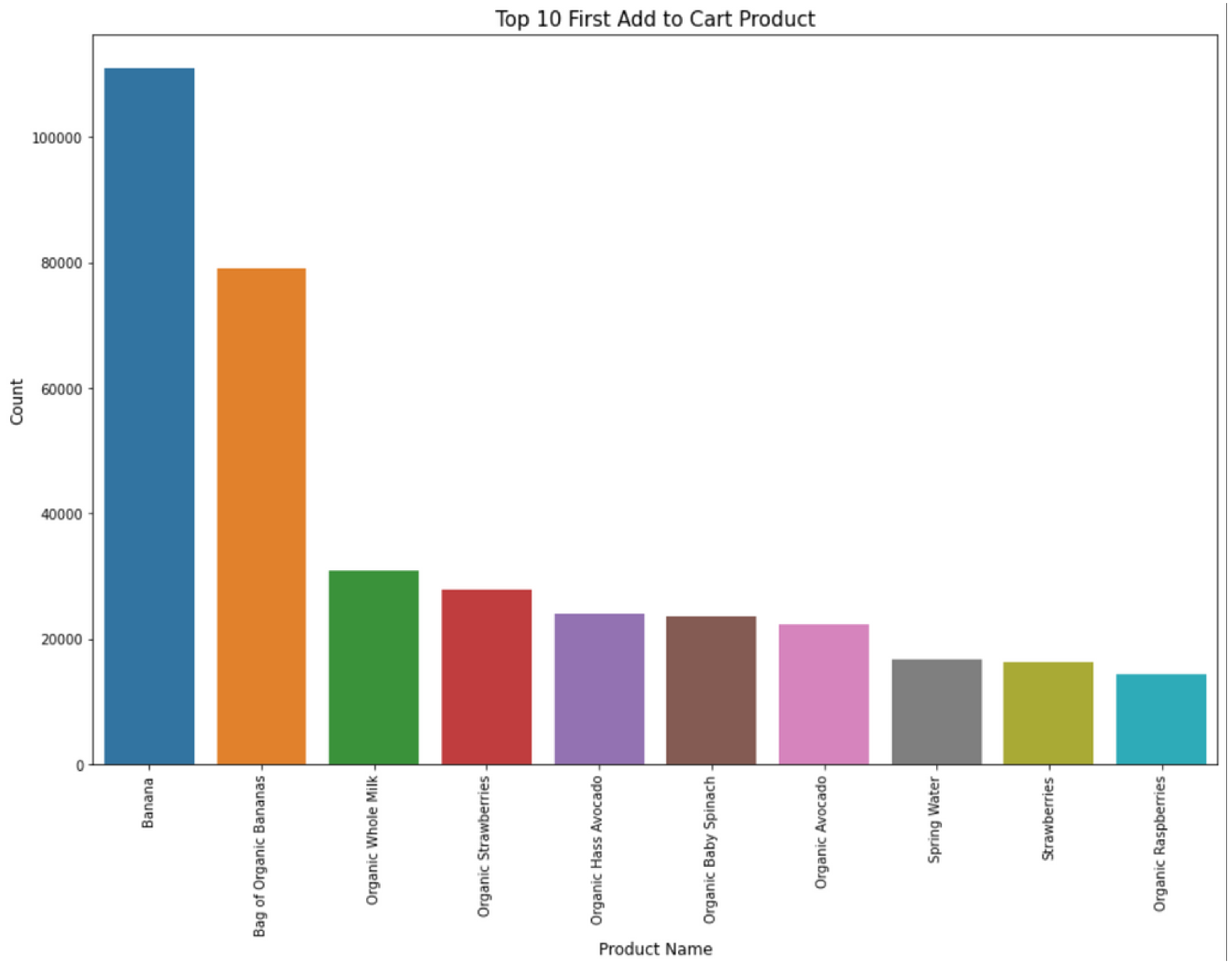
Top 10 Reordered products

There is high peak on banana and organic banana chips looks like that that is most loving product and reorder products among users.



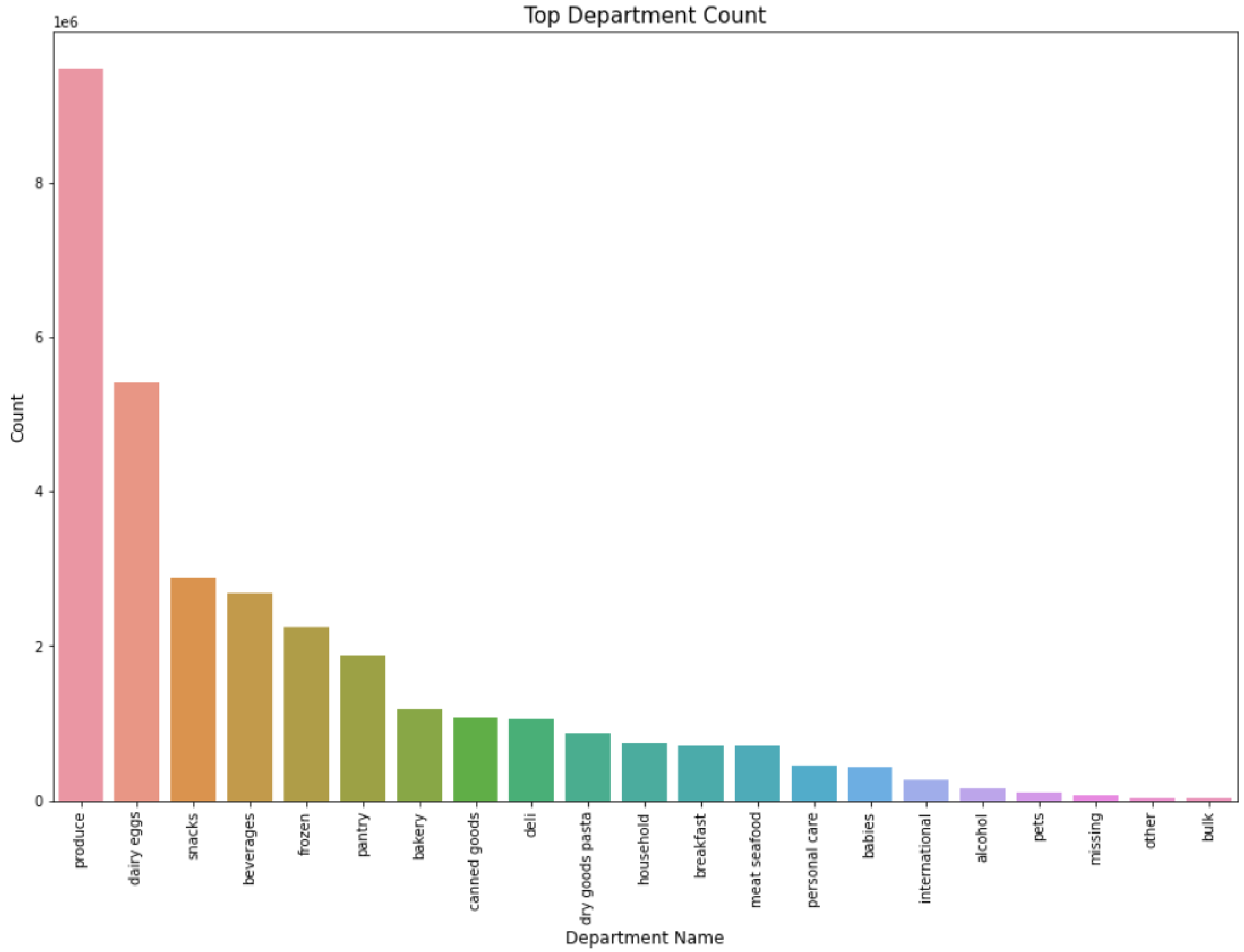
Top 10 First Add to Cart Product

Top selling, reordered and First product add to cart order product is Banana. From all this plot, the demand for produce product are high than any other Department.



