Capstone Project

Predicting whether a customer will default on his/her credit card

Team

Abhijeet Kulkarni , Kundan Lal
pankaj Ganjare , Akshay Auti

Problem Statement:

This project is aimed at predicting the case of customers default payments in Taiwan. From the perspective of risk management, the result of predictive accuracy of the estimated probability of default will be more valuable than the binary result of classification - credible or not credible clients

Content

- Data Description
- Attribute Information
- Summary of Data
- Data Preprocessing
- Data Visualization
- Heat Map
- Standard Normalization
- Data Training & Testing
- Algorithms For Machine Learning
- Hyper tuning
- Conclusion

Data Description:

This research employed a binary variable, default payment (Yes = 1, No = 0), as the response variable. This study reviewed the literature and used the following 23 variables as explanatory variables:
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(properly pay); 0 = not delay;, 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.

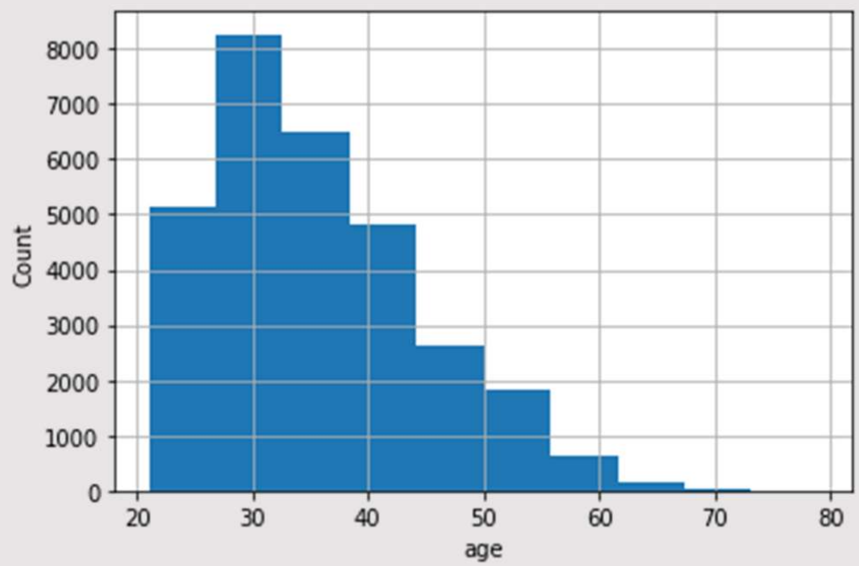
X12-X17: Amount of bill statement (NT dollar). X12 = amount of bill statement in September, 2005; X13 = amount of bill statement in August, 2005; . . .; X17 = amount of bill statement in April, 2005.
X18-X23: Amount of previous payment (NT dollar). X18 = amount paid in September, 2005; X19 = amount paid in August, 2005; . . .;X23 = amount paid in April, 2005.

