1. Introduction

1.1 **Artificial Intelligence**

Artificial Intelligence refers to the computer ability to perform complex task that only a human can do by mimicking human reasoning, decision making and problem-solving capabilities (Coursera, 2024). AI is a includes wide variety of technologies such as machine learning, deep learning, Computer vision, data mining, and natural language processing (Coursera, 2024). AI learn through the huge amount of dataset collected by different organizations for variety of sources which can be used to train AI model so that it can assist the day-to-day business operation (AWS, 2024). AI has been used in business for task automation that can be done by human being to reduce manpower cost, task such as customer service, fraud detection, insight generation, etc. (Craig, 2024).

1.2 **Machine learning**

Machine learning is the field of artificial intelligence which can mimic human intelligence for Performing a complex task using the problem-solving approach used by humans, and with more exposure to learning data it gains experience to improve their performance and accuracy (Brown, 2021). As per the UC Berkeley School of Information, a machine learning algorithm comprises three main components that are decision process, error function and model optimization (datascience@berkeley, 2022).

- **Decision process:** The decision process is about how the machine learning model makes the prediction, classifies the correct category and groups the common data based on the given dataset which is mapped with the input feature to the output/ target feature that is both labelled and unlabeled.
- **Error Function:** The Error function in machine learning describes the performance of the model on unseen data. It compares the predicted data and actual data to evaluate its overall accuracy and performance.
- **Model Optimization:** Model optimization is concerned with improving the performance of the machine learning model by finding out the best model parameter and adjusting the weight to the model to minimize the error function.

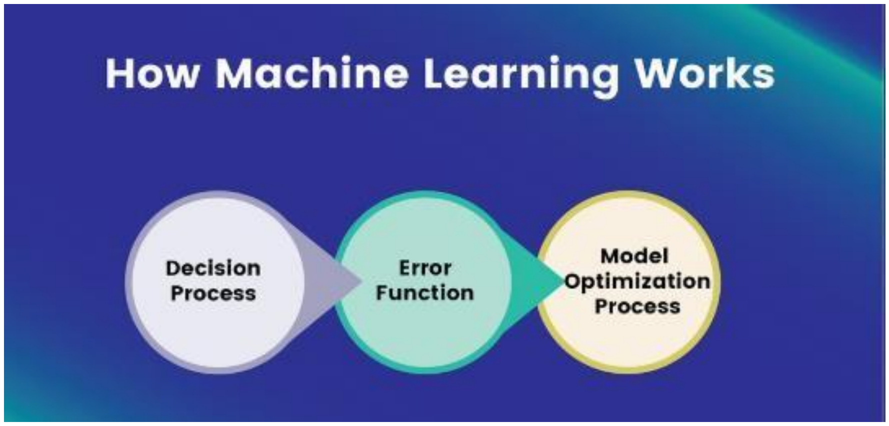


Figure 1 Main Component of Machine Learning (SALAH, 2024)

The machine learning algorithm is further categorized into supervised, unsupervised learning, semi-supervised and reinforcement learning.

Supervised learning: Supervised learning is a sub-group of a machine learning algorithm which predicts the numerical data and classifies the category of the data based on the training of the model by using the labelled data set. The label dataset includes both feature variable data and target variable data which is to be predicted (IBM, 2024). It is further classified into classification and regression tasks.

Classification: It is the process of classifying the data into a specific category or predicting the category of the data based on the given input features variable. Examples of classification algorithms are logistic regression, KNN, Support vector machine, Random Forest classifier etc.

1.3 Problem Domain

According to the Federal Reserve, the ratio of credit card default is increasing due to various reason which includes an increase in health care service, job scarcity, student debt etc. (Streaks, 2024). A Credit default happens when a credit card holder can't pay their debt amount for a certain period which degrades their credit score, legal action from the issuing authorities and their credit card might be closed by their respective bank branch (Streaks, 2024). Most people fall into the credit debt trap as they accumulate high amounts of credit debt to pay for their expenses, mortgage, car payments etc. which might affect the bank's economic growth. For the Banking sector, credit card is a major source of income which generates some revenue for them. so, they may consider insight generated from their customer credit score and default payment status to track the customer with default payment activity and cancel their credit card to prevent financial loss and maintain healthy market valuation (Junhong Li, 2024). A better-performing machine learning model with good accuracy will help the banking institution to decide whom to lend money to customers waiting for credit card approval, it also aware the customer about the consequences of credit card default which can affect their credit score (Florentin Butaru, 2016).

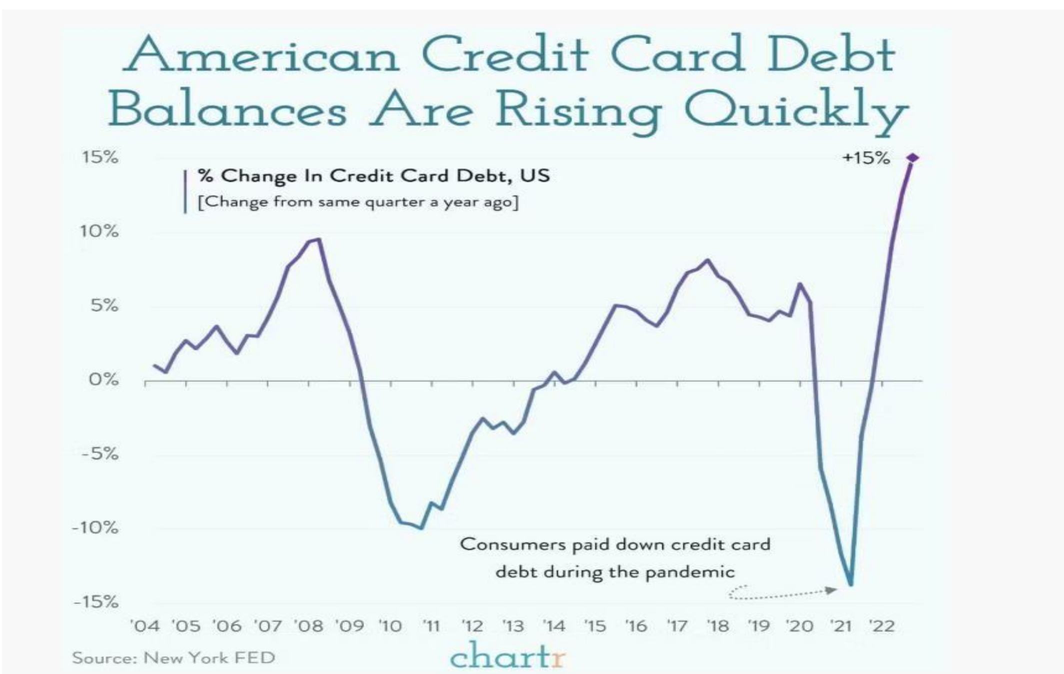


Figure 2 Trend of American Credit card debt (Crowther, 2022)

1.4 Objective

- To carry out research on the problem of credit card default along with the implementation of ml model to solve the problem.
- To carry out Data preprocessing and Exploratory data analysis on the data set.
- To Deal with imbalanced data set.
- To achieve the better performance in machine learning model above 70 percent.
- To compare the model using its respective evaluation metrics.
- To determine the best performing model.

2 Background

2.1 Research on the chosen problem domain

According to Steve Bucci, in this uncertain economic period, the customers of a credit card provided by the banking institution who have been negatively impacted by job scarcity, due to mortgages, student debt and other situations must focus on their expense activity as, due in credit card payment for several months can reduce their credit score which can cause difficulty for approval of new credit in the future (Bucci, 2023). Credit card default usually happens when the monthly payment is due for six months which might lead to the closure of the bank account by the legal authorities (Bucci, 2023).

As per Tania Jalee ET Online, the credit card default amount was approximately 2.7 lakh crore in June 2024, leading from 2.6 lakh crore in March 2024 and the annual growth rate over the five years was 24 per cent (Jaleel, 2024). Most youngster nowadays use their entire credit card limit amount for making larger purchases such as cars, luxury goods, and property without prior analysis of their income which leads to their credit being defaulted which might affect chance of approval of the future loan or new credit by the institution that lends money to them (Jaleel, 2024).