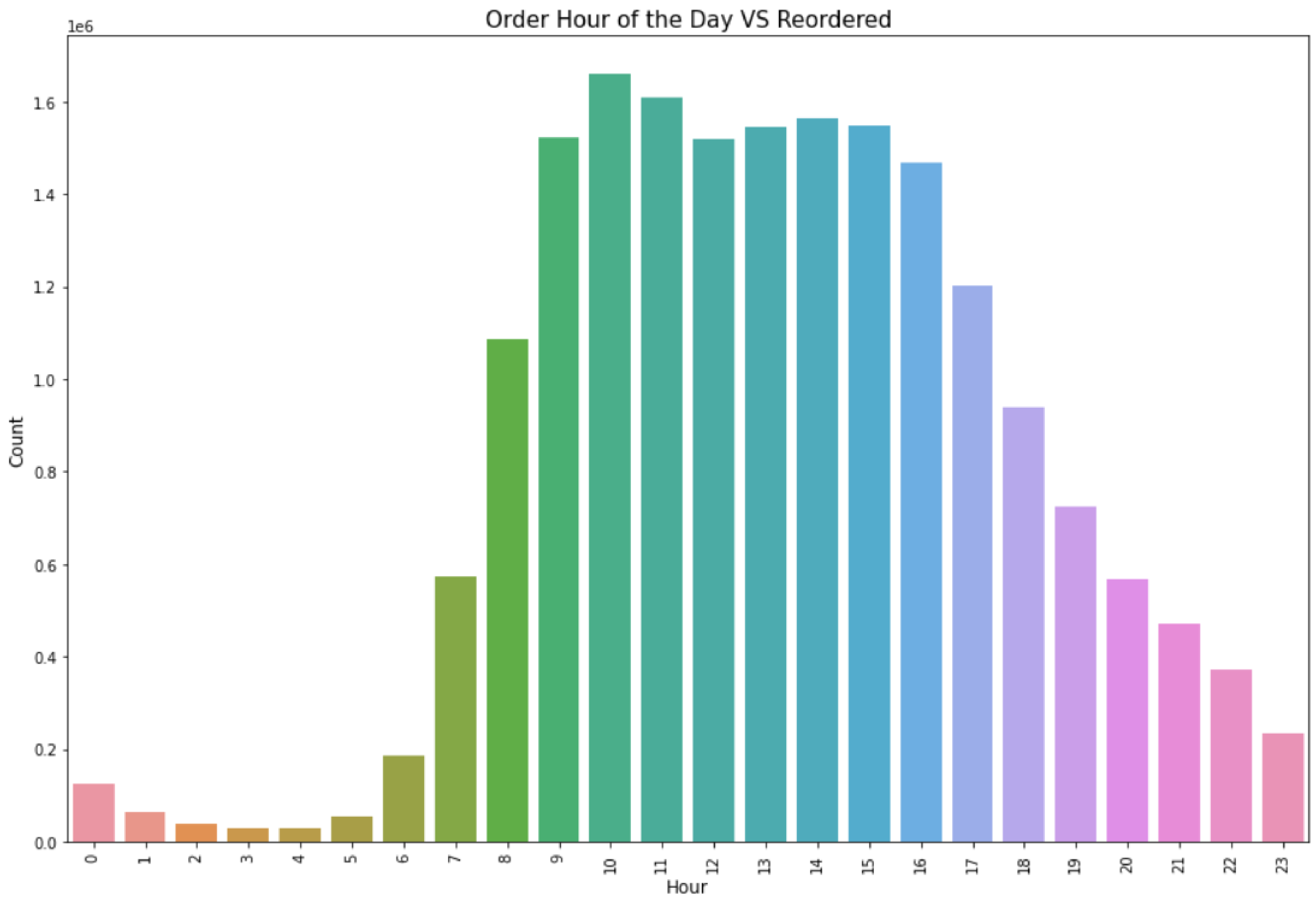
Let's look at the distribution of Top departments

As the First 2 Department are most important for any users because the Produce Department are about farm, crops and fruits and eggs are essential for the breakfast.



