Next Basket Recommendation

**Introduction**

On a visit to a store either offline or online, Customers purchase a list of products. Their subsequent visits to the stores is spaced over a period.

The store managers can review historical transaction of each user to analyze their purchase needs and subsequently give personalized recommendation on list of products that each user is likely to purchase in their next visit.

**Problem Description**

Assist each customer with a shopping list for the next purchase according to their current needs We will predict the product list the user is most likely to have in the next basket

**Exploratory Data Analysis**

To be precise, the data consists of catalog of products with names, ids, aisle and department wise marked with names included. We have about 3.4 million orders with user ids and other details given below. And we have for each order, products ordered and re order status. Some of the numbers for the data look as follows.

|  |  |
| --- | --- |
| **Type of Dataset** | **Number of orders** |
| Number of Users | 206209 |
| Number of Unique Products | 49677 |
| Total Number of Orders | 3421083 |
| Number of Orders in Priors | 3214874 |
| Number of Orders in Train | 131209 |
| Number of Orders in Test | 75000 |

The data frame wise information is given below,

|  |  |
| --- | --- |
| **Data Frame** | **Attributes** |
| aisles.csv | aisle\_id, aisle |
| departments.csv | department\_id, department |
| order\_products.csv | 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 |

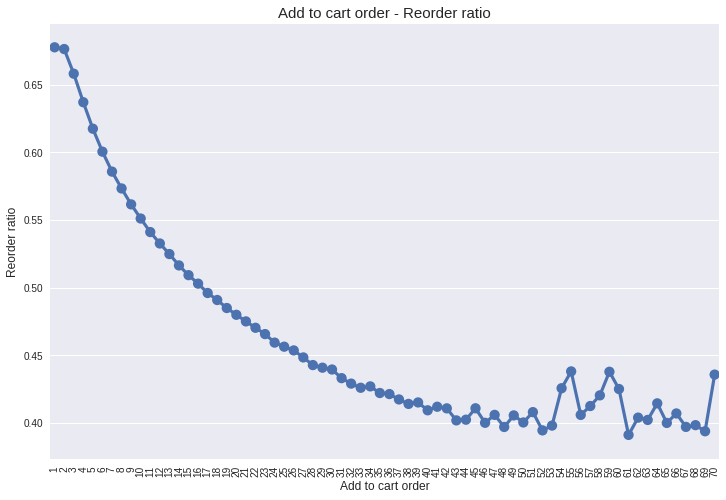
Then, we had a hypothesis that a very small percentage of products are being purchased most of times and we did a quick analysis and the results are as follows.

|  |  |
| --- | --- |
| **Number of Products chosen**  **Total - 49877** | **% Volume among total volume of purchases made** |
| Top 50 | 16.57 |
| Top 100 | 23.07 |
| Top 250 | 33.35 |
| Top 500 | 42.89 |
| Top 750 | 49.22 |
| Top 1000 | 54.0 |
| Top 2500 | 69.96 |
| Top 5000 | 81.53 |
| Top 7500 | 87.45 |
| Top 10000 | 91.05 |
| Top 15000 | 95.1 |

We could conclude that top 10% of products account for 81% purchases. Based on this observation, we finetuned our focus that we could predict on that top 10% most ordered.

1. **Add to Cart Vs Reorder Ratio**

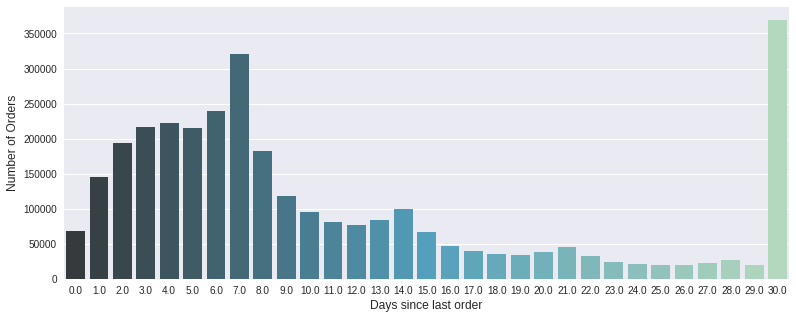
We test another hypothesis which is based on general grocery buyer pattern i.e. they order a product which they most need which is actually most reordered. We mapped add to cart index with re order ratio and the results are clear



As we can see most re ordered items are added to cart first.