Figure 3 Credit card default (NewsDrum, 2023)

2.2 Review and analysis of existing work in the problem domain

2.2.1 Research on Credit Card Default Prediction Based on k-Means SMOTE and BP Neural Network

Author: Ying Chen and Ruirui Zhang

This research paper purpose a prediction of a credit card default based on the k-means SMOTE (Synthetic Minority Over Sampling Technique) and BP neural network where k-mean SMOTE was used to change the data distribution, then the importance feature of the data was calculated using random forest where the initial weight of Back propagation neural networks substitute for the accurate prediction on the dataset of the default credit card client on Kaggle. (Ying Chen, 2021). Other five common machine learning approach logic regression, SVM, Random Forest, KNN, and Tree was trained, and their accuracy was compared with each other, it is found out that the problem of data imbalance was solved by k-Mean SMOTE which improve the prediction and performance of the model (Ying Chen, 2021). SVM have the highest accuracy i.e. 0.881, highest recall score of 0.885 and backpropagation neural network have the highest F1 score of 0.880 and precision score of 0.923.

2.2.2 Credit card default prediction using machine learning models

Author: Hritwiz, Yash, Affan,Kumar Saurav and Kumar Saurav

In this research project, logistic regression, SVM and ANN was been used to develop a predictive machine learning model for accurately predicting the credit card default which can help the bank to mitigate risk, optimize their lending process to customer, and decrease their loss using the data set of credit card client from the Taiwan (Hritwiz Yash, 2023). These three models were trained, and their performance was evaluated using performance metric, where most of the algorithms model accuracy was in the range of 80-82.03 percent. Support vector machine classifier outperform all the other model with the accuracy of 82.03 percent. SVM ability to capture the complex pattern in large dataset, handle outlier and noise have supported to greater performance in the prediction (Hritwiz Yash, 2023).

2.3 Datasets

The dataset was chosen from UCI machine learning repository of Default of credit card clients and its description of is given below:

S. N	Name of the columns	Description	Data type
1.	ID	It is a unique identifier for each row of the dataset.	Int64
2.	SEX	These columns indicate the gender with two categories where the “1” value denotes male and “2” denotes female. The Female has the highest number of row count i.e. 18112 and the male is 11888 which tells us that the female ratio of using the credit card is more than male.	Int64
3.	EDUCATION	These columns indicate the education status where values range from 0 – 6. The “1” value denotes graduate school, “2” denotes university, “3” denotes high school and “4” denotes other. “5”,”0” and “6” values information is unknown and will be replace in “other” or “4” category. The education status with “2” or “university” has the highest number of row entries which indicates that most of the student from the university uses the credit card for their daily expense and they might be in the credit default.	Int64
4.	MARRIAGE	These columns indicate the marital status of the credit card holder where the values range from 0 to 3. “1” value denotes “married”, “2” value denotes “single” and “3” value denotes “others”. The “0” value information is unknown and will be determined in the “others” or “3” category in the MARRIAGE column. The “2” value has the highest number of row counts (15964) which tells us that married people mostly use credit cards.	Int64
5.	AGE	These columns indicate the Age of the credit card holder. The age	Int64

6.	PAY_0 TO PAY_6	These six columns indicate the payment that is already made by the credit amount taker in the month from April to September in 2005. The value ranges for -1 to 8 where "-1" indicates payment duly, "1" indicates payment delay for a month and "2" indicates payment delay for 2 month and so on.	Int64
7	BILL_AMT1 to BILLAMT6	These six columns indicate the bill amount that need to settle by the customer in the month from April to September in 2005.	Int 64
8.	PAY_AMT1 to PAY_AMT6	These six columns indication the amount that is payed by the customer in the month from April to September in 2005	Int64
9.	Default payment next month	It is the target variable which indicate of the customer credit card is being defaulted or not "1" denotes defaulted and "0" denotes not defaulted	Int64

3. Solution

3.1 AI Algorithm

Support vector machine

Support vector machine is a supervised machine learning algorithm that is used for the both the linear and nonlinear classification including regression task (geeksforgeeks, 2024). It can handle data with outlier and noise. A support vector machine for classification forms a hyperplane with maximum margin which lies in a transformed input space and separated the input data classes by maximizing the distance between two categories or classes which is extracted from a quadratic optimization problem (Shmilovici, 2024). The hyperplane with the maximum margin is select based on the greater distance between the hyperplane and the closest data point on each side of the hyperplane.

The equation of the hyperplane (linear):

$$w^{\top}x + b = 0$$

Where w denotes the vector which direction is perpendicular to the hyperplane, the b is the parameter which denotes distance of the hyperplane from the origin with respect to the normal vector w. The equation for the distance between the datapoint x and hyperplane

$$d_i = \frac{w^{\top}x_i + b}{||w||}$$

Where ||w|| is the Euclidean norm of the vector w

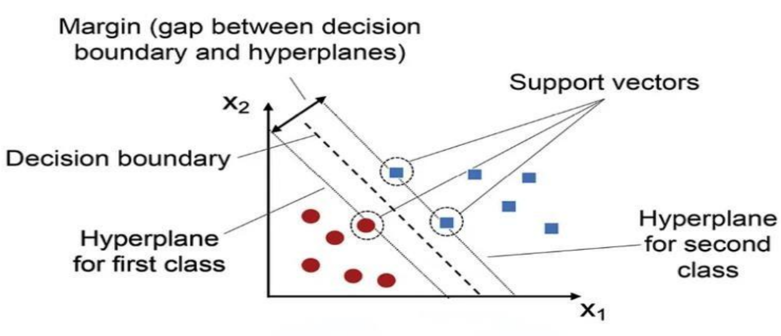


Figure 4 Support Vector Machiner (Jainvidip, 2024)

Random forest Classifier

Random forest is a supervised learning algorithm which is used for both classification and regression task which is based on the concept of merging the prediction of several decision tree to generate a final output with higher prediction accuracy and less error (School, 2024). Each tree in random forest a bootstrapping process is used to randomly create a sub sample of the input data with replacement which ensure that separate decision trees are not same, and prediction of each tree are different for increase in model strength. It can handle complex pattern in data, deal with outlier and noisy data, and reduce overfitting. After all the separated trees have been developed then it aggregates overall result of the tree into int final output prediction, in case of classification task major voting system is used to predict the correct task (School, 2024).

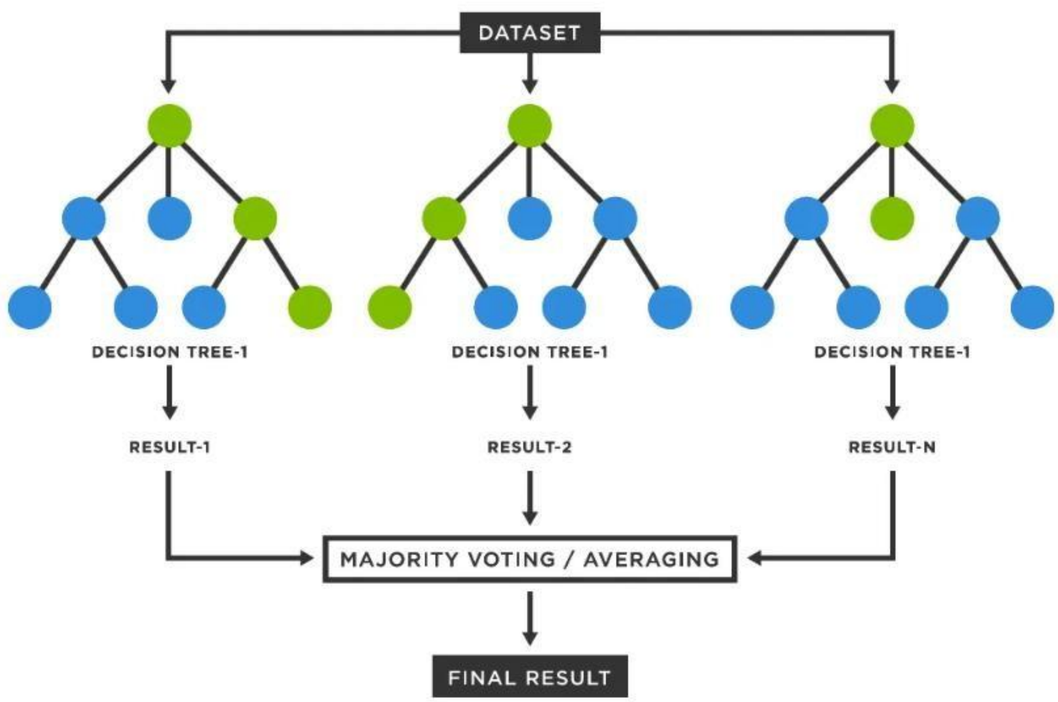


Figure 5 Random Forest diagram (Gunay, 2023)

