Instacart Market Basket Analysis

Aim: Which products will an Instacart consumer purchase again?

About: Instacart, a grocery ordering and delivery app, aims to make it easy to fill refrigerator and pantry with user’s personal favorites and staples when they need them. After selecting products through the Instacart app, personal shoppers review user’s orders and do the in-store shopping and delivery. The aim of the project assigned is to use this anonymized data on customer orders over time to predict which previously purchased products will be in a user’s next order.

[Exploratory Analysis - Instacart](https://www.kaggle.com/philippsp/exploratory-analysis-instacart)

ABOUT DATA:

**Orders.CSV**

This file contain all the orders in the dataset( 1 row/order). In this data we can clearly see that for user 1 we have 11 orders(10 for train and 1 for prior). Though it is not showing what actual orders is to be ordered.

kable(head(orders,3))

| | order\_id| user\_id|eval\_set | order\_number| order\_dow| order\_hour\_of\_day| days\_since\_prior\_order|

|:-------|--------:|-------:|:--------|------------:|---------:|-----------------:|----------------------:|

|416562 | 418705| 25088|prior | 3| 0| 12| 17|

|988438 | 3244398| 59411|prior | 19| 6| 14| 5|

|2269208 | 1889551| 136638|prior | 32| 1| 17| 12|

|50977 | 1574004| 3178|prior | 2| 1| 8| 10|

Detail of orders:

data.frame': 300000 obs. of 7 variables:

$ order\_id : int 418705 3244398 1889551 1574004 145096 392413 1045366 851938

$ user\_id : int 25088 59411 136638 3178 62823 98329 11361 106510 49790 141250 ...

$ eval\_set : Factor w/ 3 levels "prior","test",..: 1 1 1 1 1 1 1 1 1 1 ...

$ order\_number : int 3 19 32 2 19 3 5 34 3 15 ...

$ order\_dow : int 0 6 1 1 6 5 2 4 3 2 ...

$ order\_hour\_of\_day : int 12 14 17 8 18 2 13 15 15 11 ...

$ days\_since\_prior\_order: num 17 5 12 10 6 30 8 3 30 14 ...

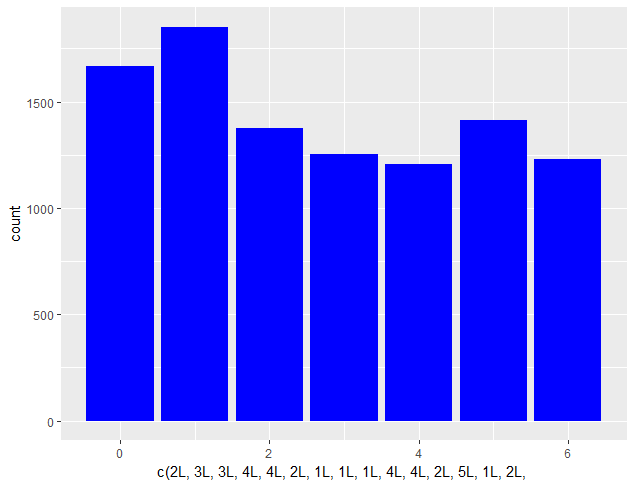
orders (3.4m rows, 206k users):

* order\_id: order identifier
* user\_id: customer identifier
* eval\_set: which evaluation set this order belongs in (see SET described below)
* order\_number: the order sequence number for this user (1 = first, n = nth)
* order\_dow: the day of the week the order was placed on
* order\_hour\_of\_day: the hour of the day the order was placed on
* days\_since\_prior: days since the last order, capped at 30 (with NAs for order\_number = 1)

visualization

histogram for order\_dow

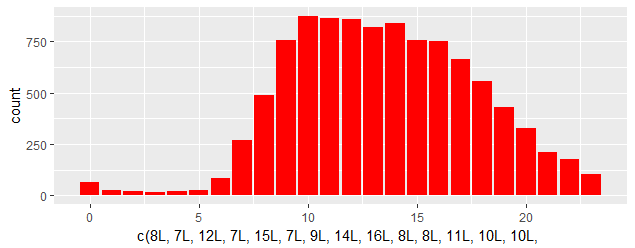
ggplot(orders,aes(x=order\_dow)) + geom\_histogram(stat="count",fill="blue")



There is a clear effect of day of the week. Most orders are on days 0 and 1.Unfortunately there is no info regarding which values represent which day, but one would assume that this is the weekend.