Attribute Information : Null Values and Dtypes:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30000 entries, 0 to 29999 Data columns (total 25 columns):
# Column                                    Non-Null Count Dtype
--- ------                                  -------------- -----
0 ID                                        30000 non-null int64
1 LIMIT_BAL                                 30000 non-null int64
2 SEX                                       30000 non-null int64
3 EDUCATION                                 30000 non-null int64
4 MARRIAGE                                  30000 non-null int64
5 AGE                                       30000 non-null int64
6 PAY_0                                     30000 non-null int64
7 PAY_2                                     30000 non-null int64
8 PAY_3                                     30000 non-null int64
9 PAY_4                                     30000 non-null int64
10 PAY_5                                    30000 non-null int64
11 PAY_6                                    30000 non-null int64
12 BILL_AMT1                                30000 non-null int64
13 BILL_AMT2                                30000 non-null int64
14 BILL_AMT3                                30000 non-null int64
15 BILL_AMT4                                30000 non-null int64
16 BILL_AMT5                                30000 non-null int64
17 BILL_AMT6                                30000 non-null int64
18 PAY_AMT1                                 30000 non-null int64
19 PAY_AMT2                                 30000 non-null int64
20 PAY_AMT3                                 30000 non-null int64
21 PAY_AMT4                                 30000 non-null int64
22 PAY_AMT5                                 30000 non-null int64
23 PAY_AMT6                                 30000 non-null int64
24 default payment next month               30000 non-null int64
dtypes: int64(25) memory usage: 5.7 MB
```

Data Visualization

From graph we know that Credit Card holders whose age is between 28 to 40 are highest in numbers

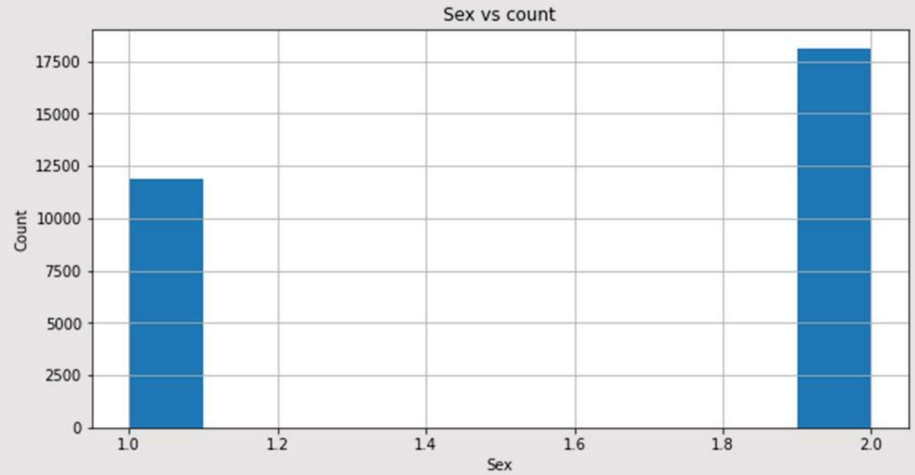


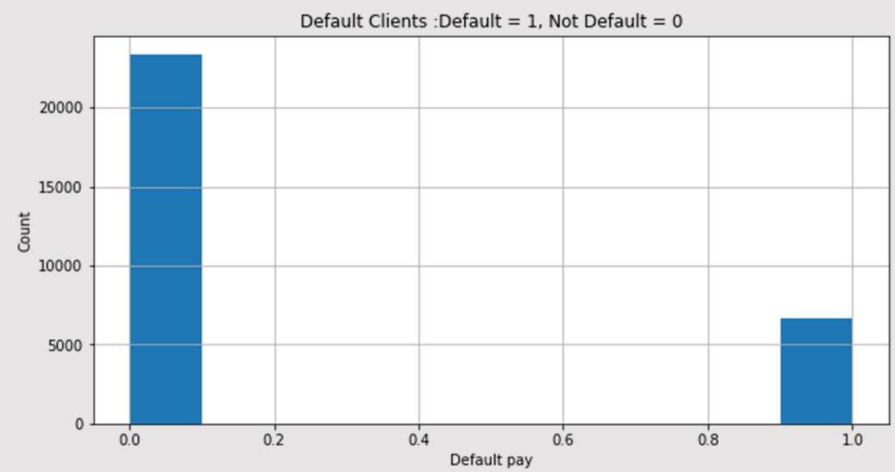
Analysis based on Gender (1 = m ale; 2 = female)

Females contains more numbers of credit cards as compare to males

#Numbers of Default and Not Def ault credit card holders

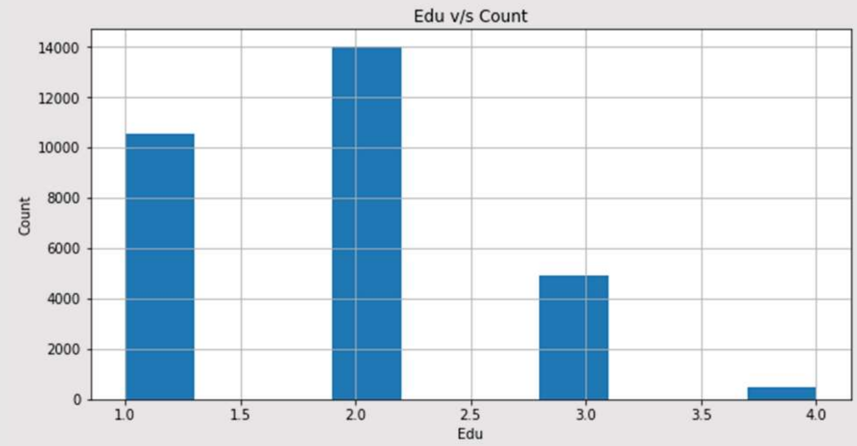
Percentage of Defaulters are smaller than the Non Defaulters in the given dataset





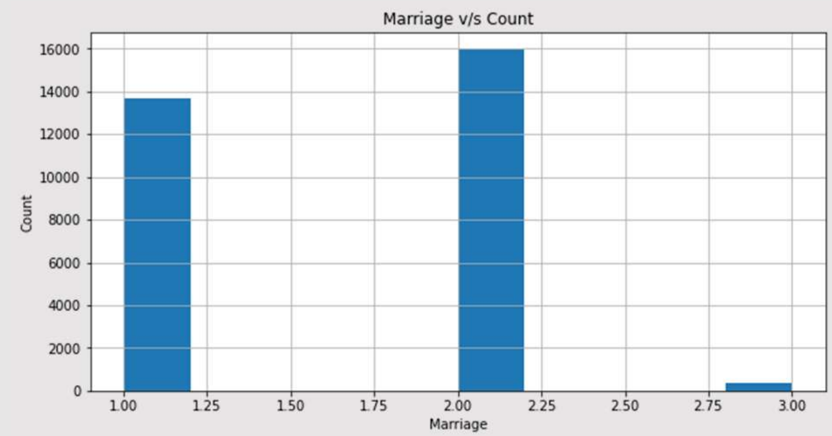
#Analysing on Education Basis (1 = graduate school; 2 = university; 3 = high school; 4 = others)

University has highest numbers of credit card holders followed by Graduate School.



#Analysing on Marriage Basis (1 = married; 2 = single; 3 = others)

More number of credit cards holder are Singles followed by Married ones.

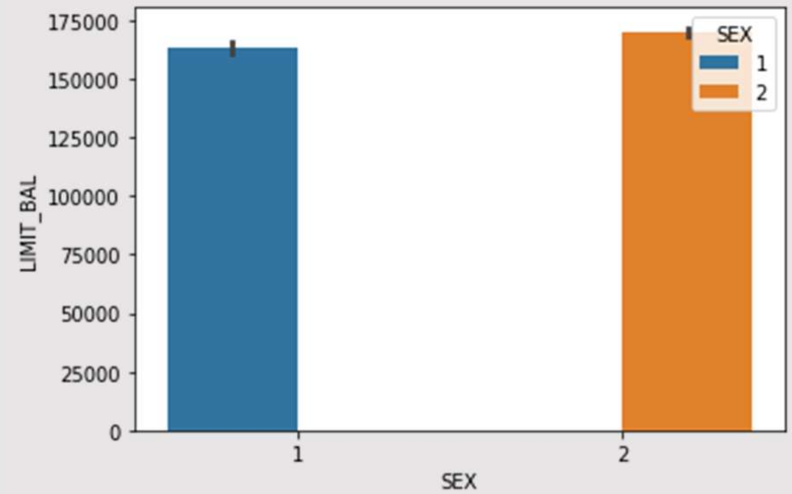


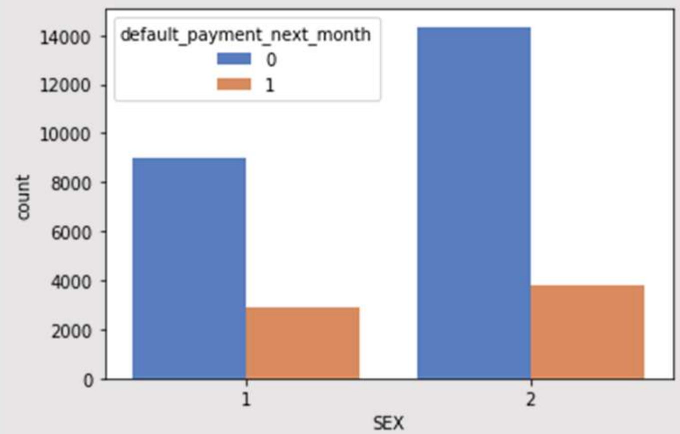
