User Reorder Analysis of Instacart Users

(Project of MTH552A)

Submitted by:

Soumyadip Sarkar (201431)
Saumyadip Bhowmick (201408)

Supervised by,

Dr. Amit Mitra

Contents

1 Introduction

Ordering food supplies online is a new way of restocking groceries and other necessary items. But problem occurs when customers forget some items when adding items to the cart or they want to improve their suggestion on their items. To tackle such situations, users are provided with suggestions based on their previous orders or user preferences.
Instacart is a grocery order and the delivery app with over 500 Million products and 40000 stores that serves across U.S. & Canada. Instacart provides a user experience where customers will get product recommendation based on your previous orders.

2 Objective

The dataset we have got provided us with transactional data of customer orders over time to predict which previously purchased products will be in a user's next order. Also we will try to analyze the user previous transactions for any insights .

3 Data Description

The dataset contains a sample of over 1 million grocery orders from more than 200,000 Instacart users. The data consist mainly 2 parts:

- **Prior Data**: Previous order history of each user. This data contains almost 3–100 past orders of the user.
- **Train data** : Present order data of every user . This data consists of only the last order of the user.

orders.csv — Consists of order details placed by any user. Shape of this data is (3421083,7)

- Order_id : Unique for every order
- User_id : Unique for every user
- Eval_set : (prior / train)
- 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

order_products__prior.csv— Consists of all product details for any prior order. Shape of this data is (10000000,4)

- 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 ? (1/0)

order_products__train.csv — Consists of all product details for a train order. Shape of this data is (1384617,4)

- 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 ? (1/0)

products.csv — Details of a product belonging to a particular aisle and department. Shape of this data is (49688,4)

- product_id : product ID of item
- product_name : name of product
- aisle_id : aisle id of the product
- department_id : department id of the product

Aisles.csv — Details of aisles of products. Shape of this data is (134,2)

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

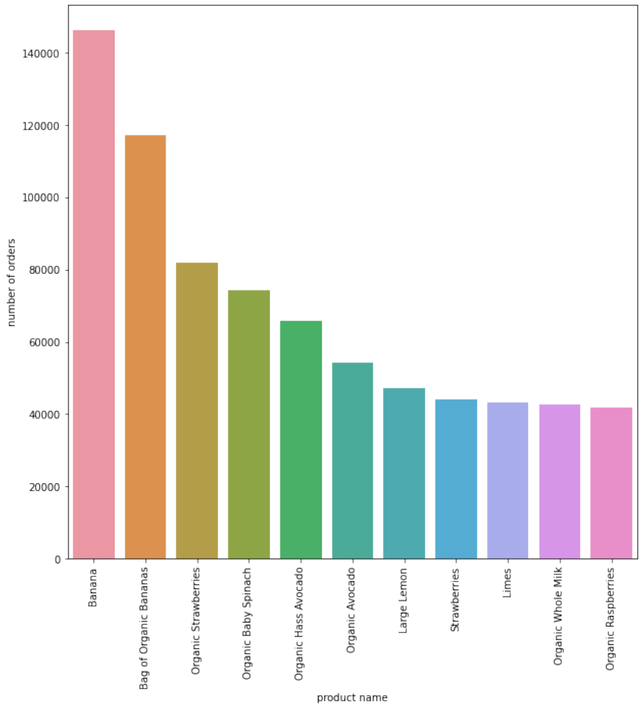
department.csv — Details of department of products.Shape of this data is (21,2)