Histogram for hour\_of\_day

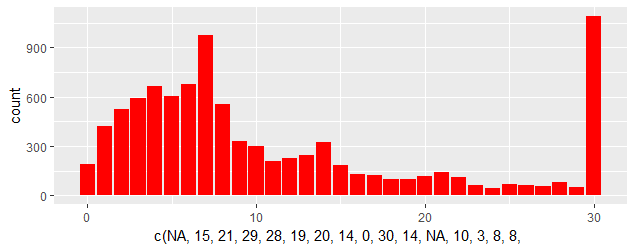
ggplot(orders ,aes(x=order\_hour\_of\_day)) + geom\_histogram(stat="count",fill="red")



We can clearly see most orders are between 8.00-18.00.

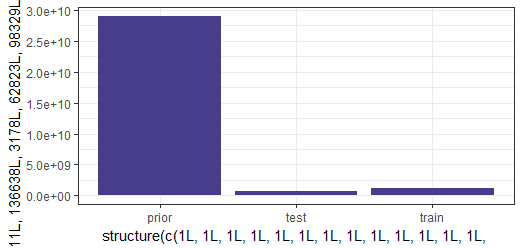
days\_since\_prior\_order histogram:

ggplot(orders,aes(x=days\_since\_prior\_order)) + geom\_histogram(stat="count",fill="red")



People seem to order more often after exactly 1 week.

order\_eval\_set bar plot



We can clearly see that most of the eval\_set values of prior and least of them belongs to test.

**aisles.CSV**

kable(head(aisles,5))

|  |
| --- |
| | aisle\_id|aisle |  |--------:|:--------------------------|  | 1|prepared soups salads |  | 2|specialty cheeses |  | 3|energy granola bars |  | 4|instant foods |  | 5|marinades meat preparation | |
|  |
| |  | | --- | |  | |

str(aisles)

'data.frame': 134 obs. of 2 variables:

$ aisle\_id: int 1 2 3 4 5 6 7 8 9 10 ...

$ aisle : Factor w/ 134 levels "air fresheners candles",..: 107 121 43 75 82 92 95 6 101 77

aisles (134 rows):

* aisle\_id: aisle identifier
* aisle: the name of the aisle

**departments.CSV**

kable(head(departments,4))

| department\_id|department |

|-------------:|:----------|

| 1|frozen |

| 2|other |

| 3|bakery |

| 4|produce |

Str(departments)

data.frame': 21 obs. of 2 variables:

$ department\_id: int 1 2 3 4 5 6 7 8 9 10 ...

$ department : Factor w/ 21 levels "alcohol","babies",..: 11 16 3 20 1 13 4 19 10 6 ...

deptartments (21 rows):

* department\_id: department identifier
* department: the name of the department

**Order\_prior.CSV**

kable(head(order\_prior,5))

| | order\_id| product\_id| add\_to\_cart\_order| reordered|

|:--------|--------:|----------:|-----------------:|---------:|

|9682007 | 1022154| 17317| 10| 0|

|5519877 | 582621| 47209| 2| 1|

|17750258 | 1872049| 24933| 7| 1|

|17778641 | 1875077| 31952| 22| 0|

|2688225 | 283558| 48679| 17| 1|

|  |
| --- |
| str(order\_prior)  data.frame': 1000000 obs. of 4 variables:  $ order\_id : int 1022154 582621 1872049 1875077 283558 2090805 1781362 1668699 1242080  $ product\_id : int 17317 47209 24933 31952 48679 5479 17600 47516 260 26209 ...  $ add\_to\_cart\_order: int 10 2 7 22 17 6 10 20 7 9 ...  $ reordered : int 0 1 1 0 1 1 1 0 0 1 ...  order\_products\_\_SET :   * order\_id: foreign key * product\_id: foreign key * add\_to\_cart\_order: order in which each product was added to cart * reordered: 1 if this product has been ordered by this user in the past, 0 otherwise   where SET is one of the four following evaluation sets (eval\_set in orders):   * "prior": orders prior to that users most recent order (~3.2m orders) * "train": training data supplied to participants (~131k orders) * "test": test data reserved for machine learning competitions (~75k orders) |
| Visualization for orders\_prior and orders\_train  CAN WE SEE HOW MANY PRODUCT DO USERS BUY ????  Ok,lets try for it  ##orders\_prior  x=summarize(group\_by(order\_prior,order\_id),  n\_items = last(add\_to\_cart\_order))  ggplot(x,aes(x=n\_items))+  geom\_histogram(stat="count",fill="red") +  geom\_rug() +  coord\_cartesian(xlim=c(0,80))  C:\Users\mdsha\OneDrive\Documents\R\win-library\3.3\readr\instacart_data\order_prior_histogram.png |
| |  | | --- | |  | |

