AI maBetter

Capstone Project

CREDIT CARD DEFAULT PREDICTION

By – Lavanya Shinde

CONTENTS

01

INTRODUCTION & PROBLEM STATEMENT

1. As we know in today's times, credit cards have huge risks behind the high returns of banks. The increasing number of credit card users is all about an increase in the number of credit card defaults and that's why the result is amounts of bills & repayment information data have chances to create a risk.

2. The Credit card default prediction is based on the data of all credit card customers. The method which we use to predict and analyze credit card customer default behavior is a typical classification problem.

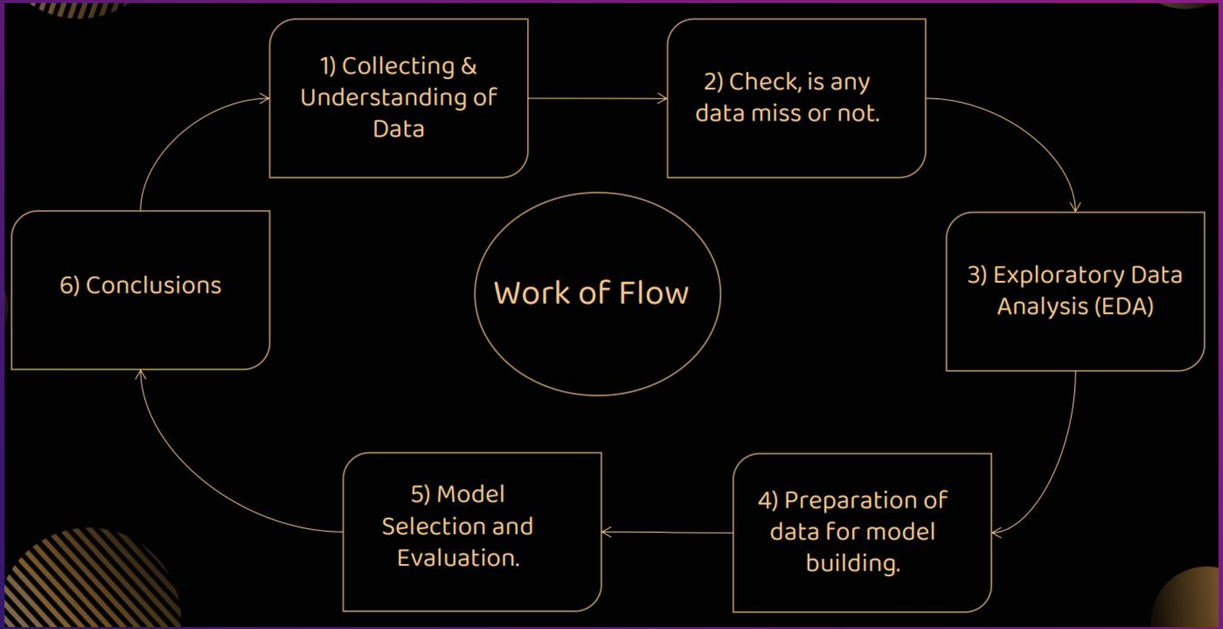
3. According to the Federal Reserve economic data, the default rate on credit loans across all commercial banks is at an all-time high for the past 66 months and it is likely to continue to climb throughout 2020.

4. That’s why, banks must have a risk prediction model and be able to classify the most relative characteristics that are indicative of people who have a higher probability of default on credit.

5. The main purpose is to build a model that allows us to effectively combine static and dynamic features to provide superior predictive performance for financial data.

02 WORKFLOW

- Here is the Simple Work of Flow used for Project :-



03 DATA REVIEW

Let’s understand every columns which is contain in dataset :--

1. ID :- Contain Id Number of Credit Card Users.
2. Limit Bal :- Include the information of Limit Balance.
3. Sex :- Include the information of users is Male or Female.
4. Education :- Include the information of Education of Users.
5. Marriage :- Is user single or married.
6. Age :- Age information of users.
7. Pay-0 to Pay-6 :-History of past payments from April to September.
8. Bill-Amt1 to Bill-Amt6 :- Amount of bill statement from April to September.
9. Pay-Amt1 to Pay-Amt6 :- Amount of Previous Payment from April to September.
10. Default Payment Next Month : - Default payment information.

Exploring the Data

```python
# Check The First Five rows
credit_data.head()
```

	ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	...	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2	PAY_AMT3	PAY_AMT4	PAY_AMT5	PAY_AMT6	default payment next month
0	1	20000	2	2	1	24	2	2	-1	-1	...	0	0	0	0	689	0	0	0	0	1
1	2	120000	2	2	2	26	-1	2	0	0	...	3272	3455	3261	0	1000	1000	1000	0	2000	1
2	3	90000	2	2	2	34	0	0	0	0	...	14331	14948	15549	1518	1500	1000	1000	1000	5000	0
3	4	50000	2	2	1	37	0	0	0	0	...	28314	28959	29547	2000	2019	1200	1100	1069	1000	0
4	5	50000	1	2	1	57	-1	0	-1	0	...	20940	19146	19131	2000	36681	10000	9000	689	679	0

5 rows × 25 columns

```python
# last five rows
credit_data.tail()
```

	ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	...	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2	PAY_AMT3	PAY_AMT4	PAY_AMT5	PAY_AMT6	default payment next month
29995	29996	220000	1	3	1	39	0	0	0	0	...	88004	31237	15980	8500	20000	5003	3047	5000	1000	0
29996	29997	150000	1	3	2	43	-1	-1	-1	-1	...	8979	5190	0	1837	3526	8998	129	0	0	0
29997	29998	30000	1	2	2	37	4	3	2	-1	...	20878	20582	19357	0	0	22000	4200	2000	3100	1
29998	29999	80000	1	3	1	41	1	-1	0	0	...	52774	11855	48944	85900	3409	1178	1926	52964	1804	1
29999	30000	50000	1	2	1	46	0	0	0	0	...	36535	32428	15313	2078	1800	1430	1000	1000	1000	1