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

4 Exploratory Data Analysis

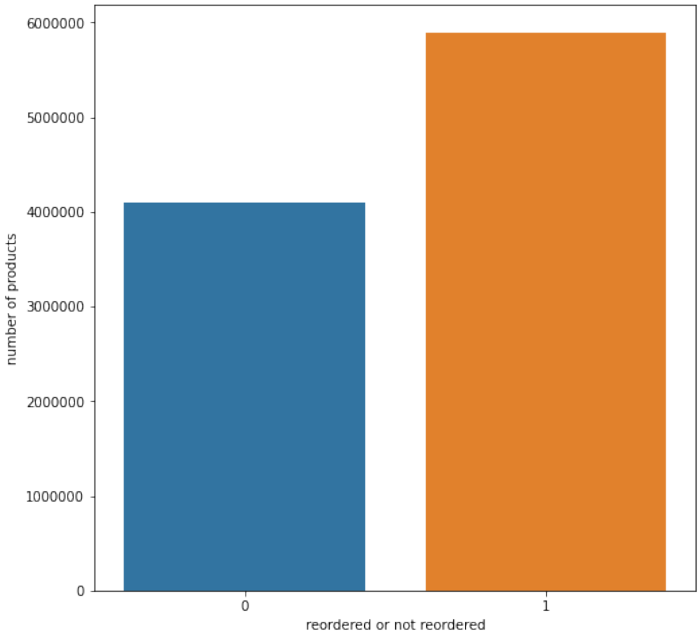
In the dataset we have got data of 991714 orders and 48520 unique products. Now we perform some exploratory data analysis on the basis of a final merged dataset we have obtained combining the data from different datasets mentioned above.

1. At first we want look at the most frequently ordered product.



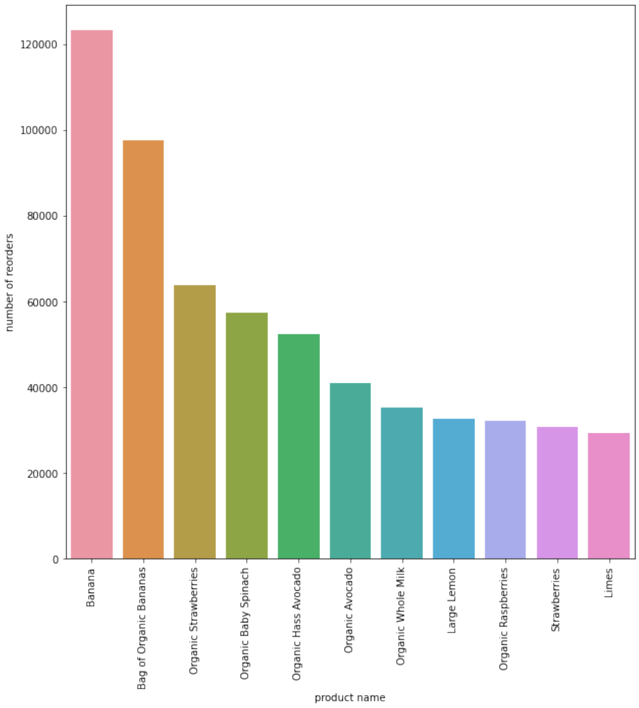
Here we have plotted bar diagram for top 10 of the most **frequently ordered** products and found that orders for banana and bag of organic bananas are significantly more than others.

2. We want to know the number of times products have been reordered or not reordered.



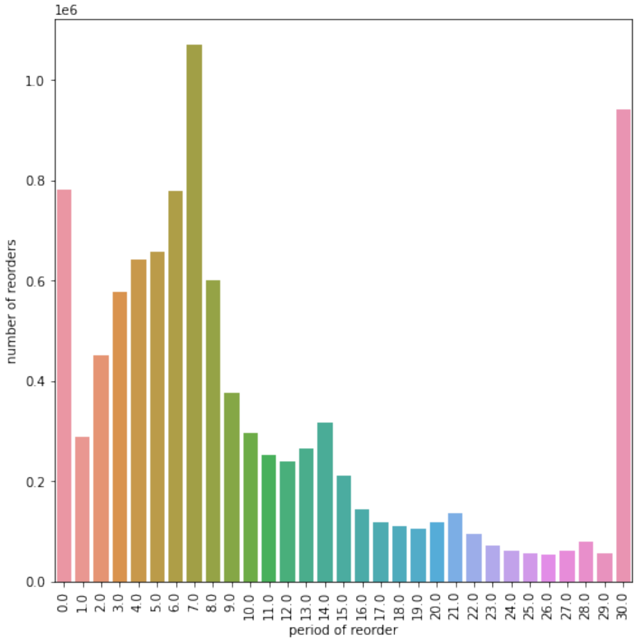
The plot above gives us the number of products which have been reordered or not.Almost 4000000 times products have been reordered and 6000000 times products have not been reordered.

