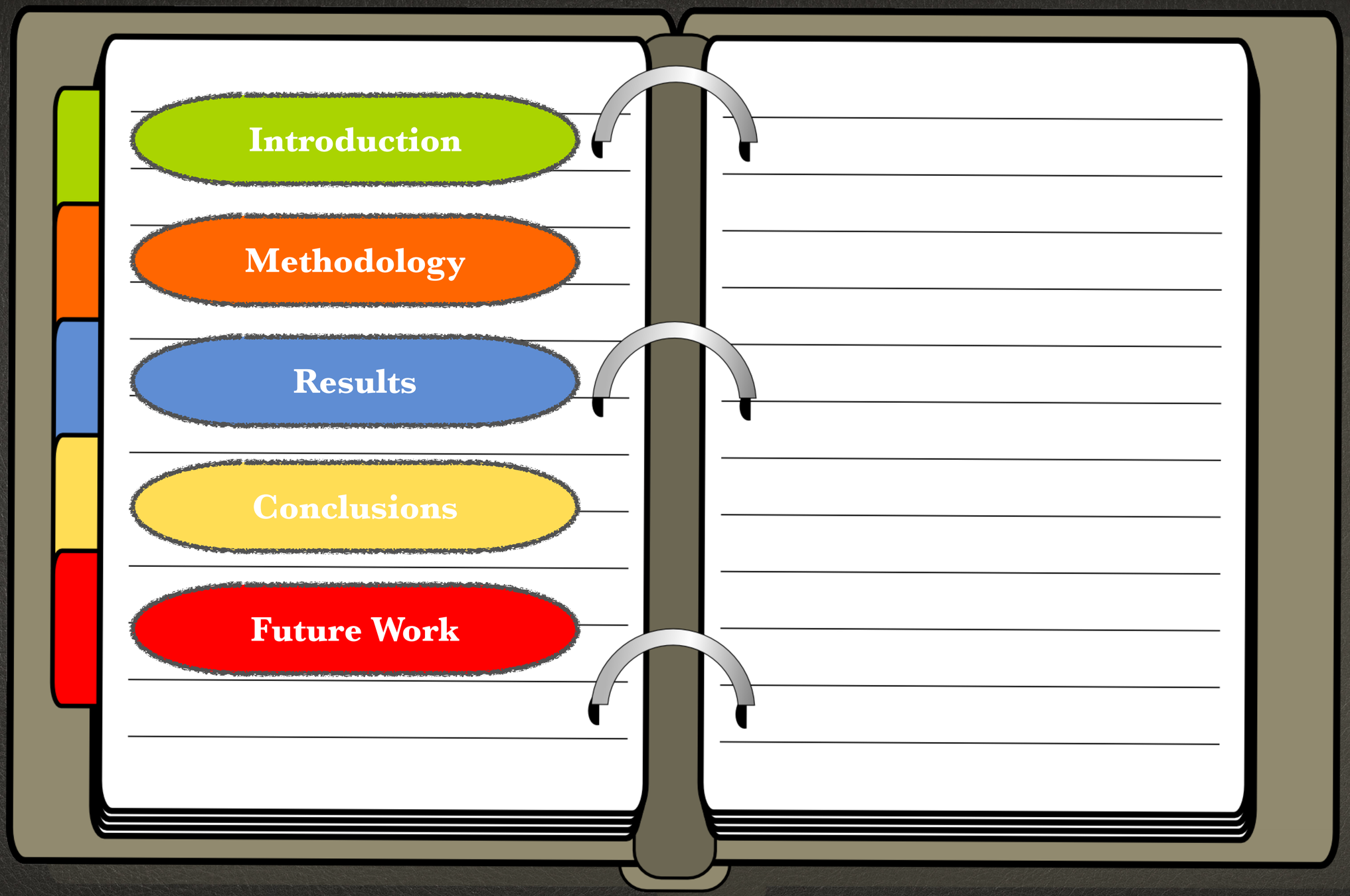
PREDICTING CUSTOMER BEHAVIOURS AND MARKET BASKET ANALYSIS

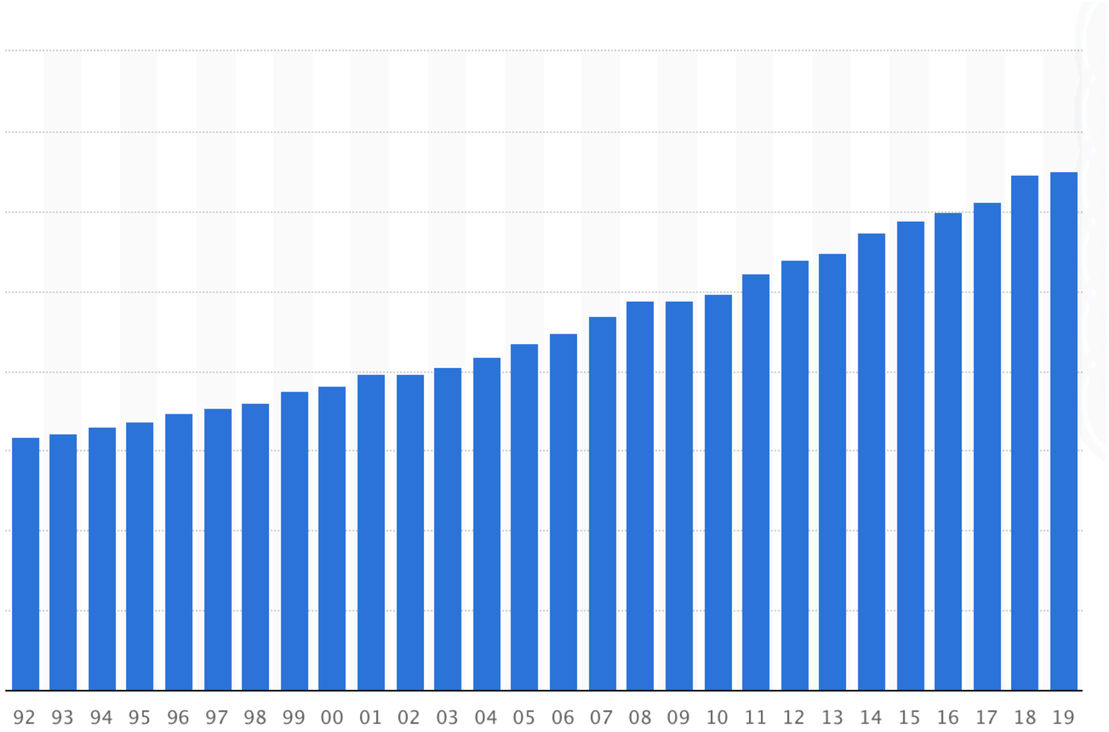
YALIN YENER

September 04, 2020


Introduction
Methodology
Results
Conclusions
Future Work

Business Need:	Predicting whether a customer will reorder a product.
Solution:	Using **Instacart** dataset, finding customer behaviour by using market basket analysis techniques and try to predict next order.
Objective and Goal:	▪ CSV to PostgreSQL. ▪ Join, Groupby, Having etc. ▪ Connect PostgreSQL with Jupyter Notebook ▪ Exploratory Data Analysis. ▪ Building ML Algorithms ▪ Feature Engineering. ▪ Evaluate Model

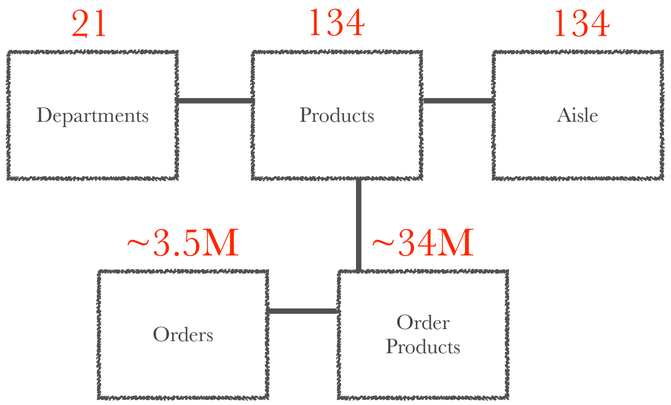
"In 2019, U.S. supermarket and grocery store sales amounted to about **1.5 trillion** U.S. dollars.