Logistic Regression

Logistic Regression is a supervised learning method that estimates binary outcomes and hyperplanes employing logistic functions and a probabilistic approach. In logistic regression, the outcome often gets classified in binary, indicating either 0 or 1 (geeksforgeeks, 2024). It is simple to implement, evaluate and train contrasted to other Machine Learning models. It uses a sigmoid function that transform the linear regression function f(x) continuous values into a categorical or binary output (geeksforgeeks, 2024).

$$\mathbf{f}(\boldsymbol{x}) = \frac{\mathbf{1}}{\mathbf{1} + \boldsymbol{e}^{-\boldsymbol{x}}}$$

e = natural logarithms base x = input

variable value to transform

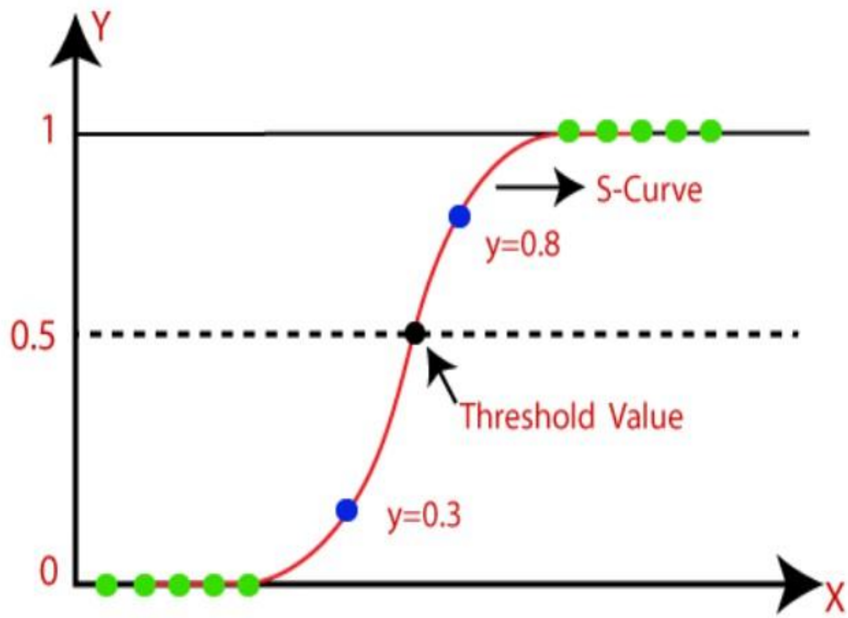


Figure 6 Logistic Regression diagram

Precision

It is the evaluation metric for classification problem which measure number of true positive predictions among all the correct positive predictions. It tells us if the prediction made by the model is correct, then how many of it are correct? (Kulkarni, 2023)

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Figure 7 Precision formula (shung , 2018)

Accuracy

It is the evaluation metric for classification problem which measure the total number of correct predictions made by the model. (Kulkarni, 2023).

$$Accuracy = \frac{TN + TP}{TN + FP + TP + FN}$$

Figure 8 Accuracy formula (NB, 2019)

Recall

It is the evaluation metric which measures the instance of true positive prediction among all the actual positive prediction, it tells us how many instances the model predicted correctly. (Kulkarni, 2023)

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Figure 9 Recall formula (shung , 2018)

Confusion matrix

It is an evaluation metric which shows the table of summarization of the model performance which include true positive, true negative, false negative and false positive number of the model (Kulkarni, 2023).

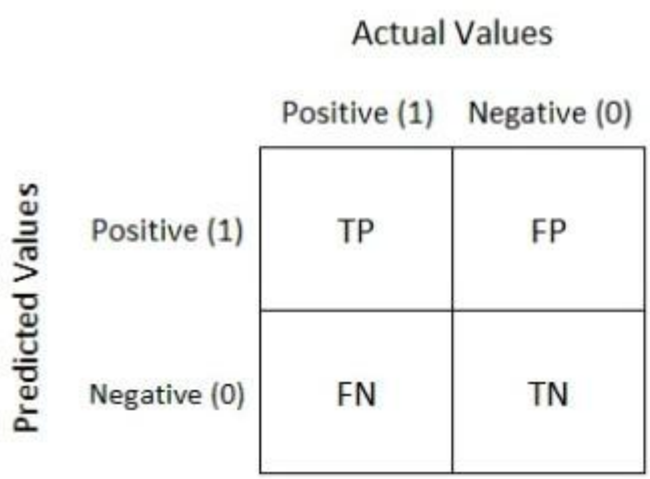


Figure 10 Confusion matrix (Narkhede, 2018)

3.2 Pseudocode

START

IMPORT necessary libraries

LOAD dataset

DO Data preprocessing

PERFORM removal of missing values

PERFORM duplicates value removal

PERFORM Encoding for categorical data

PERFORM DATA Normalization and Scaling

END DO

DO Exploratory data analysis

CREATE univariate analysis include visualization

CREATE bivariate analysis including visualization

CREATE correlation analysis including heatmap

PERFORM Outlier detection including boxplot

END DO

IF dataset is imbalance

CONDUCT under sampling or over sampling

END IF

DECLARE feature variable

DECLARE target variable

CONDUCT train test split

INITIALIZE random forest classifier model

INITIALIZE support vector classifier model

INITIALIZE Logistic regression model

TRAIN random forest classifier model, support vector classifier model and logistic regression model

GENERATE classification report for random forest classifier model, support vector classifier model and logistic regression model

