Capstone Project - 3

Credit Card Default Prediction

ML Supervised Classification

Individual Project:
Soumya Ranjan Mishra

Contents

- Problem Statement
- Data Summary
- Data Analysis
- Analysis Details
- Outlier Treatment
- Feature Selection
- Data Preparation
- Implementing Various Classification Algorithms
- Challenges
- Conclusions

Problem Statements

- **Predict that customers will default on their credit card payments or not.**

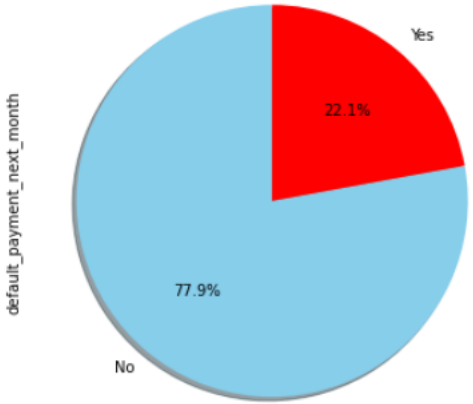
- **Create K-S Chart to evaluate which customers will default on their credit card payments .**

Data Summary

- Dependent Variable: default payment (Yes = 1, No = 0), as the response variable.
- X1: Amount of the given credit (NT dollar): it includes both the individual consumer credit and his/her family (supplementary) credit.
- X2: Gender (1 = male; 2 = female).
- X3: Education (1 = graduate school; 2 = university; 3 = high school; 4 = others).
- X4: Marital status (1 = married; 2 = single; 3 = others).
- X5: Age (year).
- X6 - X11: History of past payment. We tracked the past monthly payment records (from April to September, 2005) as follows: X6 = the repayment status in September, 2005; X7 = the repayment status in August, 2005; . . .;X11 = the repayment status in April, 2005. The measurement scale for the repayment status is: -1 = pay duly; 1 = payment delay for one month; 2 = payment delay for two months; . . .; 8 = payment delay for eight months; 9 = payment delay for nine months and above.
- X12-X17: Amount of bill statement (NT dollar). X12 = amount of bill statement in September, 2005; X13 = amount of bill statement in August, 2005; . . .; X17 = amount of bill statement in April, 2005.
- X18-X23: Amount of previous payment (NT dollar). X18 = amount paid in September, 2005; X19 = amount paid in August, 2005; . . .;X23 = amount paid in April, 2005.

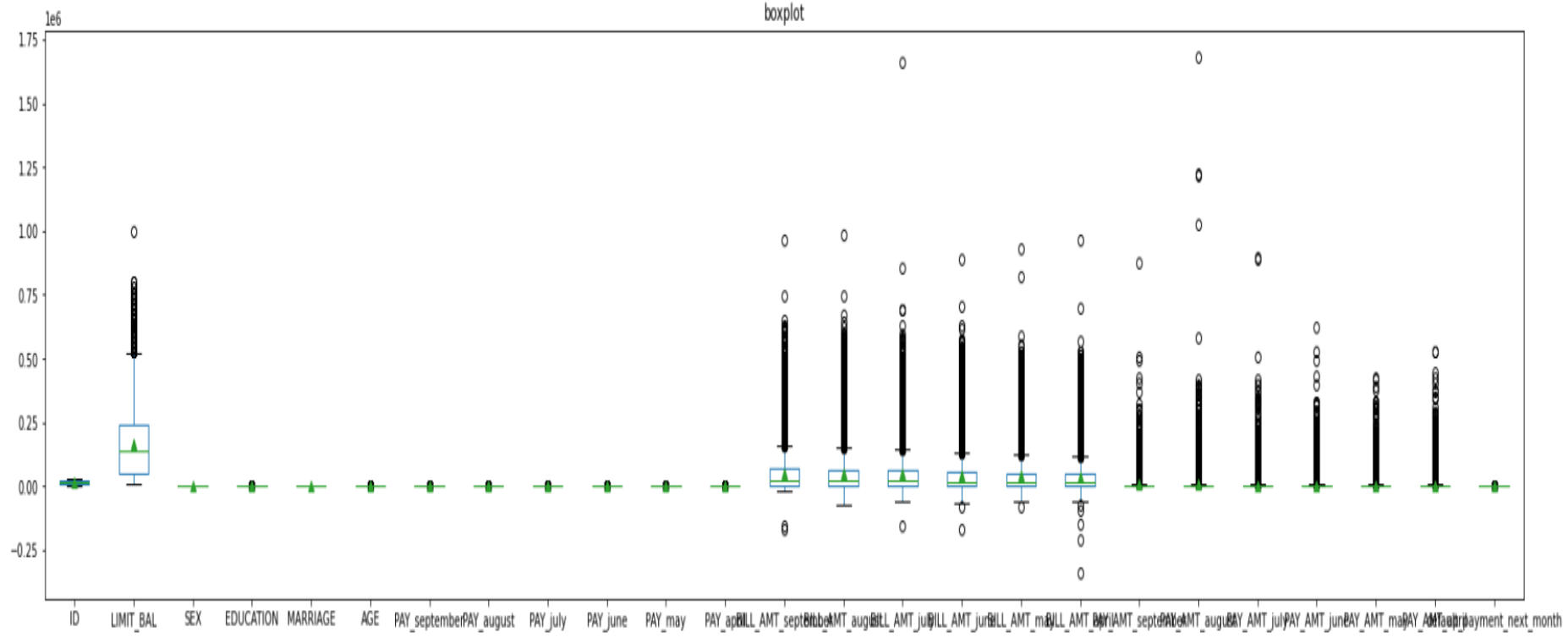
Basic Data Exploration

- The dataset has 30000 rows and 25 features(columns).
- No categorical features as encoding done before.
- Outliers present
- No Missing Values.
- No Duplicated values.
- No null values.