sns.barplot sex vs limit_bal

Credit Limit of Male members are less as compare to females.

sns.countplot vs default payment next month

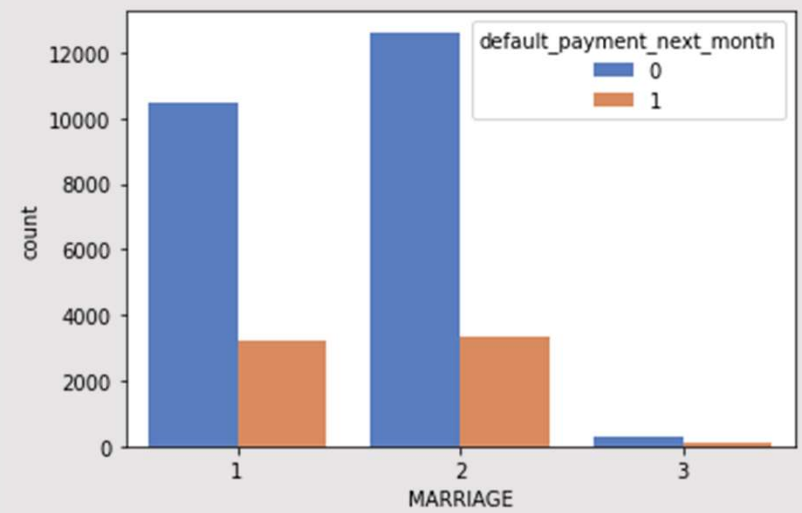
In Males, Non Default credit card holders has highest numbers present. In Females, Non Default credit card holders has highest numbers present.





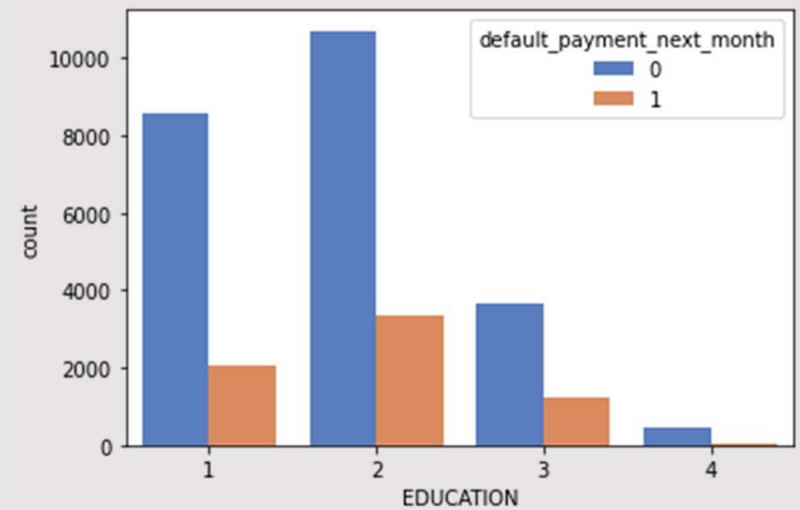
sns.countplot marriagre vs default_payment_next_month

From plot it is clear that people who have marital as status single have more default payment wrt married status people

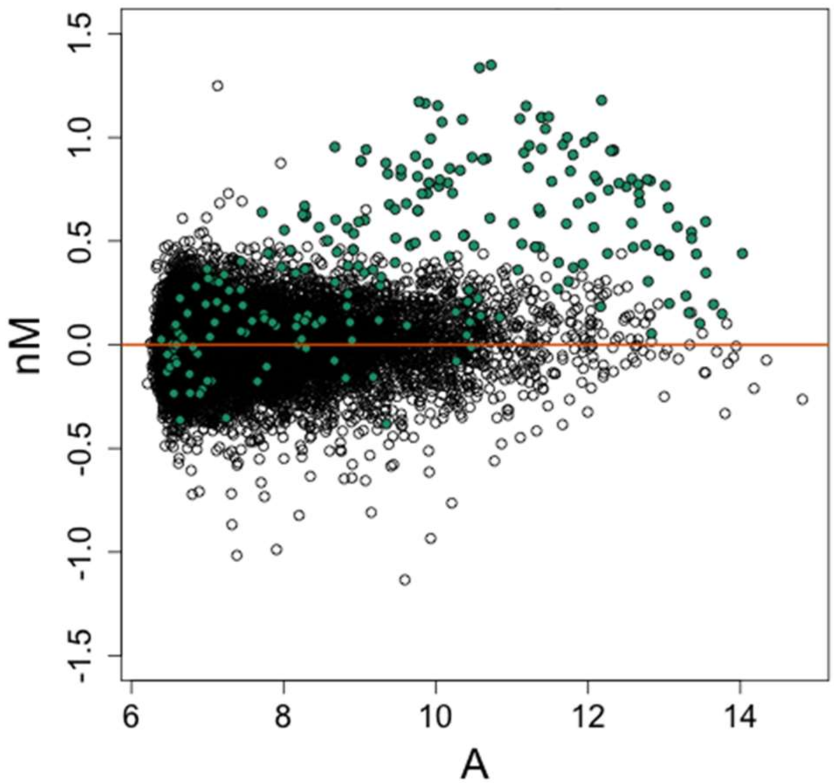


countplot education vs default_payment_next_month

From plot it is clear that people from university have more default payment wrt to all other

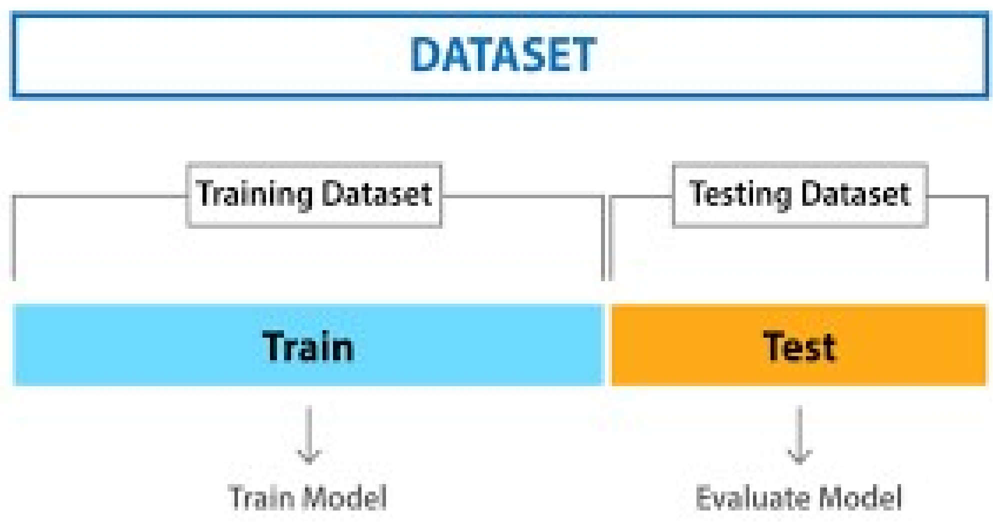


Process of Normalization

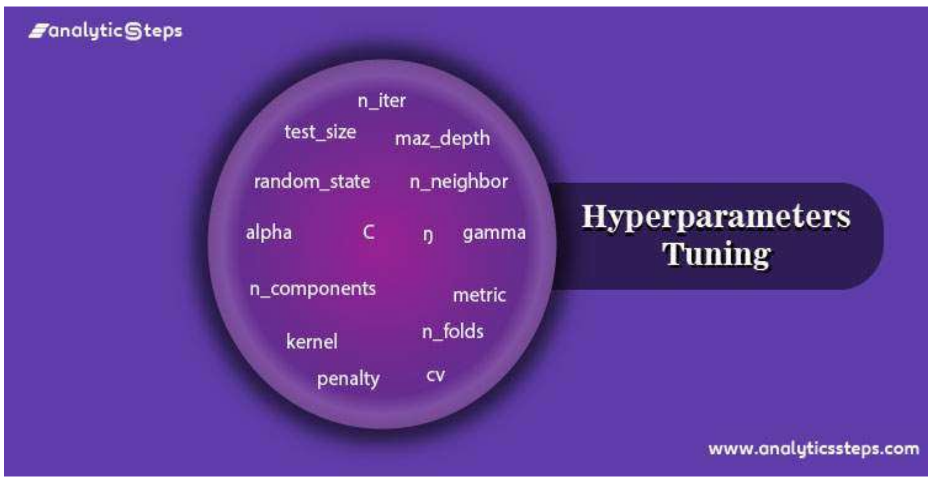


- **Normalization is a scaling technique in Machine Learning** applied during data preparation to change the values of numeric columns in the dataset to use a common scale. It is not necessary for all datasets in a model. It is required only when features of machine learning models have different ranges

Train Test Splitting of Data

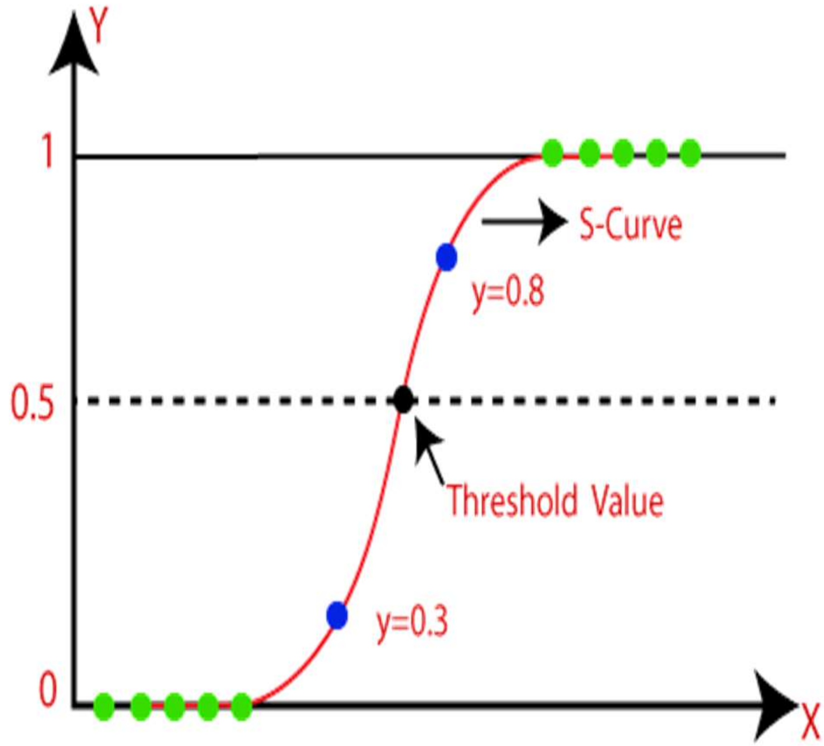


Hyper Parameter Tuning & its Importance



- **Hyper parameter tuning** is an essential part of controlling the behavior of a machine learning model. If we don't correctly tune our hyperparameters, our estimated model parameters produce suboptimal results, as they don't minimize the loss function. This means our model will makes more errors if not tuned.