3. Now we want look at the most frequently reordered product.



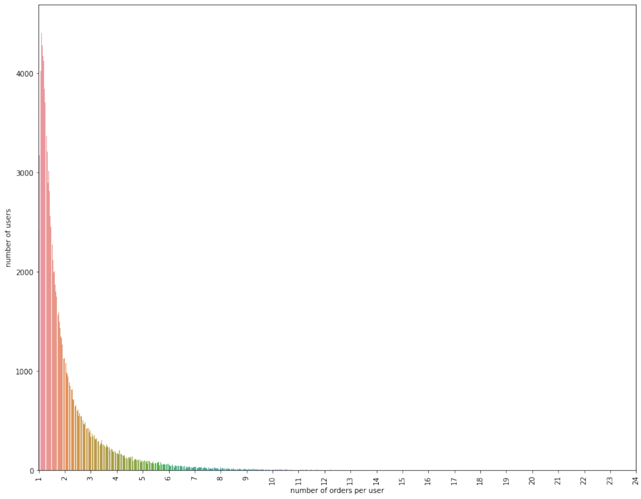
Here also reorders for banana and bag of organic bananas are significantly more than others as we see from the bar-diagrams and another thing to be noticed that the most ordered and reordered products are similar for top 10 products. It is just that they have permuted among themselves for last few products.

4. We are also interested to know how frequent the reorders are.



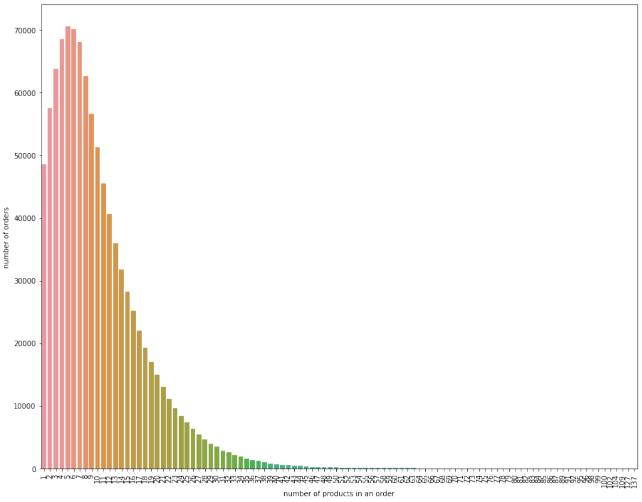
We see that the number of reorders are significantly good on weekly and monthly basis. Frequency is high at 0 may be interpreted as there may be more than one units ordered at a time.

5. Now someone may be interested to know the frequency of users for number of orders per user.



The number of users decrease as the average number of order per user increases.

6. Now we want to see number of orders for different cart size.



In this cases the frequency decreases as the cart size increases after the value of 9 which is intuitively logical as well.

5 Feature selection

Now we are using the `data_orders.csv` dataset to incorporate the prior data for our analysis. Therefore some more features have been extracted using those prior data which we are going to use as predictor variable along with the predictor variables we have got in `order_products_train .csv` dataset. These extracted features are of 3 kinds:

1. Product only features
2. User only features
3. User product features

5.1 Product only features

Some product based features have been extracted to use them for classification.

- Feature_1:
  reordered_ratio:How frequently a particular product was reordered =

$$\frac{\text{Number of times a product was reordered}}{\text{Number of times a product was ordered}}$$

- Feature_2:
  mean_position: Average position of a product in the cart when ordered.

- Feature_3:
  dept_reorder_rate: How frequently a particular product from a department was reordered =

$$\frac{\text{Number of times a product from a department was reordered}}{\text{Number of times a product from a department was ordered}}$$

- Feature_4:
  aisle_reorder_rate: How frequently a particular product from a aisle category was reordered =

$$\frac{\text{Number of times a product from a aisle was reordered}}{\text{Number of times a product from a aisle was ordered}}$$

5.2 User only features

Some user based features have been extracted to use them for classification.

- Feature_1:
  user_reorder_rate: Average reorder rate on order placed =

- Feature_2:
  user_unique_products: Distinct products ordered by a user =

$$\frac{\text{Number of times a user reordered}}{\text{Number of times a users ordered}}$$

- Feature_3:
  user_total_products: Total product ordered by a user.