5 rows × 25 columns

```python
# Check Total Number of rows and Columns in dataset.
print(f' The shape of dataset is ({(credit_data.shape[0])} x {(credit_data.shape[1])})\n Total Number of Rows are : {(credit_data.shape)[0]}\n Total Number of Columns are : {(credit_data.shape)[1]}')
```

```
The shape of dataset is (30000 x 25)
Total Number of Rows are : 30000
Total Number of Columns are : 25
```

```
[ ] credit_data.describe().T
```

	count	mean	std	min	25%	50%	75%	max
ID	30000.0	15000.500000	8660.398374	1.0	7500.75	15000.5	22500.25	30000.0
LIMIT_BAL	30000.0	167484.322667	129747.661567	10000.0	50000.00	140000.0	240000.00	1000000.0
SEX	30000.0	1.603733	0.489129	1.0	1.00	2.0	2.00	2.0
EDUCATION	30000.0	1.853133	0.790349	0.0	1.00	2.0	2.00	6.0
MARRIAGE	30000.0	1.551867	0.521970	0.0	1.00	2.0	2.00	3.0
AGE	30000.0	35.485500	9.217904	21.0	28.00	34.0	41.00	79.0
PAY_0	30000.0	-0.016700	1.123802	-2.0	-1.00	0.0	0.00	8.0
PAY_2	30000.0	-0.133767	1.197186	-2.0	-1.00	0.0	0.00	8.0
PAY_3	30000.0	-0.166200	1.196868	-2.0	-1.00	0.0	0.00	8.0
PAY_4	30000.0	-0.220667	1.169139	-2.0	-1.00	0.0	0.00	8.0
PAY_5	30000.0	-0.266200	1.133187	-2.0	-1.00	0.0	0.00	8.0
PAY_6	30000.0	-0.291100	1.149988	-2.0	-1.00	0.0	0.00	8.0
BILL_AMT1	30000.0	51223.330900	73635.860576	-165580.0	3558.75	22381.5	67091.00	964511.0
BILL_AMT2	30000.0	49179.075167	71173.768783	-69777.0	2984.75	21200.0	64006.25	983931.0
BILL_AMT3	30000.0	47013.154800	69349.387427	-157264.0	2666.25	20088.5	60164.75	1664089.0
BILL_AMT4	30000.0	43262.948967	64332.856134	-170000.0	2326.75	19052.0	54506.00	891586.0
BILL_AMT5	30000.0	40311.400967	60797.155770	-81334.0	1763.00	18104.5	50190.50	927171.0
BILL_AMT6	30000.0	38871.760400	59554.107537	-339603.0	1256.00	17071.0	49198.25	961664.0
PAY_AMT1	30000.0	5663.580500	16563.280354	0.0	1000.00	2100.0	5006.00	873552.0
PAY_AMT2	30000.0	5921.163500	23040.870402	0.0	833.00	2009.0	5000.00	1684259.0
PAY_AMT3	30000.0	5225.681500	17606.961470	0.0	390.00	1800.0	4505.00	896040.0
PAY_AMT4	30000.0	4826.076867	15666.159744	0.0	296.00	1500.0	4013.25	621000.0
PAY_AMT5	30000.0	4799.387633	15278.305679	0.0	252.50	1500.0	4031.50	426529.0
PAY_AMT6	30000.0	5215.502567	17777.465775	0.0	117.75	1500.0	4000.00	528666.0
default payment next month	30000.0	0.221200	0.415062	0.0	0.00	0.0	0.00	1.0

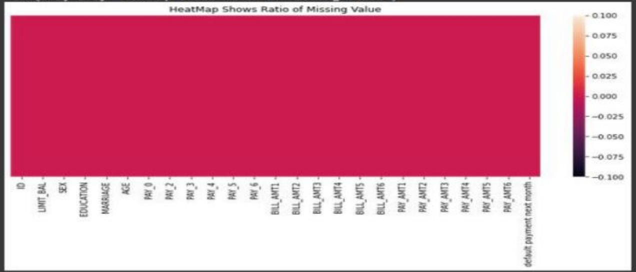
Let's Check Missing Value in Dataset. If some value is null in dataset , then we target every missing value to fill & make data complete .

```
## Find the missing value, show the total null values for each column and sort it in descending order
credit_data.isnull().sum().sort_values(ascending=False)[:10]
```

```
ID           0
BILL_AMT2    0
PAY_AMT6     0
PAY_AMT5     0
PAY_AMT4     0
PAY_AMT3     0
PAY_AMT2     0
PAY_AMT1     0
BILL_AMT6    0
BILL_AMT5    0
dtype: int64
```

```
# Visulaizing the null values.
plt.figure(figsize=(13,5))
sns.heatmap(credit_data.isnull(),yticklabels=False)
plt.title("HeatMap Shows Ratio of Missing Value")
```

```
Text(0.5, 1.0, 'HeatMap Shows Ratio of Missing Value')
```



But there are no null value in our dataset. So, data is perfect for start the project .

POINTS FOUND FROM DATA REVIEW.

- There are No Missing Values present in Dataset
- There are No Duplicate values present in Dataset
- There are No null values. ➢ 9 Categorical variables present.
- 6 Months payment and bill data available.
- In our Dataset There are total 30000 rows and 25 columns

```
# Check Total Number of rows and Columns in dataset.
print(f' The shape of dataset is ({(credit_data.shape[0])} x {(credit_data.shape[1])})\n Total Number of Rows are : {(credit_data.shape)[0]}\n Total Number of Columns are : {(credit_data.shape)[1]}')
```

```
The shape of dataset is (30000 x 25)
Total Number of Rows are : 30000
Total Number of Columns are : 25
```

```
# Checking Duplicate Values
value= credit_data.duplicated().sum()
print("The Total number of duplicate values in the data set is =",value)
```

```
The Total number of duplicate values in the data set is = 0
```

- Count the value from default_payment_next_month, sex, education, marriage & age

Count the value from default_payment_next_month, sex, education, marriage & age

```
# counts the values of default payment next month cloumns
pd.DataFrame(credit_data['default_payment_next_month'].value_counts()).T
```

	0	1
default_payment_next_month	23364	6636

default_payment_next_month :-

> 0 - Non-Defaulter

> 1 - Defaulter

```
# Get the proportion of customers who had default payment in the next month
# About 22% customers had default payment next month

credit_data['default_payment_next_month'].value_counts(normalize=True)
```

```
0    0.7788
1    0.2212
Name: default_payment_next_month, dtype: float64
```

```
# Counts the values of SEX variable data set
pd.DataFrame(credit_data['SEX'].value_counts()).T
```

	2	1
SEX	18112	11888

Sex :-

> 1 - Male

> 2 - Female

- Count the value from default_payment_next_month, sex, education, marriage & age

```
# Counts the values of Education
pd.DataFrame(credit_data['EDUCATION'].value_counts()).T
```