Introduction to Logistics Regression



- Logistic regression is an example of supervised learning. It is used to calculate or predict the probability of a binary (yes/no) event occurring.

Logistic Regression Metric Values

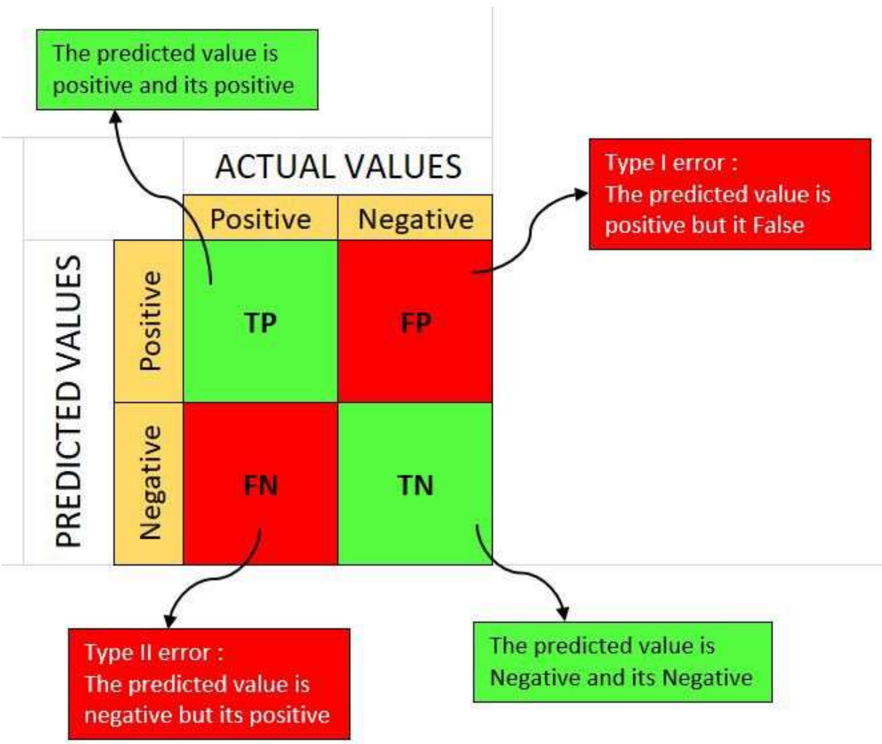
- Y Train Values

Model	Accuracy	Precision	Recall	F1 Score	ROC
Logistic Regression	0.807429	0.70529	0.238501	0.356461	0.604898

- Y Test Values

Model	Accuracy	Precision	Recall	F1 Score	ROC
Logistic Regression	0.816556	0.738056	0.230928	0.351786	0.604203

Confusion Matrix



- **F1-score is the harmonic mean of precision and recall**. It combines precision and recall into a single number using the following formula: This formula can also be equivalently written as, Notice that F1-score takes both precision and recall into account, which also means it accounts for both FPs and FNs
- **Accuracy is a metric** for classification models that

$$Precision = \frac{\#\ of\ True\ Positives}{\#\ of\ True\ Positives + \#\ of\ False\ Positives}$$

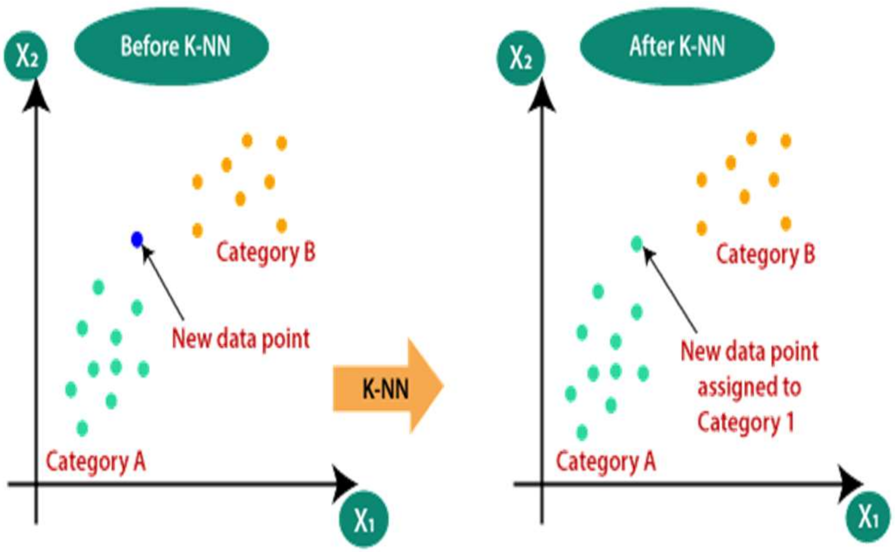
$$Recall = \frac{\#\ of\ True\ Positives}{\#\ of\ True\ Positives + \#\ of\ False\ Negatives}$$

Classification Report & Confusion matrix

	precision	recall	f1-score	support
0	0.82	0.98	0.89	7060
1	0.74	0.23	0.35	1940
accuracy			0.82	9000
macro avg	0.78	0.60	0.62	9000
weighted avg	0.80	0.82	0.78	9000