**Order\_train.CSV**

kable(head(order\_train,5))

| | order\_id| product\_id| add\_to\_cart\_order| reordered|

|:-------|--------:|----------:|-----------------:|---------:|

|636132 | 1566529| 36854| 4| 0|

|1227091 | 3032001| 3262| 3| 1|

|225427 | 551676| 4462| 5| 0|

|1277685 | 3156638| 39121| 10| 1|

|270344 | 658135| 15290| 10| 0|

str(order\_train)

|  |
| --- |
| data.frame': 100000 obs. of 4 variables:  $ order\_id : int 1566529 3032001 551676 3156638 658135 1142439 1476639 203564 358668 37689  $ product\_id : int 36854 3262 4462 39121 15290 1181 36374 33245 27966 47518 ...  $ add\_to\_cart\_order: int 4 3 5 10 10 4 16 5 6 17 ...  $ reordered : int 0 1 0 1 0 0 0 0 1 0 ... |
|  |
| https://www.kaggle.io/svf/1295774/d3fe903c47938a466a38f223e9abdf0c/__results___files/figure-html/unnamed-chunk-14-1.png   |  | | --- | | In both the histograms we can easily observe most users are near to 5. | |

**products.CSV**

kable(head(products,5))

|  |
| --- |
| | | product\_id|product\_name | aisle\_id| department\_id|  |:-----|----------:|:----------------------------------|--------:|-------------:|  |30062 | 30062|White Select-A-Size Paper Towels | 54| 17|  |23963 | 23963|Organic Shrimp & Crab Grill & Boil | 104| 13|  |11071 | 11071|MATZO BALL MIX | 33| 6|  |49087 | 49087|Hard Candy Awesome Reds | 45| 19|  |42678 | 42678|Multigrain Corn Tortilla Chips | 107| 19| |
|  |
| |  | | --- | | str(products) | |
| data.frame': 5000 obs. of 4 variables:  $ product\_id : int 30062 23963 11071 49087 42678 15148 10871 23781 5123 44883 ...  $ product\_name : Factor w/ 49688 levels "'Swingtop' Premium Lager",..: 48518 31709 24629 19281 26544  $ aisle\_id : int 54 104 33 45 107 54 50 134 106 105 ...  $ department\_id: int 17 13 6 19 19 17 19 5 12 13 ... |
|  |
| |  | | --- | |  | |

products (50k rows):

* product\_id: product identifier
* product\_name: name of the product
* aisle\_id: foreign key
* department\_id: foreign key

which product purchased by the user most?

x=group\_by(order\_train,product\_id)

y=summarize(x,count = n())

top\_products = top\_n(y,10, wt = count) %>%

left\_join(select(products,product\_id,product\_name),by="product\_id") %>%

arrange(desc(count))

kable(top\_products)

| product\_id| count|product\_name |

|----------:|-----:|:----------------------|

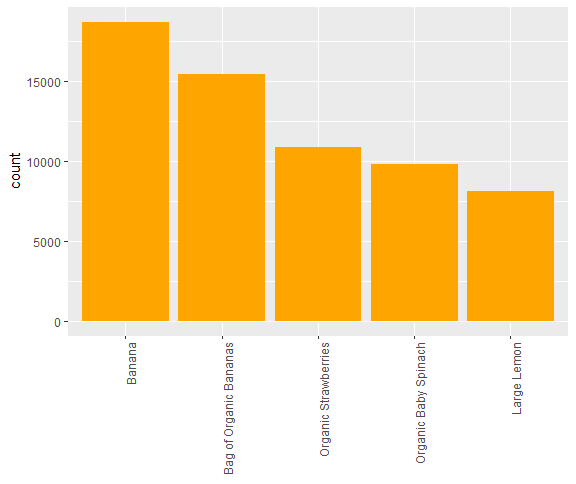
| 24852| 18726|Banana |

| 13176| 15480|Bag of Organic Bananas |

| 21137| 10894|Organic Strawberries |

Banana is the most purchased product by the user,bag of organic gains second position.