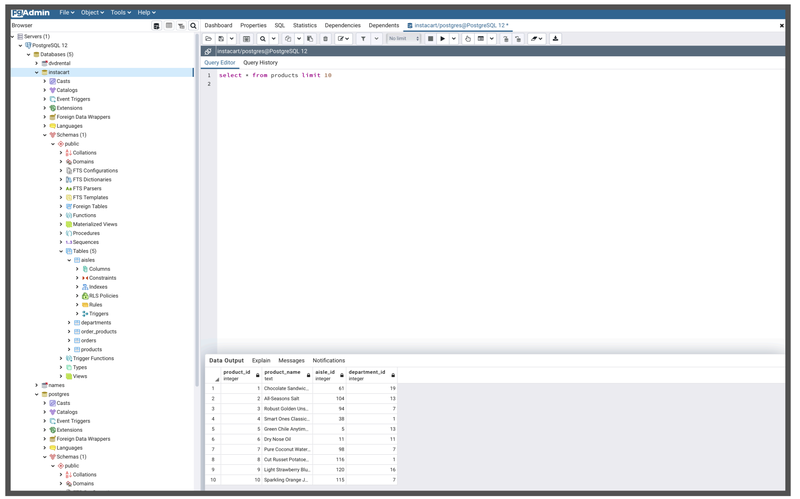


–IGD Retail Analysis, 2019

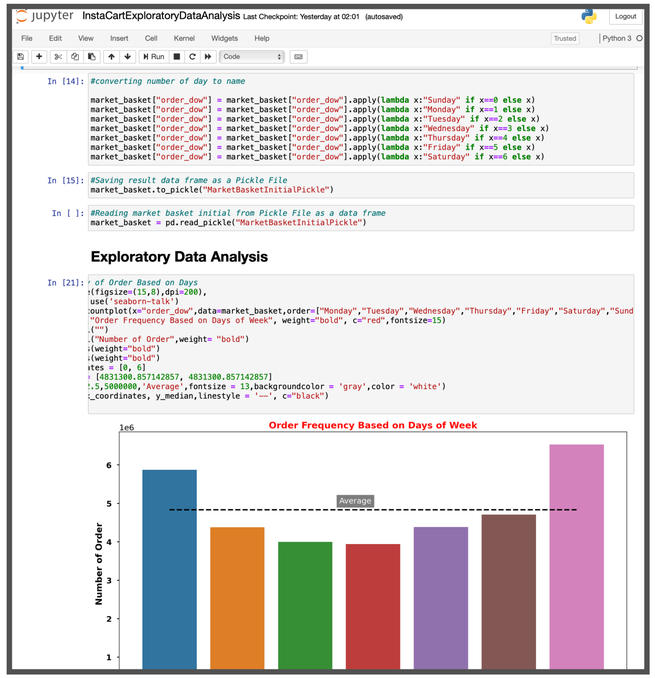

instacart

CSV


21
134
134
Departments
Products
Aisle
~3.5M
~34M
Orders
Order Products


pgAdmin
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
select * from products limit 10
Data Output
Explain
Messages
Notifications

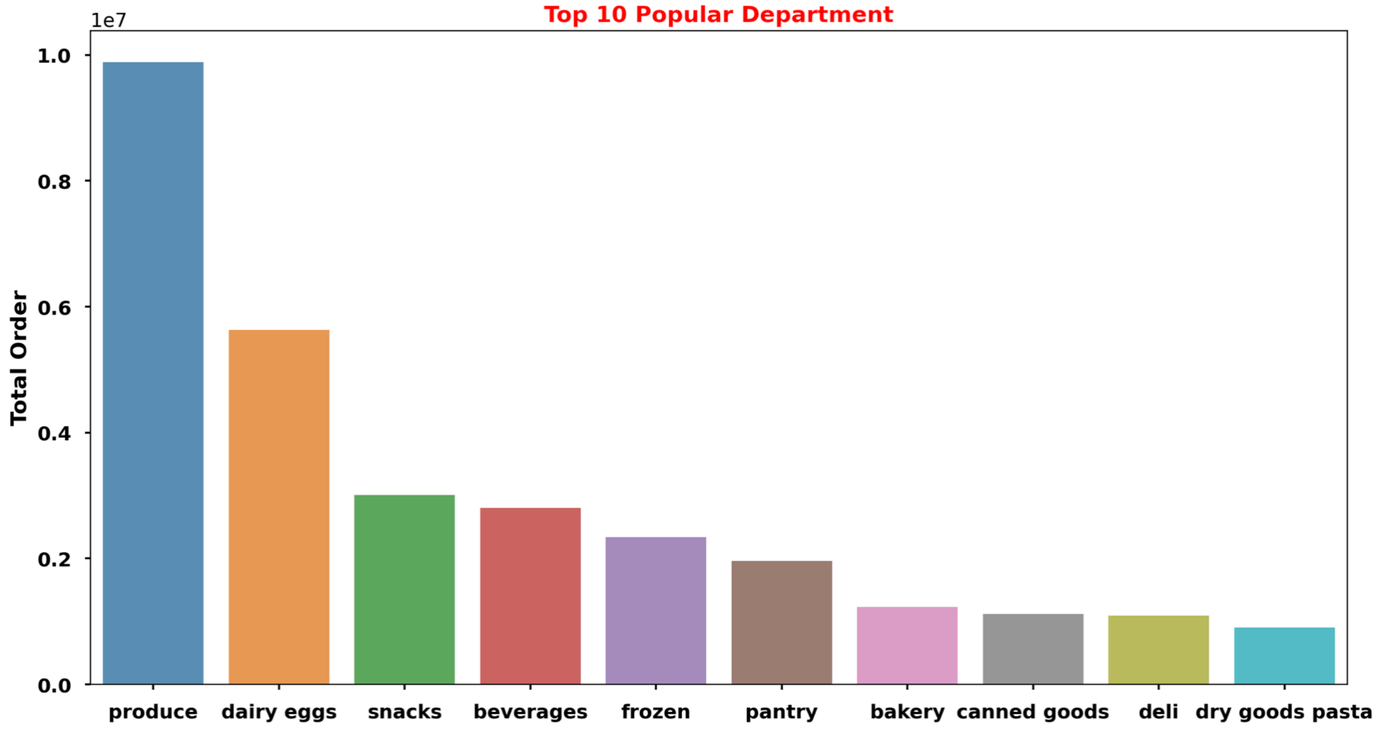

jupyter


jupyter InstaCartExploratoryDataAnalysis Last Checkpoint: Yesterday at 02:01 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
In [14]: #converting number of day to name
market_basket["order_dow"] = market_basket["order_dow"].apply(lambda x:"Sunday" if x==0 else x)
market_basket["order_dow"] = market_basket["order_dow"].apply(lambda x:"Monday" if x==1 else x)
market_basket["order_dow"] = market_basket["order_dow"].apply(lambda x:"Tuesday" if x==2 else x)
market_basket["order_dow"] = market_basket["order_dow"].apply(lambda x:"Wednesday" if x==3 else x)
market_basket["order_dow"] = market_basket["order_dow"].apply(lambda x:"Thursday" if x==4 else x)
market_basket["order_dow"] = market_basket["order_dow"].apply(lambda x:"Friday" if x==5 else x)
market_basket["order_dow"] = market_basket["order_dow"].apply(lambda x:"Saturday" if x==6 else x)
In [15]: #Saving result data frame as a Pickle File
market_basket.to_pickle("MarketBasketInitialPickle")
In [ ]: #Reading market basket initial from Pickle File as a data frame
market_basket = pd.read_pickle("MarketBasketInitialPickle")
Exploratory Data Analysis
In [21]: of Order Based on Days
(figsize=(15,8),dpi=200),
use('seaborn-talk')
ountplot(x="order_dow",data=market_basket,order=["Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sund
"Order Frequency Based on Days of Week", weight="bold", c="red",fontsize=15)
("")
("Number of Order",weight= "bold")
(weight="bold")
(weight="bold")
tes = [0, 6]
[4831300.857142857, 4831300.857142857]
5,5000000,'Average',fontsize = 13,backgroundcolor = 'gray',color = 'white')
coordinates, y_median,linestyle = '--', c="black")
Order Frequency Based on Days of Week
1e6
Average
Number of Order
1
2
3
4
5
6

