kaggle

Santander

Presented by (Group 9) –

BHURJI Prineet Kaur
TIRUMALE LAKSHMANA RAO Kiran

About the Company

To help people and businesses prosper by looking for ways to **help customers understand their financial health** and identify which products and services might **help them achieve their monetary goals**.

16th Largest banking institution in the World

Problem Statement

Through this Kaggle challenge, our aim is to help the Company identify which customers will **make a specific transaction in the future**, irrespective of the amount of money transacted.

Classification Problem, Supervised Learning

Table of Contents –

STEP 1 - EDA

```
Here are the data types:
ID_code     object
target       int64
var_0      float64
var_1      float64
var_2      float64
            ...
var_195    float64
var_196    float64
var_197    float64
var_198    float64
var_199    float64
Length: 202, dtype: object
```

- Checking for the Datatypes for all the Variables
- Checking for the Shape of the Train and Test Sets
- Checking for the Missing Values & Duplicates
    - Both Train as well as Test set doesn't have any missing or duplicate values
- Data was Anonymized so couldn't create any New Features

```
train.shape
```

```
(200000, 202)
```

```
test.shape
```

```
(200000, 201)
```

```
train.isna().sum().sum()
```

```
0
```

```
test.isna().sum().sum()
```

```
0
```

```
# Check for duplicates on both train and test datasets
train.loc[:,'var_0':].duplicated().sum()
```

```
0
```

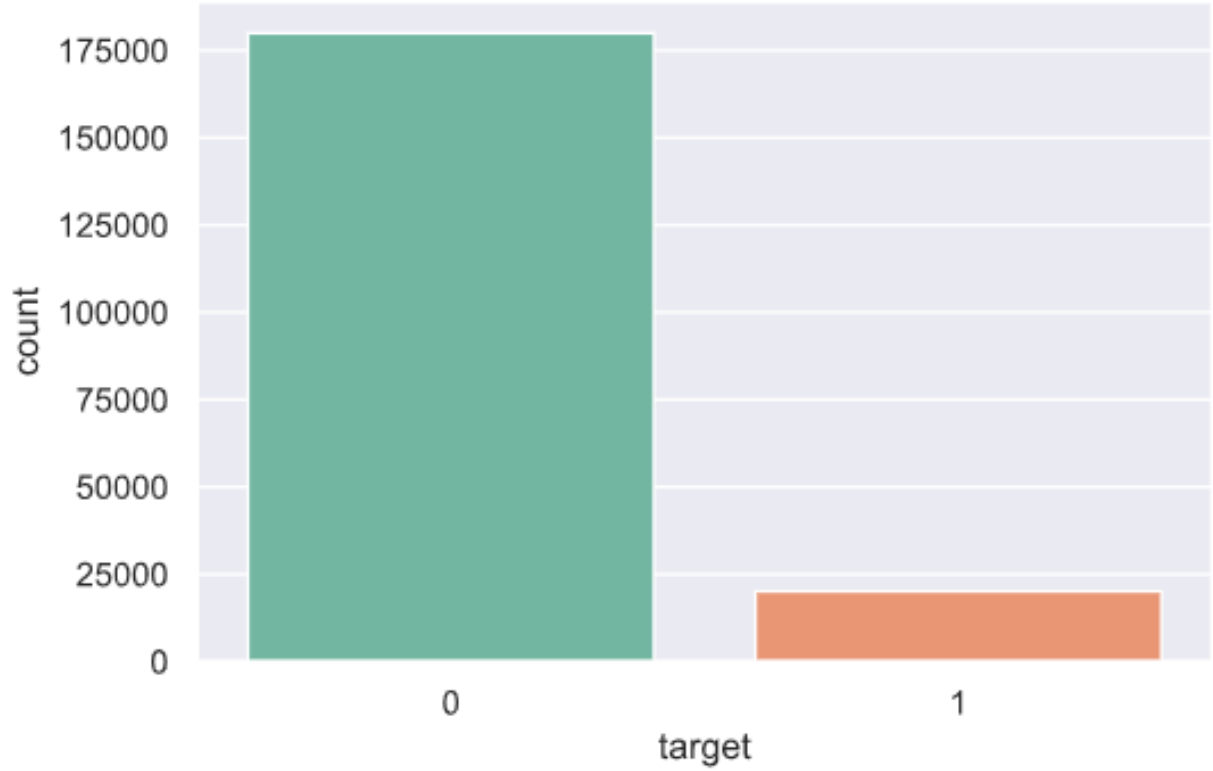
```
test.loc[:,'var_0':].duplicated().sum()
```

```
0
```