```
[[6901  159]
 [1492  448]]
```

Introduction to KNN



- The abbreviation KNN stands for “K-Nearest Neighbour ”. **It is a supervised machine learning algorithm**. The algorithm can be used to solve both classification and regression problem statements. The number of nearest neighbors to a new unknown variable that has to be predicted or classified is denoted by the symbol 'K'

KNN Metric Values

- Y Train Values

Model	Accuracy	Precision	Recall	F1 Score	ROC
KNN Classifier	0.843238	0.727184	0.478492	0.57719	0.713394

- Y Test Values

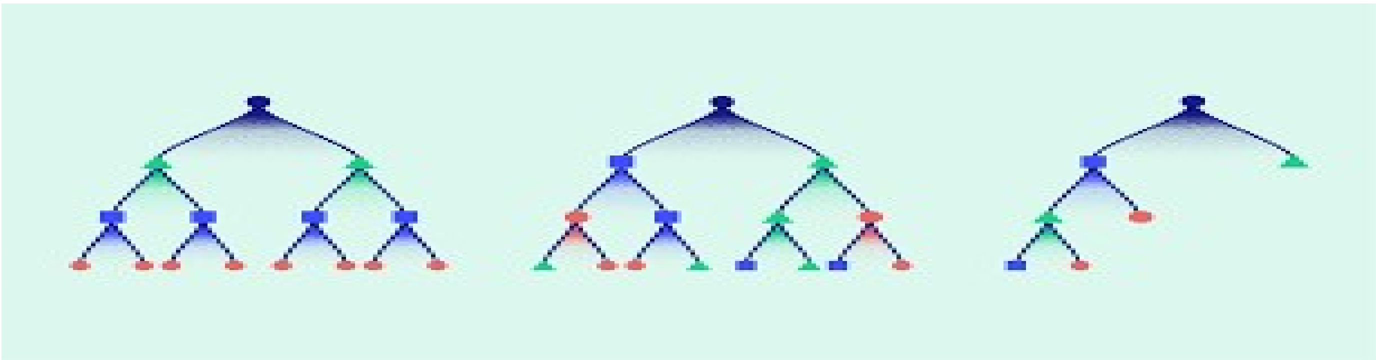
Model	Accuracy	Precision	Recall	F1 Score	ROC
KNN Classifier	0.789444	0.5179	0.335567	0.407257	0.624866

Classification Report & Confusion matrix