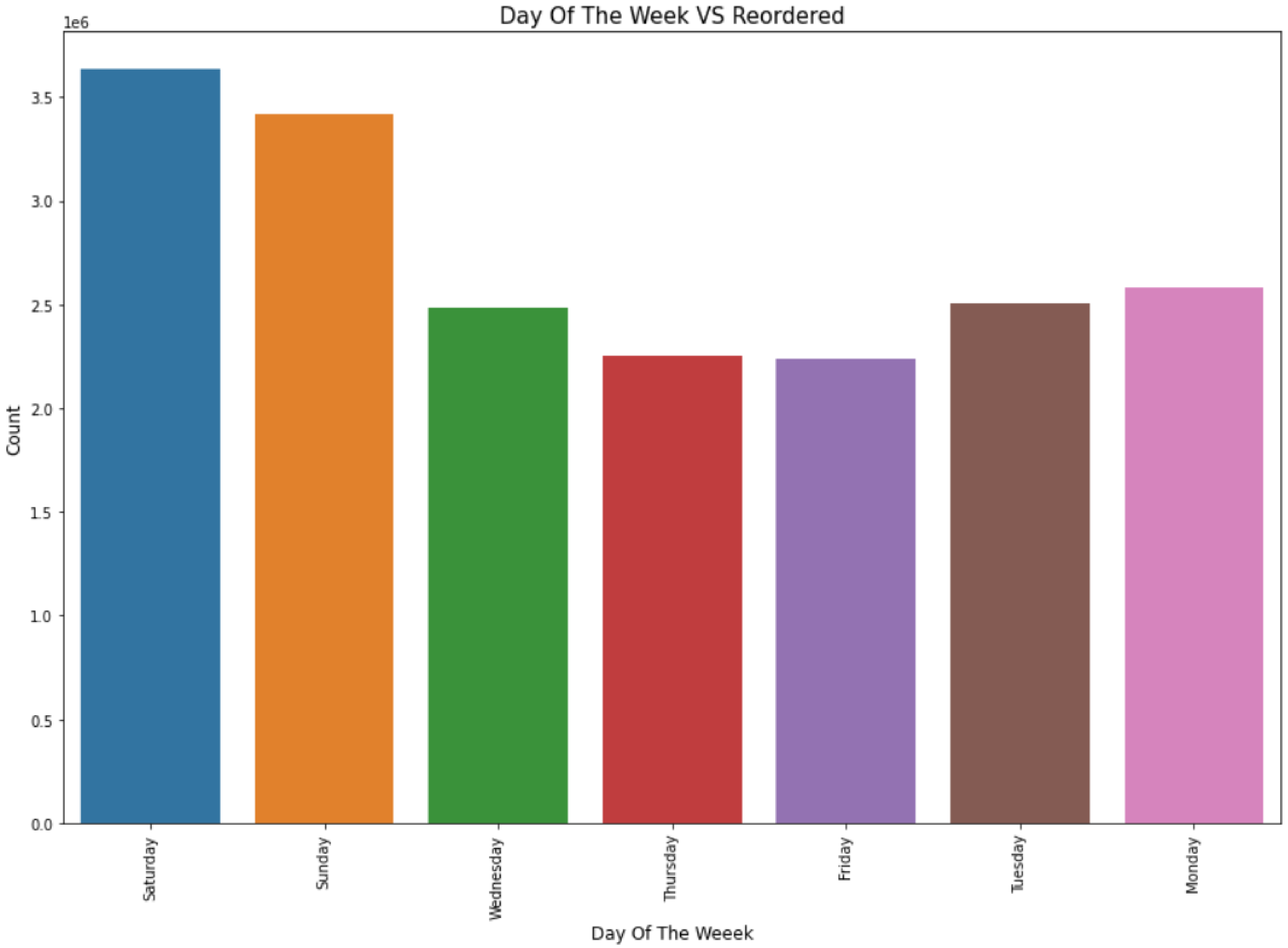
At what time user place the order again

Here I plot at what time most reordered products are placed. Most of the orders are placed between 8'0 clock in the morning and 5'0 clock in the evening.
This plot is almost same as Frequency of the order day. It shows user tends to order/reorder between this hours.



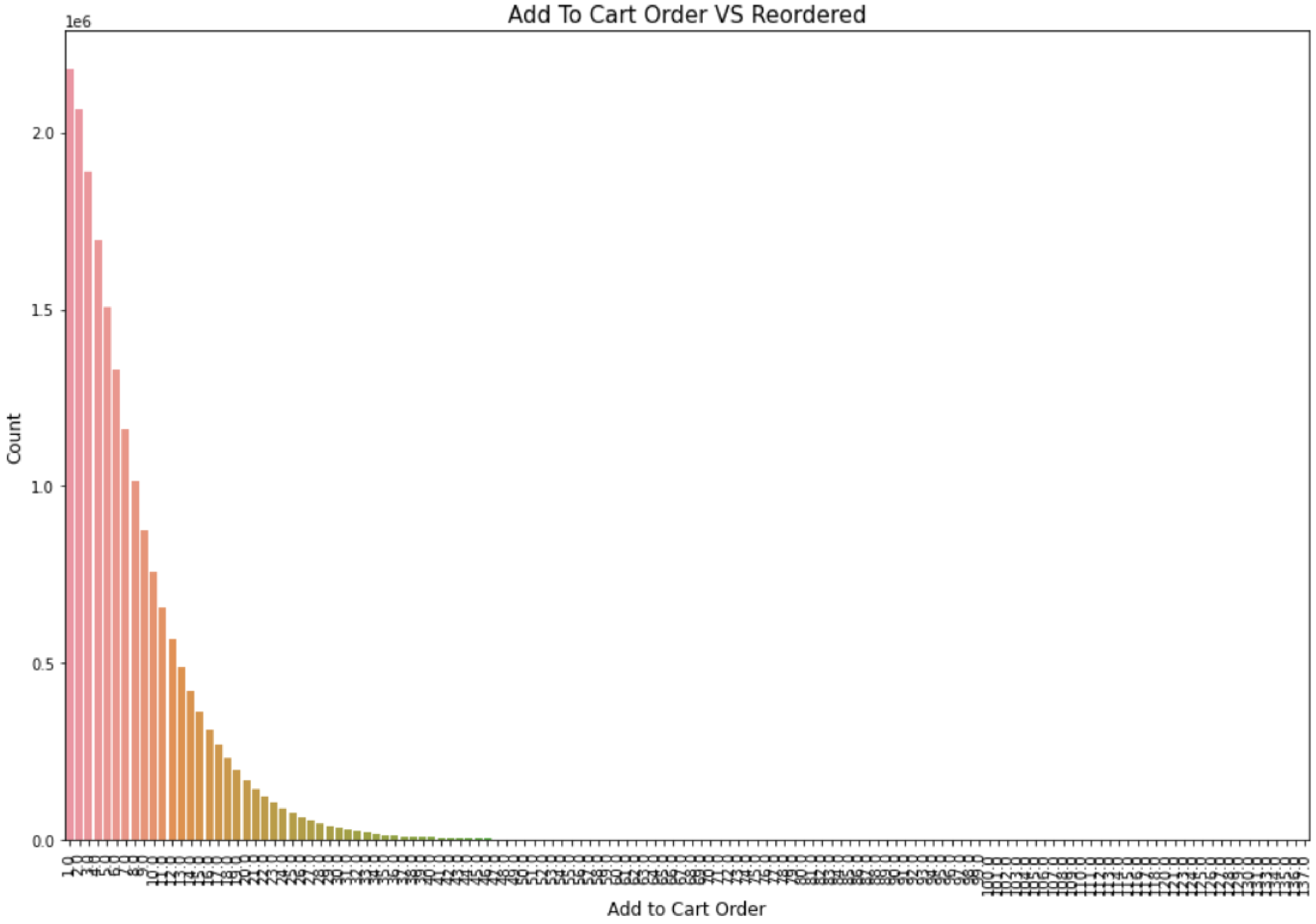
At what Day Most User place order again?

It was observed usually the people place the order again on weekends i.e on Saturday and Sunday.
The Order and Re-Order ration are almost same. The user tends to place order again on Saturday and Sunday. Saturday and Sunday have the high ratio of Order/Re-Order getting placed.