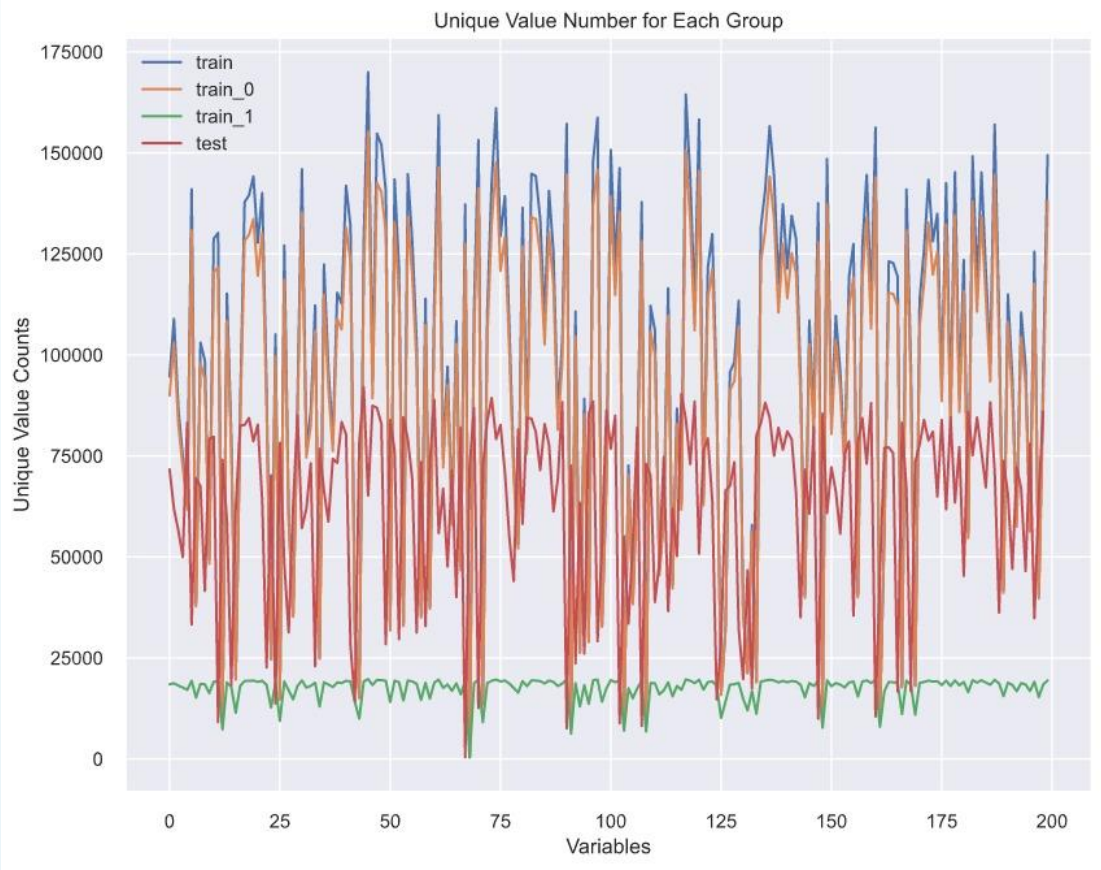
- Checking if the Dataset is Balanced

```
# Checking if the dataset is balanced
mylst = list(train["target"].value_counts())
zeroes = round(float((mylst[0]/sum(mylst))*100),2)
ones = round(float((mylst[1]/sum(mylst))*100),2)
print('The dataset has {zero} % of target 0 and {one} % of target 1'.format(zero=zeroes, one=ones))
```

```
The dataset has 89.95 % of target 0 and 10.05 % of target 1
```