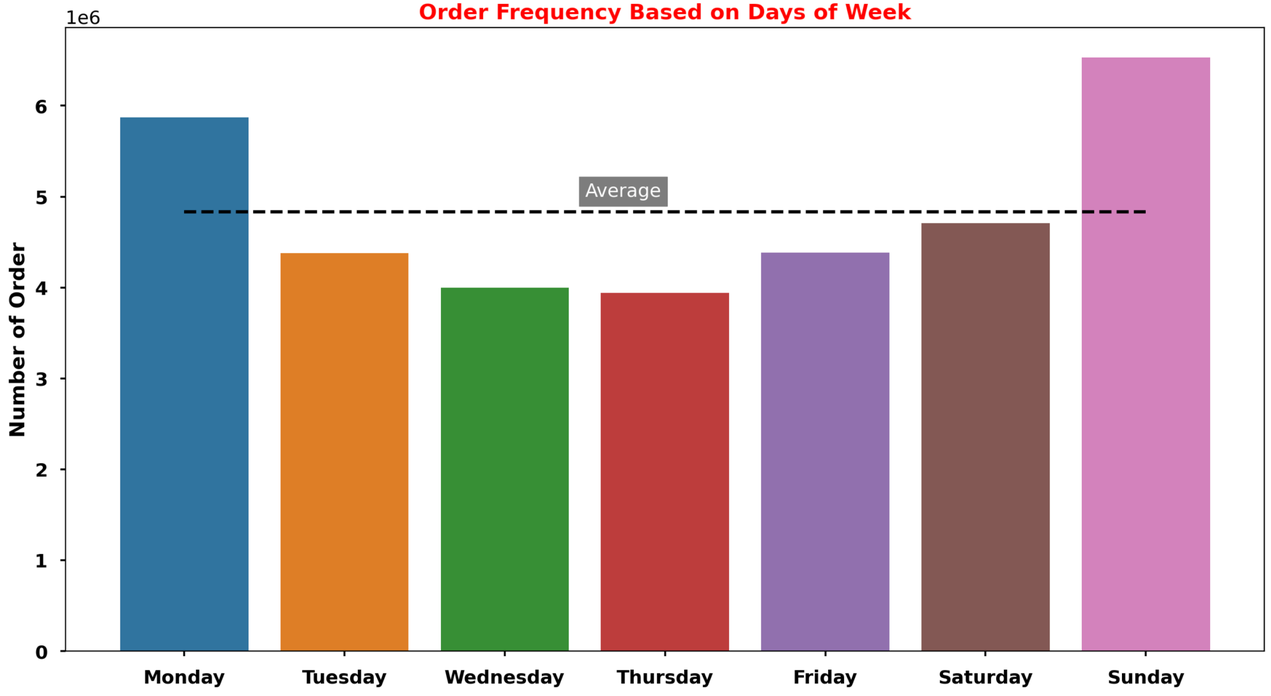
user_id	order_id	order_number	order_dow	order_hour_of_day	days_since_prior_order	product_name	add_to_cart_order	reordered	department	aisle
112108	1	4	4	10	9.0	Organic Celery Hearts	3	0	produce	fresh vegetables
112108	1	4	4	10	9.0	Organic 4% Milk Fat Whole Milk Cottage Cheese	2	1	dairy eggs	other creams cheeses
112108	1	4	4	10	9.0	Bag of Organic Bananas	6	0	produce	fresh fruits
112108	1	4	4	10	9.0	Organic Whole String Cheese	8	1	dairy eggs	packaged cheese
112108	1	4	4	10	9.0	Lightly Smoked Sardines in Olive Oil	5	1	canned goods	canned meat seafood

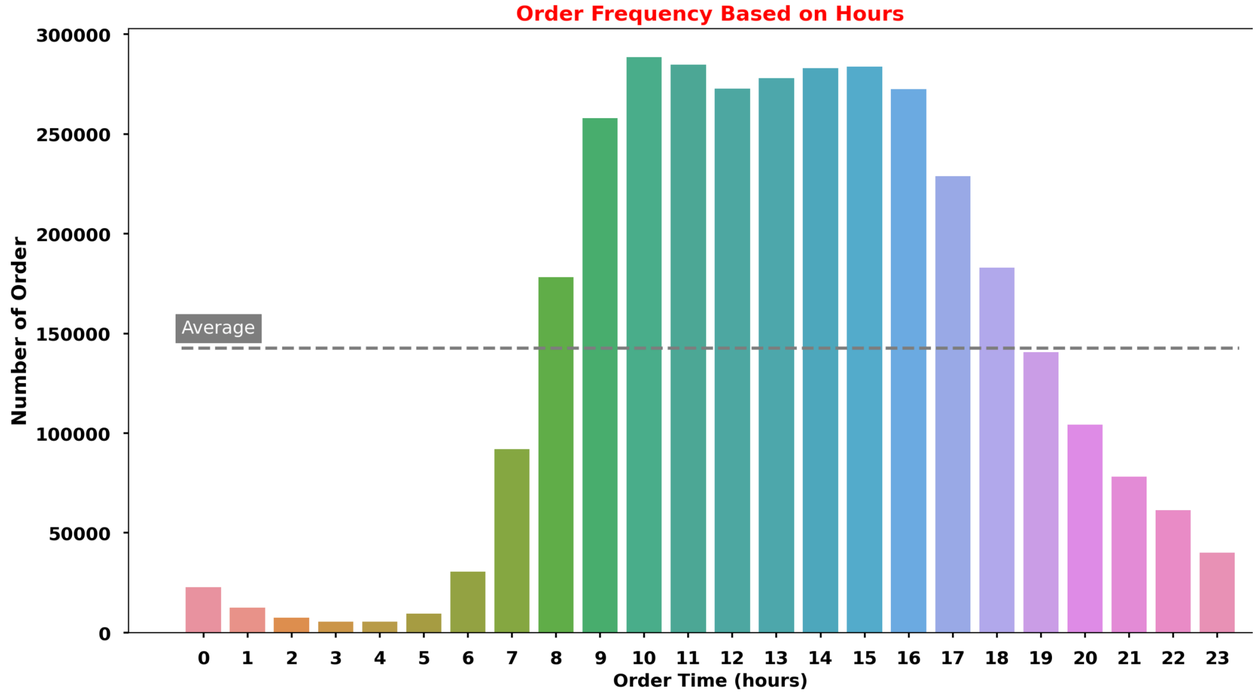
Exploratory Data Analysis

	product_name	frequency_count
0	Banana	491291
1	Bag of Organic Bananas	394930
2	Organic Strawberries	275577
3	Organic Baby Spinach	251705
4	Organic Hass Avocado	220877
5	Organic Avocado	184224
6	Large Lemon	160792
7	Strawberries	149445
8	Limes	146660
9	Organic Whole Milk	142813