Lets plot a bar graph for the top 5 products



Which product will the user pick into the the cart first..???

x=group\_by(order\_train ,product\_id, add\_to\_cart\_order)

y=summarize(x,count = n())

z=mutate(y,count\_rate=count/sum(count))

p=filter(z,add\_to\_cart\_order == 1, count>10)

q=arrange(p,desc(count\_rate))

cart\_first =left\_join(q,products,by="product\_id") %>%

select(product\_name, count\_rate, count) %>%

ungroup() %>%

top\_n(10, wt=count\_rate)

| product\_id|product\_name | count\_rate| count|

|----------:|:-----------------------------------------------|----------:|-----:|

| 45004|White Multifold Towels | 0.6610169| 39|

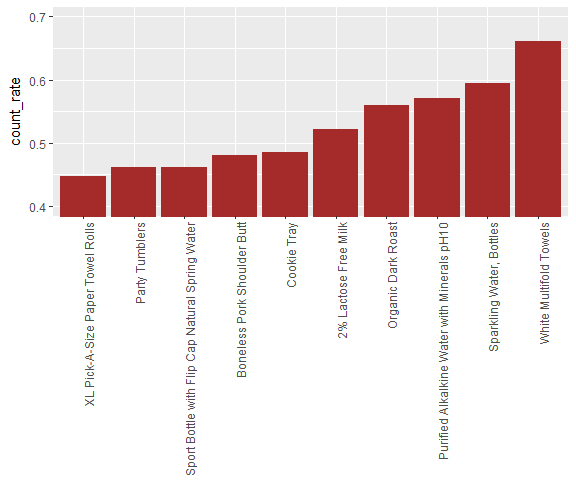
| 11885|Sparkling Water, Bottles | 0.5942029| 41|

| 13128|Purified Alkalkine Water with Minerals pH10 | 0.5714286| 12|

| 4100|Organic Dark Roast | 0.5600000| 14|

Clearly white Multifold Towels would be the user’s first choice.

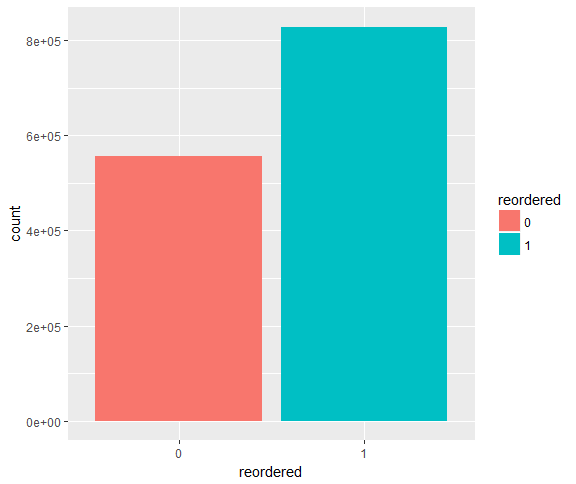
Lets plot the bar graph for the above data.



How frequently a user go for the same product?

|  |  |  |
| --- | --- | --- |
| x=group\_by(order\_train,reordered) y=summarize(x,count = n())  same\_product <- mutate(y,reordered = as.factor(reordered)) %>%  mutate(proportion = count/sum(count))  kable(same\_product)   |  |  | | --- | --- | | |  | | --- | |  | |   |reordered | count| proportion|  |:---------|------:|----------:|  |0 | 555793| 0.4014056|  |1 | 828824| 0.5985944| |
|  |

Bar plot for reordered



**Most frequently reordered product**

x=group\_by(order\_train,product\_id)

y=summarize(x,proportion\_reordered = mean(reordered), n=n())

most\_reordered = filter(y,n>40) %>%

top\_n(10,wt=proportion\_reordered) %>%

arrange(desc(proportion\_reordered)) %>%

left\_join(products,by="product\_id")