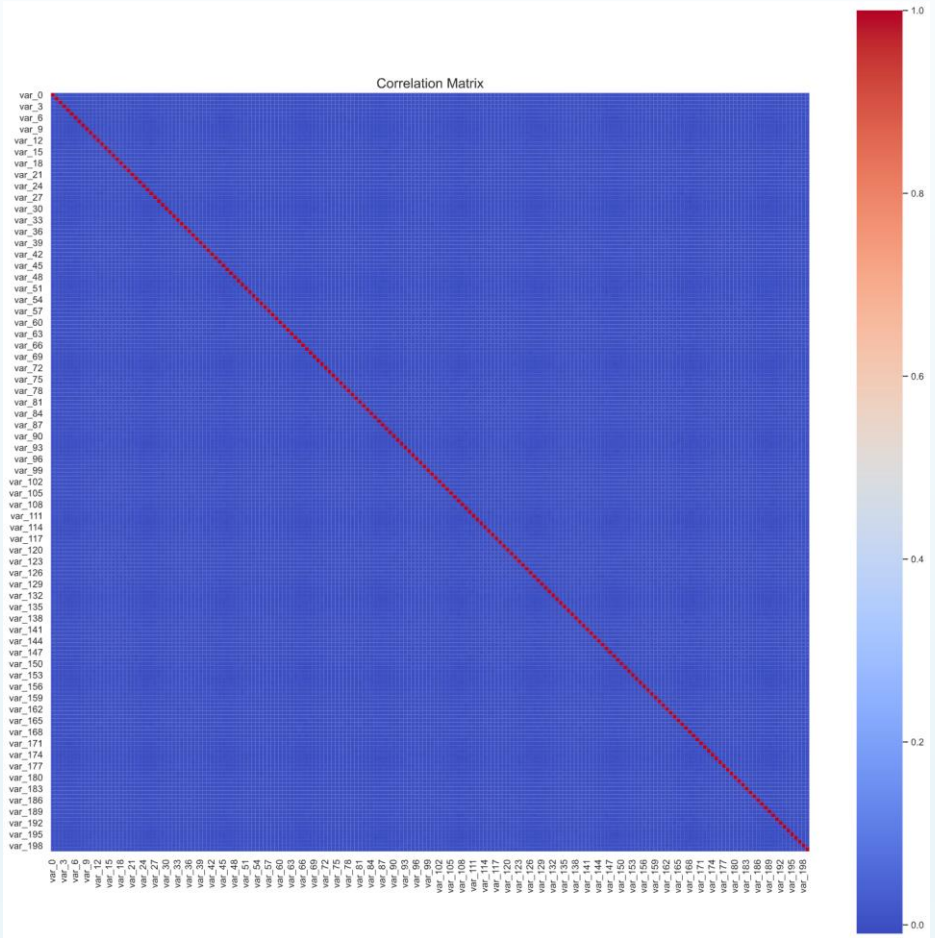
- Understanding the Unique values in Train & Test set



- Checking the Correlation between various Variables

None of the variables show any significant Correlation

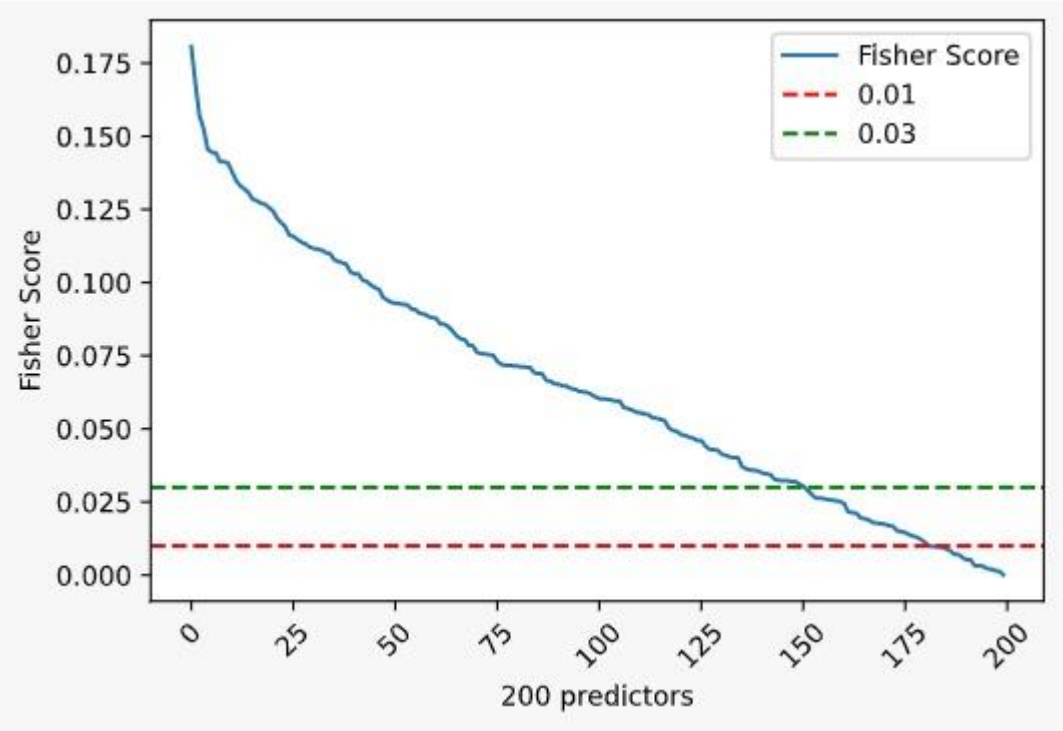
We can say that the Data doesn’t need any Pre-Processing