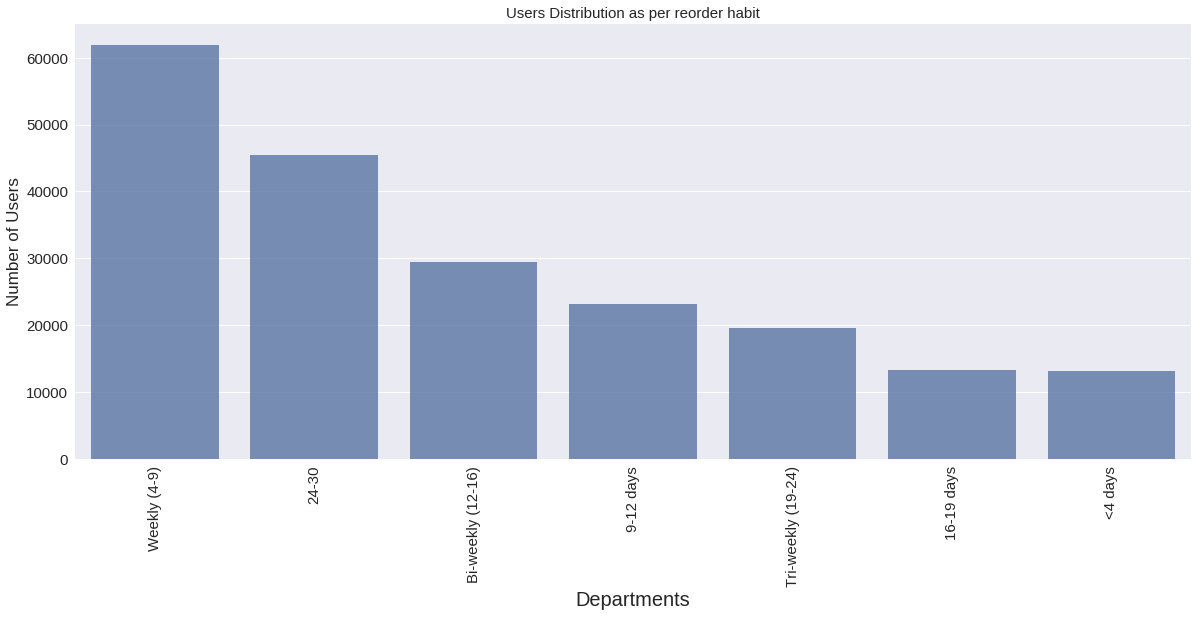
1. **Orders Vs Days Since last order**

We also would like to analyze the how frequent are the orders being published. Hence, we used counts of how many orders vs days since last order



1. **User Distribution as per reorder frequency**

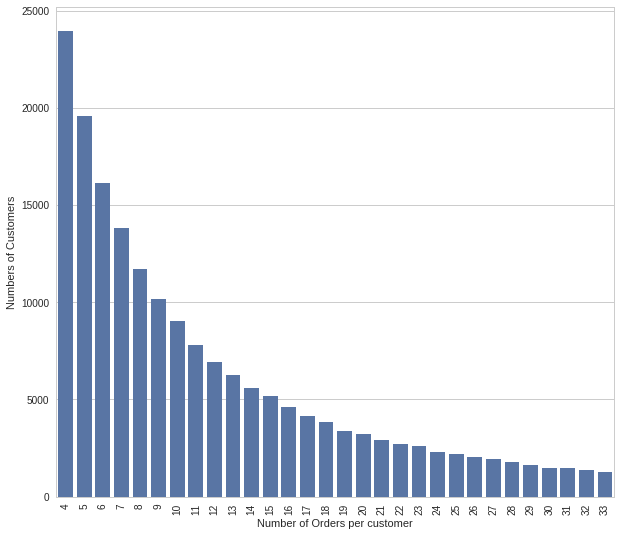
We can see a peak at 30 days since Instacart clipped all the orders with days since at 30. Also, we see local peaks at 7,14,21 days indicating weekly, bi weekly and tri weekly users. Armed with this, we decided to segment users based on their frequency of orders. And the result is as follows



We noticed that majority of users are weekly and followed by tri weekly.