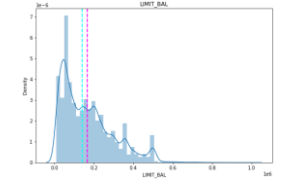


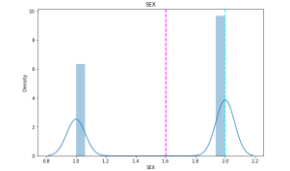
- Imbalanced Dataset as “No” count is 23364 as 77.88% and “Yes” count is 6636 as 22.12% of the whole dataset.

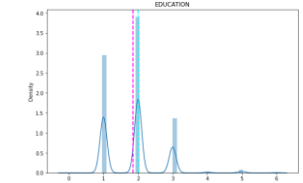
Outliers in the features

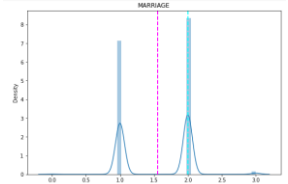


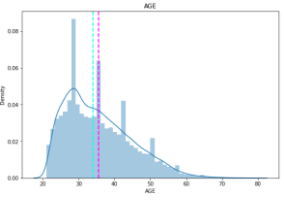
Mean Distribution of Various Features

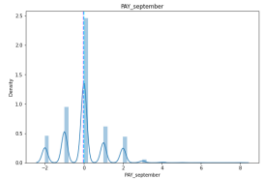


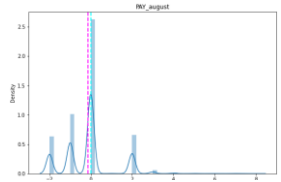


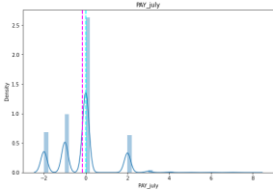


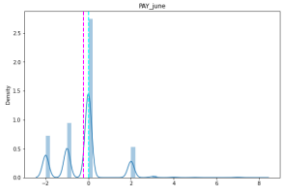


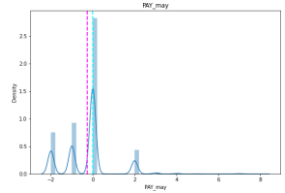


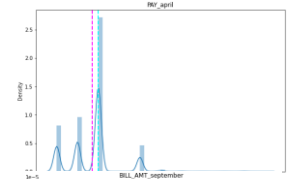


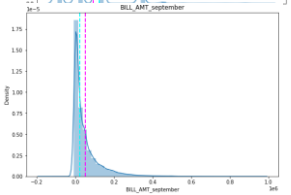


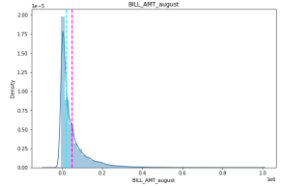