STEP 2 – Feature Selection

- For selecting the top features we explored the following 3 methods –
  - Fisher Score
  - Logistic Regression
  - Using Light GBM

- Using Fisher Score



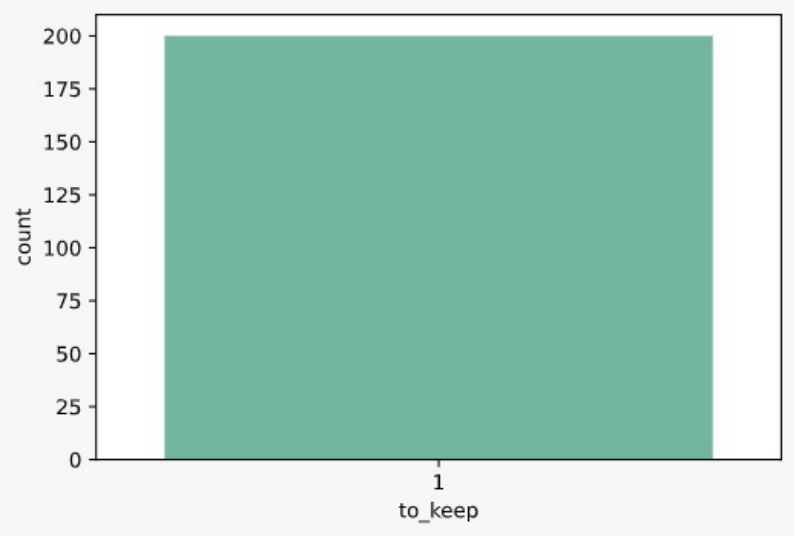
After analysing the above visualization for Fisher Score, we found that we can go ahead with around 150 variables.

- Logistic Regression

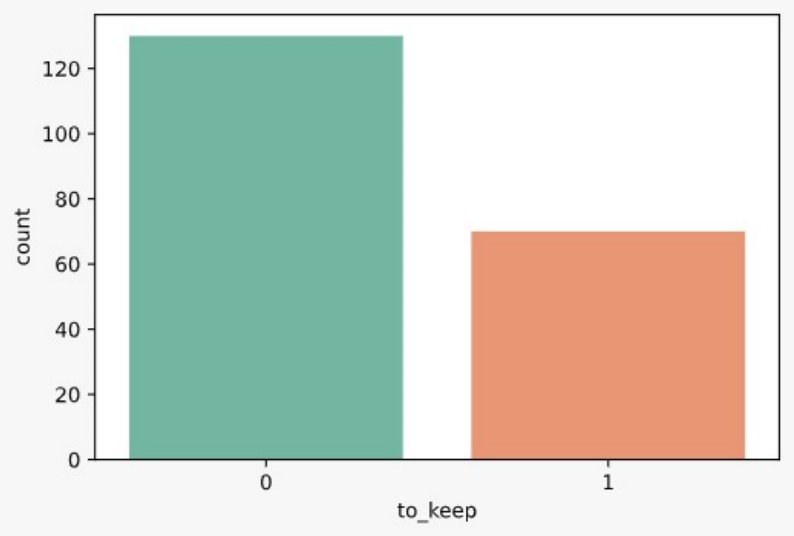
Using logistic regression computed AUC of both Train and Test sets