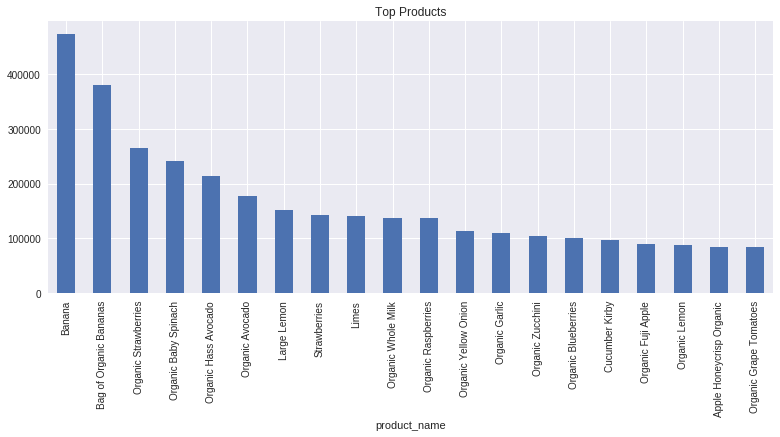
1. **Number of Orders per customer**

Next, we worked on distribution of number of orders of every customer and graph is as follows. We can see that there is a minimum of four orders for any user. And it is exponentially decreasing.



