**Analysis**

I used Two algorithms to solve this classiﬁcation problem

**XGBoost**: XGBoost stands for eXtreme Gradient Boosting. It is a scalable and accurate implementation of gradient boosting machines. It was developed to optimize the model performance and computational speed. Thus, it has become really popular tool among Kaggle competitors and also widely used by Data Scientists in industry.

I divided the entire data set into Training and Cross Validation data sets. I then trained a baseline model with default hyperparameters and all the features mentioned in the ’Feature Engineering’ section. I found determined importance for all the features (Fig 6). I then ran a loop to test for which set of top features model is giving the best f-score on cross validation data.

As GridCV was throwing memory error due to large size of data, I tuned each hyperparameter separately to get highest f-score. After training the model on best features, I tested various threshold to get highest f-score,

**Light GBM**: Light GBM is a gradient framework that uses tree based learning algorithms. Its is designed to be distributed and efﬁcient. It has faster training speed and higher efﬁciency, consumes less memory, and produce better accuray.

I started with training a baseline model with all the constructed features mentioned in the ’Feature Engineering’ section and the default parameters. To select the best features and hyperparameters, I divided the entire data set into Training and Cross Validation data sets. Next I found out the importance for all the features and tested for top features for which top features the validation f-score was the highest

After that I changed the hyperparameters given while training the model and checked for best f-score for validation dataset. Please note that for checking best hyperparameters I changed one hyperparameter at a time as running a GridCV by changing all hyperparameters at a time was not possible due to size of the dataset. After I obtained the best features and parameters, I checked for which threshold I obtained the best f1 score on the validation dataset. After obtaining the best hyperparameters, features, and threshold, I predicted the products for an order in the test set.

**Clustering**

I performed Clustering Analysis using K-Means Clustering method to segment similar users together. K-Means Clustering is a form of unsupervised learning which aims to partition the observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as prototype for the cluster.

Doing a clustering analysis gave us a better insight into the kind of products more frequently bought by similar user. Using this data I were able to build our recommendation tool. Our original idea was to segment users on the basis of pairs of products bought by them. Since our data set is too large, I decided to perform clustering analysis on random 2500 users. I calculated the frequency of product pair bought for all the 2500 users.

The issue faced when I were performing this analysis was that the number of product pairs were ¡enter number of product pairs¿, which created a data set of dimensions 2500 x 242793, I couldn’t perform Principal Component Analysis on this data set as it gave any Memory Error on Kaggle Kernels as well as BigRed2. I tried cutting down the number of product pairs by half by taking the top 50 percentile product pairs by frequency. This too threw a Memory Error. In the end I had to change our approach and perform clustering analysis by the number of products bought by a user from a given aisle. The clustering analysis was performed in following steps:

1) I performed customer segmentation on the original 2500 users I selected for clustering by product pairs

2) I created a 2500 x 134 matrix of number of users by aisles(since there are 134 aisles). The value of the matrix were the number of times an user has bought a product from that aisle.

3) I performed the PCA on this matrix to reduce the number of features. I selected the top 12 components obtained from PCA as they explained 85% of the variance in the data Next I chose the pairs(1st component paired with another component since it explains the maximum variance) for which the data was most spread out, these pairs are the potential pairs which can be used to perform clustering analysis. Selection of K To select the suitable value of K for K-Means clustering, I looked at sum of squared distances for each point from the cluster centroid (Elbow Method). The Elbow Curve is shown in the Fig 1. From the ﬁgure I can that curve shows an elbow at k=2. But since I wanted more number of clusters to obtain more and diverse segments of customers, I went ahead with k=4 4) Next I looked at pair of components with the 1st component(since it explains the maximum variance) to check which pair can be chosen to perform clustering analysis. I discovered the best pair was of component 1 and component 9 since it gave the best silhouette score (0.724) for k=4. The cluster are obtained for components 1 and 9 as shown in the ﬁg 2. The cluster are as follows

**Apriori**

Apriori is a association rules mining technique. Apriori algorithm is used to identify frequent item sets (in our case, item pairs). It does so using a ”bottom up” approach, ﬁrst identifying individual items that satisfy a minimum occurrence threshold. It then extends the item set, adding one item at a time and checking if the resulting item set still satisﬁes the speciﬁed threshold. The algorithm stops when there are no more items to add that meet the minimum occurrence requirement.