- Feature_4:
  user_avg_cart_size: average number of products in the cart for a user.

- Feature_5:
  user_avg_days_between_orders: average gap between 2 consecutive purchases.

- Feature_6:
  user_reordered_products_ratio: user product reorder ratio =

$$\frac{\text{number of unique products reordered}}{\text{number of unique products ordered}}$$

5.3 User-Product features

Some user product combination based features have been extracted to use them for classification.

- Feature_1:
  u_p_order_rate: How frequently a user has ordered a a particular product.

- Feature_2:
  u_p_reorder_rate: How frequently a user has reordered a particular product.

- Feature_3:
  u_p_avg_position: average position of a product in the list of cart of a user.

- Feature_4:
  u_p_orders_since_last: number of orders placed by a user since the product was last ordered.

- Feature_5:
  days_since_prior_reorder_rate: how frequently user reordered a particular product given difference between 2 orders in days.

6 Train-test Split

Once the extracted features columns were constructed they were merged with `order_products_train.csv` dataframe and then merged dataframe is split into train dataset and test dataset in the proportion of 9:1.

7 Classification Models

7.1 Logistic Regression

We here use Logistic Regression for binary classification. Here we use the sigmoid function to predict the class for a data point. For a model with k parameters and if we have to classify datapoints into two classes classes viz 0 and 1 . We denote p = P(Y=1|X=**x**) where Y is predictor variable ,We can write

$$p = \frac{1}{1 + e^{-(\beta_0 + \sum_{i=1}^{k} \beta_i x_i)}}$$

Now, Y is predicted to belong to class 1 if p $\geq$ 0.5 or predicted to belong to class 0 otherwise.

.

7.2 Decision Tree

A decision tree is a flowchart-like structure in which each internal node represents a “test” on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label. The paths from root to leaf represent classification rules.
It is like an example of a multistage decision process. Rather than using the complete set of

features jointly to make a output decision, different subsets of features are used at different levels of the tree.

7.3 Random Forest

A random forest algorithm consists of many decision trees. A random forest is a machine learning technique that's used to solve regression and classification problems. It utilizes ensemble learning, which is a technique that combines many classifiers to provide solutions to complex problems. In other words, in building a random forest, at each split in the tree, the algorithm is not even allowed to consider a majority of the available predictors.

Classification in random forests employs an ensemble methodology to attain the outcome. The training data is fed to train various decision trees.

Decision trees suffer from high variance problem. To get rid of this high variance problem, we use random forest. In this method, we use the concept of **"Averaging independent observations reduce variance"**. In random forest, we take a bootstrapped sample of size N from the training data and grow a decision tree with these N samples. The only difference is that, in this case, in each split, we take a random sample of total number of predictors, preferable sample size is the square root of the number of predictors. We repeat this process, say, M number of times.

For classification setting, we note down the class predicted by each of the 'M' trees and for final prediction, we take the majority of these 'M' outputs.

The advantage of Random Forest lies in the fact that it **decorrelates** the trees. Suppose there is a strong predictor in our dataset. Then there will be a tendency to use this predictor in each top split. So the trees may look similar. In Random Forest, we are using a random sample of predictors and in some cases, the most important variables may even not be considered. We can think this as decorrelating the trees, thereby making predictions less variable.

8 Model evaluation

8.1 ROC Curve and AUC

ROC Curve or receiver Operating Characteristic curve is a curve which is used to measure the performance of a classification model at all possible classification threshold.

The curve is drawn by plotting **true positive rate** against **false positive rate**

True positive rate is defined as

$$TPR = \frac{TP}{TP + FN}$$

And false positive rate is defined as

$$FPR = \frac{FP}{FP + TN}$$

AUC stands for Area Under the ROC Curve. If 0.5<AUC<1 there is a high possibility that the classifier will be able to separate the values of different classes from each other.