EVALULATE metric

PEFROM hyperparameter tuning

PERFROM Stacking

COMPARE performance of the model

END

3.3 Flowchart

3.3.1 Overall flowchart

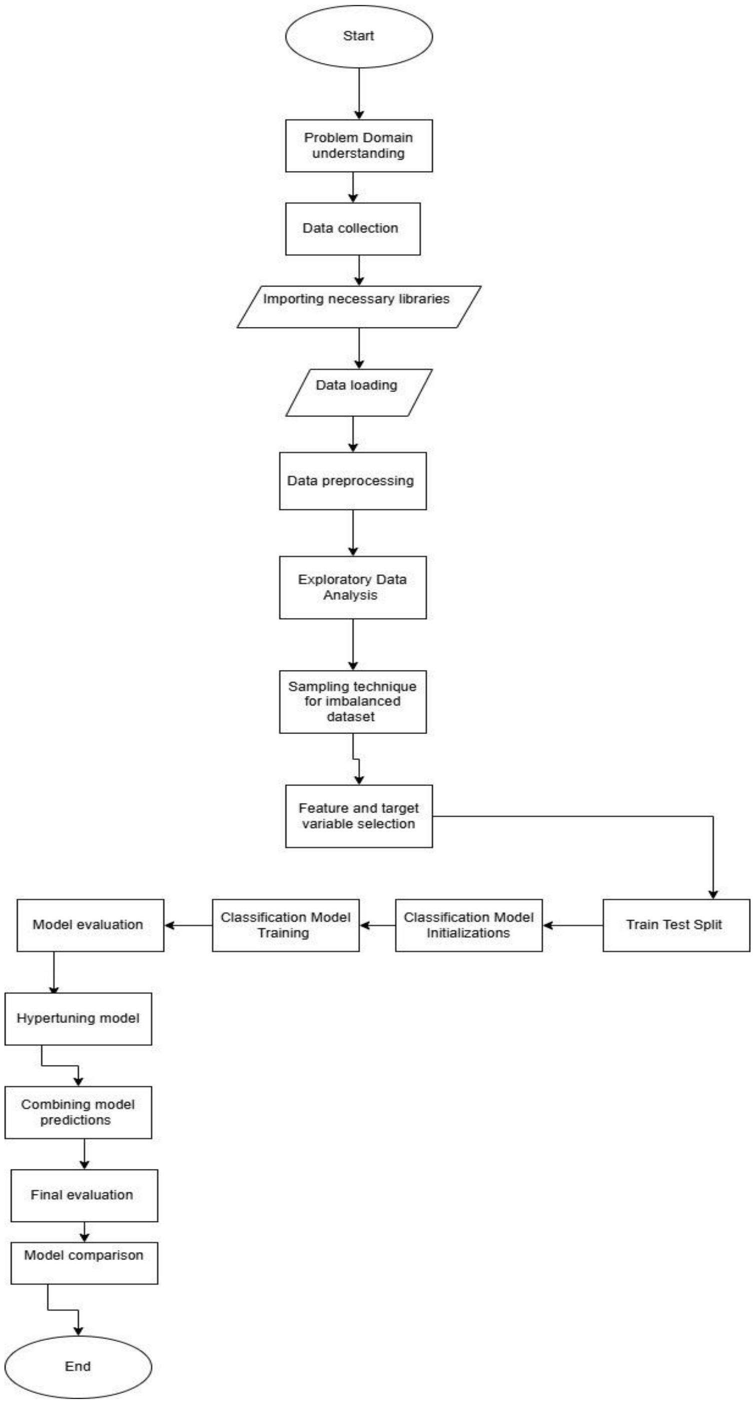


Figure 11 Flowchart of system

3.3.2 Logistic Regression

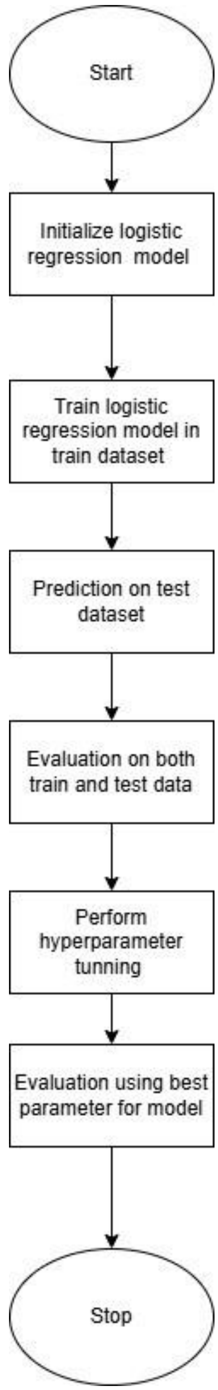


Figure 12 Flow chart for logistic regression model

3.3.3Random forest classifier

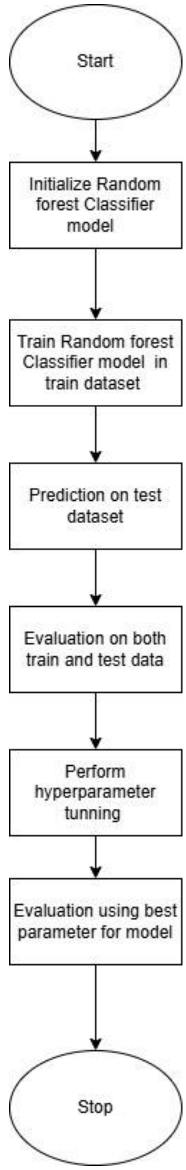


Figure 13 Flowchart for random forest classifier model

3.3.4 Support vector classifier

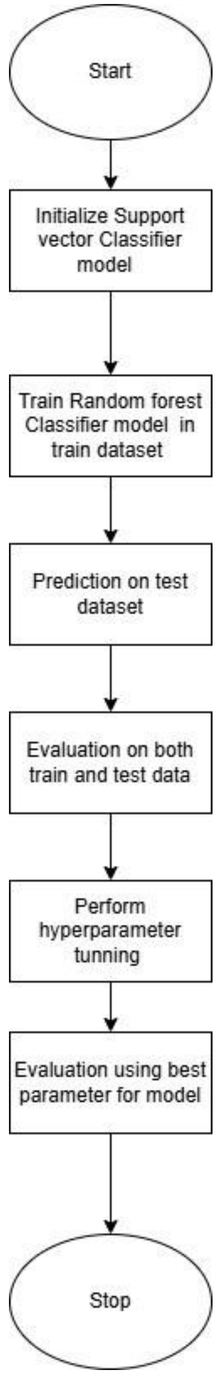


Figure 14 Flow chart for support vector classifier

3.4 Development

3.4.1 Importing necessary libraries

The required libraries required to perform the task is imported at first.

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report,confusion_matrix, roc_auc_score
from sklearn.svm import SVC
from sklearn.model_selection import cross_val_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import StandardScaler
from imblearn.over_sampling import SMOTE
from imblearn.under_sampling import RandomUnderSampler
```

Figure 15 Screenshot of importing libraries

3.4.2 Data loading

The excel dataset file was loaded to pandas data frame using read_excel() method where the path of the file and header is set to 1 which indicates that second row the the excel file is the main header for the dataset.

```python
# Loading the excel file using read_excel() of pandas
df_default_credit = pd.read_excel("defaultofcreditcard.xls",header=1)
```

Figure 16 Screenshot of Loading the excel file

3.4.3 Data understanding

Displaying the first 5 row of the data frame

In pandas, head() method was is to display the first 5 row of the data frame where we can specify the number of rows to print. No parameter was used, which results in display first 5 rows by default.

```
[6]: pd.set_option('display.max_columns', None) # display all the columns
df_default_credit.head() # display first five row of the dataframe
```

	ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	PAY_6	BILL_AMT1	BILL_AMT2	BILL_AMT3	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	-2	-2	3913	3102	689	0	0	0	0	689	0	0	0	0	1
1	2	120000	2	2	2	26	-1	2	0	0	0	2	2682	1725	2682	3272	3455	3261	0	1000	1000	1000	0	2000	1
2	3	90000	2	2	2	34	0	0	0	0	0	0	29239	14027	13559	14331	14948	15549	1518	1500	1000	1000	1000	5000	0
3	4	50000	2	2	1	37	0	0	0	0	0	0	46990	48233	49291	28314	28959	29547	2000	2019	1200	1100	1069	1000	0
4	5	50000	1	2	1	57	-1	0	-1	0	0	0	8617	5670	35835	20940	19146	19131	2000	36681	10000	9000	689	679	0

Figure 17 Screenshot of Display first 5 rows of data frame

Displaying the last 5 row of the data frame

In pandas, tail() method was is to display the last 5 row of the data frame where we can specify the number of rows to print. No parameter was used, which results in display last 5 row by default.

```
[8]: df_default_credit.tail() # display last five rows of the dataset
```

	ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	PAY_6	BILL_AMT1	BILL_AMT2	BILL_AMT3	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	0	0	188948	192815	208365	88004	31237	15980	8500	20000	5003	3047	5000	1000	0
29996	29997	150000	1	3	2	43	-1	-1	-1	-1	0	0	1683	1828	3502	8979	5190	0	1837	3526	8998	129	0	0	0
29997	29998	30000	1	2	2	37	4	3	2	-1	0	0	3565	3356	2758	20878	20582	19357	0	0	22000	4200	2000	3100	1
29998	29999	80000	1	3	1	41	1	-1	0	0	0	-1	-1645	78379	76304	52774	11855	48944	85900	3409	1178	1926	52964	1804	1
29999	30000	50000	1	2	1	46	0	0	0	0	0	0	47929	48905	49764	36535	32428	15313	2078	1800	1430	1000	1000	1000	1

Figure 18 Screenshot of displaying last 5 rows of data frame

Random 10 rows of the dataset

Random 10 rows of the data frame is displayed using .sample(10) method of pandas, we can specify required number of rows to randomly display by passing the number in parameter.

```
[10]: df_default_credit.sample(10) #Display random 10 rows for the dataframe
```

[10]:

	ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	PAY_6	BILL_AMT1	BILL_AMT2	BILL_AMT3	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2	PAY_AMT3	PAY_AMT4	PAY_AMT5	PAY_AMT6	default payment next month
5785	5786	60000	1	2	2	26	2	2	2	2	0	0	61103	61380	62158	120142	60428	58192	3000	2746	0	2297	1875	1158	1
1573	1574	50000	1	1	2	24	0	0	0	0	0	0	50716	50606	33265	29300	20703	30101	3029	2000	1021	1004	10000	2500	1
20183	20184	210000	2	1	2	28	-1	-1	-1	-1	-1	0	2670	262	3771	5474	69536	76188	262	3771	5474	69536	10000	4000	0
15657	15658	220000	2	1	1	37	-2	-2	-2	-2	-2	-2	14298	100606	11202	21683	10860	12534	101405	11202	21778	10860	12534	7923	0
3064	3065	430000	2	2	1	42	0	0	0	0	0	0	89395	90052	90604	91200	92134	92834	3243	3200	3185	3500	3500	3420	0
3451	3452	50000	2	1	2	26	0	0	2	0	0	0	39052	44416	43407	44382	45672	46758	6000	0	2000	2000	2000	2000	0
19687	19688	210000	2	3	1	40	0	-1	-1	-1	-1	-1	2616	7537	6730	3447	16220	5566	7537	6730	3447	16220	7564	2632	0
14544	14545	80000	1	2	1	46	0	0	0	0	0	0	81283	80390	79080	77850	80330	80000	2892	3000	5390	4106	3000	0	1
21568	21569	80000	2	1	2	25	2	0	0	0	0	0	69614	68836	69714	57318	46921	41072	3238	3103	3000	1800	1500	1700	1
27167	27168	10000	1	2	1	29	0	0	0	0	0	0	8275	8409	8600	9470	6690	9690	2800	2000	1500	900	3000	0	1

Figure 19 Screenshot of 10 random rows of the data frame

Columns of the data frame

The columns of the data frame is displayed by calling . columns method of pandas.