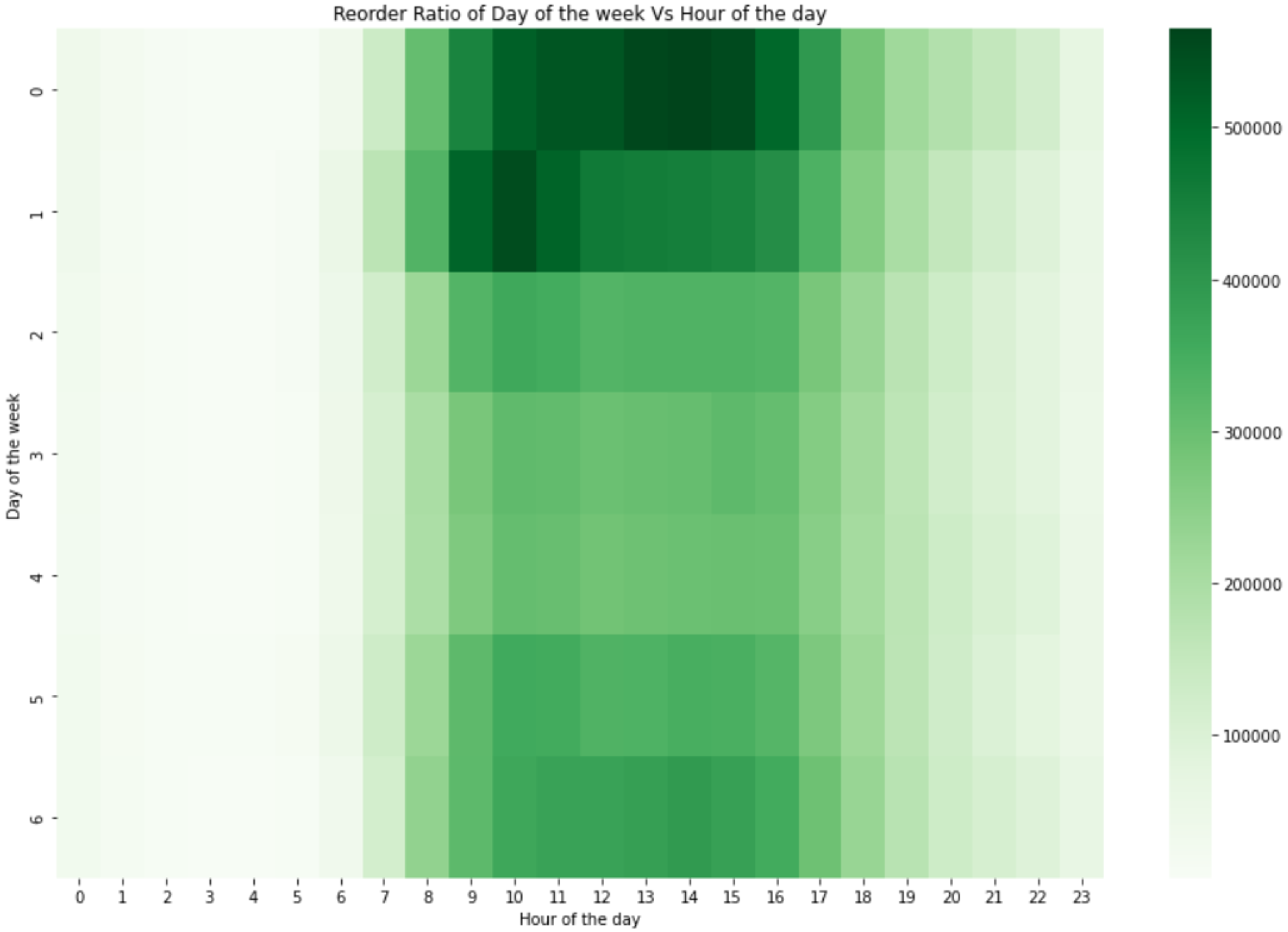
Add To Cart Order VS Reordered

The order which are added at 1th and 4th position in the cart have high chance of Re-Order by users



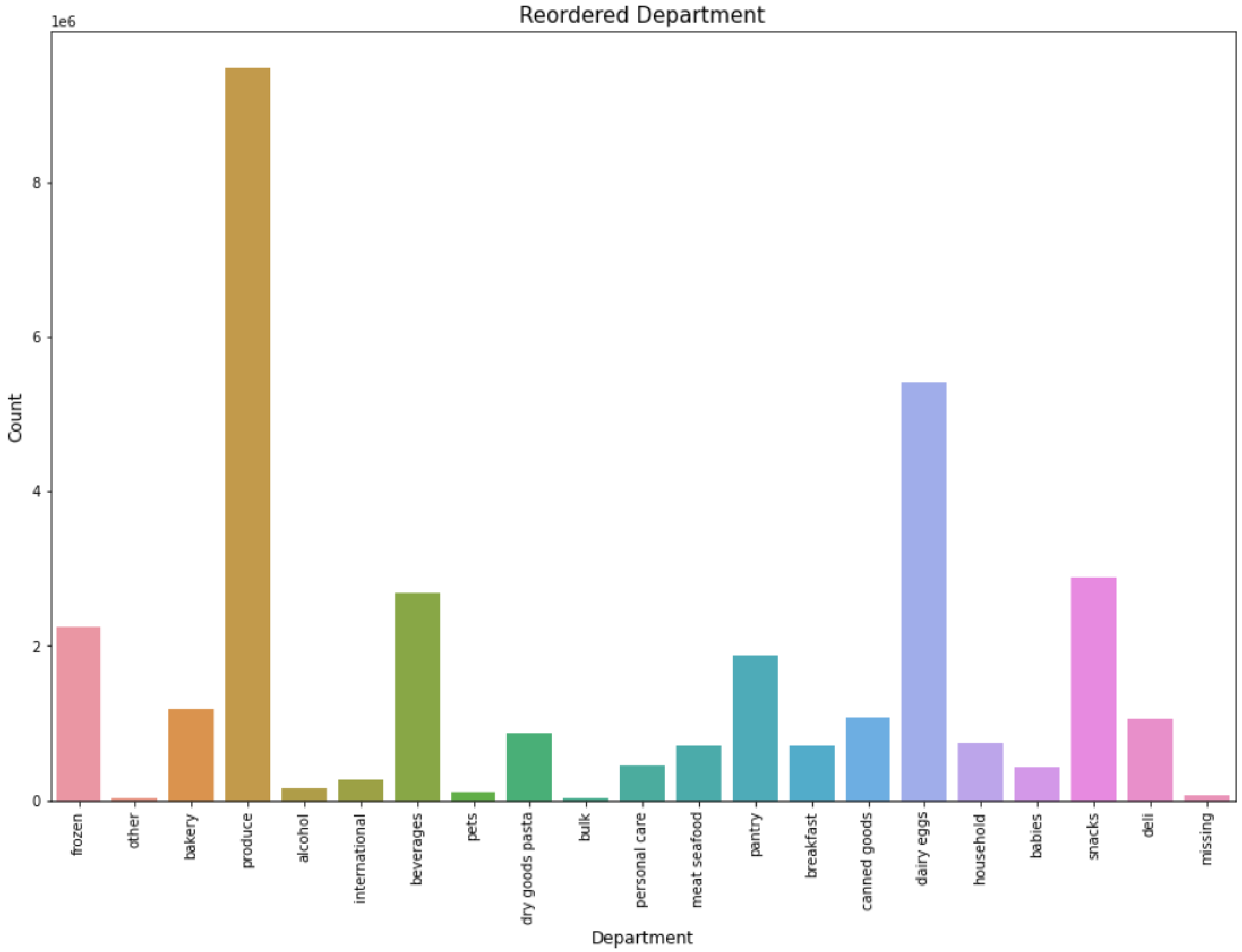
Reorder Ratio of Day of the Week Vs Hour of the Day

This Heatmap tells everything, Lot of users place order on Saturday and Sunday in between 10'clock to 15'clock.