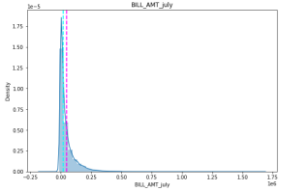


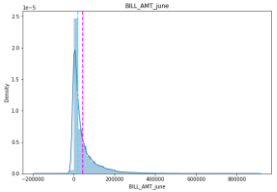


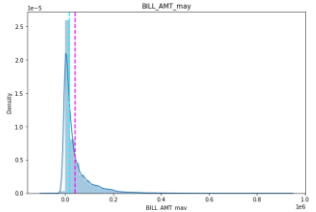


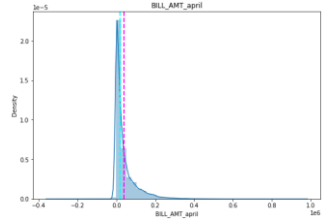


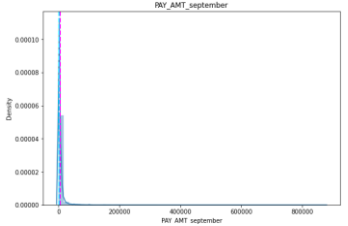


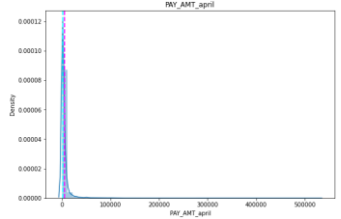


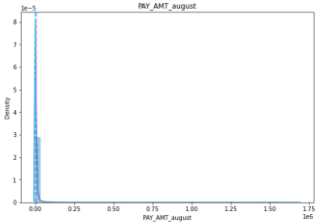


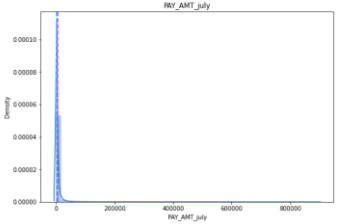


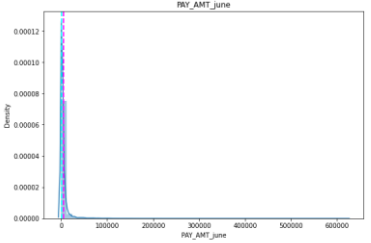


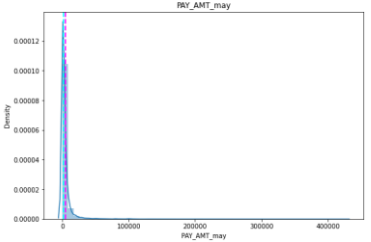




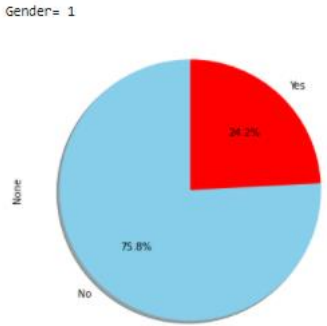


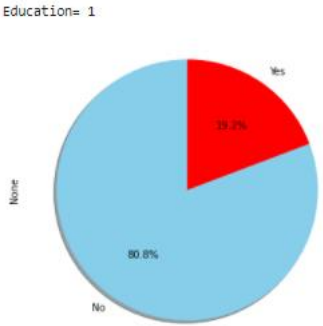


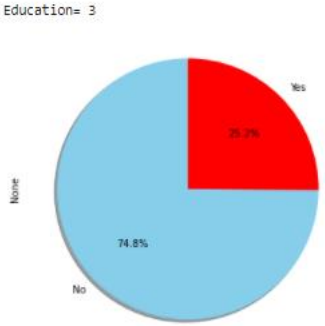


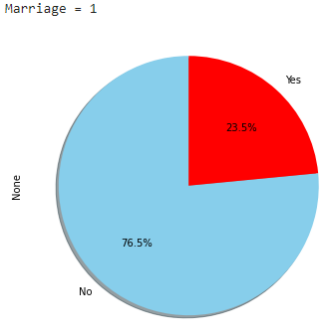


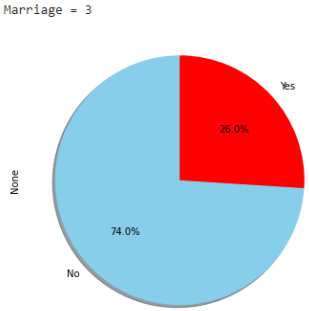
Distribution of Gender, Education , Marital Status