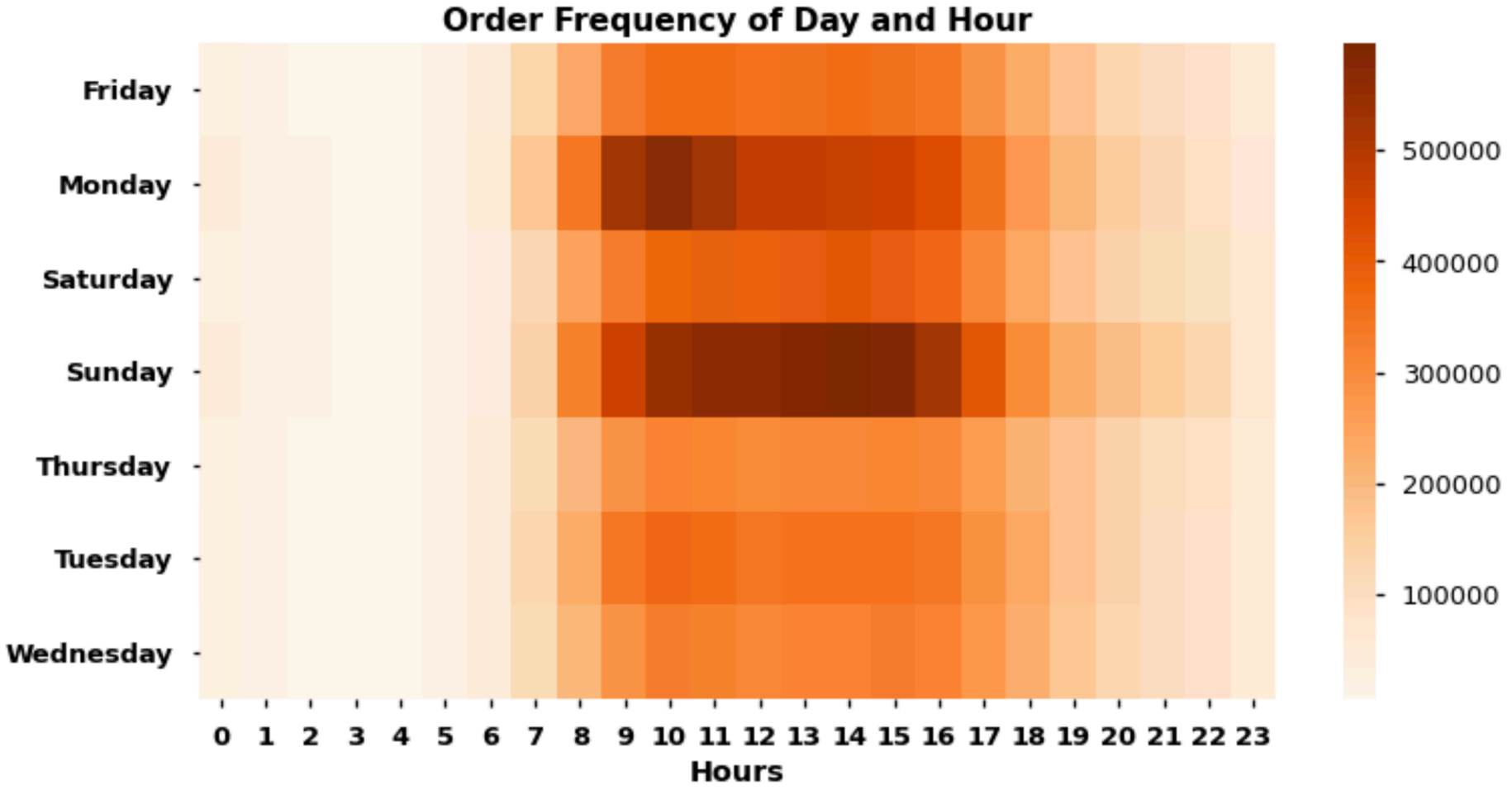


Exploratory Data Analysis

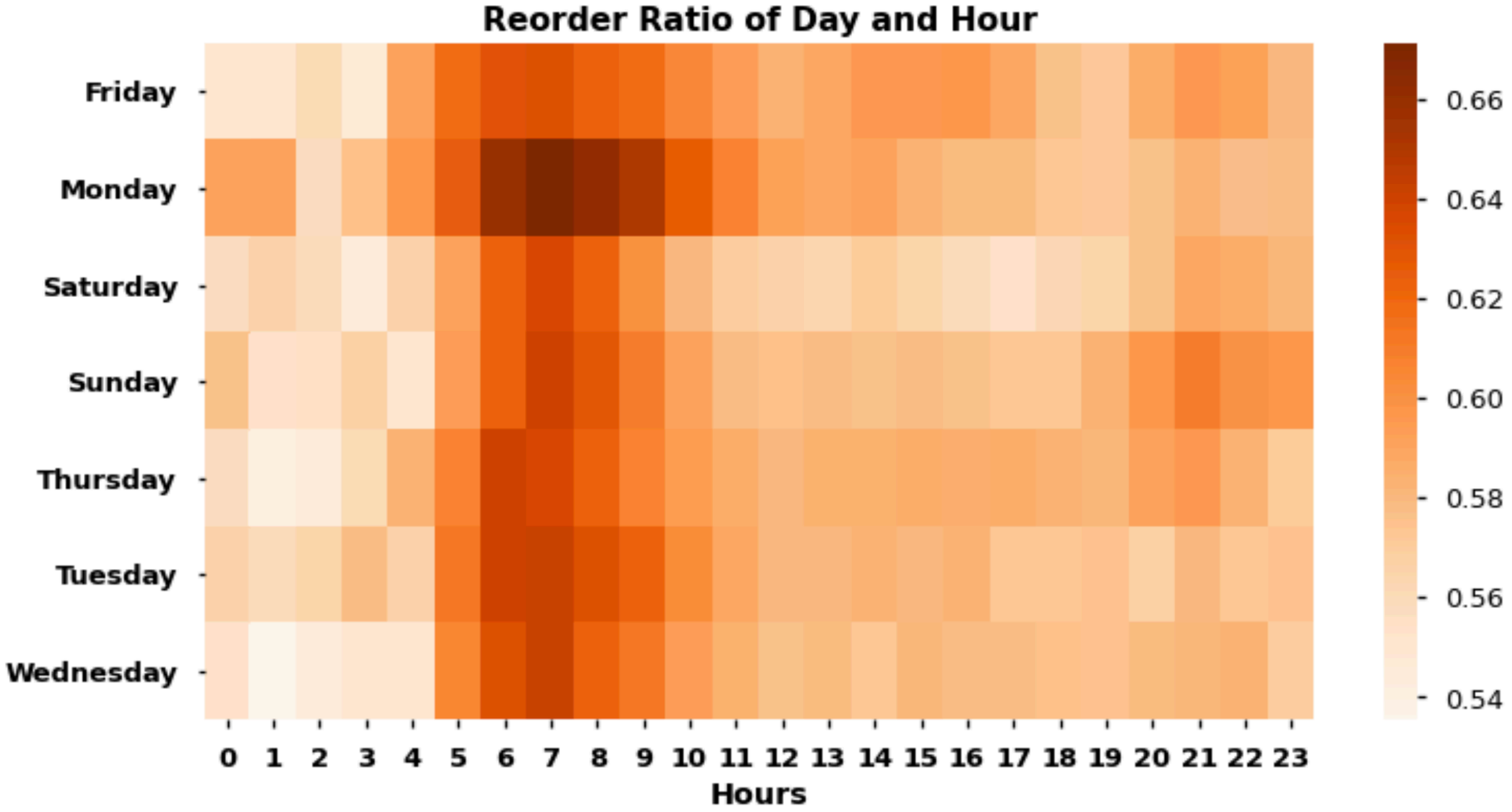




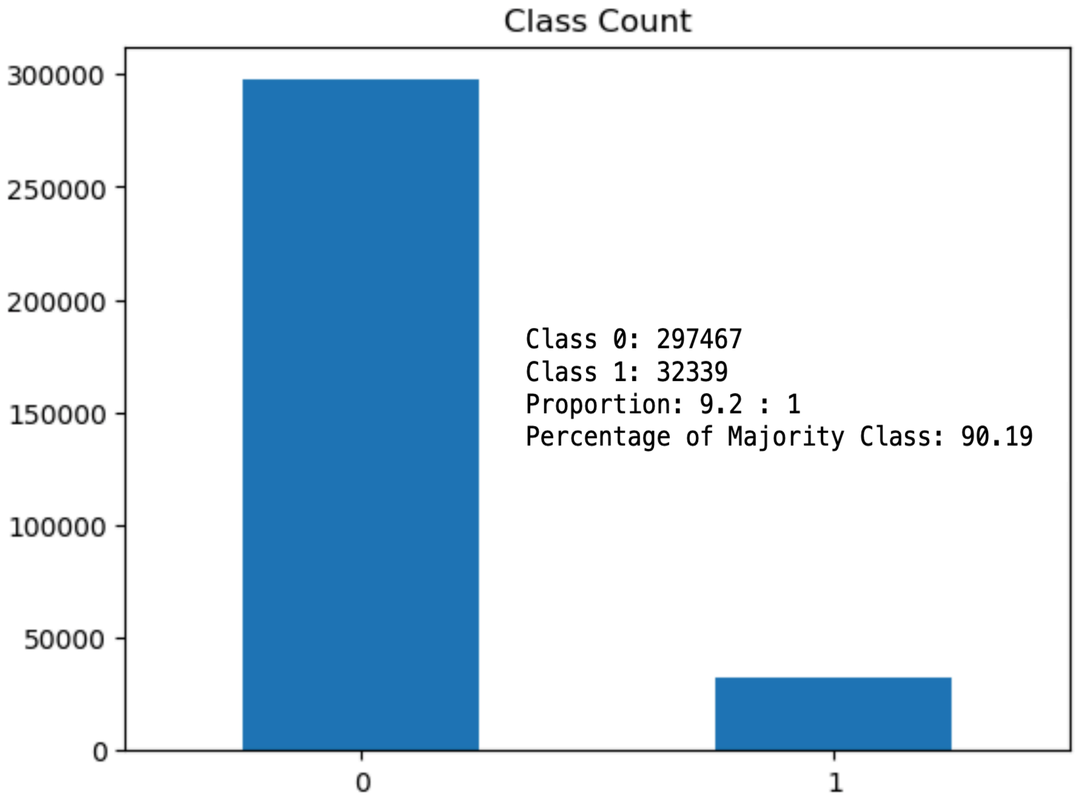
Exploratory Data Analysis



Exploratory Data Analysis



```
['product_id', 'user_id', 'user_product_total_orders', 'latest_cart',
'in_cart'  'product_total_orders', 'product_avg_add_to_cart_order',
 'user_total_orders', 'user_avg_cartsize', 'user_total_products',
...
'spreads', 'tea', 'tofu meat alternatives', 'tortillas flat bread',
'trail mix snack mix', 'trash bags liners', 'vitamins supplements',
'water seltzer sparkling water', 'white wines', 'yogurt'],
```