For the following orders

• order 1: chips,banana bread

• order 2: milk,banana

• order 3: chips, milk,bread

• order 4: banana, chips

• Cluster 0: Yellow

• Cluster 1: Blue

• Cluster 2: Purple

• Cluster 3: Green

• order 5: banana, bread, milk

Apriori will generate these pairs

• pairs: (chips, banana), (chips, bread), (banana, bread)

• pairs: (milk, banana)

A few observations from the cluster analysis:

• Most of the users lie in Cluster0 or Cluster 1

• ’Fresh Fruits’ and ’Fresh vegetables’ are the most important product in all the cluster

• ’Yogurt’ and ’Refrigerated’ products are the most common products for Cluster 3

• Top 10 to 15 products show the distinction between clusters

• Users in Cluster 2 buy the most number of products. they are the most proﬁtable customers

After the Clusters are obtained for each user, I can assign top aisles to each user. Each user is more likely to buy products from the top aisle of his/her cluster

• pairs: (chips, milk), (chips, bread), (milk, bread)

• pairs: (banana, chips)

• pairs: (banana, bread), (banana, milk), (bread, milk)

Once the item sets have been generated using apriori, I can start mining association rules. Given that I are only looking at item sets of size 2, the association rules I will generate will be of the form A - B.

The terms associated with Apriori algorithm areSupport: The percentage of transactions that contain all of the items in an item set. The higher the support the more frequently the item set occurs.

support(banana,chips) = 2/5 or 40%

Conﬁdence: Given two items, A and B, conﬁdence says how likely item B is purchased when item A is purchased, conﬁdence(A-B) = support(A,B) / support(A) conﬁdence(banana-milk) = support(banana,milk) / support(banana) = (2/5) / (4/5)= 50% Lift: Given two items, A and B, lift indicates whether there is a relationship between A and B, or whether the two items are occurring together in the same orders simply by chance. A lift value greater than 1 means that item B is likely to be bought if item A is bought, while a value less than 1 means that item B is unlikely to be bought if item A is bought. lift(A,B) = support(A,B) / (support(A) \* support(B)) lift(banana, bread)= sup(banana,milk)/(sup(banana)\*sup(bread))

=(2/5)/((4/5)\*(3/5))= 5/6= 83%

Our original idea was to implement apriori on user level, but since our data set was too large, I decided to perform apriori on the same 2500 users that was chosen during clustering The algorithm gave support, conﬁdence and lift considering the entire products ordered by the 2500 users,I wanted to calculate these metrics for each user given his orders, but this was not feasible due to memory issue, so instead it was calculated for the product pair for all the orders of the 2500 users The top 3 product pairs are:

• Yogurt, Sheep Milk Strawberry AND Blueberry Sheep Milk Yogurt

• Bamba Peanut Snack AND Bissli Pizza Flavor Snack

• Iced Bhakti Chai Coffee Blend AND Apple Mango Passion Fruit Fruit Snack

E. Recommendation

Our recommendation engine comprises of product recommendations from algorithms: Apriori, light GBM and Segmentation Analysis. Recommendation model is only build for the 2,500 customers on which the Segmentation and Apriori analysis have been performed. In the Instacart app, every product accompanies 11 recommendations therefore I will also recommend 11 products to user for each item placed in cart. Recommendation for a user is performed using following steps-

• Recommending products when user has not placed any order using light GBM. This will tell us if the user is expected to buy a particular product in an order.

• Recommending 11 products to the user based on the last product user has placed in the cart. I take 9 products with highest lift for the products placed in the cart using apriroi algorithm. For a particular order id, I check to which user the order id belongs, I then recommend the top 2 products which have been obtained from each cluster using Clustering Analysis. I combine this two set of products to ﬁnally recommend 11 products to the user for each product placed in the cart.

11 products for each item placed in cart. If user actually bought one or more recommended product then I are considering that set of recommendations as success. Final accuracy will be total success by total number of products ordered. So, a 0.20 accuracy will indicates that 20% of the products bought by the user were also recommended.

 