ML Project Report

▼ PROJECT DESCRIPTION

Instacart has open-sourced anonymized data on customer orders over time in 2017. The dataset can be accessed from the link given in the references.

The data that Instacart opened up include orders of 200,000 Instacart users with each user having between 4 and 100 orders. Instacart indicates each order in the data as prior, train or test. Prior orders describe the past behavior of a user while train and test orders regard the future behavior that we need to predict.

The aim of this project is to predict which previously purchased products (prior orders) will be in a user's next order (train and test orders).

▼ FEATURES PROVIDED IN DATASET

Below are the features which are provided in the dataset.

1. Order number = This represents the order number of the orders placed by the user
2. Order Day of Week = On which day of the week the order was placed
3. Order Hour of Day = In which hour was the order placed
4. Days Since Prior Order = How many days ago was the last order placed by the user

▼ FEATURE ENGINEERING

We have created 3 features from the data provided to us. They are as below.

1. User Reorder Ratio = Number of times the user has purchased products which were reorders divided by total number of times the users has purchased products
2. Product Reorder Ratio = Number of times the product was purchased as a reorder divided by the total number of times the product was purchased
3. User Product Reorder Ratio = For a given user and product combination, how many times has the user reordered this product divided by the total number of times the user has purchased the product

We have also used PCA and then K-Means Clustering to cluster users as per their purchasing behaviors

▼ GENERAL APPROACH

We have created feature vectors for each user_id and product_id pair and added to the training data along with the reordered column which represents reordered with 1 and not reordered with 0.

We have trained a XGBoost classifier using the above data

▼ Experiment 1

Approach	XGBoost	XGBoost
Accuracy %	89.71	89.99
User Data for Training	50,000	100,000
aisles	134	134
departments	21	21
products	49,688	49,688
orders	831,792	1,665,732
order_products_prior	7,882,503	15,806,241
order_products_train	334,239	672,294
max tree depth	6	6
Feature Vectors	u_reordered_ratio	u_reordered_ratio
	p_reordered_ratio	p_reordered_ratio
	uxp_reordered_ratio	uxp_reordered_ratio