(most\_reordered)

| product\_id| reordered\_ratio| n|product\_name | aisle\_id| department\_id|

|----------:|---------------:|-----:|:-----------------------|--------:|-------------:|

| 1729| 0.9347826| 92|2% Lactose Free Milk | 84| 16|

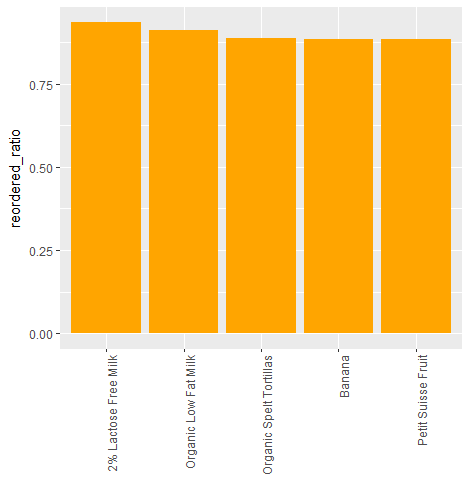
| 20940| 0.9130435| 368|Organic Low Fat Milk | 84| 16|

| 21038| 0.8888889| 81|Organic Spelt Tortillas | 128| 3|

| 24852| 0.8841717| 18726|Banana | 24| 4|

| 117| 0.8833333| 120|Petit Suisse Fruit | 2| 16|

Lets try to plot the bar\_plot for the product having highest probability of being reordered.



We can observe from this plot that 2% lactose free milk having the highest probability of being reordered.

### Link between time of last order and probability of reorder

order\_train %>%

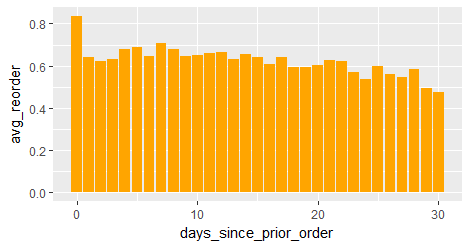
left\_join(orders,by="order\_id") %>%

group\_by(days\_since\_prior\_order) %>%

summarize(avg\_reorder = mean(reordered)) %>%

ggplot(aes(x=days\_since\_prior\_order,y=avg\_reorder))+

geom\_bar(stat="identity",fill="orange")

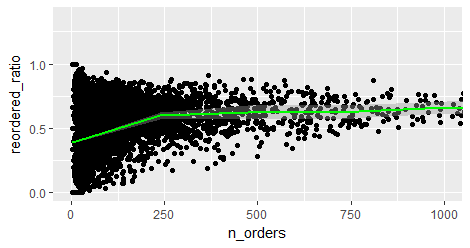


This plot is implying if the customer reorder a product on the same day,he will reorder the same product more frequently.

### How number of orders and probability of reordering are relating?

Lets see,

Smooth plot



In this plot,product with high number of orders is more likely to be reordered.

### Organic vs Non-organic Ratio

### prod\_cat <- products %>%

### mutate(organic=ifelse(str\_detect(str\_to\_lower(products$product\_name),

### 'organic'),"organic","not organic"), organic= as.factor(organic))

### or\_nor <- order\_train %>%

### left\_join(prod\_cat, by="product\_id") %>%

### group\_by(organic) %>%

### summarize(count = n()) %>%

### mutate(proportion = count/sum(count))

### kable(or\_nor)

|organic | count| proportion|

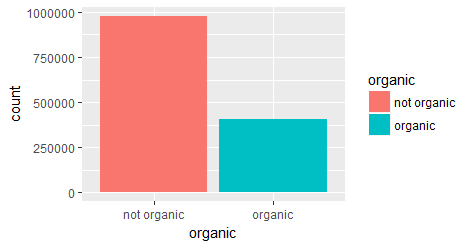
|:-----------|------:|----------:|

|not organic | 979000| 0.7070547|

|organic | 405617| 0.2929453|

Almost 70% products are not organic.

Lets plot the bar\_plot:



### Reordering Organic vs Non-Organic

### r\_or\_nor <- order\_train %>% left\_join(products,by="product\_id") %>% group\_by(organic) %>% summarize(avg\_reordered = mean(reordered))

### kable(r\_or\_nor)

|organic | avg\_reordered|

|:-----------|-------------:|

|not organic | 0.5784985|

|organic | 0.6470981|

### Lets plot the bar graph for the above stat

### C:\Users\mdsha\OneDrive\Documents\R\win-library\3.3\readr\instacart_data\org_non_organic_reordering.png

### Data Preparation:

### After finding all the insights from the exploration part, I move towards Data Preparation. I upload all the data files into the R. I start merging aisles.csv(2 varaibles) and products.csv(4 varaibles) by “aisles\_id”.In result I get a table having 5 varaibles. The resulting table is then merged with the department.csv(2 var) by “department\_id” ,which gives a table product\_new comprising 4 variables after excluding the “aisles\_id” and “department\_id” columns.