Top Reordered Department

Most Users reorder from Produce and Dairy Eggs Department as they are essential for every day that's why they have high peak than any other department.



Feature Engineering

1. user features
2. product features
3. user product interaction features

Users features

1. Taking the maximum of the order numbers placed by each user .
2. Average number of products bought in each orders.
3. Day of the week the users orders the most.
4. Hour of the day the user has placed most of the orders.
5. Reordered ratio of each user.
6. Average days since prior order.
7. Total items bought by user.
8. Merging all the created features into the users dataset.

Product Features

1. Number of times the product has been purchased by the users

2. Reorder ratio of each products. .

3. Average add to cart order for each product.

4. Merging all the created features into the prd dataset.

User product interaction features

1. How many times a User has bought a product.

2. How many times a user bought a product after its first purchase.

3. Finding when the user has bought a product for the first time.

4. Merging all the created features into the uxp dataset.

5. How many times a customer bought a product on its last 5 orders.

6. product bought by users in the last_five orders.

7. Ratio of the products bought in the last_five orders.

Training and Testing

We split the data using the train_test_split into 70% Training and 30% Testing with the random state of 10.

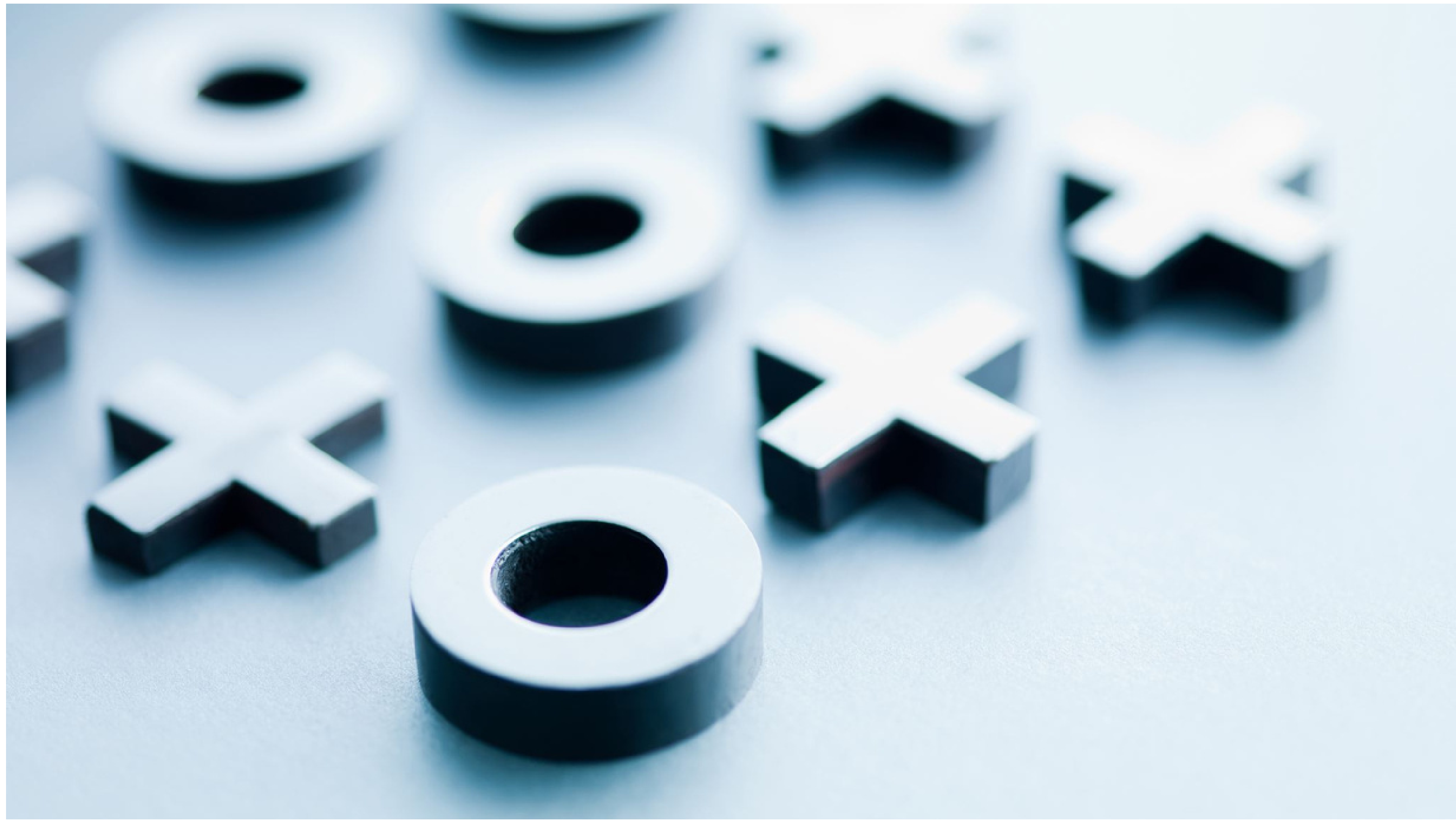
Model Building

The models used for Binary classification problems are :