	2	1	3	5	4	6	0
EDUCATION	14030	10585	4917	280	123	51	14

Education :-

> 1 - Graduate School
>
> 2 - University
>
> 3 - High School
>
> 0,4,5,6 - Others

```
# Counts the values of Marriage
pd.DataFrame(credit_data['MARRIAGE'].value_counts()).T
```

	2	1	3	0
MARRIAGE	15964	13659	323	54

+ Code + Text

Marriage :-

> 1 - Married
>
> 2 - Single
>
> 3 & 0 - Others

```
# Counts the values of Age
pd.DataFrame(credit_data['AGE'].value_counts()).T
```

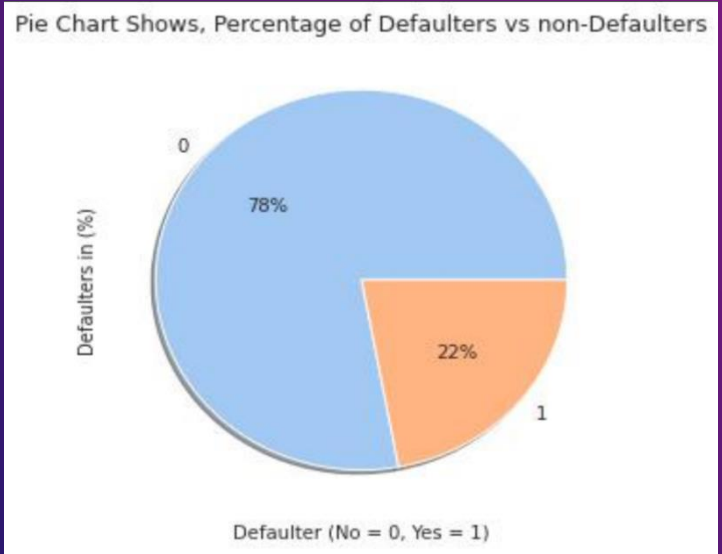
	29	27	28	30	26	31	25	34	32	33	...	67	69	70	68	73	72	75	71	79	74
AGE	1605	1477	1409	1395	1256	1217	1186	1162	1158	1146	...	16	15	10	5	4	3	3	3	1	1

1 rows × 56 columns

04

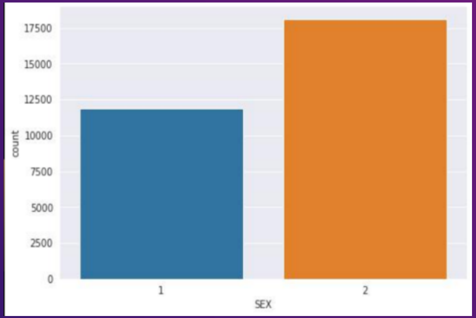
EXPLORATORY DATA ANALYSIS

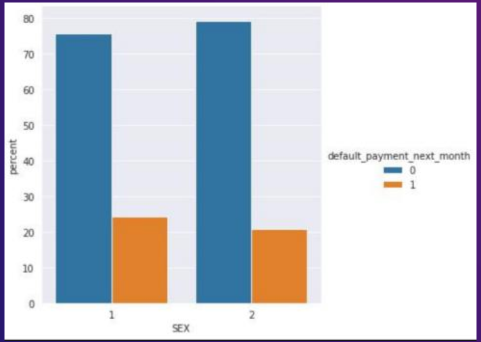
1) Visualize the data of Defaulters vs Non-Defaulters



Key Insights:
So , According to our pie chart visualization , we can say that 22% is Defaulters & 78% is Non-Defaulters

2) Visualize the data of Male vs Female for Credit





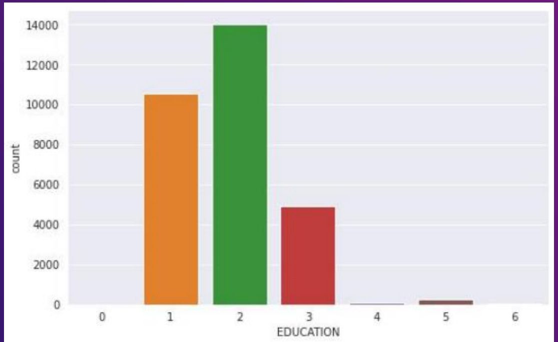
Key Insights:

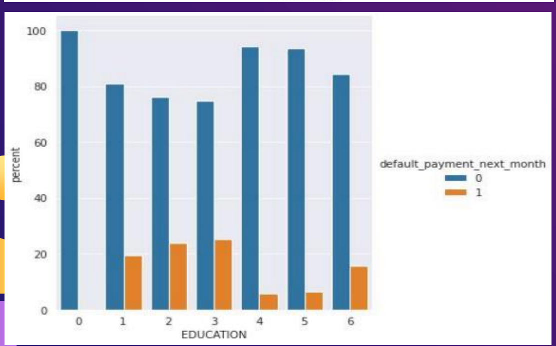
Chart shows, Male credit holder is less Than Female Credit Card Holder. In Another Chart we can see that In defaulters list male credit holder is Higher than Female Credit Holder

Now Here We Observe From the above chart, There are Female Credit Holder
is More than Male Credit Holder

- 1 :- Male
- 2 :- Female

3) Visualize the data of Education of Credit Card Holders



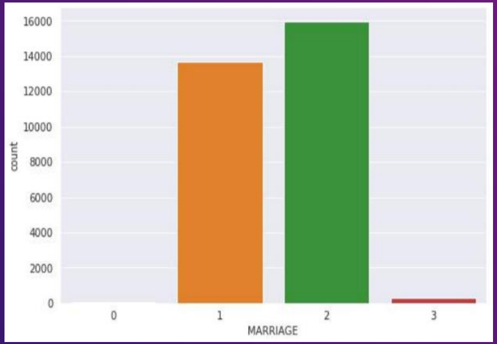


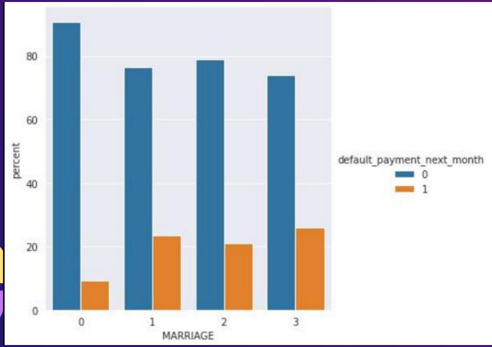
Key Insights:

From the above visualization data, we see Highest Number of credit holders are university students then 2ndHighest are Graduate Students then 3rdHighest from High school students & Remaining from Others. In below chart, we can say that other category students have higher number of default payment with the comparison of graduate, university &high school students.