### aisles.csv & department.csv

### C:\Users\mdsha\Desktop\project\pics\products.PNG C:\Users\mdsha\Desktop\project\pics\department.PNG

### Products.csv

### C:\Users\mdsha\Desktop\project\pics\aisles.PNG

### Products\_new

### C:\Users\mdsha\Desktop\project\pics\products_new.PNG

### Due to the lack of primary memory I regularly drop the tables which are not in further use. So, I drop the tables aisles.csv and department.csv as their necessary columns have fetched into the new table product\_new tables.

### I create a new variable ‘user\_id’ in order\_train(4 var) table by fetch the entries which are exactly same in order\_train and orders dataset using match() function.Now updated order\_train consist 5 variables.

### Order\_train(4 var) and updated order\_train(5 var)

### C:\Users\mdsha\Desktop\project\pics\order_train.PNG C:\Users\mdsha\Desktop\project\pics\updated_order_train.PNG

### Then I joined order dataset(7 var) and order\_prior dataset(4 var) by = "order\_id" into new table called as order\_detail(10 var).Then I removed order\_prior table.

### Orders and oder\_prior dataset.

### C:\Users\mdsha\Desktop\project\pics\orders.PNG C:\Users\mdsha\Desktop\project\pics\order_prior.PNG

### Order\_detail dataset(obtained after merging order and order\_prior)

### C:\Users\mdsha\Desktop\project\pics\order_detail.PNG

### Then I organize the order\_detail data by ‘user\_id’, ‘order\_number’, ‘product\_id’ column through arrange() function. This result is further grouped by ‘user\_id’, ‘product\_id’ columns. Using this grouped data I created a new column ‘product\_occurance’ in which each entry will tell the total occurances of a particular order being placed.All the changes are taking place into product\_detail dataset, it means this dataset is now comprising 11 variables.

### Product\_detail(comprising 11 varaibles after 1st update)

### C:\Users\mdsha\Desktop\project\pics\product_detail_1st_update.PNG

### Then the data is ungrouped using ungroup() function. The same data is further grouped by ‘product\_id’ and the resulting data is placed into the summarize function where I create 4 new columns. Now the resulting product\_detail consist 5 var(1 product\_id,4 new columns).

### Product\_detail(comprising 5 varaibles after 2nd update)

### C:\Users\mdsha\Desktop\project\pics\product_detail_2nd_update.PNG

### Product\_detail (comprising 8 varaibles after 3rd update)

### C:\Users\mdsha\Desktop\project\pics\product_detail_ration_3rd_updatePNG.PNG

### Then I create few probability and ratio columns in product\_detail (8 var) and deselect those columns which are not in further use. Now product\_detail consisting 5 variables..

### Product\_detail(comprising 5 varaibles after final update)

### C:\Users\mdsha\Desktop\project\pics\product_detail_final_update.PNG

### Now I pick the order dataset(7 var) and filters out the observations depending on the ‘prior’ entry of eval\_set column and then grouped the filtered data using ‘user\_id’,in result we get a new table called users\_summary.

### Users\_summary(comprising 7 var after 1st update)

### C:\Users\mdsha\Desktop\project\pics\users_summary_1st_update.PNG

### The resulting data is used to summarize the variables ‘order\_number’ as maximum and variable ‘days\_since\_prior\_order’ as the summation and the average by ‘user\_id’.The final user\_summary dataset results in 4 variables namely ‘user\_id’, ‘max\_user\_orders’, ‘user\_period\_of\_order’, ‘user\_mean\_days\_prior’.

### C:\Users\mdsha\Desktop\project\pics\users_summary_final_update.PNG

### The order\_detail(10 var) dataset is first grouped by user\_id and then summarized,which results in 3 new variables and 1 existing one.I named this table as user\_group.