Baseline Classification - V1

Product Features -V2

- Product total orders
- Product average add to cart order

User Features -V3

- User total order
- User average chart size
- User total product
- User average days since last order

User - Product - V4

- User product average add to cart order
- User product order frequency

Category - Dummy Variables -V5

Day of Week - Dummy Variables -V6

Features Importance -V7

Resampled -V8

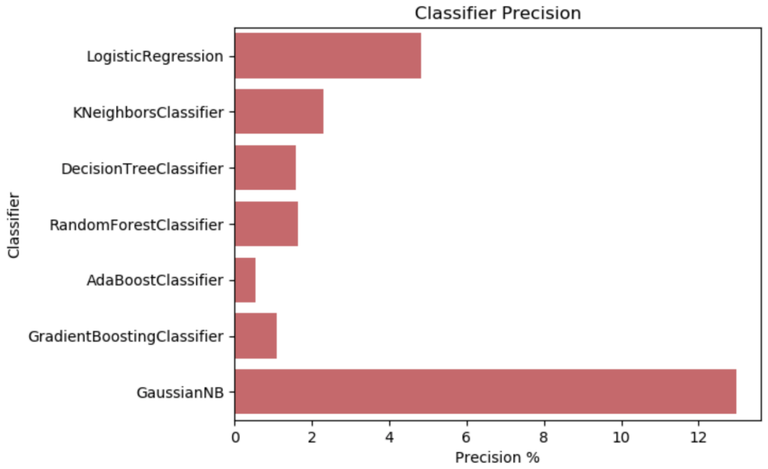
Smoted - V9

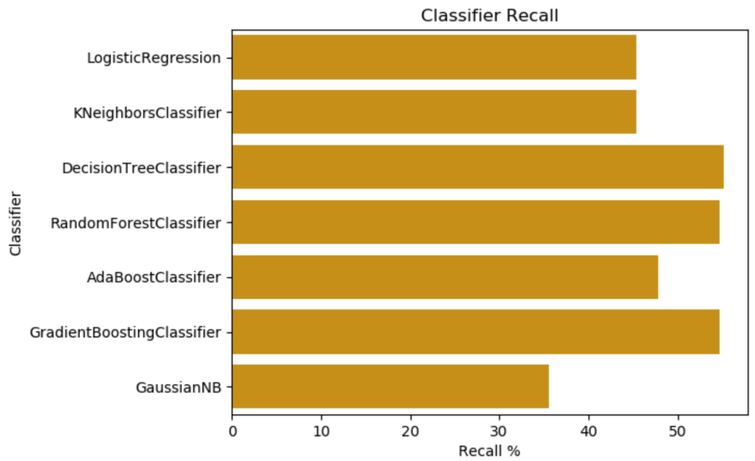
Cross Validation - V10

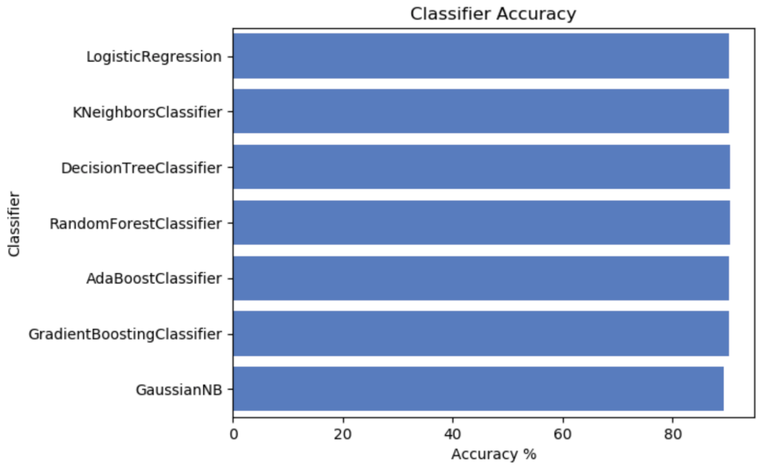
Baseline Classification - V1- Metrics Comparasion

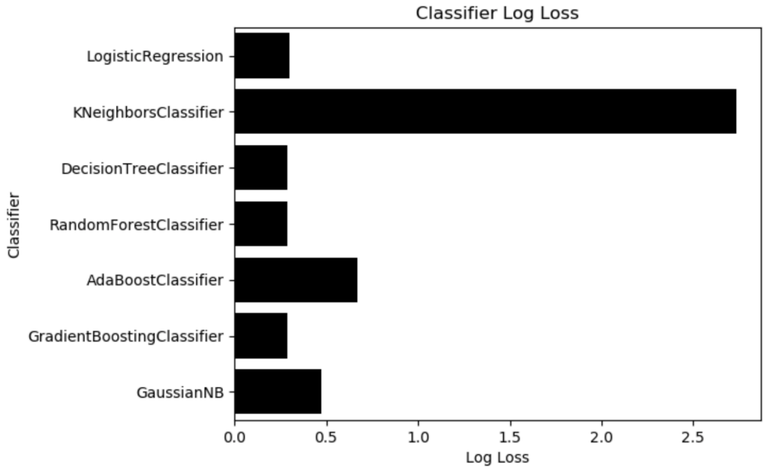
Classifier	Accuracy	Precision	Recall	F1Score	ROC AUC	Log Loss
LogisticRegression	90.24	4.83	45.43	8.73	68.06	0.30
KNeighborsClassifier	90.28	2.30	45.37	4.39	67.94	2.73
DecisionTreeClassifier	90.36	1.60	55.14	3.11	72.80	0.29
RandomForestClassifier	90.36	1.65	54.69	3.20	72.57	0.29
AdaBoostClassifier	90.32	0.53	47.89	1.05	69.13	0.67
GradientBoostingClassifier	90.35	1.08	54.76	2.12	72.59	0.29
GaussianNB	89.31	12.96	35.57	19.00	63.42	0.48

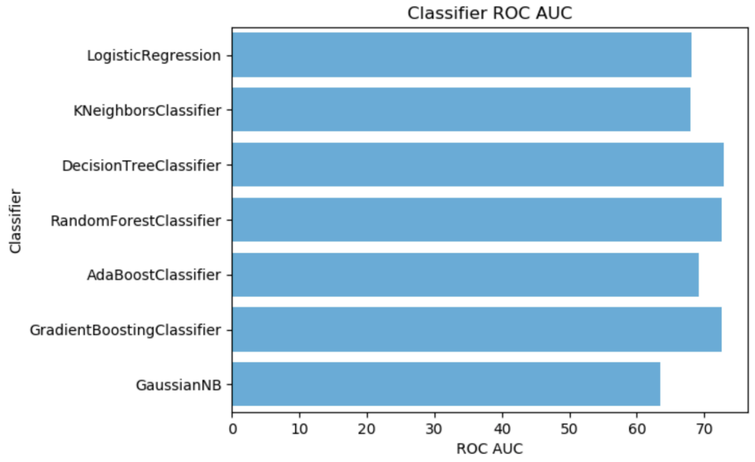
Baseline Classification - V1- Metrics Comparison