1 = graduate school; 2 = university; 3 = high school; 0 = others

4) Visualize the data From Marriage Column



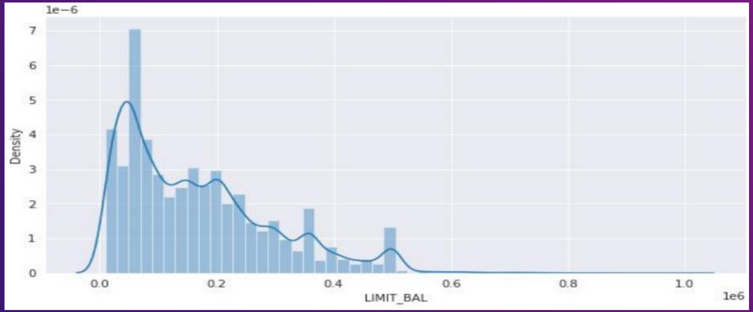


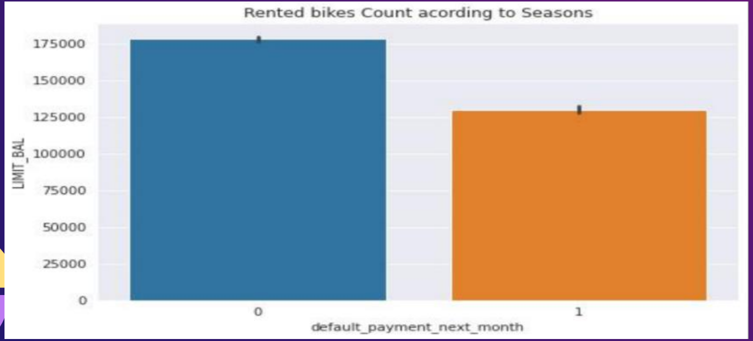
Key Insights:

Here Chart shows

- 1 - married
- 2 - single
- 3 & 0 - others
- From the above visualization data we see the Highest Number of credit holders are Single, then 2nd Highest are Married then 3rd & 0 from Others.
- In below chart, we can say that married people have less number of defaulters with the comparison of other marriage person category lists.

5) Visualize the data of default payment next month with limit Balance

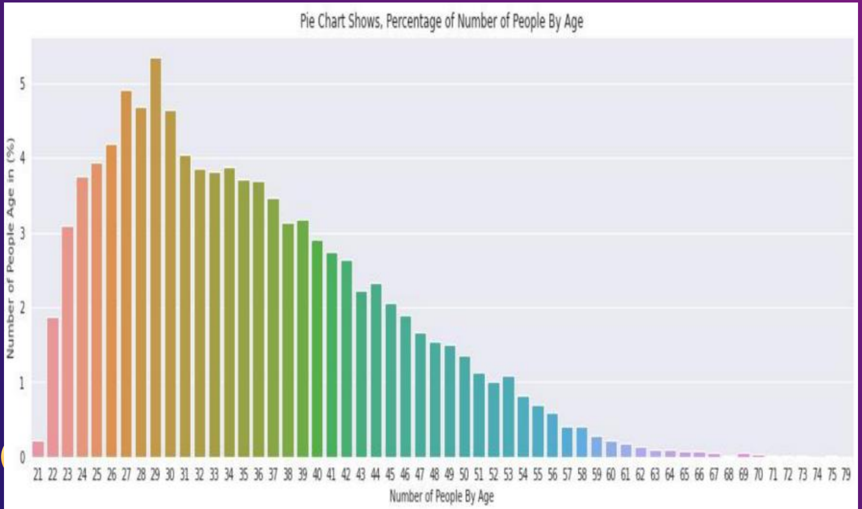




Key Insights:

In this chart we clearly see that The Maximum amount of given credit in NT dollars is 50,000 followed by 30,000 and 20,000.

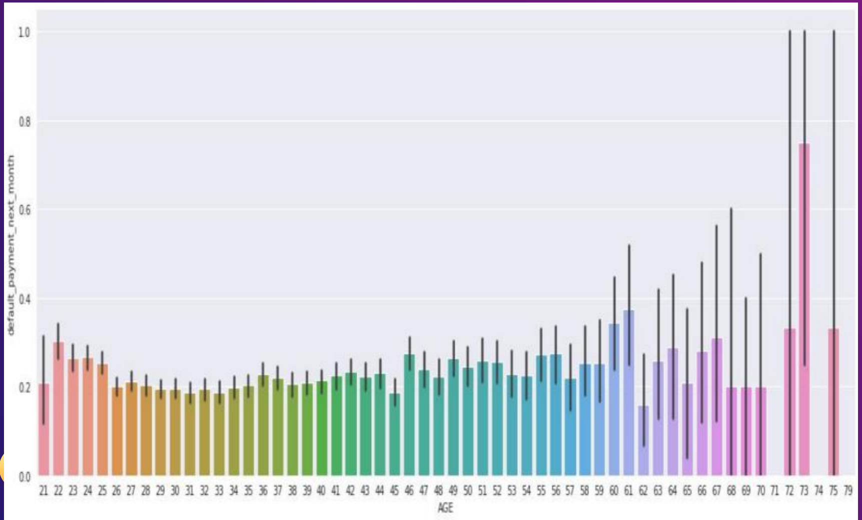
6) Visualize the data of Number of People By Age



Key Insights:

From the above Age Data Visualization, We observe that Most of credit card holders age start from 24-32 Years old and people above 61year old use credit cards very rarely.

7) Visualize the data of default payment next month with Age Column.