```
[12]: df_default_credit.columns # Columns of the dataframe

[12]: Index(['ID', 'LIMIT_BAL', 'SEX', 'EDUCATION', 'MARRIAGE', 'AGE', 'PAY_0',
             'PAY_2', 'PAY_3', 'PAY_4', 'PAY_5', 'PAY_6', 'BILL_AMT1', 'BILL_AMT2',
             'BILL_AMT3', 'BILL_AMT4', 'BILL_AMT5', 'BILL_AMT6', 'PAY_AMT1',
             'PAY_AMT2', 'PAY_AMT3', 'PAY_AMT4', 'PAY_AMT5', 'PAY_AMT6',
             'default payment next month'],
            dtype='object')
```

Figure 20 Screenshot of columns of data frame

Dimension of the data frame

The dimension represents the number of rows and columns in data frame which is displayed using. shape method of pandas. There are total of 30000 rows and 25 columns in data frame

```
5]: df_default_credit.shape # Number of rows and columns in dataframe

5]: (30000, 25)
```

Total rows : 30000 and Total Columns : 25

Figure 21 Screenshot of number of rows and columns of data frame

Data type of each column

In pandas, to display the data type of each column of data frame .dtypes method is called.

```
[18]: df_default_credit.dtypes # Data type of each columns in the dataframe

[18]: ID                            int64
      LIMIT_BAL                     int64
      SEX                           int64
      EDUCATION                     int64
      MARRIAGE                      int64
      AGE                           int64
      PAY_0                         int64
      PAY_2                         int64
      PAY_3                         int64
      PAY_4                         int64
      PAY_5                         int64
      PAY_6                         int64
      BILL_AMT1                     int64
      BILL_AMT2                     int64
      BILL_AMT3                     int64
      BILL_AMT4                     int64
      BILL_AMT5                     int64
      BILL_AMT6                     int64
      PAY_AMT1                      int64
      PAY_AMT2                      int64
      PAY_AMT3                      int64
      PAY_AMT4                      int64
      PAY_AMT5                      int64
      PAY_AMT6                      int64
      default payment next month    int64
      dtype: object
```

Figure 22 Screenshot of data types of each column of data frame

Information about each column

.info() method is used to display the information like index range, columns name and their data types and null value counts, memory usage.

```
•[20]: df_default_credit.info() # Information regarding each columns in the dataframe

<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
```

Figure 23 Screenshot of information about the columns of the data frame

Summary statistics of each column describe() method of pandas is used to display the summary statistics of each column like maximum values ,minimum values, quartiles, standard deviation , count rows and mean values.

```
•[22]: df_default_credit.describe() # perform and display statistical summary of each columns which includes mean, standard deviation,max and min values, quartiles, count
```

[22]:

	ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3
count	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000
mean	15000.500000	167484.322667	1.603733	1.853133	1.551867	35.485500	-0.016700	-0.133767	-0.166200
std	8660.398374	129747.661567	0.489129	0.790349	0.521970	9.217904	1.123802	1.197186	1.196868
min	1.000000	10000.000000	1.000000	0.000000	0.000000	21.000000	-2.000000	-2.000000	-2.000000
25%	7500.750000	50000.000000	1.000000	1.000000	1.000000	28.000000	-1.000000	-1.000000	-1.000000
50%	15000.500000	140000.000000	2.000000	2.000000	2.000000	34.000000	0.000000	0.000000	0.000000
75%	22500.250000	240000.000000	2.000000	2.000000	2.000000	41.000000	0.000000	0.000000	0.000000
max	30000.000000	1000000.000000	2.000000	6.000000	3.000000	79.000000	8.000000	8.000000	8.000000

	PAY_4	PAY_5	PAY_6	BILL_AMT1	BILL_AMT2	BILL_AMT3	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1
count	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000	3.000000e+04	30000.000000	30000.000000	30000.000000	30000.000000
mean	-0.220667	-0.266200	-0.291100	51223.330900	49179.075167	4.701315e+04	43262.948967	40311.400967	38871.760400	5663.580500
std	1.169139	1.133187	1.149988	73635.860576	71173.768783	6.934939e+04	64332.856134	60797.155770	59554.107537	16563.280354
min	-2.000000	-2.000000	-2.000000	-165580.000000	-69777.000000	-1.572640e+05	-170000.000000	-81334.000000	-339603.000000	0.000000
25%	-1.000000	-1.000000	-1.000000	3558.750000	2984.750000	2.666250e+03	2326.750000	1763.000000	1256.000000	1000.000000
50%	0.000000	0.000000	0.000000	22381.500000	21200.000000	2.008850e+04	19052.000000	18104.500000	17071.000000	2100.000000
75%	0.000000	0.000000	0.000000	67091.000000	64006.250000	6.016475e+04	54506.000000	50190.500000	49198.250000	5006.000000
max	8.000000	8.000000	8.000000	964511.000000	983931.000000	1.664089e+06	891586.000000	927171.000000	961664.000000	873552.000000

Figure 24 Summary statistic of each columns

Inspection of missing values isna() is used to return the Boolean values true in case of missing values and false incase of no missing values for each columns and sum() method is used to count the number of missing values in each column of the data frame. There are no missing values in the data frame

```
4]: df_default_credit.isna().sum() # Inspection of missing values

4]: ID                            0
    LIMIT_BAL                     0
    SEX                           0
    EDUCATION                     0
    MARRIAGE                      0
    AGE                           0
    PAY_0                         0
    PAY_2                         0
    PAY_3                         0
    PAY_4                         0
    PAY_5                         0
    PAY_6                         0
    BILL_AMT1                     0
    BILL_AMT2                     0
    BILL_AMT3                     0
    BILL_AMT4                     0
    BILL_AMT5                     0
    BILL_AMT6                     0
    PAY_AMT1                      0
    PAY_AMT2                      0
    PAY_AMT3                      0
    PAY_AMT4                      0
    PAY_AMT5                      0
    PAY_AMT6                      0
    default payment next month    0
    dtype: int64
```

Figure 25 Screen shot of missing values count of each column

Duplicates values

The duplicates rows of the data frame is displayed using the method .duplicated() and overall duplicates values count is calculated using the sum() method. There are no duplicates values in the data frame.

```
•[26]: df_default_credit.duplicated().sum() # counting the number of duplicates values

[26]: np.int64(0)
```

Figure 26 Screenshot of count of duplicate values

Unique values of EDUCATION columns

The value_counts() method is used to display the unique values of the columns along with its frequency count. There are total of 7 unique values in the education columns with the highest begin 2 which represent university.

```
•[30]: df_default_credit['EDUCATION'].value_counts() # unique values and their frequency count of the EDUCATION columns

[30]: EDUCATION
      2    14030
      1    10585
      3     4917
      5      280
      4      123
      6       51
      0       14
      Name: count, dtype: int64
```

Education columns value indication = 1 : graduate school , 2: university. 3: high school, 4 : others

5,6 and 0 are unknown

Figure 27 Screenshot of value counts of EDUCATION columns

Unique value of MARRIAGE columns

There are total of 4 unique values in MARRIAGE columns where ‘2’ single has highest number of counts and ‘0’ unknown has the lowest number of counts.