8.2 F1 score

We should look for such accuracy measure that will take care of accuracy measures like precision and recall. Therefore we chose F1-score to check our model performance.
F1-score is given by,

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

9 Association Rule Mining

In the context of recommendation of products to the user, Association rule mining has a significant role since it can find association between different itemsets that customers place in the shopping basket.
We have used 2-step **Apriori Algorithm** to predict which product can be suggested to a particular user on the basis of his previous purchases.
The 2 basic notions for Apriori algorithm are viz **Support** and **Confidence**

9.1 Support

Support of an itemset may be defined as the fraction of transactions containing that particular itemset.If we have T number of transactions and σitemset be the number of transactions containing that itemset then Support of that itemset , S(t) is given by,

$$S(\text{itemset}) = \frac{\sigma(itemset)}{T}$$

9.2 Confidence

Confidence is related to some rule say $A \implies B$. For this rule confidence may be defined as how often B appears in the transactions containing A. Confidence for this arbitrary rule $A \implies B$ i.e ,$C(A \implies B)$is given by

$$C(A \implies B) = \frac{S(AB)}{S(A)}$$

9.3 2 step Apriori Algorithm

Step-1: Generate all frequent itemsets with support > some given threshold of minimum support
Step-2: Generate association rules using these frequent itemsets.

9.3.1 Step-1:

1. Start with k=1
2. generate itemsets of length k.
3. find their support. Prune the items having support < a minimum threshold.
4. generate itemset of length k+1 from frequent itemsets of length k.
5. repeat the step 3
6. Repeat until no frequent itemsets are found.

9.3.2 Step-2:

1. Given any frequent itemset L; find all non-empty subsets F of L.
2. output each rule $F \implies L - F$ that has the confidence more than a preassigned minimum threshold value of confidence.

10 Results and Findings

10.1 Classificaction

10.1.1 Logistic Regression

We performed a 5-fold cross validation to choose the tuning parameter C(regularising parameter) among some pre specified values, and choose the model with maximum cross validated F score. Then we trained the model with the using total training dataset, and applied it on the test dataset and obtained the following confusion matrix. For this model, we have obtained trainig accuracy of 0.65 and test accuracy of 0.64.

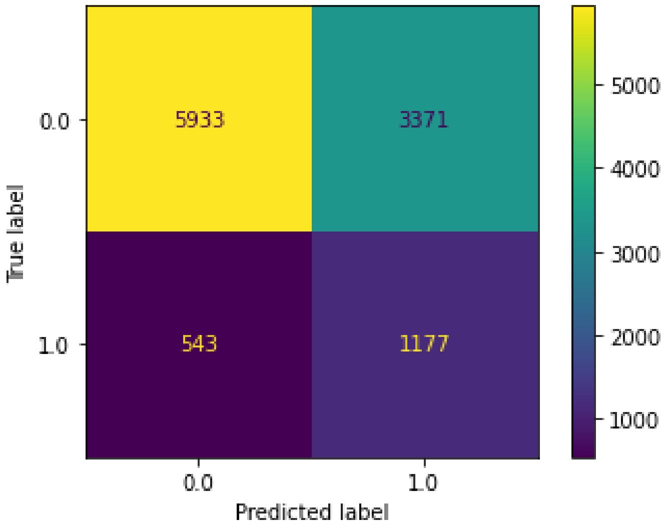


Figure 1: Confusion Matrix from applying Logistic Regression Model on Test data

We have also plotted the ROC curve. We have got F score of 0.38 and AUC of 0.72.

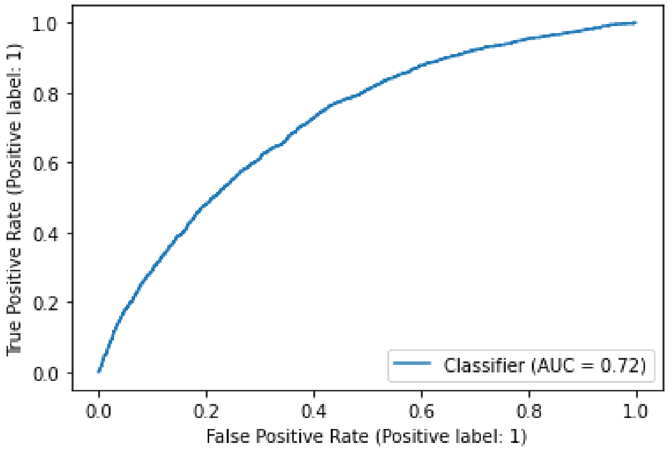


Figure 2: ROC Curve from applying Logistic Regression Model on Test data

10.1.2 Desicion Tree

We performed a 5-fold cross validation to choose the tuning parameter max_depth, max_features, n_estimators among some pre specified values, and choose the model with maximum cross validated F score. Then we trained the model with the using total training dataset, and applied it on the test dataset and obtained the following confusion matrix. For this model, we have obtained trainig accuracy of 0.71 and test accuracy of 0.68.