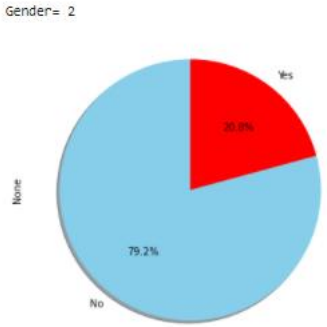


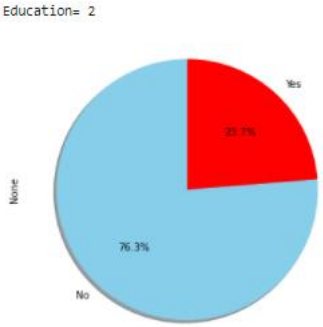


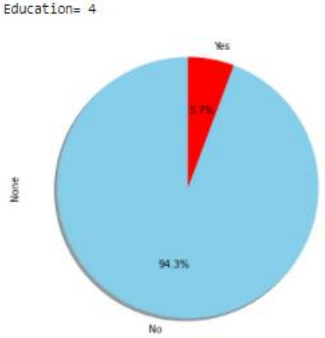


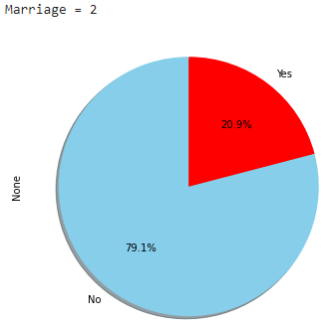






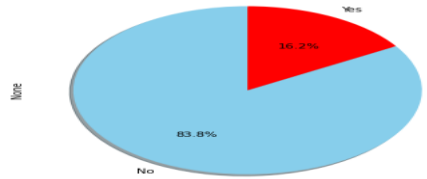




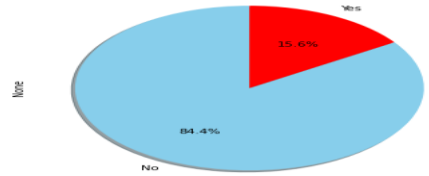


Distribution of PAY_X

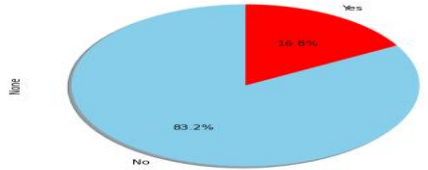
PAY_may = -1



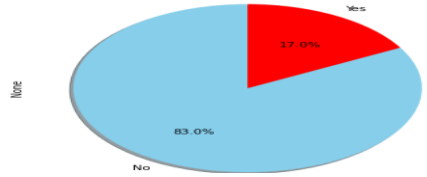
PAY_july = -1



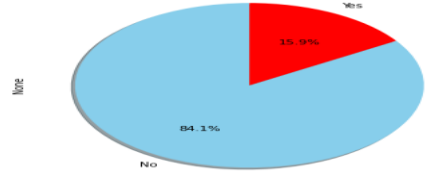
PAY_september = -1



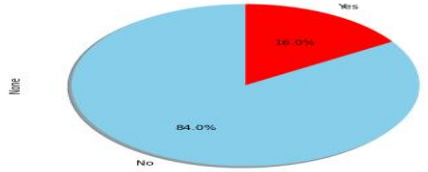
PAY_april = -1



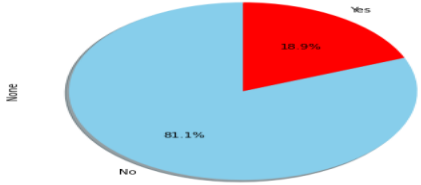
PAY_june = -1



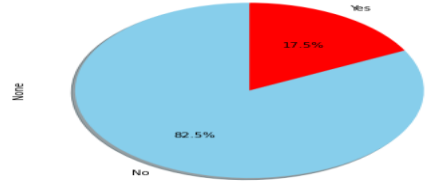
PAY_august = -1



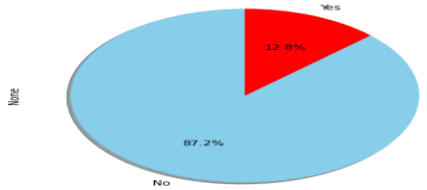
PAY_may = 0



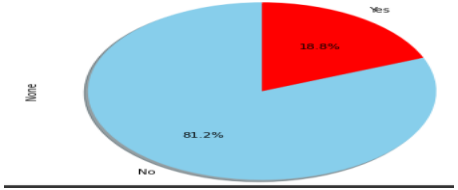
PAY_july = 0



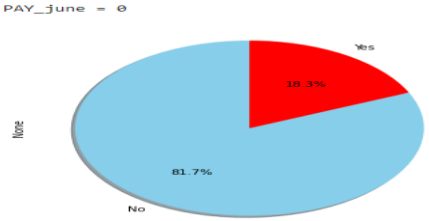
PAY_september = 0



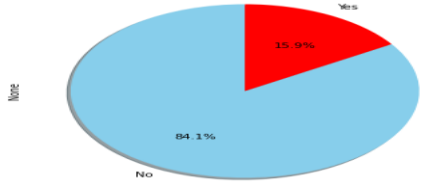
PAY_april = 0



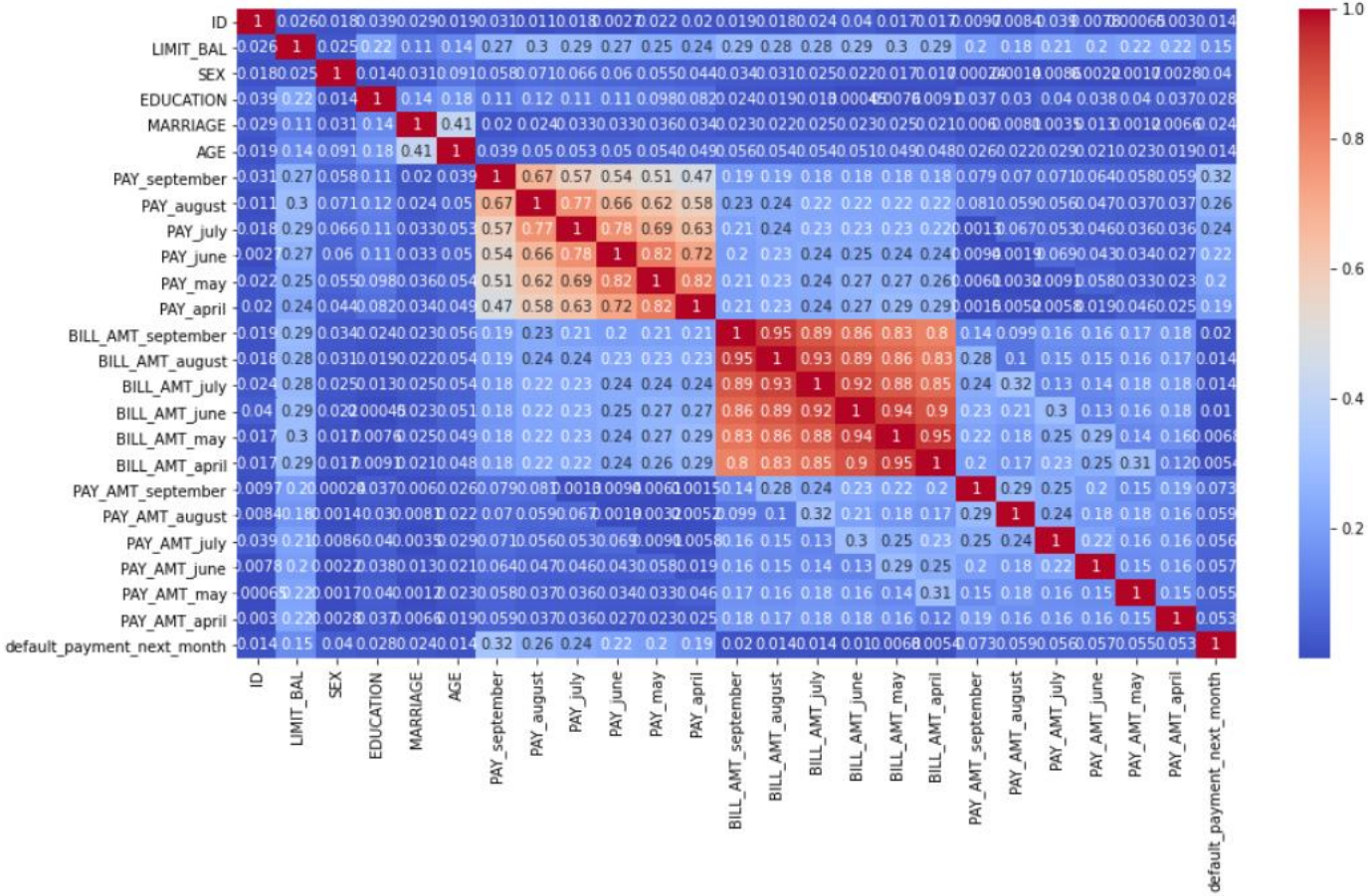
PAY_june = 0



PAY_august = 0



Correlation Matrix



Analysis Details

- Defaults have a higher proportion of Lower LIMIT_BAL values
- NonDefaults have a higher proportion of Females (Sex=2)
- NonDefaults have a higher proportion of More Educated (EDUCATION=1 or 2).
- NonDefaults have a higher proportion of Singles (MARRIAGE=2)
- NonDefaults have a higher proportion of people 30-40years
- NonDefaults have a MUCH higher proportion of zero or negative PAY_X variables (this means that being current or ahead of payments is associated with not defaulting in the following month).

Outlier Treatment

- For Skewed Features using IQR

- For Quite Symmetric Features using Standard Deviation.

Feature Selection