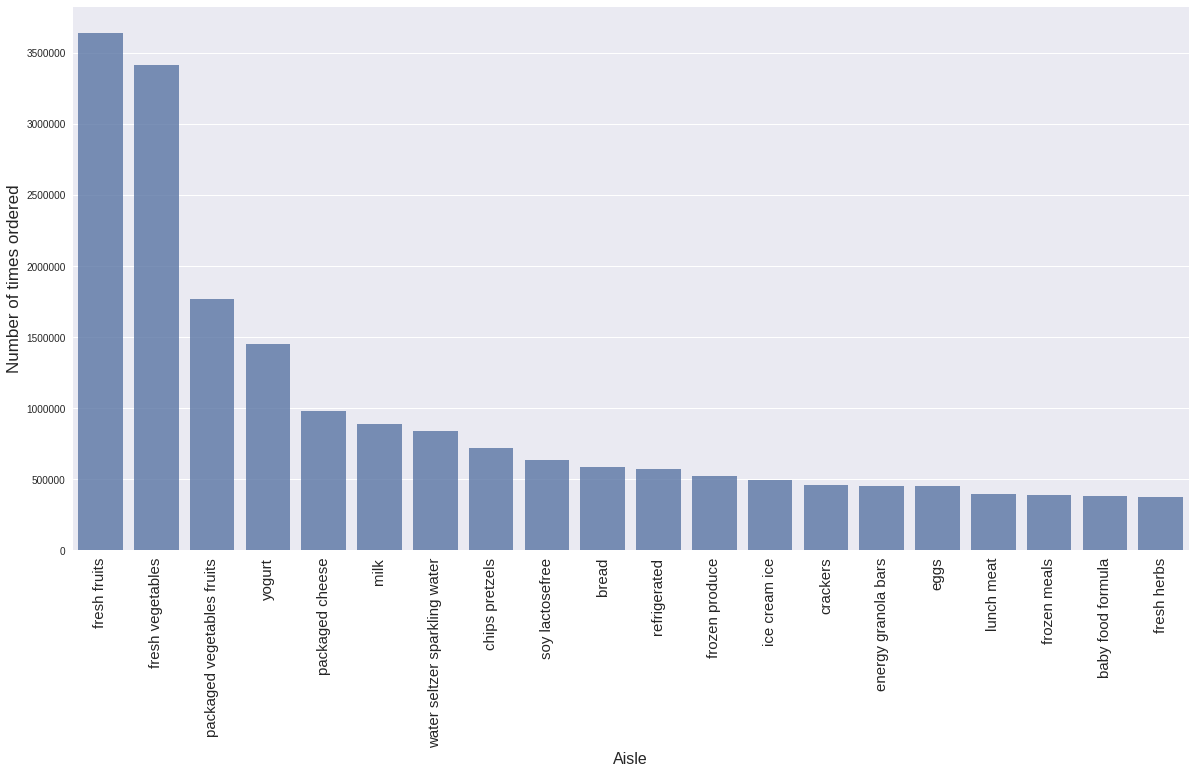
1. **Top Products Sold**

We have mapped the most sold products on the entire catalog and it is as follows

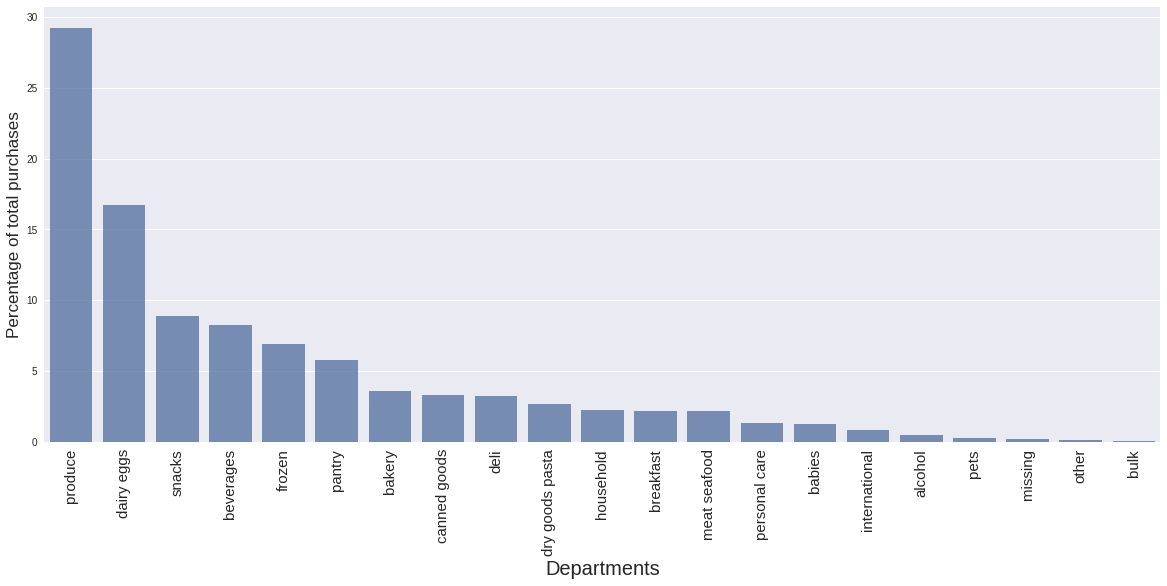


1. **Aisle Wise distribution**

We have analyzed the aisle wise distribution for all purchases. And we could notice fresh fruits and vegetables were most sold aisles.