```
              precision    recall  f1-score   support

           0       0.83      0.91      0.87      7060
           1       0.52      0.34      0.41      1940

    accuracy                           0.79      9000
   macro avg       0.68      0.62      0.64      9000
weighted avg       0.77      0.79      0.77      9000

[[6454  606]
 [1289  651]]
```

XGBoost Classifier:

- XGBoost is an algorithm that has recently been dominating applied machine learning and Kaggle competitions for structured or tabular data.
- XGBoost is an implementation of gradient boosted decision trees designed for speed and performance.



XGBoost Metric Values

- Y Train Values

Model	Accuracy	Precision	Recall	F1 Score	ROC
XGBOOST Classifier	0.823714	0.696443	0.375213	0.487683	0.664054

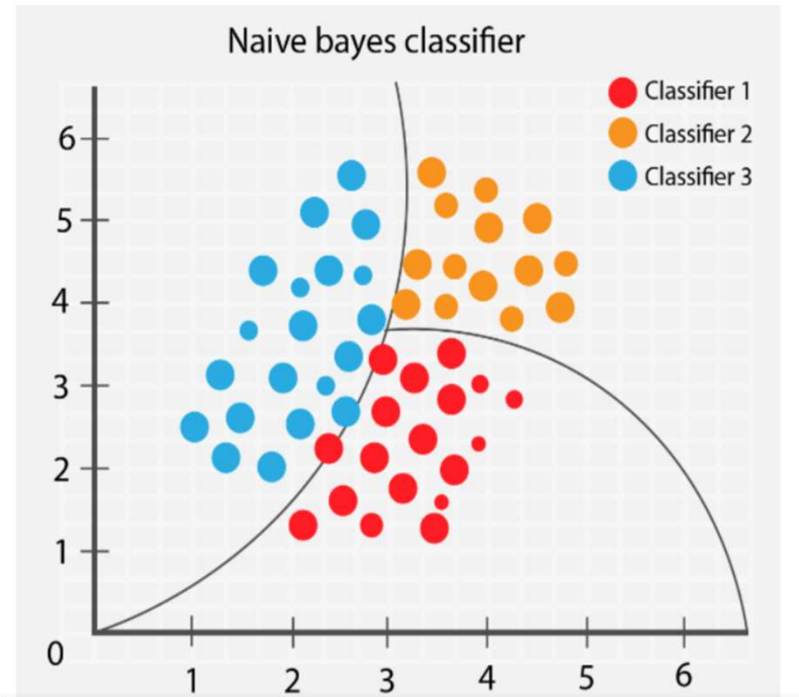
- Y Test Values

Model	Accuracy	Precision	Recall	F1 Score	ROC
XGBOOST Classifier	0.824778	0.676043	0.359278	0.469202	0.655985

Classification Report & Confusion matrix

```
              precision    recall  f1-score   support

           0       0.84      0.95      0.90      7060
           1       0.68      0.36      0.47      1940

    accuracy                           0.82      9000
   macro avg       0.76      0.66      0.68      9000
weighted avg       0.81      0.82      0.80      9000

[[6726  334]
 [1243  697]]
```

Naive Bayes



- Naive Bayes Classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong independence assumptions between the features. They are among the simplest Bayesian network models, but coupled with kernel density estimation, they can achieve high accuracy levels.

Naive Bayes Metric Values

- Y Train Values

Model	Accuracy	Precision	Recall	F1 Score	ROC
Gaussian Naive Bayes	0.588571	0.321506	0.756388	0.45122	0.648312

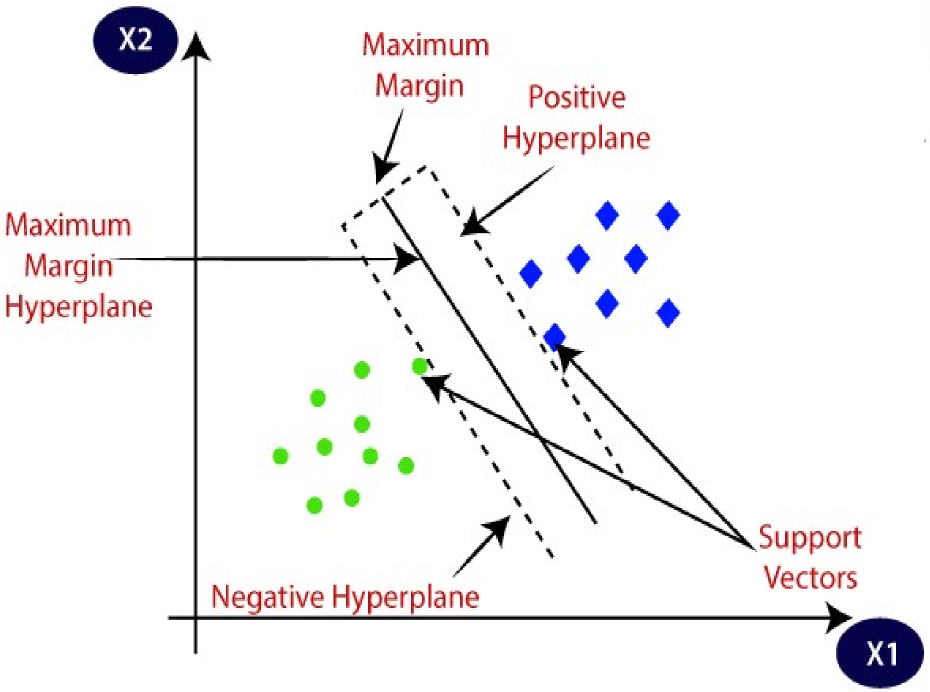
- Y Test Values

Model	Accuracy	Precision	Recall	F1 Score	ROC
Gaussian Naive Bayes	0.584778	0.309276	0.751031	0.43813	0.645062

Classification Report & Confusion matrix