- Dropping Constant Feature Using Variance Threshold

- Feature Selection with Pearson Correlation

Data Preparation

- **Handling Imbalance set by Synthetic Minority Oversampling Technique (SMOTE)**

- **Train Test Split (**test_size=0.3, random_state=0**)**

Logistic Regression

Train Set Metrics

```
[[ 9321  9430]
 [ 5002 13629]]
```

	precision	recall	f1-score	support
0	0.50	0.65	0.56	14323
1	0.73	0.59	0.65	23059
accuracy			0.61	37382
macro avg	0.61	0.62	0.61	37382
weighted avg	0.64	0.61	0.62	37382

```
roc_auc_score
0.6528013916798787
```

Test Set Metrics

```
[[2274 2339]
 [1266 3467]]
```

	precision	recall	f1-score	support
0	0.49	0.64	0.56	3540
1	0.73	0.60	0.66	5806
accuracy			0.61	9346
macro avg	0.61	0.62	0.61	9346
weighted avg	0.64	0.61	0.62	9346

```
roc_auc_score
0.6518238469268705
```

Decision Tree

Train Set Metrics

```
[[17884,  4816],
 [  867, 13815]]
```

	precision	recall	f1-score	support
0	0.95	0.79	0.86	22529
1	0.75	0.94	0.83	14853
accuracy			0.85	37382
macro avg	0.85	0.86	0.85	37382
weighted avg	0.87	0.85	0.85	37382

```
roc_auc_score
0.8489597682158649
```

Test Set Metrics

```
[[4351, 1214],
 [ 262, 3519]]
```

	precision	recall	f1-score	support
0	0.94	0.78	0.85	5531
1	0.75	0.93	0.83	3815
accuracy			0.84	9346
macro avg	0.84	0.86	0.84	9346
weighted avg	0.86	0.84	0.84	9346

```
roc_auc_score
0.8433068085952445
```

Hyper parameter

```
(ccp_alpha=0.0, class_weight=None, criterion='entropy',
 max_depth=8, max_features=14, max_leaf_nodes=100,
 min_impurity_decrease=0.0, min_impurity_split=None,
 min_samples_leaf=1, min_samples_split=2,
 min_weight_fraction_leaf=0.0, presort='deprecated',
 random_state=0, splitter='best')
```