1. Logistic Regression
2. Decision Trees
3. Random Forest Classifier
4. XGBoost

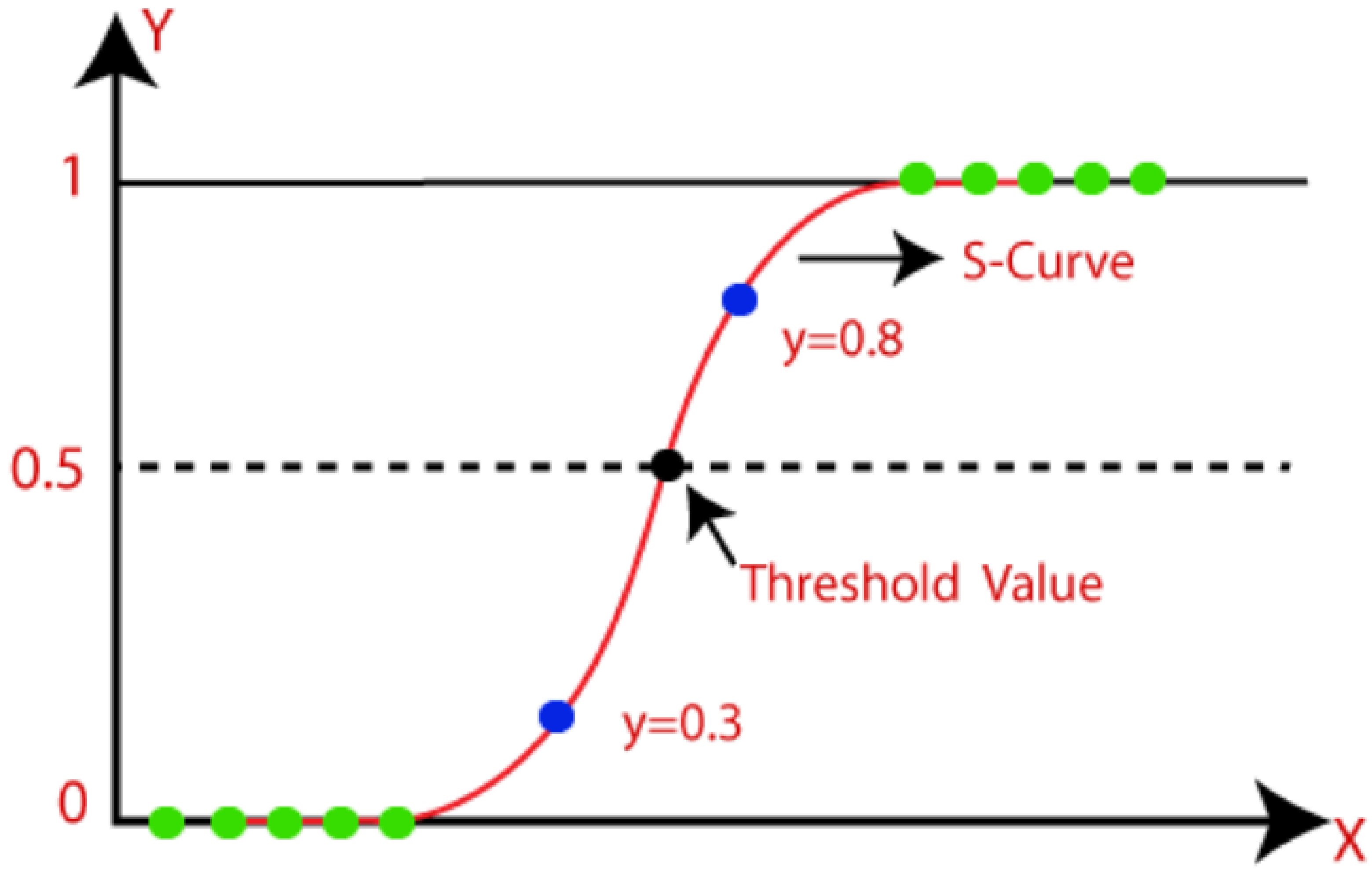
Threshold

In order to choose a threshold value Let's do some analysis by checking the range of threshold values (0.18,0.19,0.21).It was observed the threshold greater than 0.21 is giving good F1 score.



Logistic Regression

- Classification algorithm
- predicts the output of a categorical dependent variable(Yes/No).
- It gives the probabilistic values which lie between 0 and 1.
- It uses the sigmoid function

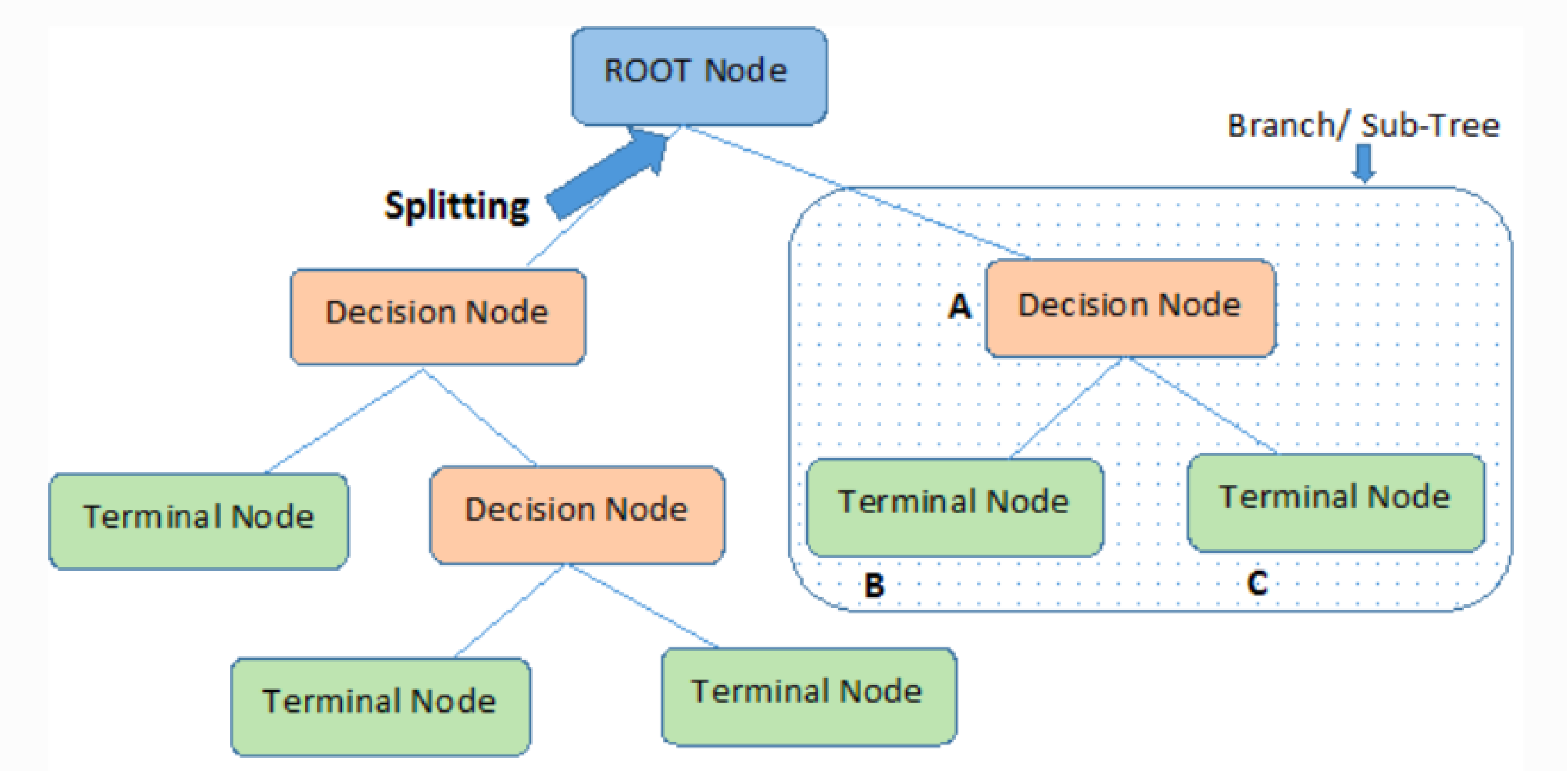


Results and Observations

- Accuracy: 0.75
- Precision for 0 class: 0.81
- Precision for 1 class: 0.36
- Recall for 0 class: 0.89
- Recall for 1 class: 0.23
- **F1 score for 0 class: 0.85**
- **F1 score for 1 class: 0.28**