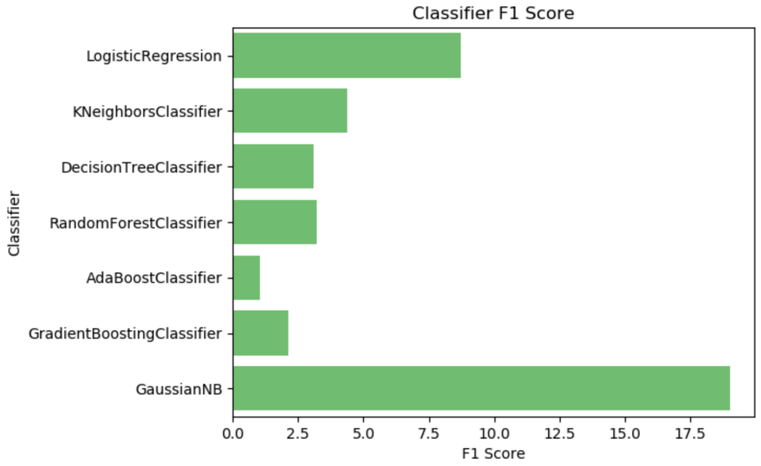




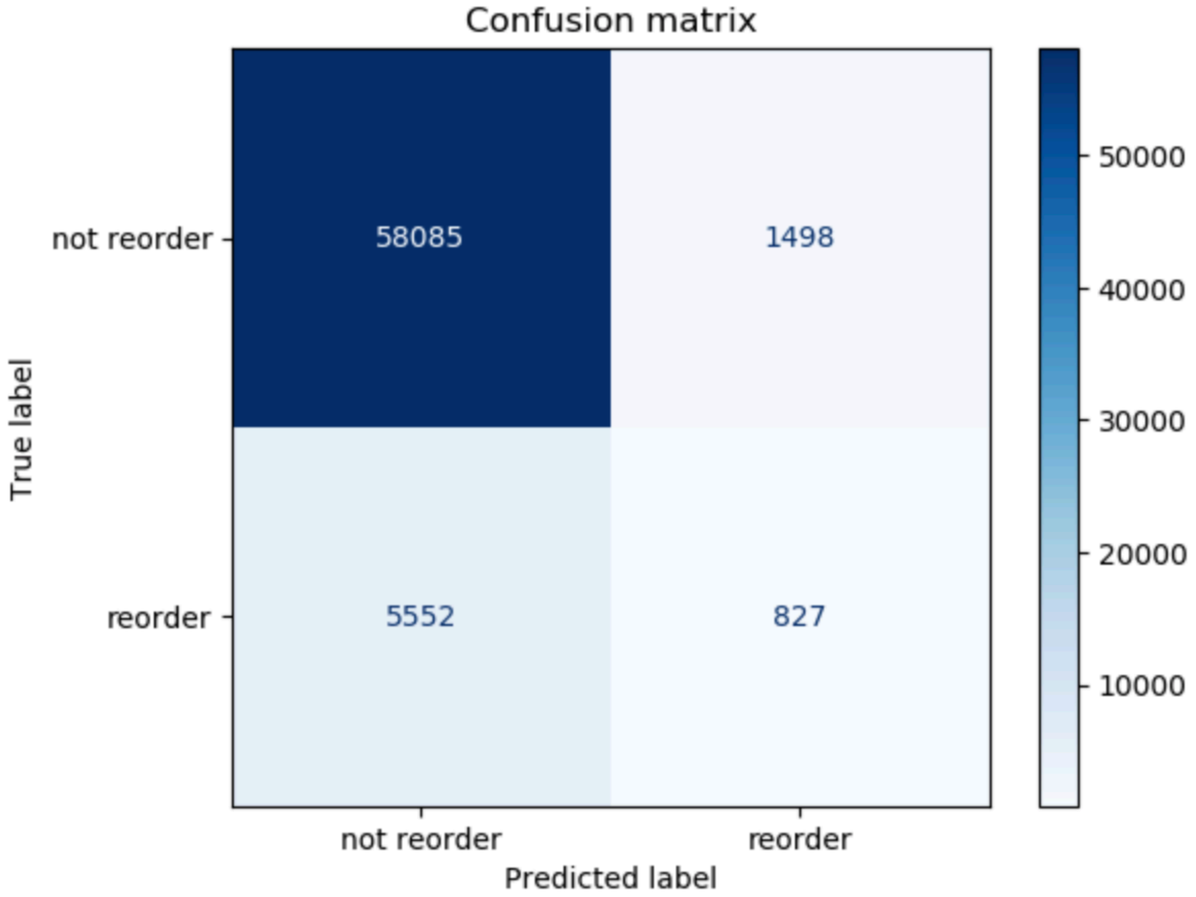


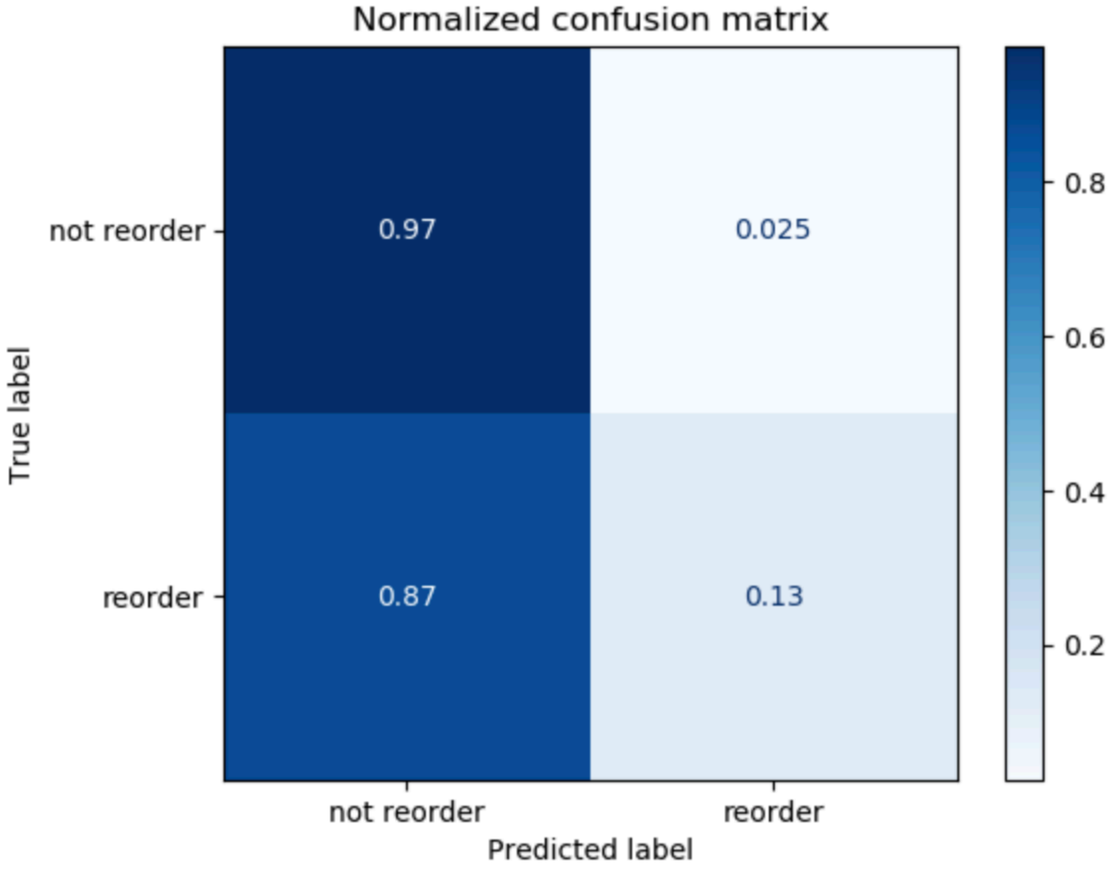






Baseline Classification - V1 - Confusion Matrix





Baseline Classification - V1

Product Features -V2

- Product total orders
- Product average add to cart order

User Features -V3

- User total order
- User average chart size
- User total product
- User average days since last order