Key Insights:

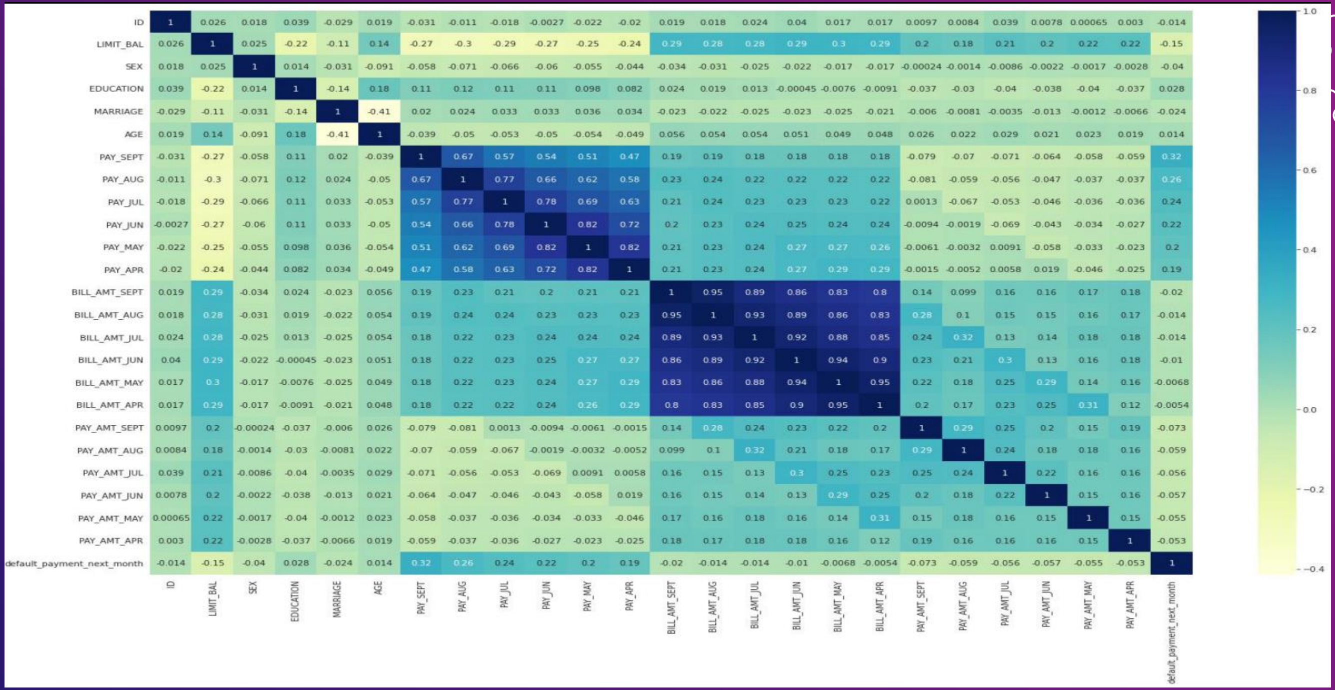
From the chart, we find the relationship between age and defaulters. We can say that people who are 60 years or older may not use their credit card frequently.

8) Smote Operation.

Key Insights:

SMOTE stands for Synthetic Minority Oversampling Technique. It is a statistical technique for increasing the number of cases in our dataset in a balanced way. The component works by generating new instances minority cases that we supply as input. After performing the SMOTE operation, we get this balance Dataset.

Checking the Correlation between dependent and independent variable.



Now we Start Model Building For :-

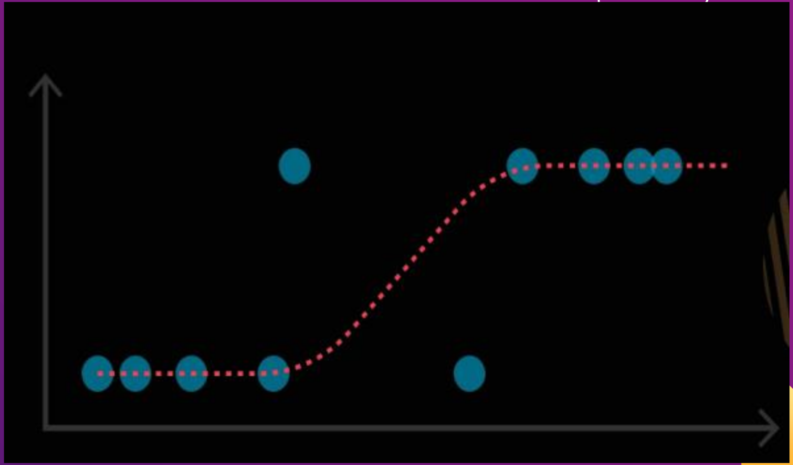
1) Logistic Regression
2) Random Forest Classifiers
3) Support Vector Classifier
4) XGBoost Classifiers
5) Model Evaluation
6) AUC-ROC Curve Comparison
7) Feature Importance

1) Logistic Regression

What is Logistic Regression ?

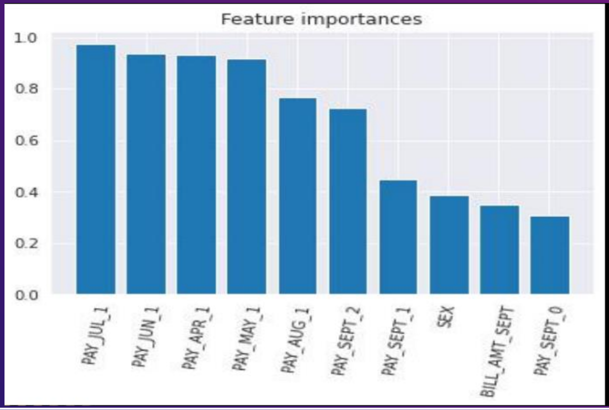
:- Logistic Regression is similar to Linear Regression, It is also used to find the relationship between the Dependent variable and one/more Independent Variable, also it's used to make
predictions for a categorical variable as well as used to handle the classification problems.

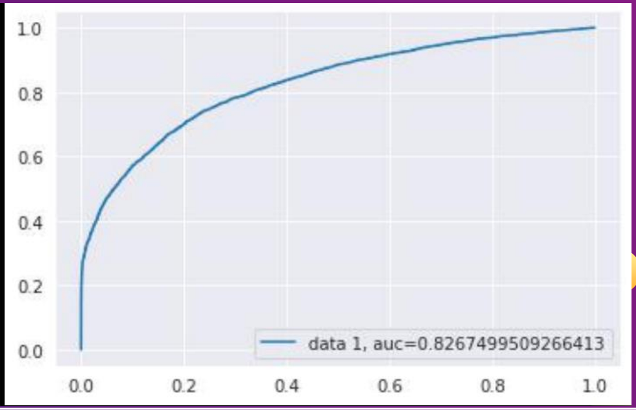
Library I used for Logistic Regression:
from.sklearn.linear_model import
LogisticRegressiom



1) Logistic Regression

Accuracy Result for Both Train & Test Data with respect to parameter :-
{'C': 0.01, 'penalty': 'l2'}
1) The accuracy score for the Train data is :- 0.752323
2) The accuracy score for the Test data is :- 0.748913
3) The precision score for the Train data is :- 0.681971
4) The recall score for the Train data is :- 0.787361
5) The f1 score for the Train data is :- 0.730886
6) The roc score for the Train data is :- 0.753454