Decision Trees

- Classification and Regression problems
- Predicting the categorical target variable
- Best parameter - RandomizedSearchCV

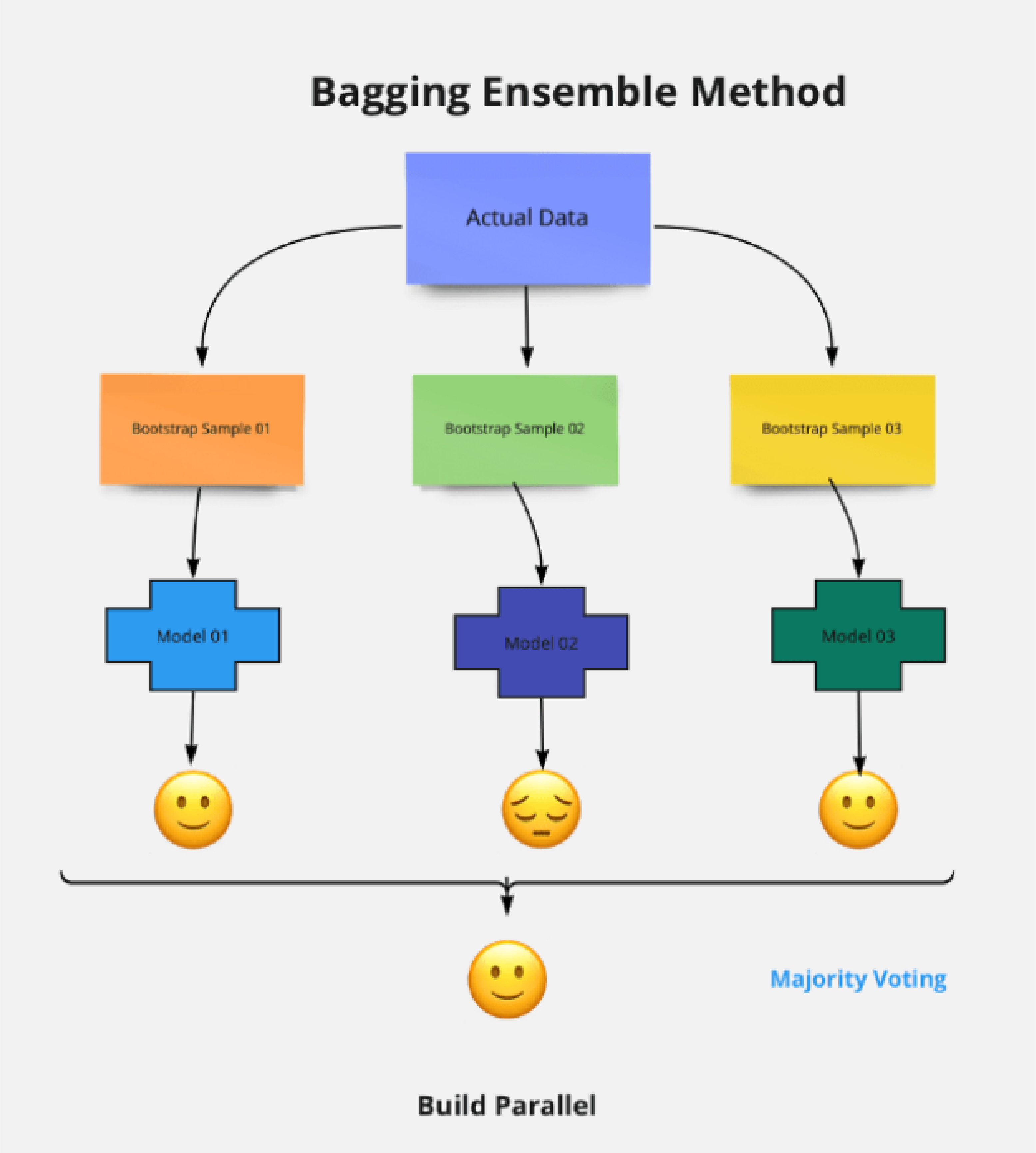


Results and Observations

- Accuracy: 0.83
- Precision for 0 class: 0.91
- Precision for 1 class: 0.34
- Recall for 0 class: 0.90
- Recall for 1 class: 0.37
- **F1 score for 0 class: 0.91**
- **F1 score for 1 class: 0.35**

Random Forest Classifier

- Classification and Regression Problems
- Predicting categorical and continuous variables
- Ensemble technique - Bagging
- CalibratedClassifierCV