Random Forest

Train Set Metrics

```
[[18746,     5],
 [    9, 18622]]
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	18755
1	1.00	1.00	1.00	18627
accuracy			1.00	37382
macro avg	1.00	1.00	1.00	37382
weighted avg	1.00	1.00	1.00	37382

```
roc_auc_score
0.9999979333040048
```

Test Set MetricS

```
[[4265,  348],
 [ 786, 3947]]
```

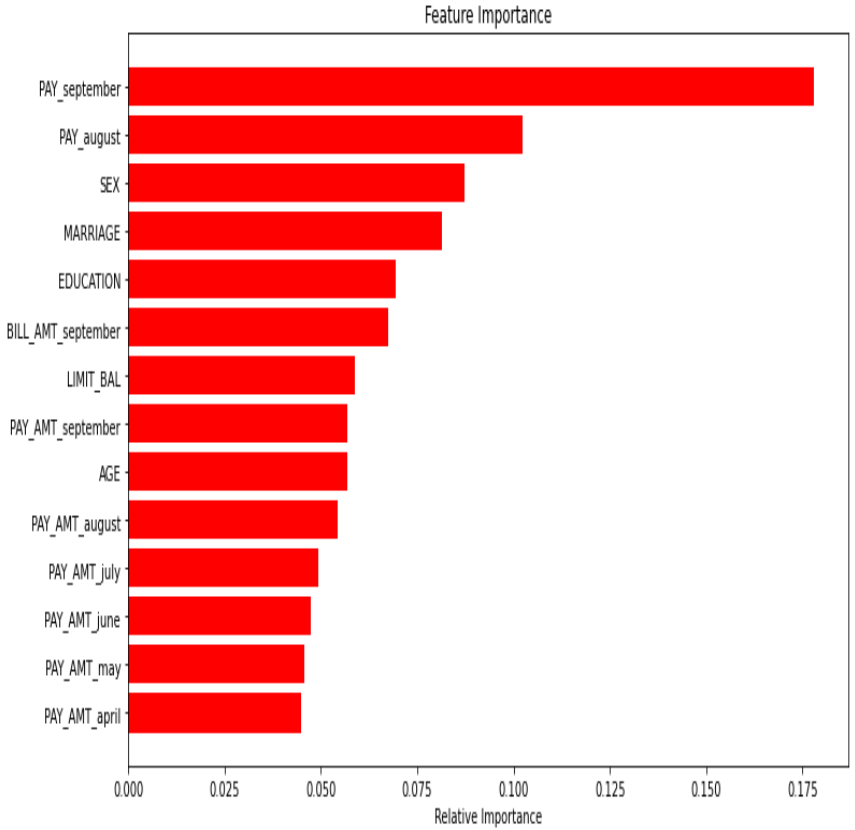
	precision	recall	f1-score	support
0	0.92	0.84	0.88	5051
1	0.83	0.92	0.87	4295
accuracy			0.88	9346
macro avg	0.88	0.88	0.88	9346
weighted avg	0.88	0.88	0.88	9346

```
roc_auc_score
0.938280300727388
```

Hyper parameter

```
{'max_depth': 8,
 'min_samples_leaf': 40,
 'min_samples_split': 100,
 'n_estimators': 100}
```

Feature Importance



Gradient Boosting Machine

Train Set Metrics

```
[[17976,   775],
 [ 3167, 15464]]
```

	precision	recall	f1-score	support
0	0.96	0.85	0.90	21143
1	0.83	0.95	0.89	16239
accuracy			0.89	37382
macro avg	0.89	0.90	0.89	37382
weighted avg	0.90	0.89	0.90	37382

```
roc_auc_score
0.9666025991862295
```

Test Set Metrics

```
[[4320,  293],
 [ 891, 3842]]
```

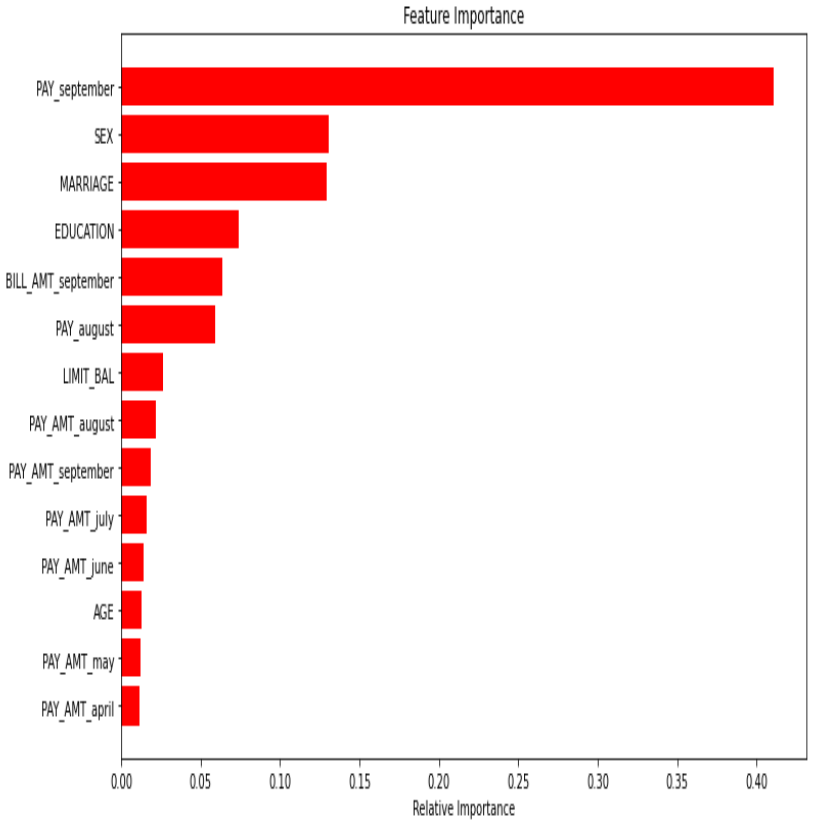
	precision	recall	f1-score	support
0	0.94	0.83	0.88	5211
1	0.81	0.93	0.87	4135
accuracy			0.87	9346
macro avg	0.87	0.88	0.87	9346
weighted avg	0.88	0.87	0.87	9346