```
[37]: df_default_credit['MARRIAGE'].value_counts() # unique values and their frequency count  of the MARRIAGE columns

[37]: MARRIAGE
      2    15964
      1    13659
      3      323
      0       54
      Name: count, dtype: int64
```

MARRIAGE columns values indication: 1 = married, 2 = single and 3 = others. 0 values is unknown

Figure 28 Screenshot of value counts of MARRIAGE columns

Unique values of Gender columns

The ratio of female gender is more than male in the data frame where the row count of female is 18112 and male is 11888.

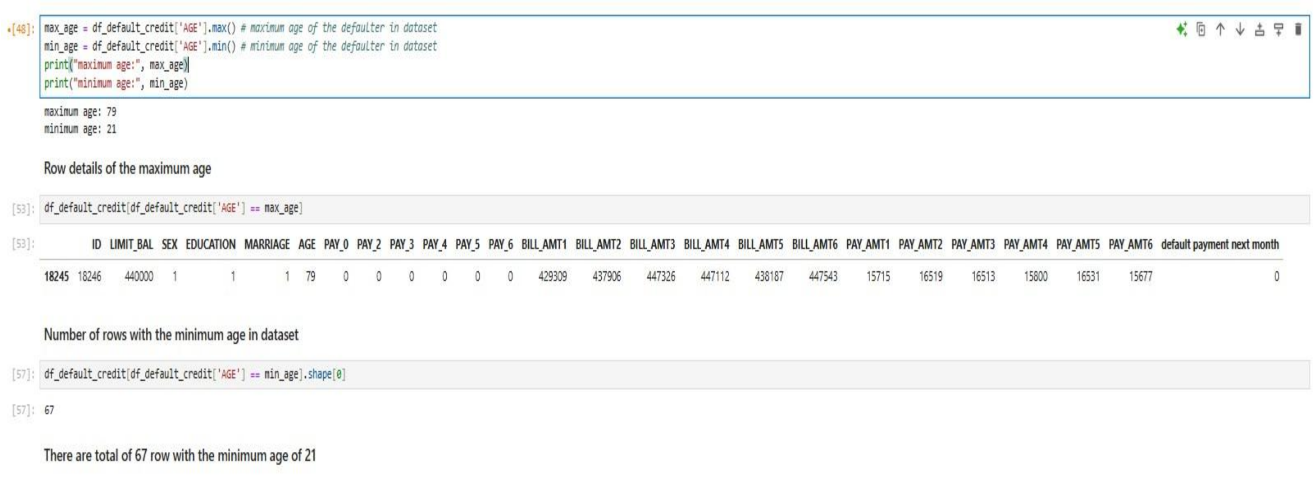
```
[42]: df_default_credit['SEX'].value_counts() # unique values and their frequency count  of the GENDER columns

[42]: SEX
      2    18112
      1    11888
      Name: count, dtype: int64
```

Gender columns indication 1: male and 2 : Female

Figure 29 Screen shot of value counts of gender columns

Age columns Maximum and minimum values

The maximum age is 79 with only 1 row count and minimum age is 21 with 67 rows count.



```
[48]: max_age = df_default_credit['AGE'].max() # maximum age of the defaulter in dataset
min_age = df_default_credit['AGE'].min() # minimum age of the defaulter in dataset
print("maximum age:", max_age)
print("minimum age:", min_age)

maximum age: 79
minimum age: 21
```

Row details of the maximum age

```
[53]: df_default_credit[df_default_credit['AGE'] == max_age]
```

[53]:

	ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	PAY_6	BILL_AMT1	BILL_AMT2	BILL_AMT3	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2	PAY_AMT3	PAY_AMT4	PAY_AMT5	PAY_AMT6	default payment next month
18245	18246	440000	1	1	1	79	0	0	0	0	0	0	429309	437906	447326	447112	438187	447543	15715	16519	16513	15800	16531	15677	0

Number of rows with the minimum age in dataset

```
[57]: df_default_credit[df_default_credit['AGE'] == min_age].shape[0]

[57]: 67
```

There are total of 67 row with the minimum age of 21

Figure 30 Screenshot of Maximum and minimum age with their rows count

Unique value of target columns

There are total of 2 unique values ‘0’ no default payment and ‘1’ default payment in target columns of the data frame. The data is imbalanced and the number of no default payments ‘0’ class row count is higher 223364 than default payments ‘1’ class 6636.

```
[65]: df_default_credit['default payment next month'].value_counts() # Unique values and their frequency count of target variable

[65]: default payment next month
0    23364
1     6636
Name: count, dtype: int64
```

values indication: 0 = no default payment and 1 = default payment

Figure 31 Screenshot of value counts of target column

Unique values of PAY_0 to PAY_6 columns

There are total of 11 unique values in each PAY columns. To perform the aggregate values count for each PAY columns. apply() function was used and the unique values was replace with data description value using map() function for easier interpretation.

[61]:
```python
# apply the aggreation function to count the unique values of the each columns PAY
PAY0_to_PAY6 = df_default_credit[['PAY_0', 'PAY_2', 'PAY_3', 'PAY_4', 'PAY_5', 'PAY_6']].apply(pd.Series.value_counts )
# map the index values to the dataset desciption
PAY0_to_PAY6.index = PAY0_to_PAY6.index.map({
    0:0,
    -2:-2,
    -1: 'Paid duly',
    1: 'Payment delayed for one month',
    2: 'Payment delayed for two months',
    3: 'Payment delayed for three months',
    4: 'Payment delayed for four months',
    5: 'Payment delayed for five months',
    6: 'Payment delayed for six months',
    7: 'Payment delayed for seven months',
    8: 'Payment delayed for eight months',
    9: 'Payment delayed for nine months and above'
})

PAY0_to_PAY6
```

[61]:

	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	PAY_6
-2	2759	3782	4085	4348	4546.0	4895.0
Paid duly	5686	6050	5938	5687	5539.0	5740.0
0	14737	15730	15764	16455	16947.0	16286.0
Payment delayed for one month	3688	28	4	2	NaN	NaN
Payment delayed for two months	2667	3927	3819	3159	2626.0	2766.0
Payment delayed for three months	322	326	240	180	178.0	184.0
Payment delayed for four months	76	99	76	69	84.0	49.0
Payment delayed for five months	26	25	21	35	17.0	13.0
Payment delayed for six months	11	12	23	5	4.0	19.0
Payment delayed for seven months	9	20	27	58	58.0	46.0
Payment delayed for eight months	19	1	3	2	1.0	2.0

The indication of -2 and 0 is not mentioned and in data description but, from further research it indicates that -2 is no consumption of credit and 0 is minimum amount paid

Figure 32 Screen shot of values counts of PAY_0 to PAY_6 columns

3.4.4 Exploratory data analysis

Visualization Gender distribution

```
[336]:
plt.figure(figsize=(10,6))
ax = sns.countplot(data=df_default_credit, x='SEX',palette='deep',width=0.6,hue='SEX')
ax.bar_label(ax.containers[0], fontsize=12, padding=5)  # adding bar label
ax.bar_label(ax.containers[1], fontsize=12, padding=5)  # adding bar label
plt.title("The Distribution of Gender in the dataset")
plt.xticks(ticks=[0,1],labels=["male",'Female'])
plt.ylim(0,20000)
plt.show()
```

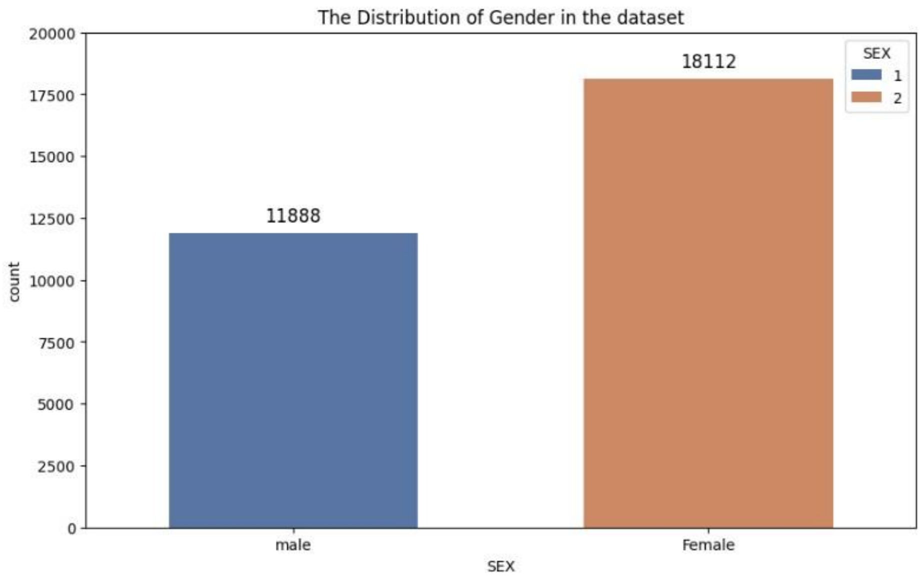


Figure 33 Visualization of gender distribution

Here is the visualization for the gender distribution of the dataset, where the number of females is highest with 18112 row count and male is 11888 row count which indicates that most of the woman tends to use credit card than man.

Visualization of Distribution of marriage status