▼ Experiment 2

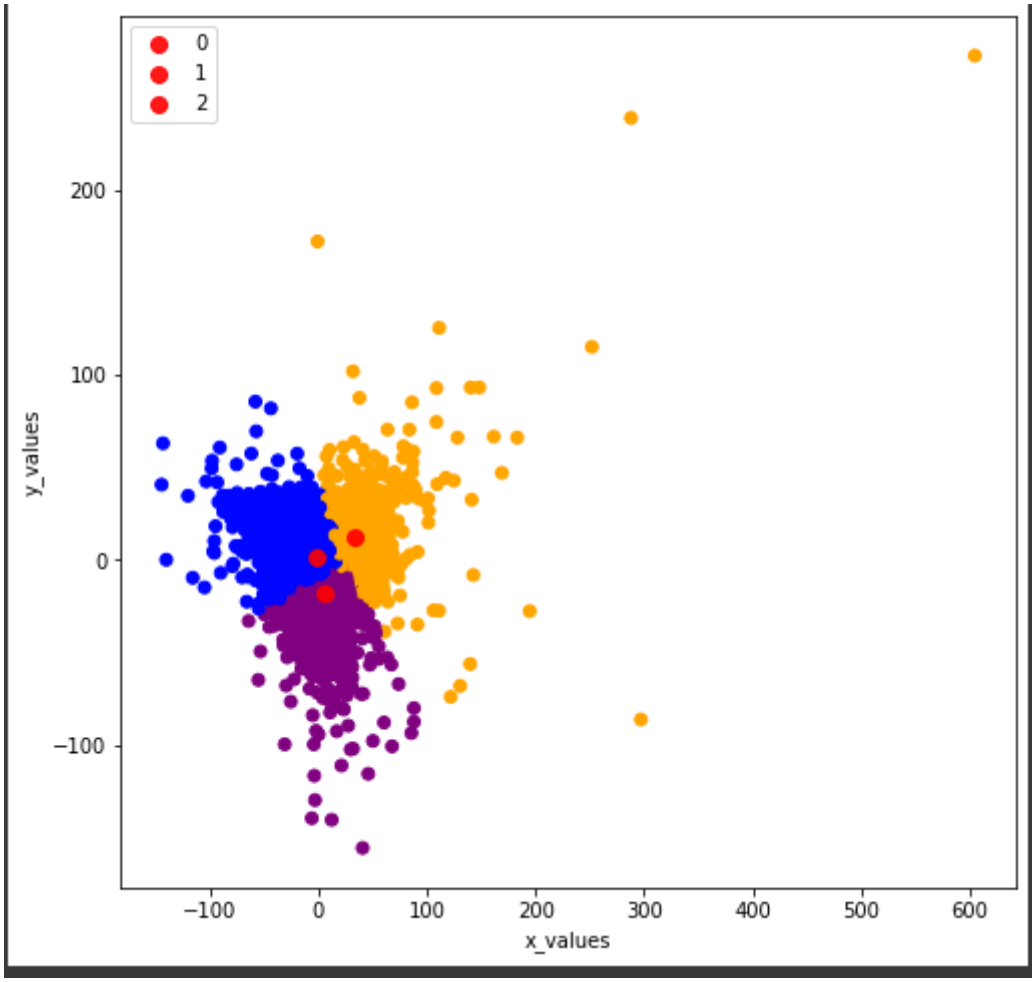
Approach	XGBoost	XGBoost
Accuracy %	95.93	95.63
User Data for Training	50,000	100,000
aisles	134	134
departments	21	21
products	49,688	49,688
orders	831,792	1,665,732
order_products_prior	7,882,503	15,806,241
order_products_train	334,239	672,294
max tree depth	6	6
Feature Vectors	u_reordered_ratio	u_reordered_ratio
	p_reordered_ratio	p_reordered_ratio
	uxp_reordered_ratio	uxp_reordered_ratio
	order_number	order_number
	order_dow	order_dow
	order_hour_of_day	order_hour_of_day
	days_since_prior_order	days_since_prior_order

▼ Experiment 3

Approach	XGBoost	XGBoost
Accuracy %	100	100
User Data for Training	50,000	100,000
aisles	134	134
departments	21	21
products	49,688	49,688
orders	831,792	1,665,732
order_products_prior	7,882,503	15,806,241
order_products_train	334,239	672,294
max tree depth	6	6
Feature Vectors		
	order_number	order_number
	order_dow	order_dow
	order_hour_of_day	order_hour_of_day
	days_since_prior_order	days_since_prior_order