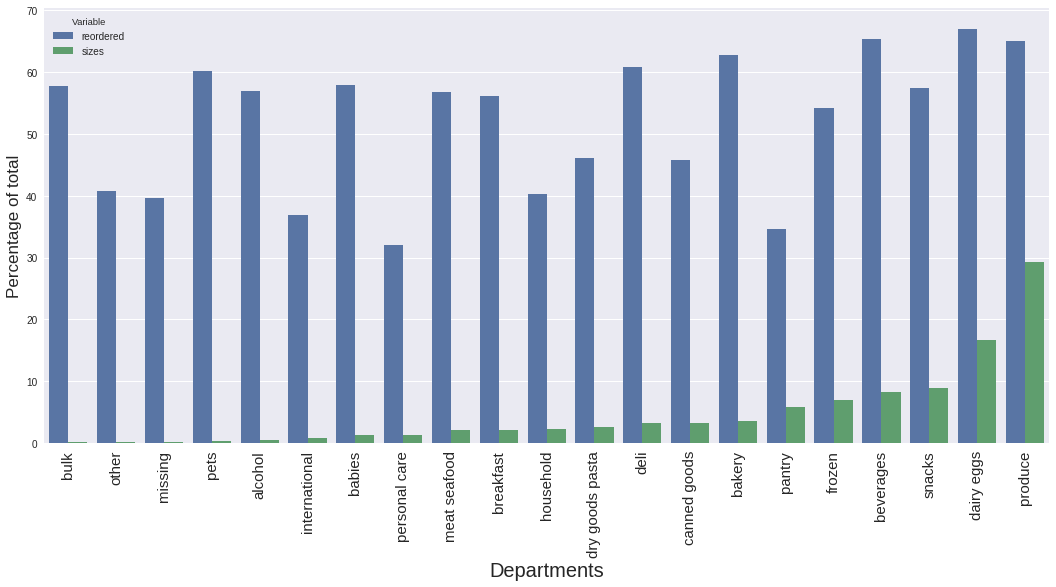


1. **Department wise distribution**



1. **Reorder ratio Vs Volume of Purchase – Department wise**

We compared the reorder ratio with % of volume of a certain department and it looked as follows. Clearly certain departments are reordered although they aren’t most sought after.



**Temporal Annotated Recurrent Sequences based Predictor**

1. **About algorithm**

There are bunch of users (U) and we have transaction history for each user. We refer each transaction as a basket (B) and items purchased in each transaction as products(P). We use Temporal Annotated Recurring Sequence Based Predictor (it’s a hybrid approach that uses sequential and pattern-based recommender)

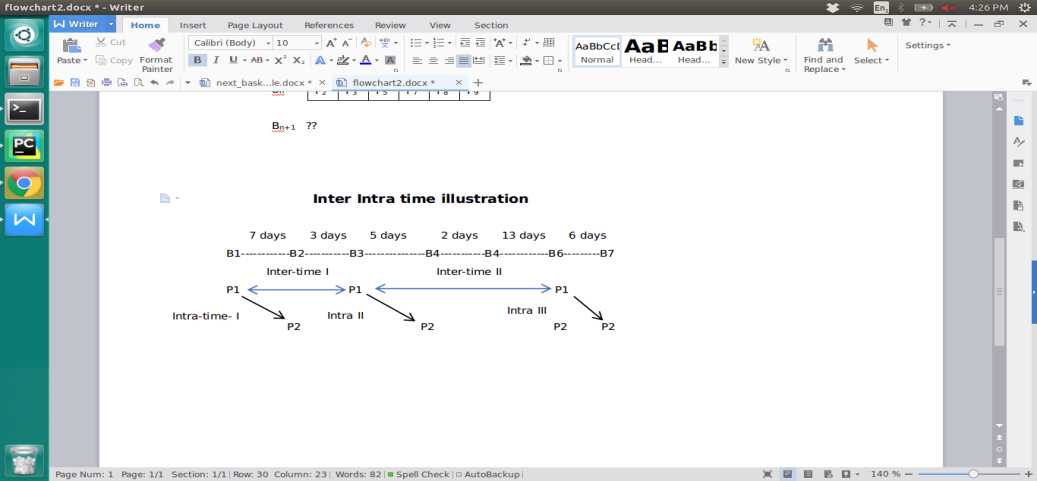
1. **Step wise approach** :
2. Capture purchasing habits – using TARS, which in turn uses FP Growth algorithm
3. Derive the recurring base sequences and get the active sequences
4. Forecast the next basket – using the TARS based predictor
5. Select active sequences and compute the score for every item. Rank the item based on the score
6. Select top k items as the basket prediction for the customer
7. **Defining TARS - Temporally Annotated Recurrent Sequences:**

Sequences are generated from Periodic FP growth algorithm and each are characterized by a head and tail.

S = Milk Curd

Inter times = [a1, a2...an], Intra times = [I1, I2, In], Med = Median number of occurrences, P = Number of periods.

1. **Defining Inter and Intra time:** Let’s have a look of Products P1, P2 occurrences in B1. B7 of User Un



Given a sequence P1 P2,

Let’s map the occurrences of P1 = [B1, B3, B6]

occurrences of P2 = [B2, B4, B6, B7]

Let’s map the occurrences of sequence P1 P2.

B1 B2, B3 B4, B6 B7

Here, we have to note that P1 occurred in B6 is not mapped with P2 in same basket because same basket items are not mapped

1. **Computing of Inter-time:**

Inter times are calculated based on difference of times between two consecutive heads of all occurrences.