User - Product - V4

- User product average add to cart order
- User product order frequency

Category - Dummy Variables -V5 ✓

Day of Week - Dummy Variables -V6

Features Importance -V7

Resampled -V8

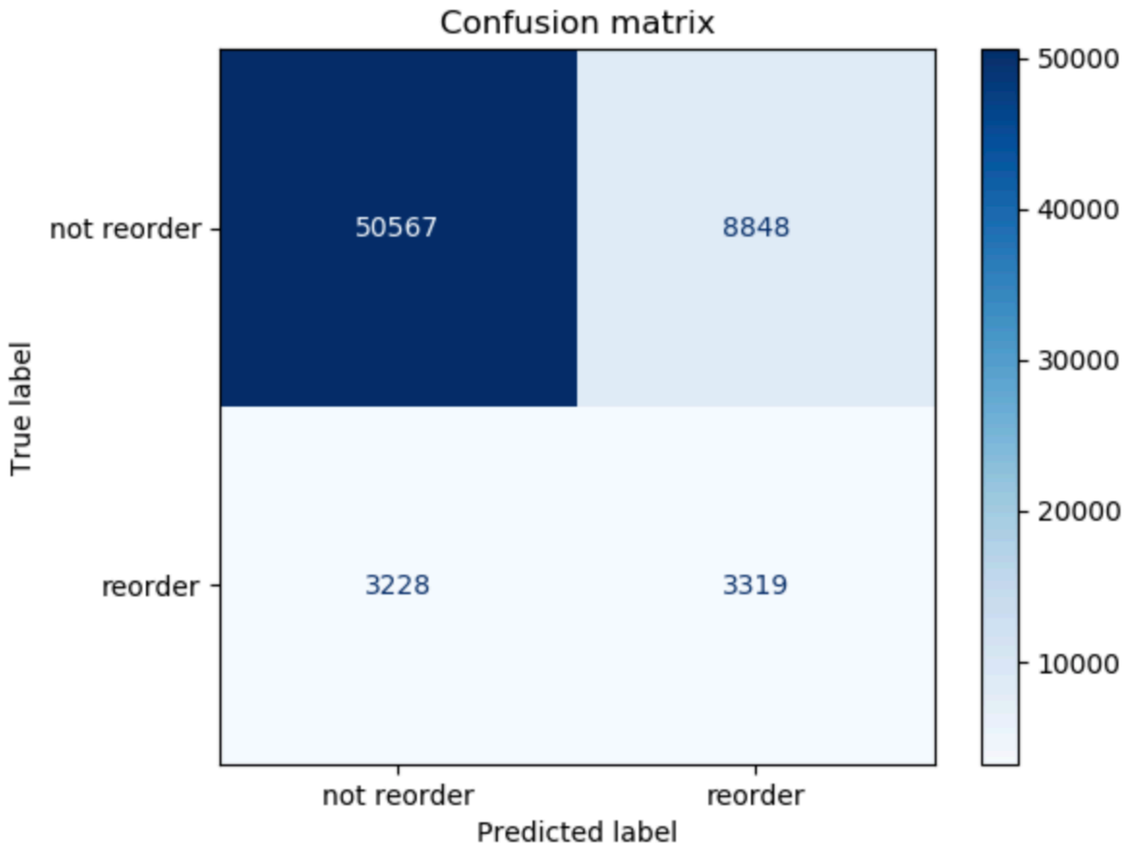
Smoted - V9

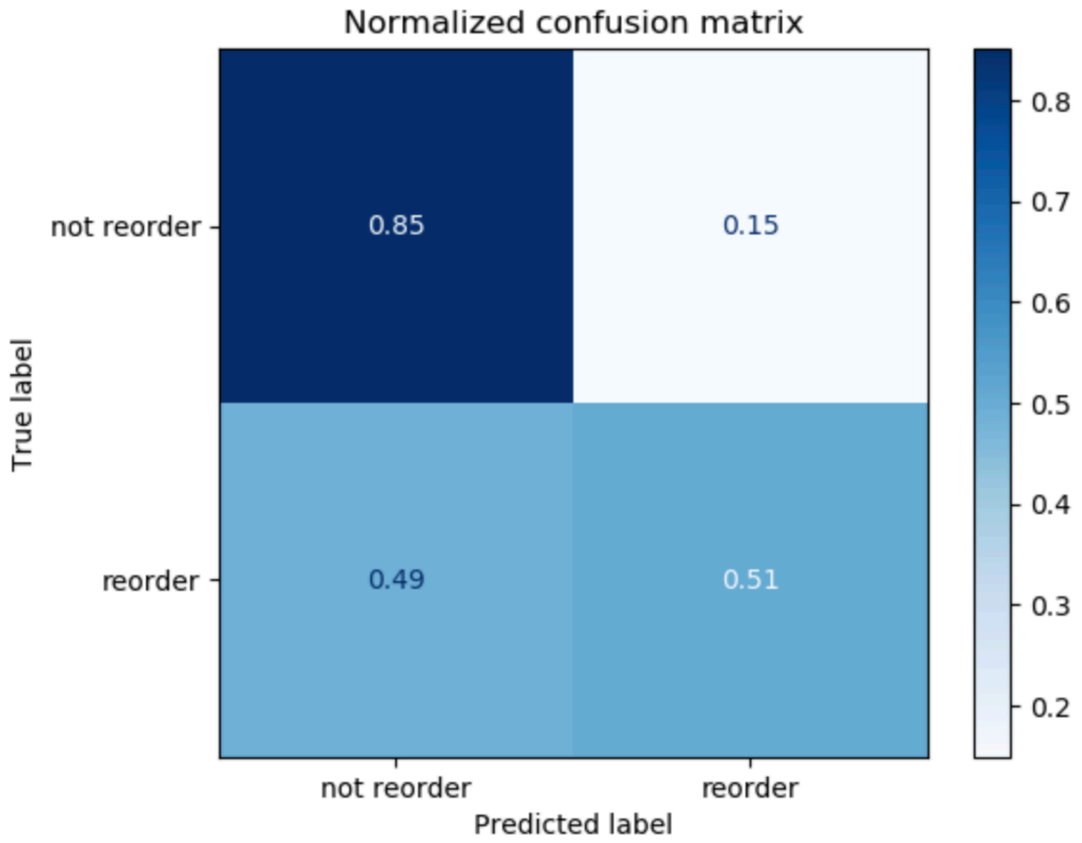
Cross Validation - V10

Dummy Variables - V5- Metrics Comparasion

Classifier	Accuracy	Precision	Recall	F1Score	ROC AUC	Log Loss
LogisticRegression	90.202	7.912	54.412	13.815	72.569	0.287
KNeighborsClassifier	88.666	11.181	30.589	16.376	60.721	2.240
DecisionTreeClassifier	84.145	26.195	23.362	24.698	57.560	5.476
RandomForestClassifier	90.460	13.991	58.048	22.548	74.651	0.317
AdaBoostClassifier	90.455	10.753	60.847	18.276	75.915	0.669
GradientBoostingClassifier	90.529	12.693	61.013	21.014	76.082	0.266
GaussianNB	81.692	50.695	27.279	35.471	60.639	1.049

Category - Dummy Variables -V5 Confusion Matrix





Features Importance -V7

```
Feature Importance:
user_product_total_orders : (0.171338)
product_total_orders : (0.138225)
product_avg_add_to_cart_order : (0.125102)
user_total_orders : (0.123267)
user_avg_cartsize : (0.110542)
user_total_products : (0.103790)
user_avg_days_since_prior_order : (0.090292)
user_product_avg_add_to_cart_order : (0.032270)
user_product_order_freq : (0.019524)
alcohol : (0.010369)
babies : (0.009049)
bakery : (0.008171)
beverages : (0.007387)
breakfast : (0.006776)
bulk : (0.005606)
canned goods : (0.005013)
dairy eggs : (0.004585)
deli : (0.004124)
dry goods pasta : (0.003988)
frozen : (0.003841)
household : (0.003613)
international : (0.003470)
meat seafood : (0.002489)
missing : (0.001819)
other : (0.001761)
pantry : (0.001071)
personal care : (0.001013)
pets : (0.000898)
produce : (0.000333)
snacks : (0.000273)
```

Classifier	Accuracy	Precision	Recall	F1Score	ROC AUC	Log Loss
LogisticRegression	90.208	7.454	54.955	13.127	72.822	0.287
KNeighborsClassifier	88.665	10.997	30.380	16.149	60.608	2.242
DecisionTreeClassifier	84.618	27.860	25.169	26.446	58.563	5.312
RandomForestClassifier	90.240	15.457	52.846	23.919	72.102	0.392
AdaBoostClassifier	90.373	8.798	60.314	15.356	75.564	0.669
GradientBoostingClassifier	90.507	12.632	60.409	20.894	75.777	0.266
GaussianNB	88.404	20.040	35.212	25.543	63.400	0.628

Resampled -V8

Classifier	Accuracy	Precision	Recall	F1Score	ROC AUC	Log Loss
LogisticRegression	72.831	68.199	21.991	33.258	58.715	0.583
KNeighborsClassifier	93.099	99.969	58.991	74.198	79.493	0.107
DecisionTreeClassifier	99.974	100.000	99.741	99.870	99.871	0.001
RandomForestClassifier	99.974	100.000	99.741	99.870	99.871	0.042
AdaBoostClassifier	72.236	70.628	22.003	33.553	58.862	0.687
GradientBoostingClassifier	73.648	69.986	22.911	34.521	59.318	0.544
GaussianNB	84.161	38.613	28.224	32.611	60.587	0.783

SMOTE -V9

Classifier	Accuracy	Precision	Recall	F1Score	ROC AUC	Log Loss
LogisticRegression	70.568	70.490	20.885	32.223	58.240	0.593
KNeighborsClassifier	89.539	99.389	48.680	65.351	74.302	0.169
DecisionTreeClassifier	99.973	99.756	99.969	99.862	99.971	0.000
RandomForestClassifier	99.973	99.832	99.893	99.862	99.937	0.090
AdaBoostClassifier	71.267	70.612	21.351	32.788	58.505	0.687
GradientBoostingClassifier	71.327	70.979	21.454	32.949	58.583	0.553
GaussianNB	82.155	43.379	26.045	32.548	59.656	0.813