```
roc_auc_score
0.9328971775215772
```

Hyper parameter

```
max_depth=8,
 min_samples_leaf= 40,
 min_samples_split=80,
 n_estimators=80
```

Feature Importance



XGBoost

Train Set Metrics

```
[[17716,  1035],
 [ 3727, 14904]]
```

	precision	recall	f1-score	support
0	0.94	0.83	0.88	21443
1	0.80	0.94	0.86	15939
accuracy			0.87	37382
macro avg	0.87	0.88	0.87	37382
weighted avg	0.88	0.87	0.87	37382

roc_auc_score
0.9311800896248195

Test Set Metrics

```
[[4329,  284],
 [ 955, 3778]]
```

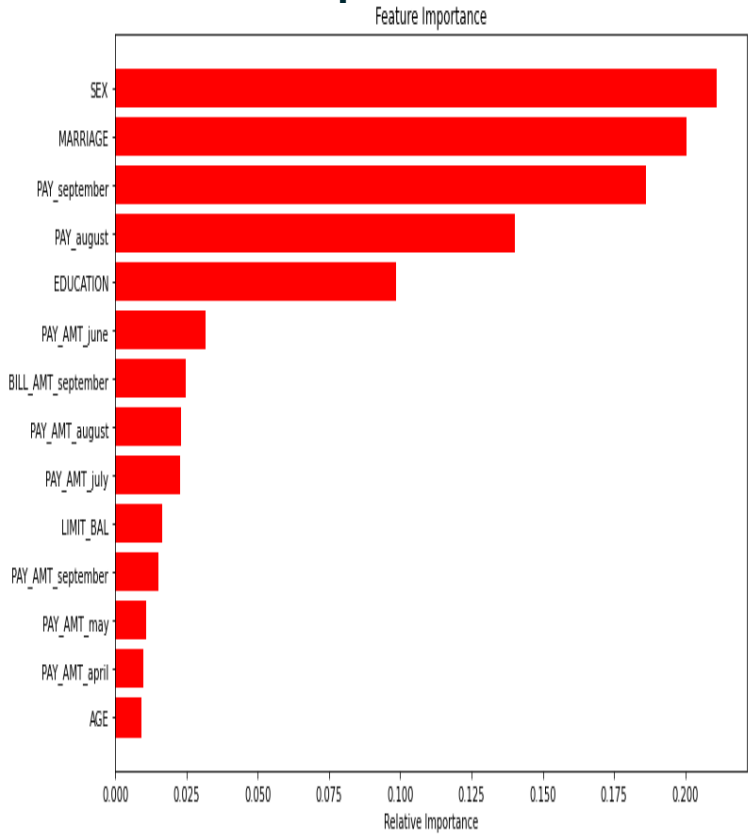
	precision	recall	f1-score	support
0	0.94	0.82	0.87	5284
1	0.80	0.93	0.86	4062
accuracy			0.87	9346
macro avg	0.87	0.87	0.87	9346
weighted avg	0.88	0.87	0.87	9346

roc_auc_score
0.9252521225691237

Hyper parameter

```
{'max_depth': 8,
 'min_samples_leaf': 40,
 'min_samples_split': 50,
 'n_estimators': 100}
```

Feature Importance



K-Nearest Neighbour

Train Set Metrics