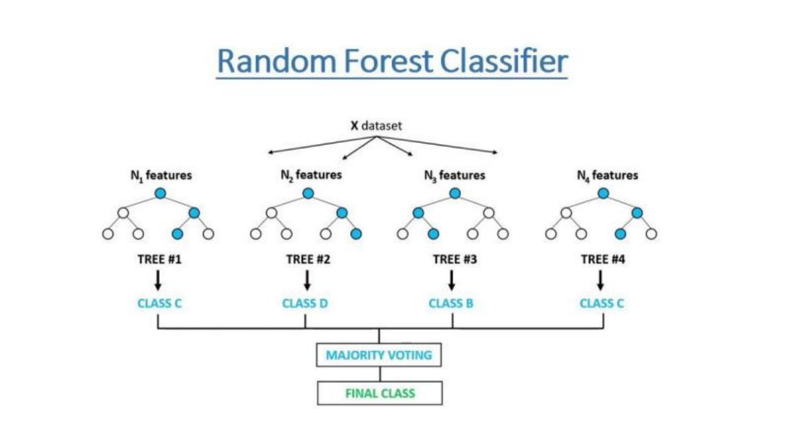


2) Random Forest Classifiers

What is Random Forest Classifiers ?

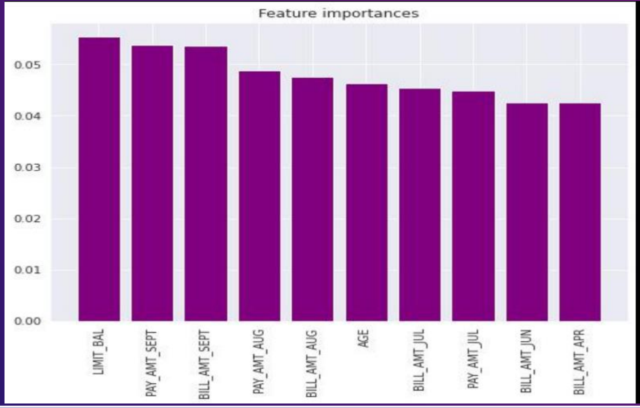
:- Random Forest Classifier is a technique that makes an aggregated prediction using a group of decision trees trained Using the bootstrap method with extra randomness, while growing trees by searching for the best features among a randomly selected feature subset.

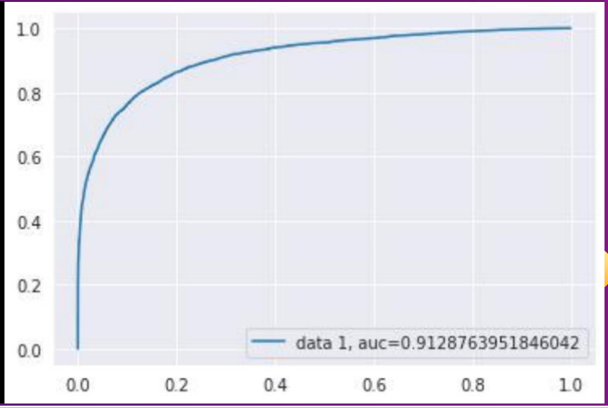
Library used for Random Forest Classifiers :-
from sklearn.ensemble import RandomForestClassifier



2) Random Forest Classifiers

Accuracy Result for Both Train & Test Data with respect to parameter :-
{‘max_depth’: 30, ‘n_estimators’:200}
1) The accuracy score for the Train data is :- 0.999393
2) The accuracy score for the Test data is :- 0.832695
3) The precision score for the Train data is :-0.801556
4) The recall score for the Train data is :- 0.854771
5) The f1 score for the Train data is :- 0.827309
6) The roc score for the Train data is :- 0.833990



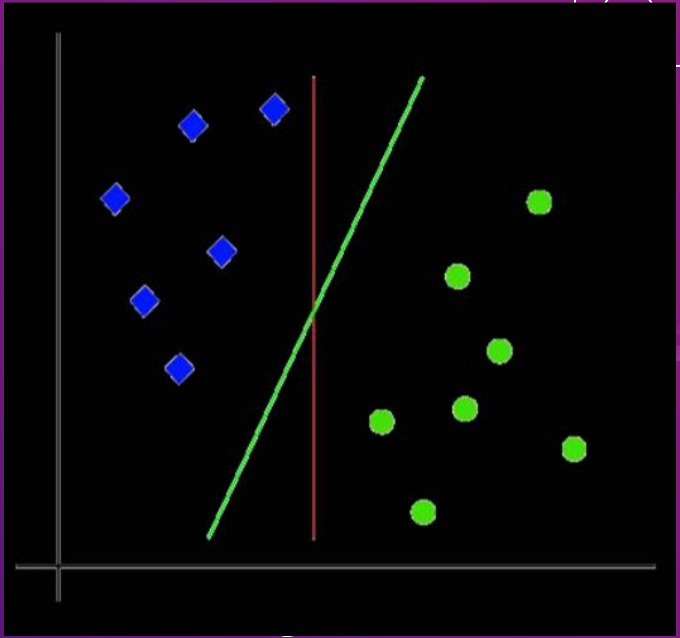


3) Support Vector Classifier

What is Support Vector Classifier ?

Support vector classifiers are a set of supervised learning methods used for classification, regression and outlier detection. The big advantage of support vector machines is that Effective in high dimensional spaces as well as it’s still effective in cases where the number of dimensions is greater than the number of samples.