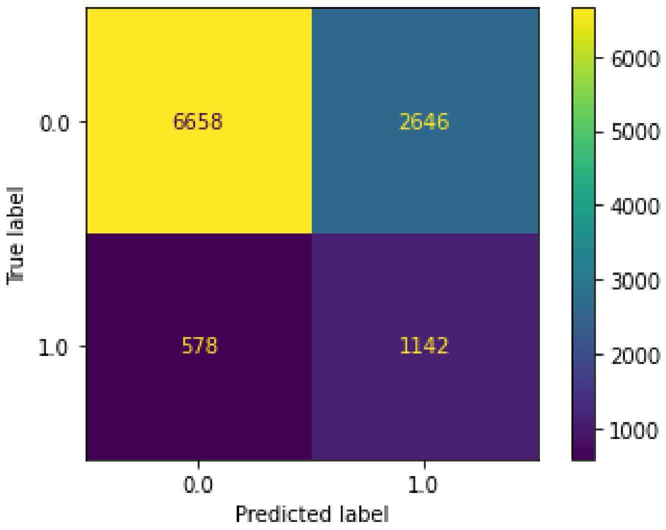


Figure 3: Confusion Matrix from applying Desicion Tree Model on Test data

We have also plotted the ROC curve. We have got F score of 0.41 and AUC of 0.71.

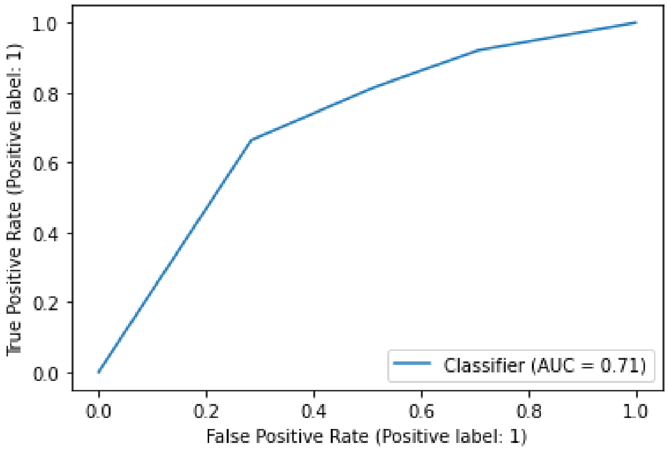


Figure 4: ROC Curve from applying Desicion Tree Model on Test data

10.1.3 Random Forest

We performed a 5-fold cross validation to choose the tuning parameter max_depth, max_features, n_estimators among some pre specified values, and choose the model with maximum cross validated F score. Then we trained the model with the using total training dataset, and applied it on the test dataset and obtained the following confusion matrix. For this model, we have obtained trainig accuracy of 0.90 and test accuracy of 0.81.

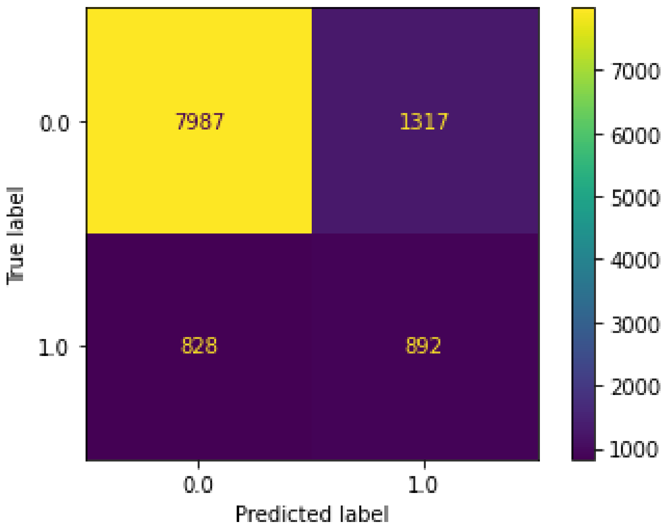


Figure 5: Confusion Matrix from applying Random Forest Model on Test data

We have also plotted the ROC curve. We have got F score of 0.45 and AUC of 0.77.