▼ Experiment 4

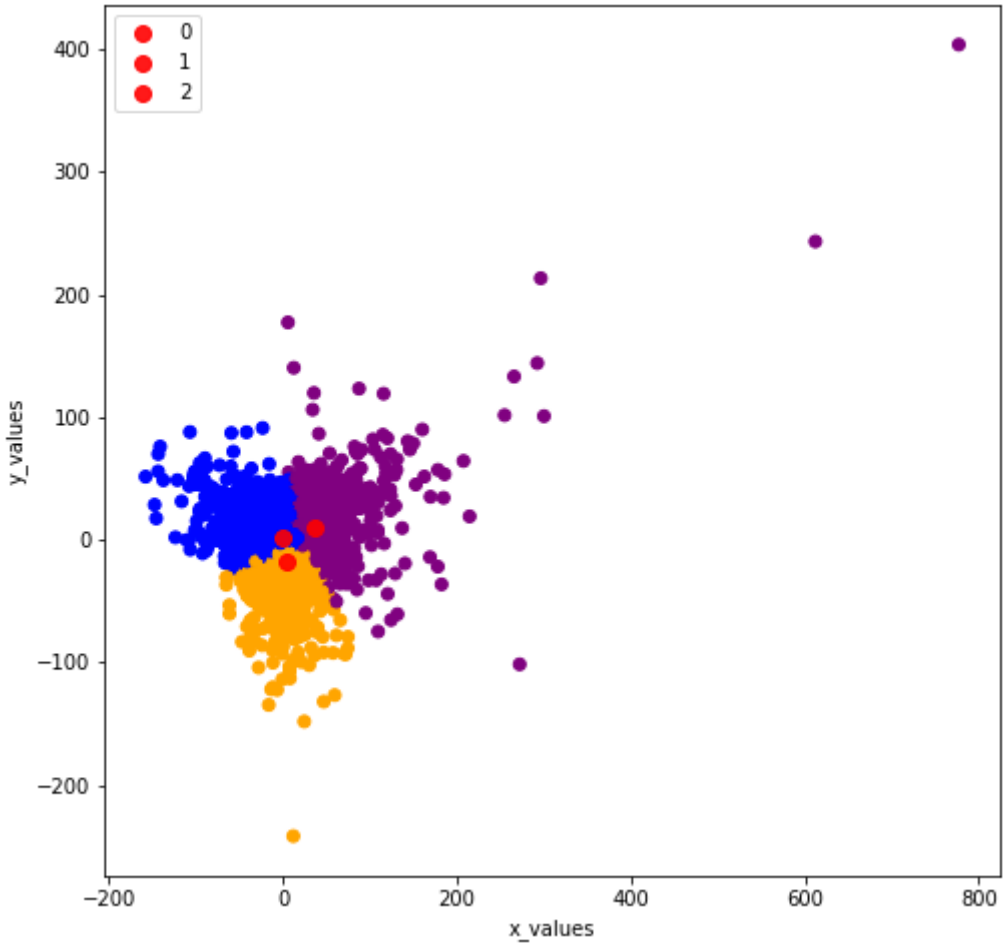
I have used PCA, K-Means Clustering and XGBoost for this experiment.



Approach	PCA + XGBoost	PCA + XGBoost	PCA + XGBoost
Accuracy %	59.7	93.39	93.55
User Data for Training	50,000	100,000	100,000
aisles	134	134	134
departments	21	21	21
products	49,688	49,688	49,688
orders	831,792	1,665,732	1,665,732
order_products_prior	7,882,503	15,806,241	15,806,241
order_products_train	334,239	672,294	672,294
max tree depth	6	6	6
cluster count	3	3	3
PCA components	4	4	4
PC1	1	1	2
PC2	3	3	3
Feature Vectors	u_reordered_ratio	u_reordered_ratio	u_reordered_ratio
	p_reordered_ratio	p_reordered_ratio	p_reordered_ratio
	uxp_reordered_ratio	uxp_reordered_ratio	uxp_reordered_ratio
	cluster_id	cluster_id	cluster_id

▼ Experiment 5

I have used PCA, K-Means Clustering and XGBoost for this experiment.


0
1
2
400
300
200
100
0
−100
−200
y_values
−200
0
200
400
600
800
x_values

Approach	PCA + XGBoost	PCA + XGBoost
Accuracy %	100	100
User Data for Training	50,000	50,000
aisles	134	134
departments	21	21
products	49,688	49,688
orders	831,792	831,792
order_products_prior	7,882,503	7,882,503
order_products_train	334,239	334,239
max tree depth	6	6
cluster count	3	3
PCA components	4	4
PC1	1	2
PC2	3	3
Feature Vectors		
	order_number	order_number
	order_dow	order_dow
	order_hour_of_day	order_hour_of_day
	days_since_prior_order	days_since_prior_order
	cluster_id	cluster_id

▼ CONCLUSION

1. The results show that the dataset is heavily biased and has an unbalanced amount of data which was reordered.
2. The number of scenarios where the label is 0 are very few. Majority of them are 1.
3. This becomes more of a single classifier problem than binary classification.
4. To fix this issue we must take an approach where we extract information about a product's frequency of purchase along with time interval between purchases. This would help to predict for the products which get reordered more frequently.
5. We could could also have used time series for this data set.

▼

REFERENCES

"The Instacart Online Grocery Shopping Dataset 2017", Accessed from https://www.instacart.com/datasets/grocery-shopping-2017 on 25th April, 2022