In our case, we have three occurrences of sequence, hence we will obtain two inter times -

Intertime I = Headoccurence I - Headoccurence II = 3+ 5 = 10 days

Intertime II = Headoccurence II - Headoccurence III = 5+2+13 = 20 days

Inter time of sequence S1 = [Inter time I, Inter time II] = [10,20]

1. **Computing of Intra-time:**

Intra time is the time taken between head and tail of a sequence occurrence. Let’s take above occurrences itself

Intra time I = B1 - B2 = 7 days

Intra time II = B3- B4 = 5 days

Intra time III = B6 - B7 = 6 days

Intra time of sequence S2 = [Intra time I, Intra time II, Intra time III] = [7,5,6]

1. **Defining periodicity for a sequence:**

Periodicity is calculated by using values of Intertimemax and qmin. We will look at how to calculate them later. Assuming we have those values, period is a set of occurrences of ‘heads’ of which each and every inter-time is less than Intertimemax and number of occurrences is greater than qmin.

Let’s assume a sequence, P1 P2 and let the head P1 occur in transactions as below,

Transaction ids of head of Sequence occurrences = [1,3,6,7,8,9,15,17,18]

Inter-times for occurrences = [5,6,5,3,22,35,3,4]

Let’s assume Inter-timemax = 10 and qmin = 4

Now to calculate periods, let’s start at starting of inter time list, keep adding all inter times to first period until we reach an inter time which is greater than Inter-timemax. Now if the length of period is greater than qmin, we retain it. Otherwise discard it and move to next.

In the above case, [5,6,5,3] are added and we come across 22 which is greater than 10 (Inter-timemax). Now we look at size of first period in in terms of transactions which will be 4 + 1 = 5 as 5 transactions will give 4 inter times. Now, 5 is greater than qmin = 4, so this is period.

I Period in terms of inter times = [5,6,5,3]

I Period in in terms of transaction ids of head of sequence = [1,3,6,7,8]

Now, let’s proceed, 22 and 35 are greater than 10, so we discard them. Proceeding further we get [3,4] as next period before we end up with list. But length = 2, this means number of occurrences will be 3 which is still less than qmin which doesn’t meet the minimum requirements.

Periods - {[1,3,6,7,8]}

Now if we assume qmin = 3, so last period will also get added. Now periods for this sequence

Periods = {[1,3,6,7,8], [15,17,18]}

1. **Defining Recurrence for a sequence:**

Recurrence for a sequence is defined as number of periods it has occurred.

Rec(S) = |Periods|

In the above case, we have recurrence as 1,2. We define a minimum Rec(S) for a sequence to be called recurrent. A sequence (S) is said to recurrent,

Rec(S) > Recmin

1. **Defining Predictor for TARS:**

We generate a list of k sequences S = {S1, S2, Sk} for every user. For all the sequences, tails are made as a list {t1, t2, ti.} and a predictor βi is initialized for each item in tails. Now each tail in list is picked and we iterate through S (sequences list). Whenever a tail tk is encountered in Sk , we find ‘increment’ value as follows

Increment = Number of occurrences of Sk in last period - Median number of occurrences in each period

βk = βk + increment

* Support of tails among sequences:

At the end, we add support of each tail ti to predictor βi. In addition, we have added a modified parameter after extensive experimentation. The parameter is proportional to difference of days since last head occurrence and median intra time

1. **Top K items to be predicted:**

Determining the top k items was a challenge. K can be taken as mean number of items in all orders across user or median as well. But we did a regression fitting the number of items as target label and days since prior order and previous order’s quantity are the dependent variables. Regression analysis gave us better scores.

1. **Flow of the algorithm:**
2. Aggregate data of orders and products together
3. Pick a user and generate data for baskets with products
4. From the data of a user, generate initial set of sequences using Periodic FP Growth Algorithm
5. Eliminate sequences which has less frequency and periodic nature.
6. Map each sequence with inter time and intra time lists
7. Calculate qmin and inter-timemax based algorithm given.
8. Based on above values, determine periodicity
9. Based on periodicity, map the recurrence and filter out sequences which has less than Recmin
10. The final filtered sequences will be used for prediction. For each sequence, tail is taken and predictor is calculated.
11. A list of products with predictor value sorted in descending order is obtained.
12. Predict the ‘k’ items to be recommended from the regression analysis

**User Specific LGBM applied with TARS based features**

1. **About algorithm**

There are bunch of users (U) and we have transaction history for each user. We refer each transaction as a basket (B) and items purchased in each transaction as products(P). We take baskets of each user and calculate certain features and there are features which are also calculated for overall data and appended here.