```
plt.figure(figsize=(10,4))
ax = sns.countplot(data=df_default_credit, x='MARRIAGE',palette='deep',width=0.5,hue="MARRIAGE")
ax.bar_label(ax.containers[0], fontsize=12, padding=5)
ax.bar_label(ax.containers[1], fontsize=12, padding=5)
ax.bar_label(ax.containers[2], fontsize=12, padding=5)
ax.bar_label(ax.containers[3], fontsize=12, padding=5)
plt.title("The Distribution of marriage status in the dataset")
plt.xticks(ticks=[0,1,2,3],labels=["unknown",'Married',"Single","others"])
plt.ylim(0,20000)
plt.show()
```

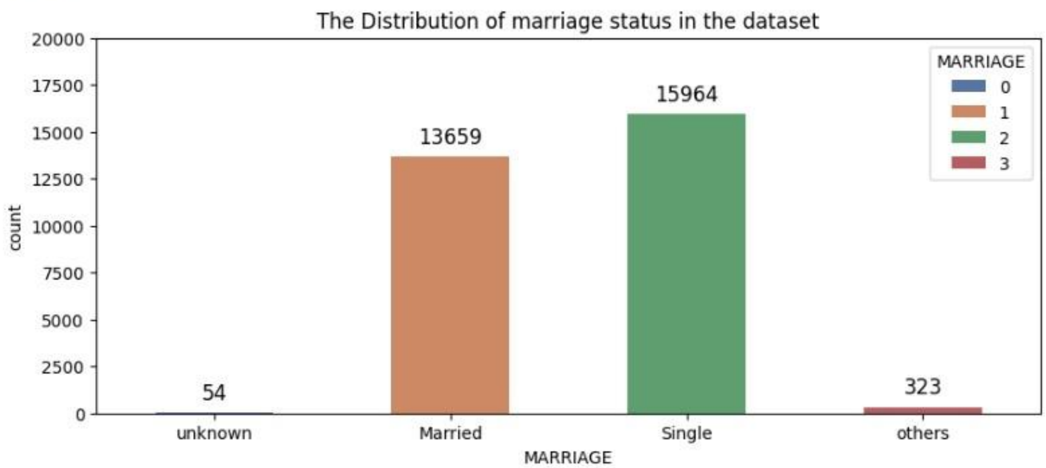


Figure 34 Visualization of marriage status distribution

Visualization of target columns class distribution

```
plt.figure(figsize=(10,11))
ax = sns.countplot(data=df_default_credit, x='default payment next month',palette='deep',width=0.5,hue='default payment next month')
ax.bar_label(ax.containers[0], fontsize=12, padding=5)
ax.bar_label(ax.containers[1], fontsize=12, padding=5)
plt.title("The Distribution of values of target variable default vs no default")
plt.xticks(ticks=[0,1],labels=["No default",'default'])
plt.ylim(0,32000)
plt.show()
```

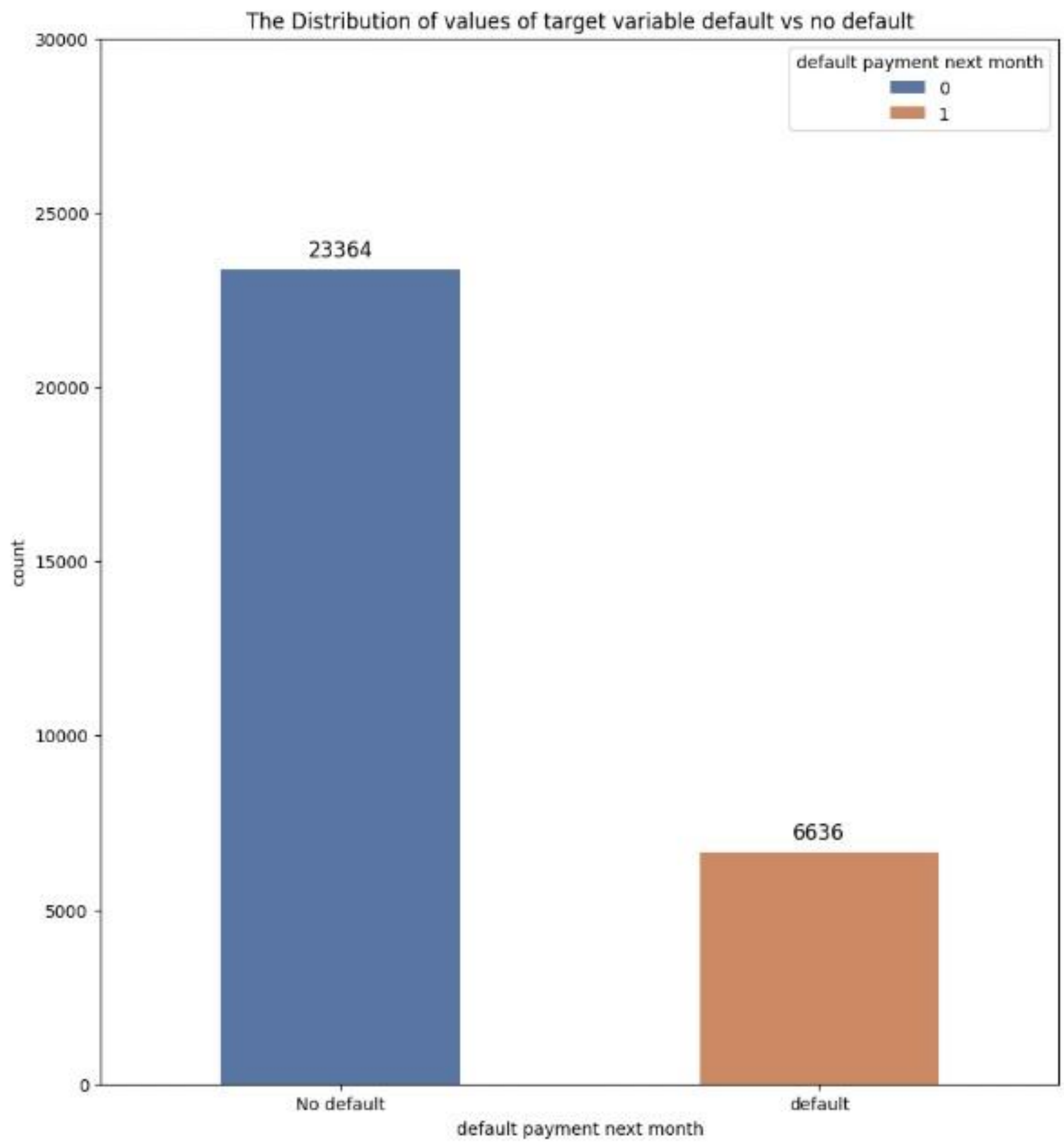


Figure 35 Visualization of distribution of target columns

Visualization of education status distribution