Max and Min values of AUC found were 0.5672 and 0.5001 respectively

AUC > 0.50	
To Keep	To Drop
200	0



AUC > 0.53	
To Keep	To Drop
70	130



- Using Light GBM

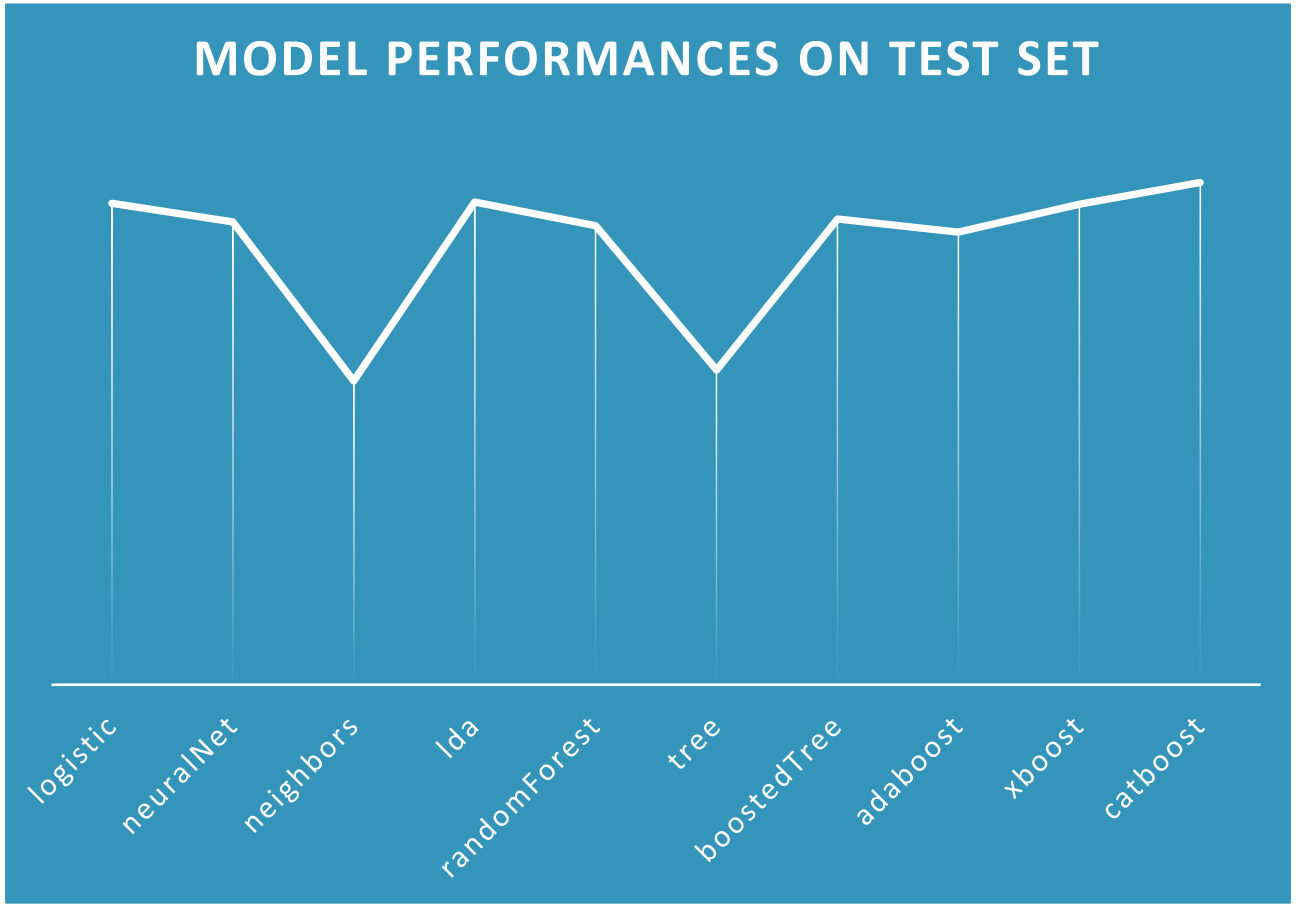
Lowest Score	Highest Score
13.00	122.00

We saw that every Feature has a significance, and nothing can be dropped!