```
[[15034,  3717]
 [  698, 17933]]
```

	precision	recall	f1-score	support
0	0.80	0.96	0.87	15732
1	0.96	0.83	0.89	21650
accuracy			0.88	37382
macro avg	0.88	0.89	0.88	37382
weighted avg	0.89	0.88	0.88	37382

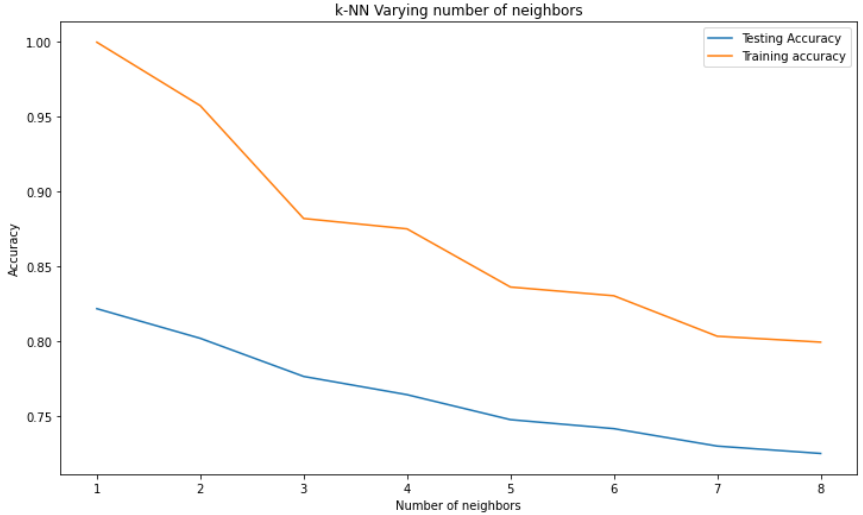
```
roc_auc_score
0.9737362197641626
```

Test Set Metrics

```
[[3050, 1563],
 [ 527, 4206]]
```

	precision	recall	f1-score	support
0	0.66	0.85	0.74	3577
1	0.89	0.73	0.80	5769
accuracy			0.78	9346
macro avg	0.77	0.79	0.77	9346
weighted avg	0.80	0.78	0.78	9346

```
roc_auc_score
0.9322817880864617
```



Support Vector Machine

Train Set Metrics

```
[[10371  8380]
 [ 6427 12204]]
```

	precision	recall	f1-score	support
0	0.25	0.68	0.37	6889
1	0.88	0.54	0.67	30493
accuracy			0.57	37382
macro avg	0.57	0.61	0.52	37382
weighted avg	0.77	0.57	0.61	37382

roc_auc_score
0.6702448010852479

Test Set MetricS

```
[[2555 2058]
 [1625 3108]]
```

	precision	recall	f1-score	support
0	0.55	0.61	0.58	4180
1	0.66	0.60	0.63	5166
accuracy			0.61	9346
macro avg	0.61	0.61	0.60	9346
weighted avg	0.61	0.61	0.61	9346

roc_auc_score
0.6052677308164962

Naïve Bayes Classification

Train Set Metrics

```
[[1187, 3426],
 [ 585, 4148]]
```

	precision	recall	f1-score	support
0	0.25	0.68	0.37	6889
1	0.88	0.54	0.67	30493
accuracy			0.57	37382
macro avg	0.57	0.61	0.52	37382
weighted avg	0.77	0.57	0.61	37382

roc_auc_score
0.6702448010852479

Test Set MetricS

```
[[1187, 3426],
 [ 585, 4148]]
```

	precision	recall	f1-score	support
0	0.26	0.67	0.37	1772
1	0.88	0.55	0.67	7574
accuracy			0.57	9346
macro avg	0.57	0.61	0.52	9346
weighted avg	0.76	0.57	0.62	9346

roc_auc_score
0.6732817748498179

K-S Chart

Decile	min_prob	max_prob	events	nonevents	event_rate	nonevent_rate
1	0.796667	1.00	2879	0	43.38%	0.00%
2	0.650833	0.79	3081	1	46.43%	0.00%
3	0.144000	0.65	676	2320	10.19%	9.93%
4	0.092500	0.14	0	3018	0.00%	12.92%
5	0.072000	0.09	0	2323	0.00%	9.94%
6	0.052500	0.07	0	3458	0.00%	14.80%
7	0.041667	0.05	0	2191	0.00%	9.38%
8	0.034000	0.04	0	2216	0.00%	9.48%
9	0.016667	0.03	0	4748	0.00%	20.32%
10	0.000000	0.01	0	3089	0.00%	13.22%

Decile	cum_eventrate	cum_noneventrate	KS
1	43.38%	0.00%	43.4
2	89.81%	0.00%	89.8
3	100.00%	9.93%	90.1
4	100.00%	22.85%	77.1
5	100.00%	32.79%	67.2
6	100.00%	47.59%	52.4
7	100.00%	56.97%	43.0
8	100.00%	66.46%	33.5
9	100.00%	86.78%	13.2
10	100.00%	100.00%	0.0

KS is 90.10000000000001% at decile 3

Ks_2sampResult(statistic=0.9986919026288261, pvalue=0.0)

Challenges

- Large Dataset to handle.
- Needs to plot lot of Graphs to analyse.
- Carefully handled Feature selection part .
- Carefully tuned Hyper parameters.
- Handled the imbalanced Dataset carefully.

Conclusion

- **No overfitting is seen.**
- **Random Forest gives the highest ROC_AUC score, Accuracy & F1 Score of 99.9%,100% & 100% respectively for Train Set and 93.8%, 88% & 88% respectively for Test set.**
- **Feature Importance value for Random Forest, Gradient Boost, and XGBoost are different.**
- **KS is 90.1% at decile 3.**
- **We can deploy this model.**

THANK YOU

Q & A