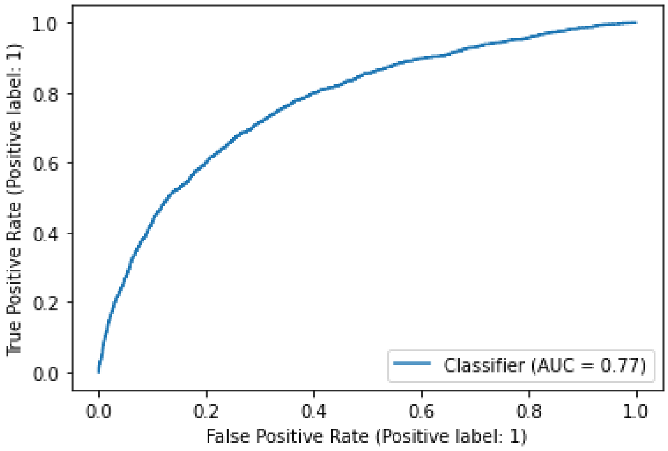


Figure 6: ROC Curve from applying Random Forest Model on Test data

Let us visualise the performances of the models for comparison from this table.

Table 1: Meaure of performance of the algorithms

Model	Training Acc.	Test Acc.	AUC	F Score
Logistic Regression	0.68	0.64	0.72	0.38
Decision Tree	0.71	0.68	0.71	0.41
Random Forest	0.90	0.81	0.77	0.45

From the above table, also, we can see that Random Forest has the highest F-Score and AUC.

10.2 ARM

10.2.1 Frequent Itemsets

Here first we have generated all frequent itemsets with support > 0.01. There were a total 123 frequent itemsets, we have tabulated here 30 most frequent itemsets.

Index	Support	Itemsets
0	0.15231591448931117	Banana
1	0.12069477434679335	Bag of Organic Bananas
2	0.08358076009501188	Organic Strawberries
3	0.07140736342042756	Organic Baby Spinach
4	0.0671021377672209	Organic Hass Avocado
5	0.057897862232779096	Organic Avocado
6	0.04735748218527316	Large Lemon
7	0.04661520190023753	Strawberries
8	0.043497624703087885	Organic Raspberries
9	0.03963776722090261	Organic Whole Milk
10	0.03963776722090261	Limes
11	0.03681710213776722	Organic Garlic
12	0.03577790973871734	Organic Yellow Onion
13	0.031621140142517816	Organic Zucchini
14	0.029097387173396674	Organic Fuji Apple
15	0.028948931116389548	Honeycrisp Apple
16	0.028652019002375295	Organic Blueberries
17	0.028058194774346793	Cucumber Kirby
18	0.027315914489311165	Organic Grape Tomatoes
19	0.027019002375296912	Apple Honeycrisp Organic
20	0.02672209026128266	Organic Half & Half
21	0.02672209026128266	Organic Lemon
22	0.024643705463182897	Organic Cucumber
23	0.024346793349168647	Organic Baby Arugula
24	0.023901425178147268	Original Hummus
25	0.02315914489311164	Organic Baby Carrots
26	0.023010688836104513	Carrots
27	0.023010688836104513	Sparkling Water Grapefruit
28	0.022416864608076008	Seedless Red Grapes
29	0.022268408551068885	Organic Gala Apples

10.2.2 Rules

Then we have generated association rules using these frequent itemsets, with confidence > 0.01.