STEP 3 - Modelling

For Modelling purpose, we decided to use the following algorithms –

- Logistic Regression
- Support Vector Machine
- Nearest Neighbours
- Random Forest
- Decision Tree
- Boosting Methods – Boosted Tree, AdaBoost, CatBoost, XBoost



	logistic	neuralNet	neighbors	linearDiscriminant	randomForest	tree	boostedTree	adaboost	xboost	catboost
Accuracy	0.914283	0.901100	0.900267	0.915117	0.900867	0.836533	0.903517	0.907800	0.914633	0.923533
AUC	0.857085	0.823957	0.540765	0.859251	0.816915	0.560192	0.828945	0.805633	0.855437	0.894053

STEP 4 – Hyperparameter Tuning

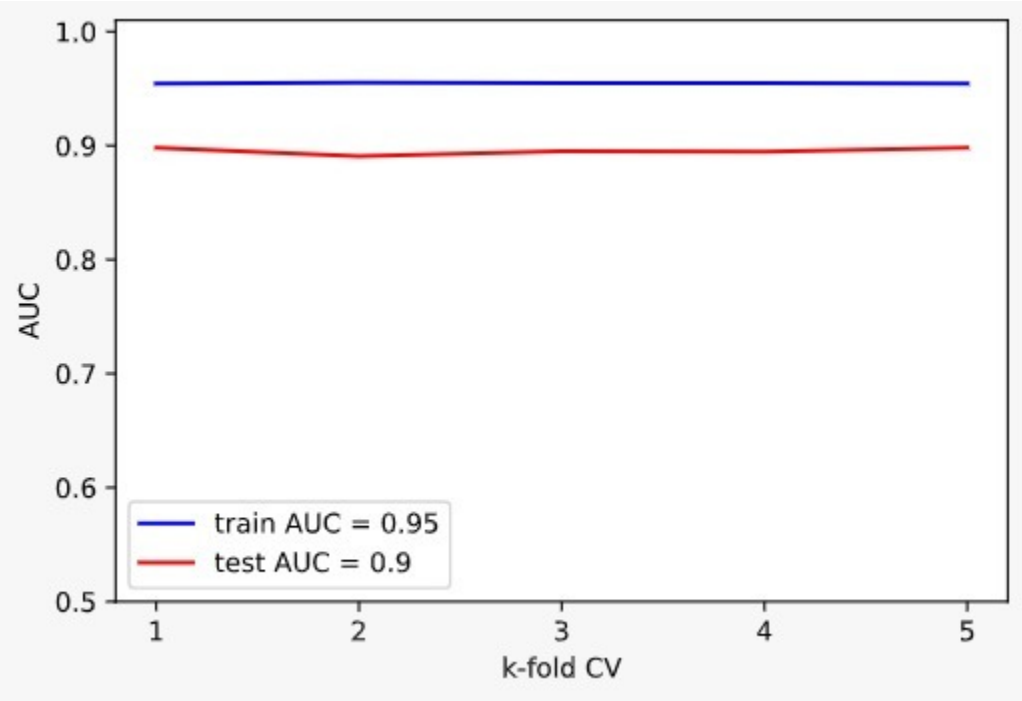
From the previously seen Accuracy and AUC scores we decided to go ahead with the following top 3 algorithms for Hyperparameter Tuning –

1) CatBoost
2) LDA
3) Logistic Regression

STEP 5 – Resampling using k-fold CV

	train	test
0	0.954265	0.898223
1	0.955341	0.890545
2	0.955064	0.895295
3	0.954918	0.894768
4	0.954351	0.897996

We applied StratifiedKfold() function with k=5 on Training Set with the Best Model i.e CatBoost



STEP 6 – Results & Conclusions

Final Score for our Best performing Model after Hyperparameter Tuning and Resampling on Kaggle was as follows -

Private Score	Public Score	Top Score
0.89162	0.89328	0.92573

Conclusions –

- Do not always depend one just one method for Feature Selection
- Hyperparameter Tuning requires a lot of time so should be planned well
- When the Data is too huge then Jupyter Notebook is not may be the best option (experimented using other options like Google Cocalc)

Thank you

Any Questions?