Library used for Random Forest Classifiers:
from sklearn.model_selection import
GridSearchCV

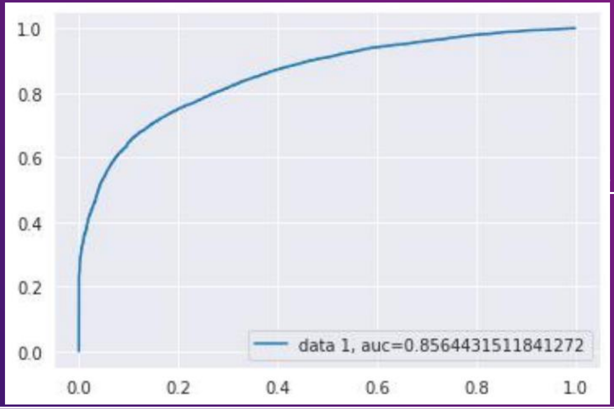


3) Support Vector Classifier

Accuracy Result for Both Train & Test Data with respect to parameter :-

{‘c’: 10, ‘kernal’: ‘rbf’}

1) The accuracy score for the Train data is :- 0.752323
2) The accuracy score for the Test data is :- 0.748913
3) The precision score for the Train data is :-0.681971
4) The recall score for the Train data is :- 0.787361
5) The f1 score for the Train data is :- 0.730886
6) The roc score for the Train data is :- 0.753454

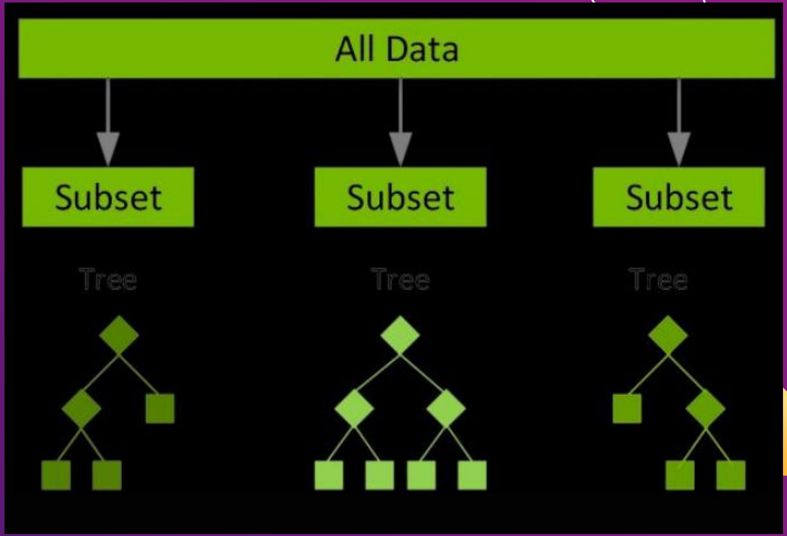


4) XGBoost Classifiers

What is XGBoost Classifiers?

:- XGBoost, which also stands for Extreme Gradient Boosting, is a scalable & distributed gradient-boosted decision tree (GBDT) machine learning library. It provides parallel tree boosting and is the leading machine learning library for regression, classification, and ranking problems.

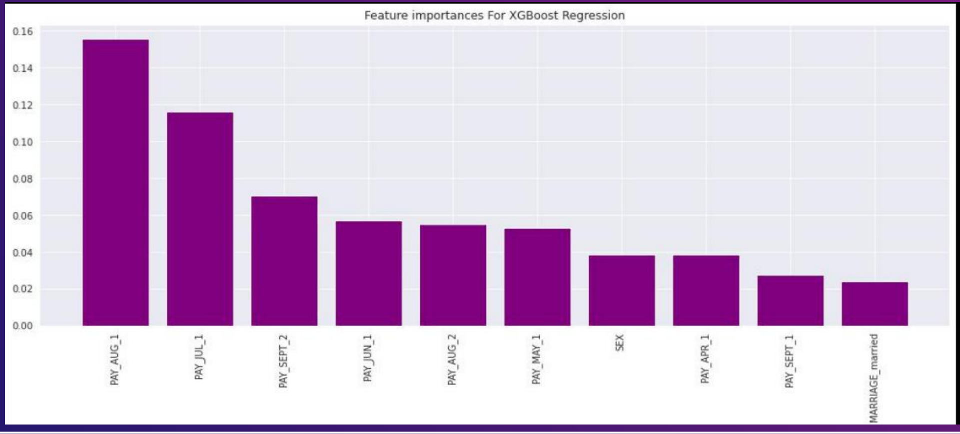
Library used for Random Forest Classifiers:
import xgboost as xgb
from xgboost import XGBClassifier

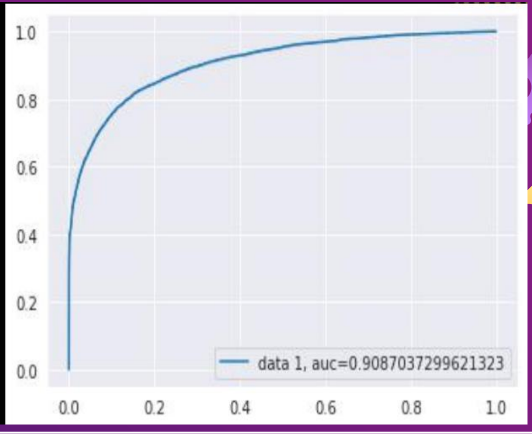


4) XGBoost Classifiers

Accuracy Result for Both Train & Test Data with respect to parameter :-

1) The accuracy score for the Train data is :- 0.785191
2) The accuracy score for the Test data is :- 0.769859
3) The precision score for the Train data is :-0.696238
4) The recall score for the Train data is :- 0.816425
5) The f1 score for the Train data is :- 0.751557
6) The roc score for the Train data is :- 0.775836





5. Evaluate the Model

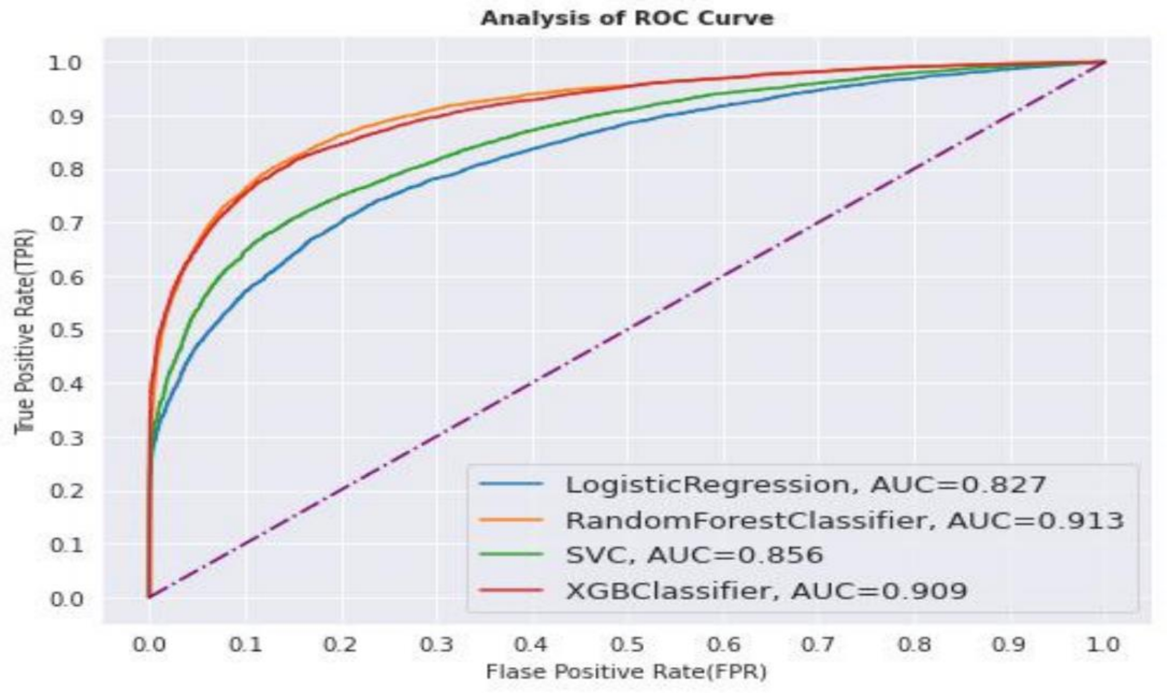
	Classifiers	Train Accuracy	Test Accuracy	Precision Score	Recall Score	F1 Score
0	Logistic Regression	0.752324	0.748914	0.681971	0.787361	0.730887
1	Support Vector Classifier	0.752324	0.748914	0.681971	0.787361	0.730887
2	Random Forest Classifier	0.998371	0.836197	0.803243	0.859900	0.830606
3	Xgboost Classifiers	0.908870	0.830232	0.788327	0.860419	0.822797