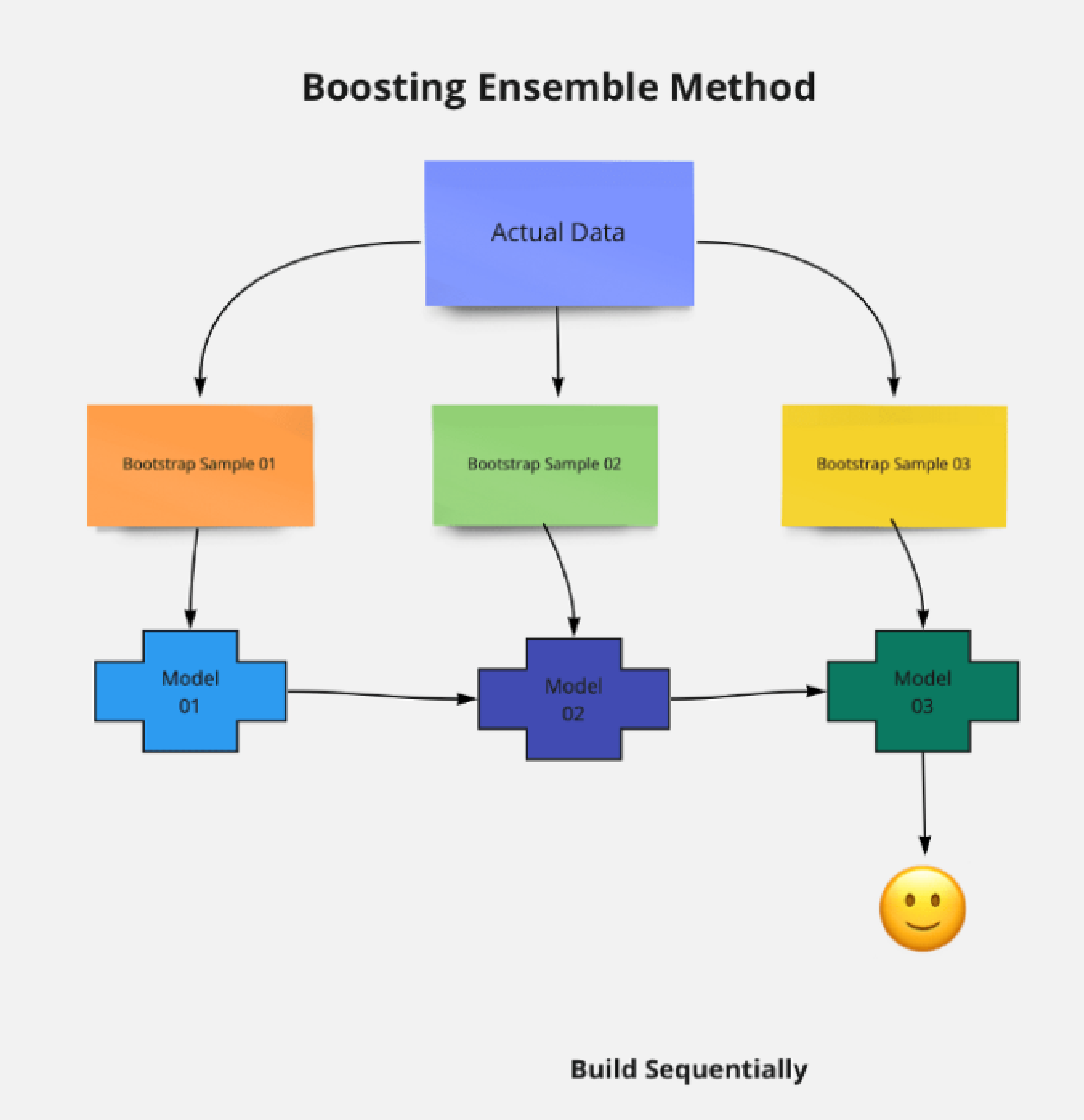


Results and Observations

- Accuracy: 0.83
- Precision for 0 class: 0.90
- Precision for 1 class: 0.39
- Recall for 0 class: 0.91
- Recall for 1 class: 0.38
- **F1 score for 0 class: 0.91**
- **F1 score for 1 class: 0.39**

XGBoost Classifier

- Classification and Regression Problems
- Extreme Gradient Boosting
- speed and Performance
- XGBClassifier



Boosting Ensemble Method

Build Sequentially

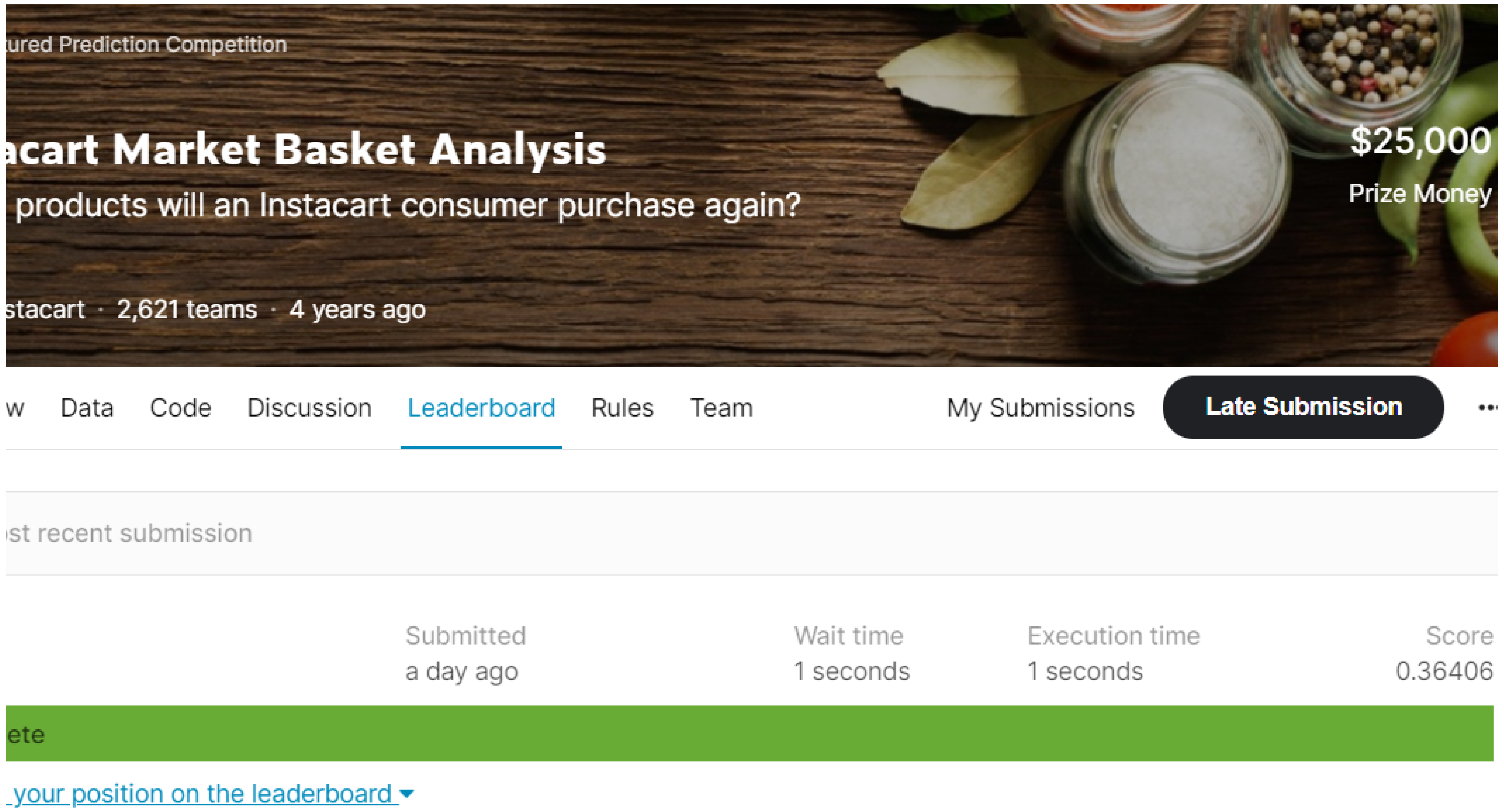
Results and Observations

- Accuracy: 0.80
- Precision for 0 class: 0.86
- Precision for 1 class: 0.46
- Recall for 0 class: 0.91
- Recall for 1 class: 0.34
- **F1 score for 0 class: 0.89**
- **F1 score for 1 class: 0.39**

Conclusions

Best Performer - XGBoost & Random Forest

Worst Performer - Logistic Regression



ured Prediction Competition

acart Market Basket Analysis

products will an Instacart consumer purchase again?

stacart · 2,621 teams · 4 years ago

st recent submission

Submitted	Wait time	Execution time	Score
a day ago	1 seconds	1 seconds	0.36406

ete

your position on the leaderboard ▾