1. **Step wise approach:**
2. Gather each user data and build
3. Build target label (purchase/ no purchase) based on index
4. Calculate user level features and append to data frame
5. Calculate product level features and append to data frame
6. Train the data frame for classifier and predict.
7. Run a FP growth on user level data.
8. Based on threshold set and classification results, print recommendations

For each user, we create a data frame with a unique id which is formed from order Id and product id.

Unique\_id = 1000000\* order\_id + product\_id

This is calculated for every order\_id and distinct\_id combinations. Suppose there three orders as follows

|  |  |
| --- | --- |
| Order Id | Products |
| 103 | [1,2,3] |
| 104 | [1,3] |
| 105 | [2,3] |
| 106 | [1,2] |

Now, let’s build the unique index as defined.

Distinct Products = [1,2,3]

Order Ids = [103,104,105,106]

Let’s simplify unique id formula as

Unique\_id = 1000\* order\_id + product\_id

Now index will be as follows

|  |
| --- |
| Unique ID |
| 103001 |
| 103002 |
| 103003 |
| 104001 |
| 104002 |
| 104003 |
| 105001 |
| 105002 |
| 105003 |
| 106001 |
| 106002 |
| 106003 |

Now let’s define target label, passing through each index, see that product is available in given order id. If it is, mark it as 1, else 0, now the labels will look as follows.

|  |  |
| --- | --- |
| Unique ID | Target Label |
| 103001 | 1 |
| 103002 | 1 |
| 103003 | 1 |
| 104001 | 1 |
| 104002 | 0 |
| 104003 | 1 |
| 105001 | 0 |
| 105002 | 1 |
| 105003 | 1 |
| 106001 | 1 |
| 106002 | 1 |
| 106003 | 0 |

**Features for LGBM**

**Product related features:**

1. Orders: Number of times specific product was ordered in whole data
2. Reorders: Number of times it is reordered in whole data
3. Re order rate: It is defined as ratio of reorders by orders.

**User related features: These are calculated for each user**

1. Average days between orders: All baskets for user are taken and average of days since last order is calculated.
2. Total unique items purchased: Count of unique products purchased from all the baskets available for user
3. Total items purchased: Count of all products purchased from all the baskets available for user
4. Average Basket Size: Average taken by summing count of all basket sizes by number of baskets.
5. Number of orders

**Computation of dynamic features:** These features value change as we proceed from one order to another order and these are calculated for each user at once.

1. Distinct items so far: For each order, we will calculate the number of distinct items ordered to date. As we proceed towards future orders, number of distinct items change.
2. Product wise orders ratio: For each product as orders proceed, counts of products ordered are calculated. The it is divided by number of orders so far placed. This is a data frame with all distinct items as row index and counts of products ordered are updated at each order. And ratio is calculated.
3. Average Days since prior orders: This is also a running feature. As orders proceed, the average is recalculated from all previous orders.
4. Product wise days since last purchase: Days since last order is given but this feature considers days since last time the product was purchased. It is equivalent to inter time
5. Product wise days since last purchase average: This is calculated by doing an average of all the inter times or product wise days since last purchase.

**Metrics**

**F1 Score:** It is the harmonic mean of precision and recall. Let’s assume we predicted next basket for a user as Bpredicted = [P3, P7, P8, P9, P10]. And the actual next basket purchased by user is Bactual = [P1, P3, P5, P7, P8, P9]

True Positives = [P3, P7, P8, P9] = 4; Predicted Positives = [P3, P7, P8, P9, P10] = 5

Actual Positives = [P1, P3, P5, P7, P8, P9] = 6

Precision = = = 0.8

Recall = = = 0.67

F1 score = =

Now this F1 score is calculated for each user and overall score is an average of all users

**Normalized F1 Score:** In this we exclude users for which True positives is zero excluding junk kind of data to get a better F1 score.