- From the Above Table Data we observe Random Forest Classifier Perform Best with the comparision of other models.

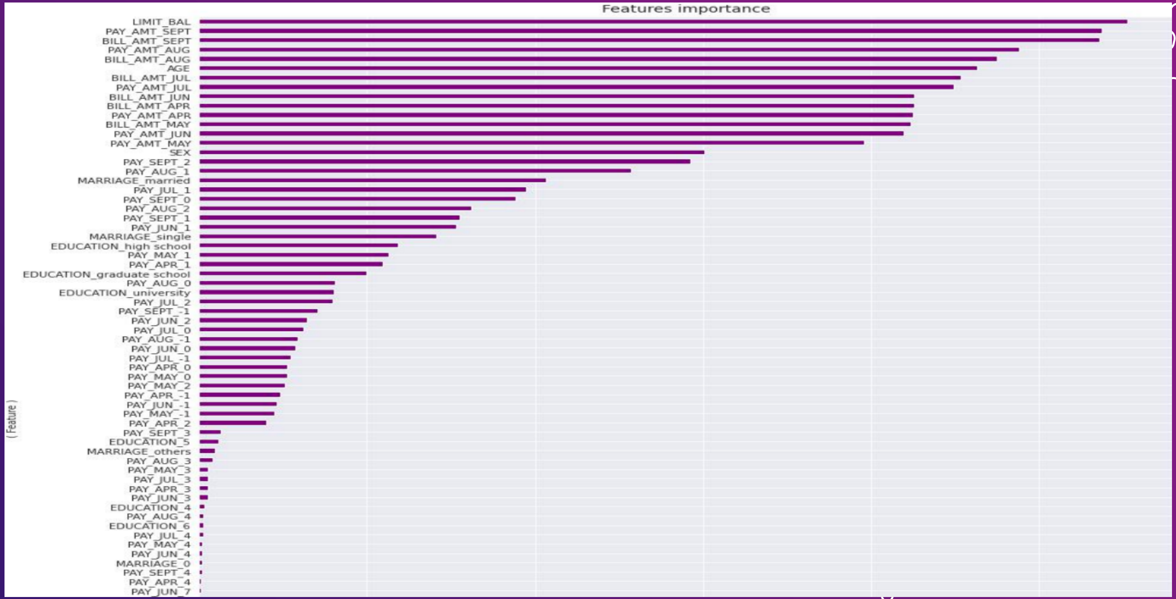
6. AUC-ROC Curve Comparison:

Classifiers	False Positive Rate	True Positive Rate	AUC
LogisticRegression	[0.0, 0.0, 0.0, 0.00012968486577616392, 0.0001...	[0.0, 0.00012970168612191958, 0.09649805447470...	0.826750
RandomForestClassifier	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...	[0.0, 0.035667963683527884, 0.0360570687418936...	0.912876
SVC	[0.0, 0.0, 0.0, 0.00012968486577616392, 0.0001...	[0.0, 0.00012970168612191958, 0.16264591439688...	0.856443
XGBClassifier	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...	[0.0, 0.00012970168612191958, 0.00194552529182...	0.908704

7. AUC-ROC Curve Comparison



Feature Importance



From the Above Graph We observer LIMT_BAL, BILL_AMT_SEPT AND PAY_AMT_SEPT are the strongest predictors of future payment default risk.

05 CONCLUSION

1. We observe 78% of people are Non-defaulters and the remaining 22% are Defaulters
2. Male credit holder is less Than Female Credit Card Holder and if we compare male/female with defaulters list we observe that, in defaulters list male credit holder is Higher than Female Credit Holder.
3. Highest Number of credit holders are university students then 2nd Highest are Graduate Students then 3rd Highest from High school Students & Remaining from Others.
4. Highest Number of credit holders are Single, then 2nd Highest are Married & remaining are from Others category. As well as we observe married people have a smaller number of defaulters with the comparison of other's marriage person category list
5. The Maximum amount of given credit in NT dollars is 50,000 followed by 30,000 and 20,000.
6. We observe Most of credit card holders' ages start from 24-32 Years and people's age above 61 year, they use credit cards very rarely.
7. We find the relationship between age, and defaulter's & we can say that people who are 60 years or older, that maybe they don’t use their credit card frequently.

8. In both cases they have a negative impact on the bank, since false positives leads to unsatisfied customers and false negative leads to Financial loss.
9. XGBoost Classifier having Recall, F1_score, and ROC Score values equals 82%, 77%, and 86% and Random forest Classifier having Re call, F1-score and ROC Score values equals 81%, 75%, and 84%.
10. XGBoost Classifier and Decision Tree Classifier are giving us the best Recall, F1_score, and ROC Score among other algorithms.
11. We observe XGBoost classifier and decision tree classifier are the best to predict whether the credit card user is defaulter or non- defaulter.
12. Random Forest is Higher Precision than Logistic Regression. That's why Random forest is better than logistic regression and it's suitable for our machine learning model.

THANK YOU!