Index	Antecedents	Consequents	Antecedent support	Consequent support	Support	Confidence
0	Honeycrisp Apple	Banana	0.028949	0.152316	0.01247	0.430769
1	Organic Fuji Apple	Banana	0.029097	0.152316	0.010837	0.372449
2	Organic Raspberries	Bag of Organic Bananas	0.043498	0.120695	0.0144	0.331058
3	Strawberries	Banana	0.046615	0.152316	0.014103	0.302548
4	Organic Hass Avocado	Bag of Organic Bananas	0.067102	0.120695	0.019596	0.292035
5	Organic Avocado	Banana	0.057898	0.152316	0.016182	0.279487
6	Organic Whole Milk	Banana	0.039638	0.152316	0.010837	0.273408
7	Organic Raspberries	Organic Strawberries	0.043498	0.083581	0.010837	0.249147
8	Organic Strawberries	Banana	0.083581	0.152316	0.020042	0.239787
9	Organic Baby Spinach	Banana	0.071407	0.152316	0.016033	0.224532
10	Organic Strawberries	Bag of Organic Bananas	0.083581	0.120695	0.018112	0.216696
11	Organic Baby Spinach	Bag of Organic Bananas	0.071407	0.120695	0.0144	0.201663
12	Organic Hass Avocado	Organic Strawberries	0.067102	0.083581	0.013361	0.199115
13	Organic Baby Spinach	Organic Strawberries	0.071407	0.083581	0.012767	0.178794
14	Organic Hass Avocado	Banana	0.067102	0.152316	0.011728	0.174779
15	Bag of Organic Bananas	Organic Hass Avocado	0.120695	0.067102	0.019596	0.162362
16	Organic Strawberries	Organic Hass Avocado	0.083581	0.067102	0.013361	0.159858
17	Organic Strawberries	Organic Baby Spinach	0.083581	0.071407	0.012767	0.152753
18	Bag of Organic Bananas	Organic Strawberries	0.120695	0.083581	0.018112	0.150062
19	Banana	Organic Strawberries	0.152316	0.083581	0.020042	0.131579

10.2.3 Recommend Product to test data users

For the test data we are first predicting the products in a user cart, and if the products are in Antecedents of the association rules, then we are recommending the Consequents of that particular rule to that user. we are able to recommend 79 users among 206 users who reordered any of the products in test data. here we tabulate 30 of them.

Index	User_id	Reorded_product_names	Recommendation
0	214.0	Organic Strawberries	Organic Raspberries
1	245.0	Organic Strawberries	Organic Raspberries
2	323.0	Organic Strawberries	Organic Raspberries
3	487.0	Bag of Organic Bananas	Organic Raspberries
4	567.0	Organic Avocado	Banana
5	641.0	Honeycrisp Apple	Banana
6	714.0	Organic Strawberries	Organic Raspberries
7	741.0	Organic Baby Spinach	Organic Strawberries
8	786.0	Organic Strawberries	Organic Raspberries
9	973.0	Strawberries	Banana
10	1172.0	Organic Avocado	Banana
11	1271.0	Bag of Organic Bananas	Organic Raspberries
12	1387.0	Organic Baby Spinach	Organic Strawberries
13	1531.0	Strawberries	Banana
14	1683.0	Bag of Organic Bananas	Organic Raspberries
15	1787.0	Organic Baby Spinach	Organic Strawberries
16	1981.0	Organic Strawberries	Organic Raspberries
17	2162.0	Organic Avocado	Banana
18	2307.0	Bag of Organic Bananas	Organic Raspberries
19	2582.0	Organic Avocado	Banana
20	2765.0	Organic Strawberries	Organic Raspberries
21	2788.0	Organic Baby Spinach	Organic Strawberries
22	2808.0	Organic Avocado	Banana
23	2887.0	Organic Avocado	Banana
24	3233.0	Bag of Organic Bananas	Organic Raspberries
25	3298.0	Organic Strawberries	Organic Raspberries
26	3312.0	Organic Baby Spinach	Organic Strawberries
27	3635.0	Bag of Organic Bananas	Organic Raspberries
28	3823.0	Bag of Organic Bananas	Organic Raspberries
29	4030.0	Honeycrisp Apple	Banana

11 Conclusion

Our final conclusions from this analysis are,

- We have found the best training accuracy of 0.90 and best test accuracy of 0.81. Clearly this is not optimal, and can be improved by using other sophisticated algorithms.
- Due to computational complexity we couldn't use total dataset, using the entire dataset may improve accuracy scores.
- The association rules generated can be modeled somehow in classification algorithms for further analysis.

12 Reference

1. Stack Overflow
2. scikit-learn Documentation
3. Wikipedia