SMOTE -V9

KNN-CV

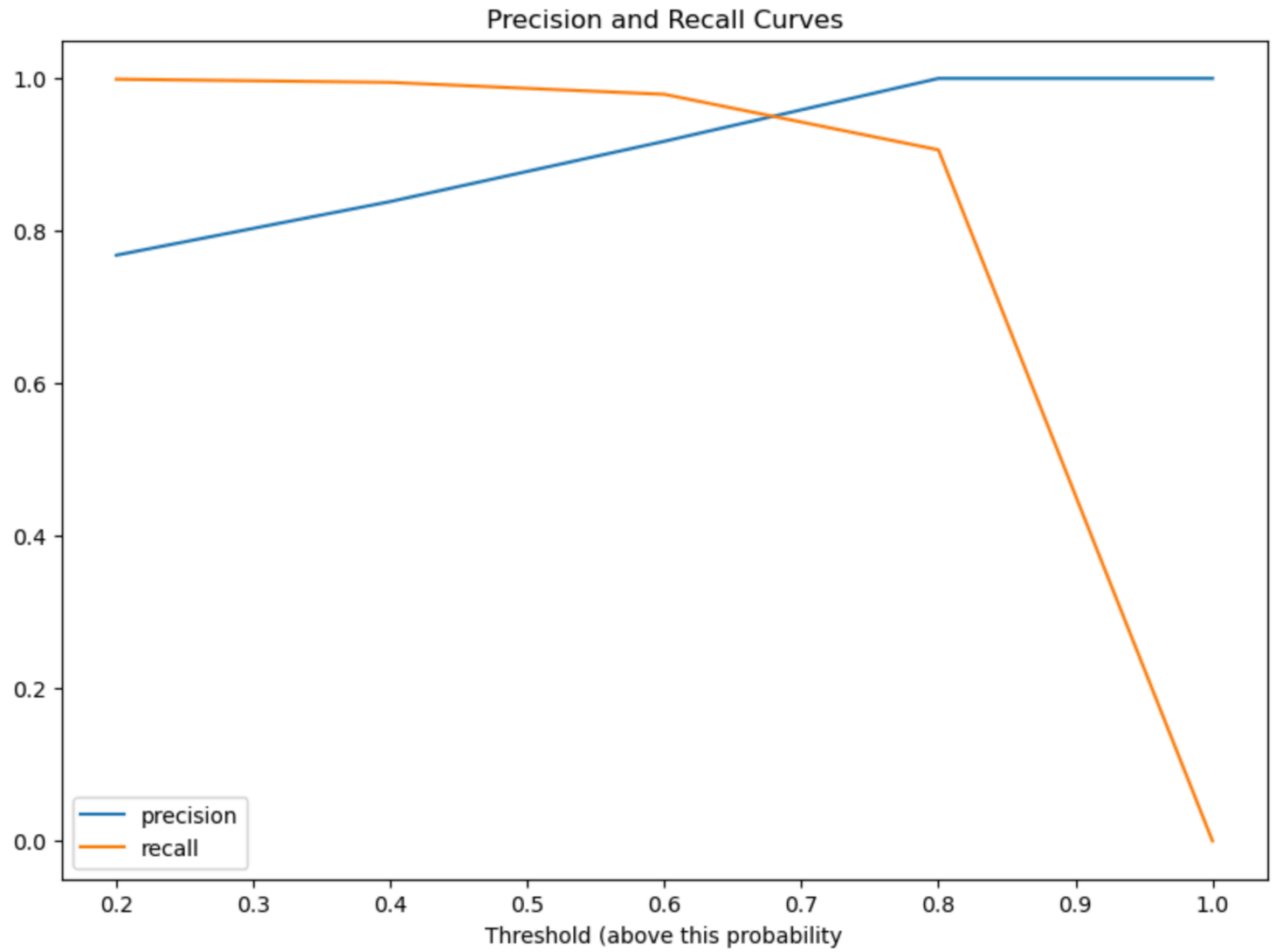
```
Mean Accuracy:  0.8052326119075038
Mean Precision:  0.7335937664342744
Mean Recall:  0.9779067890165172
Mean F1 Score:  0.7961438602004132
```

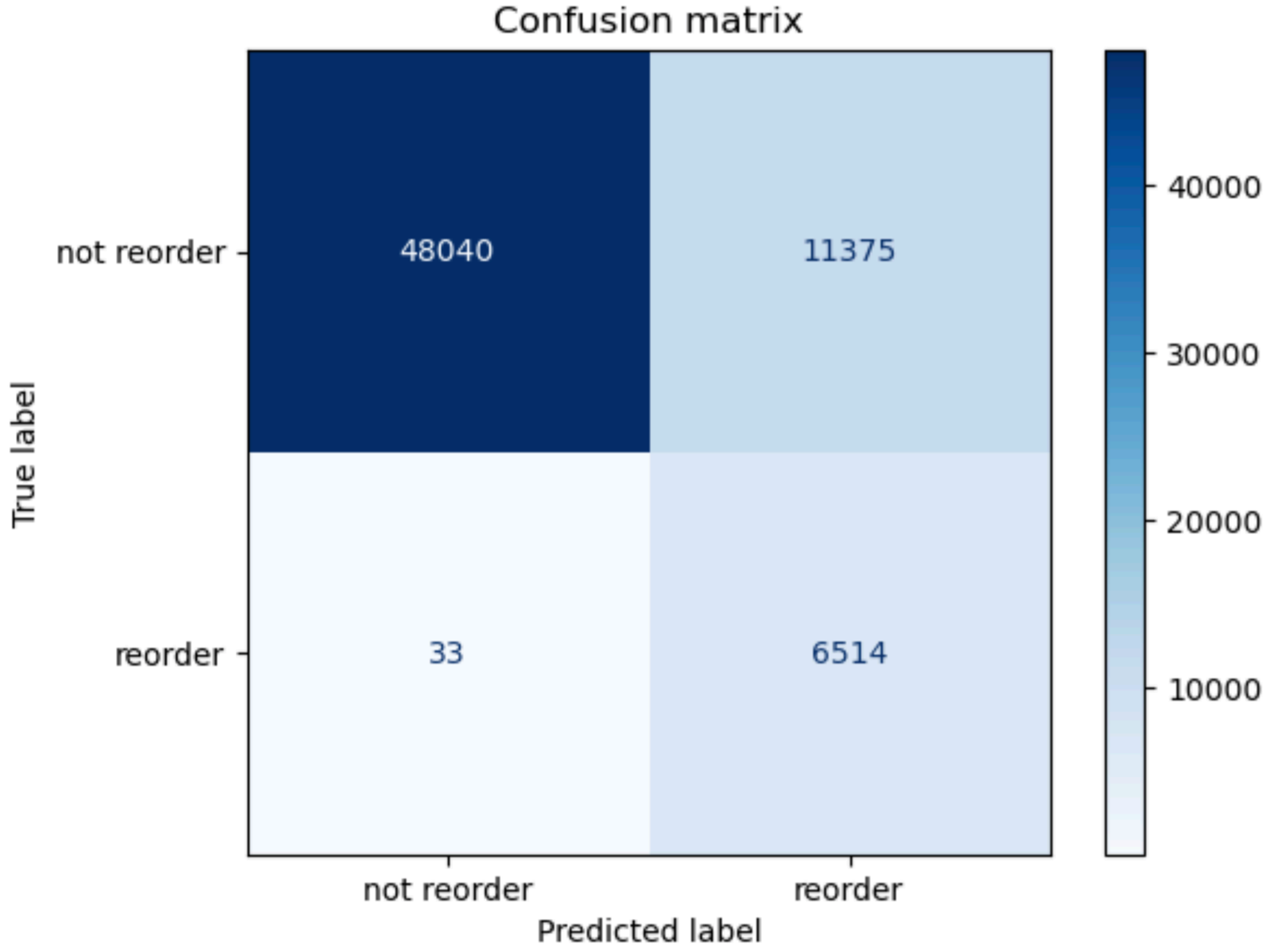
DecisionTree-CV

```
Mean Accuracy:  0.7890224639324648
Mean Precision:  0.7639411275819681
Mean Recall:  0.8512912777268173
Mean F1 Score:  0.7867491321835178
```

RandomForest-CV

```
Mean Accuracy:  0.8578535978046892
Mean Precision:  0.8212759854336298
Mean Recall:  0.9255751416610117
Mean F1 Score:  0.8550354440398003
```


Precision and Recall Curves
1.0
0.8
0.6
0.4
0.2
0.0
precision
recall
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Threshold (above this probability


Confusion matrix
not reorder
reorder
True label
48040
11375
33
6514
not reorder
reorder
Predicted label
40000
30000
20000
10000

This project can be extended further by;

- Obtain Detail Data for **clustering (customer classification)**.
- Using **cloud & big data platform** for implementing ML Algorithms with whole data

THANK YOU

YALIN YENER
+44 7786 761559
yalinyener@gmail.com
https://github.com/yalinyener
https://medium.com/@yalinyener
https://www.linkedin.com/in/yalinyener