### C:\Users\mdsha\Desktop\project\pics\user_group.PNG

### Then I merge user\_group(4 var) dataset into users\_summary(4 var) dataset using inner\_join,which results in 7 variables.

### C:\Users\mdsha\Desktop\project\pics\group_summary_merge_users_summary.PNG

### Now I want to know the average user’s cart size,so I create a new column user\_average\_bucket in the users\_summary dataset.After this users\_summary consist 8 variables.

### C:\Users\mdsha\Desktop\project\pics\users_summary_average_cart_size.PNG

### Now I pick orders(4 var) dataset again to filter out those observations having ‘test’ and ’train’ entries in the “eval\_set” variable and update the result into user\_group(4 var). Then select the variables user\_id, order\_id, eval\_set, duration\_since\_last\_order into the same dataset.

### C:\Users\mdsha\Desktop\project\pics\user_group_further_update.PNG

### Then I join user\_group dataset into users\_summary dataset using inner join.

### C:\Users\mdsha\Desktop\project\pics\users_summary_join_user_group.PNG

### Now this is the final phase of data preparation which will give me the essential features among the all the given datasets into one single dataset.

### To get my final dataset I start with order detail(10 var) dataset and start grouping using user\_id and product\_id variables.I use this result to summarize the variable order\_number as maximum and minimum and the variable add\_to\_cart\_order as mean and stores the result into new variables namely last\_order\_grow, first\_order\_grow and average\_cart\_position\_grow.I also create one new column orders\_grow which will tells us how much increment happens after 1st order. I name this dataset as final\_data which comprising 6 variables.

### Final\_data(6 var)

### C:\Users\mdsha\Desktop\project\pics\final_dataset_1st_update.PNG

### Next I merge product\_detail dataset into final\_data(6 var) by “product\_id” using inner join,which results in 11 variables.

### C:\Users\mdsha\Desktop\project\pics\final_data_join_product_detail.PNG

### Then I merge users\_summary dataset by “user\_id” using inner join into the above result,which comprising 24 variables.

### final\_dataset(24 variables)

### C:\Users\mdsha\Desktop\project\pics\final_data_join_user_summary_part1.PNG

### C:\Users\mdsha\Desktop\project\pics\final_data_join_user_summary_part2.PNG

### C:\Users\mdsha\Desktop\project\pics\final_data_join_user_summary_part3.PNG

### Dividing final\_dataset into Train/Test

### Train dataset

### I divide my final\_data into train and test on the basis of variable “eval\_set” entry ‘train’ and ‘test’. Once I select my train data I drop few variables purposely and make 0’s those enties having na’s/nan’s in column “reordered”. So finally my train dataset results in 20 variables.

### C:\Users\mdsha\Desktop\project\pics\train_1.PNG

### C:\Users\mdsha\Desktop\project\pics\train_2.PNG

### C:\Users\mdsha\Desktop\project\pics\train_3.PNG

### Test dataset

### For test data I select those observations from final data having “eval\_set” entries as ‘test’.C:\Users\mdsha\Desktop\project\pics\test_2.PNG

### C:\Users\mdsha\Desktop\project\pics\test_1.PNG

### C:\Users\mdsha\Desktop\project\pics\test_3.PNG

### Model Building

### I start building my model with default parameters using XGBoost algorithm. I obtain good accuracy, approximately upto 71%. Further, I try to find the optimum value of ‘nrounds’,I found it to be 76.

### C:\Users\mdsha\Desktop\project\pics\model\best_nround_with_default_parameters.PNG

### I use xgb.importance() function to calculate the importance of variables,and also,xgb.ggplot.importance() to visualize the result.

### C:\Users\mdsha\Desktop\project\pics\model\importance_of_variables.PNG

### Variable importance plot

### C:\Users\mdsha\OneDrive\Documents\R\win-library\3.3\readr\instacart_data\importance_plot.png

### Orders which are going to be placed next:

### C:\Users\mdsha\Desktop\project\pics\model\next_to_order.PNG

### Orders which are not going to be reorder:

### C:\Users\mdsha\Desktop\project\pics\model\missing_orders.PNG

### Then I combine the both the above results in single table,which is our Final output table

### C:\Users\mdsha\Desktop\project\pics\model\final_result.PNG Result : Predicted product id’s have been generated as output. Overall accuracy of 71% has been achieved. Further investigation for more relevant features may be required to improve the accuracy.