```
              precision    recall  f1-score   support

           0       0.89      0.54      0.67      7060
           1       0.31      0.75      0.44      1940

    accuracy                           0.58      9000
   macro avg       0.60      0.65      0.55      9000
weighted avg       0.76      0.58      0.62      9000

[[3806 3254]
 [ 483 1457]]
```

SVM



The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

-

SVM Metric Values

- Y Train metric values

Model	Accuracy	Precision	Recall	F1 Score	ROC
SVM	0.806524	0.703799	0.232751	0.349816	0.602269

- Y Test metric Values

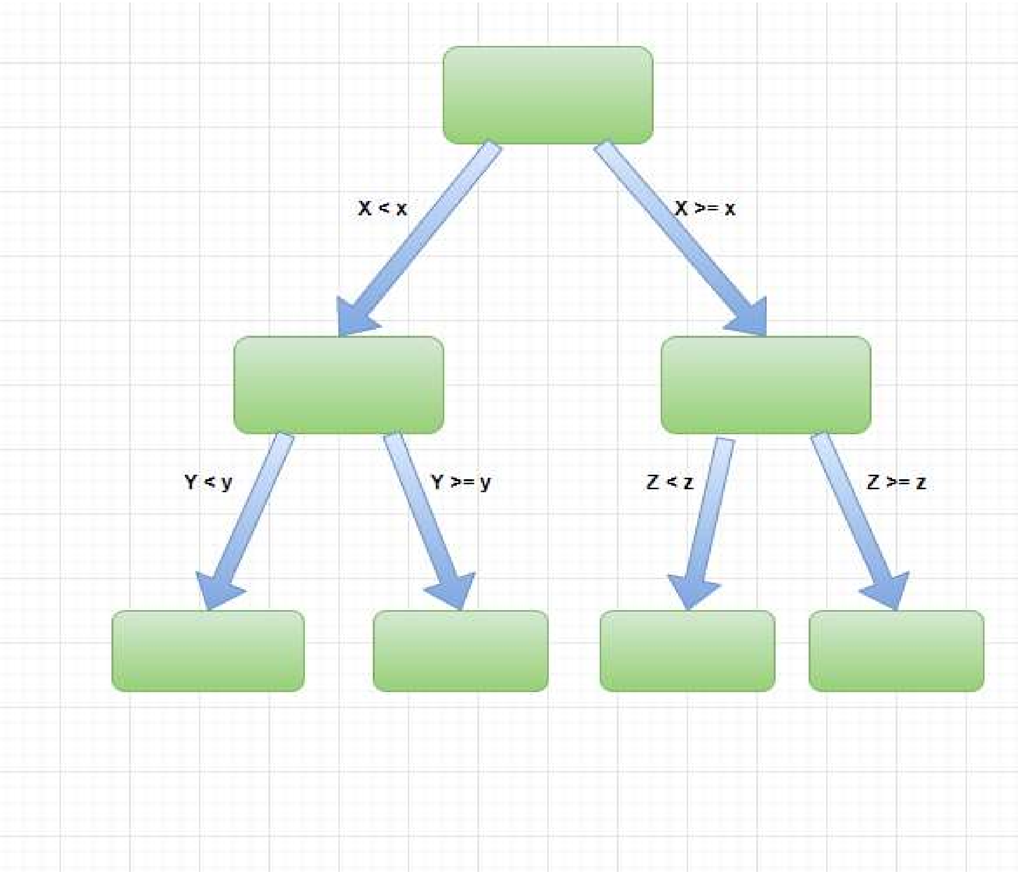
Model	Accuracy	Precision	Recall	F1 Score	ROC
SVM	0.815	0.718601	0.23299	0.351888	0.603959

Classification Report & Confusion matrix

	precision	recall	f1-score	support
0	0.82	0.97	0.89	7060
1	0.72	0.23	0.35	1940
accuracy			0.81	9000
macro avg	0.77	0.60	0.62	9000
weighted avg	0.80	0.81	0.78	9000

```
[[6883  177]
 [1488  452]]
```

Decision Tree



A decision tree is a graphical representation of possible solutions to a decision based on certain conditions. It's called a decision tree because it starts with a single box (or root), which then branches off into a number of solutions, just like a tree.

Decision Tree Metric Values

- Y Train metric values

Model	Accuracy	Precision	Recall	F1 Score	ROC
Decision Tree	0.999714	0.999787	0.998935	0.999361	0.999437

- Y Test metric Values

Model	Accuracy	Precision	Recall	F1 Score	ROC
Decision Tree	0.731111	0.385932	0.418557	0.401583	0.617777

Classification Report & Confusion matrix

```
              precision    recall  f1-score   support

           0       0.84      0.82      0.83      7060
           1       0.39      0.42      0.40      1940

    accuracy                           0.73      9000
   macro avg       0.61      0.62      0.61      9000
weighted avg       0.74      0.73      0.73      9000

[[5768 1292]
 [1128  812]]
```

Conclusion:

Technical Conclusion

- **XGBoost** was able to Predict best with **82%** accuracy score , Followed by **Logistics Regression & SV M at 81% .**

General Conclusion

- **Married ,more Educated Female Credit Card Users , whose ages are between 28-40 and are least likely to default on their payments ,**

- **Single Men , less educated whose ages are lesser than 28 or more than 40 are most likely to default on payments ,**

- **Accordingly the company can device their target customer strategy on the above Niche Market .**