```
plt.figure(figsize=(10,11))
ax = sns.countplot(data=df_default_credit, x='EDUCATION',palette='deep',width=0.5,hue='EDUCATION')
ax.bar_label(ax.containers[0], fontsize=12, padding=5)
ax.bar_label(ax.containers[1], fontsize=12, padding=5)
ax.bar_label(ax.containers[2], fontsize=12, padding=5)
ax.bar_label(ax.containers[3], fontsize=12, padding=5)
ax.bar_label(ax.containers[4], fontsize=12, padding=5)
ax.bar_label(ax.containers[5], fontsize=12, padding=5)
ax.bar_label(ax.containers[6], fontsize=12, padding=5)
plt.title("The Distribution of EDUCATION columns ")
plt.xticks(ticks=[0,1,2,3,4,5,6],labels=["unknown",'graduate school','university','high_school','others','unknown','unknown'])
plt.ylim(0,15000)
plt.show()
```

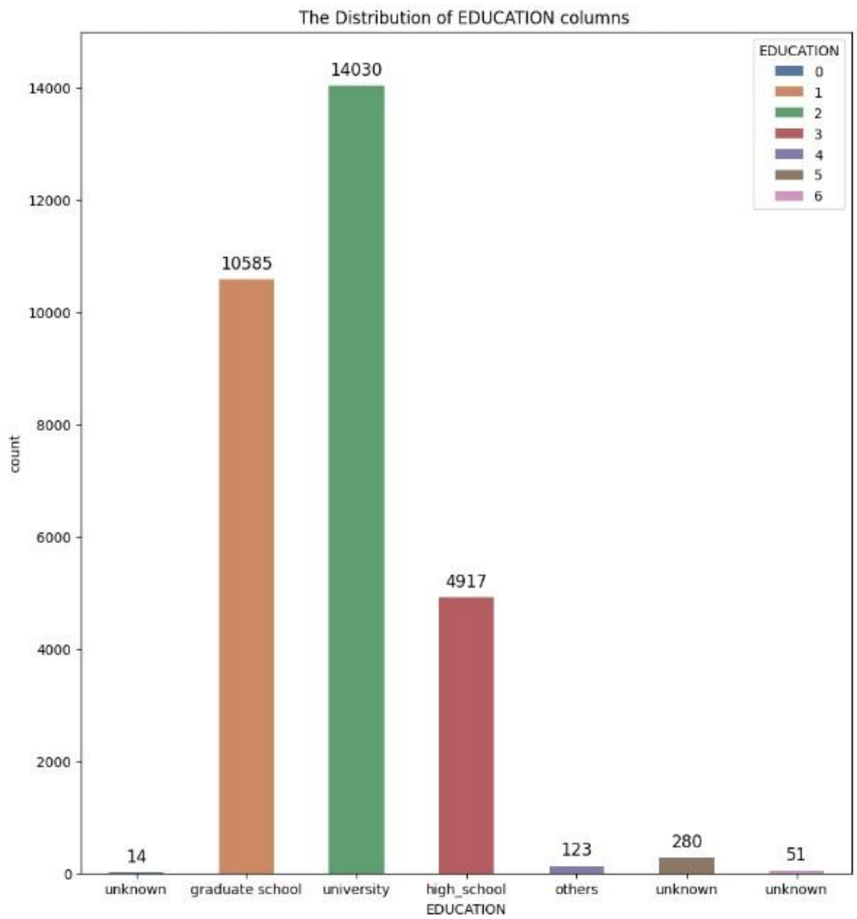


Figure 36 Visualization of education column distribution

Histogram of Age column

Here the distribution of the age column in the dataset is show using histogram. Customer with Age between 26 to 45 tends to use credit card often.

```
•[81]: # histogram seaborn
sns.histplot(df_default_credit['AGE'], kde=True, bins=30, color='blue')
plt.title("Histogram of the age column")

[81]: Text(0.5, 1.0, 'Histogram of the age column')
```

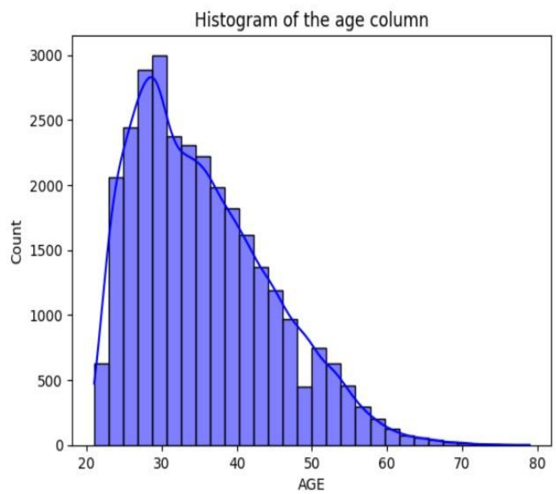


Figure 37 Histogram of AGE columns

Overall histogram of BILL_AMT columns

```
•[84]:
BILL_AMT = ['BILL_AMT1', 'BILL_AMT2', 'BILL_AMT3', 'BILL_AMT4','BILL_AMT5', 'BILL_AMT6']
# Set up the figure and subplots
fig, ax = plt.subplots(2, 3, figsize=(12, 10))  # 2 rows, 2 columns
for i, column in enumerate(BILL_AMT, 1):
    plt.subplot(2,3, i)
    sns.histplot(df_default_credit[column], kde=True, bins=40, color='green')
    plt.title(f"Histogram of the {column} column")
    plt.xlabel(column)
    plt.ylabel("Frequency")

plt.tight_layout()
plt.show()
```

Figure 38 Screenshot of overall histogram of BILL_AMT columns

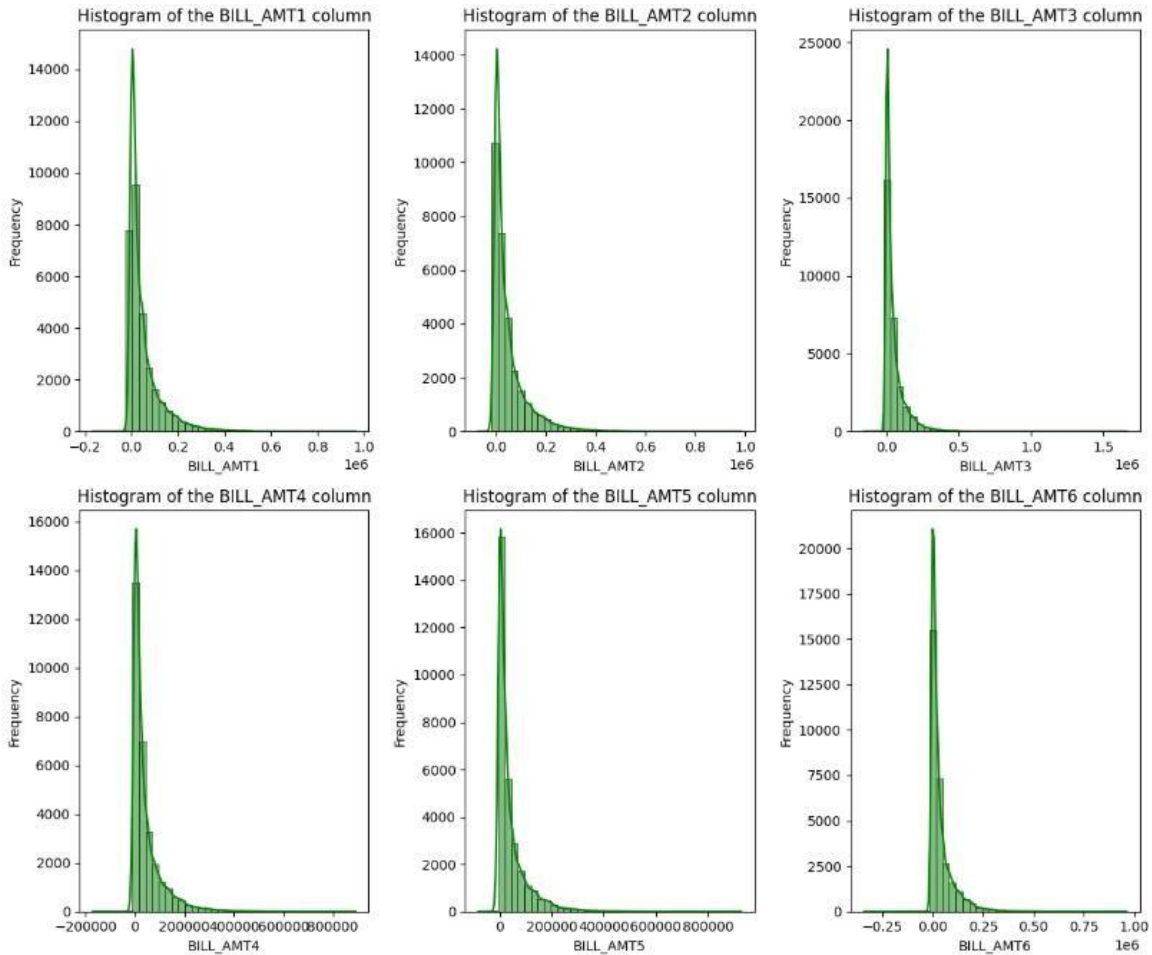


Figure 39 Histogram of overall Bill_AMT columns

Histogram of limit balance column

```
[86]: sns.histplot(df_default_credit['LIMIT_BAL'], kde=True, bins=30, color='blue')
      plt.title("Histogram of the Limit balance column")
```

```
[86]: Text(0.5, 1.0, 'Histogram of the Limit balance column')
```

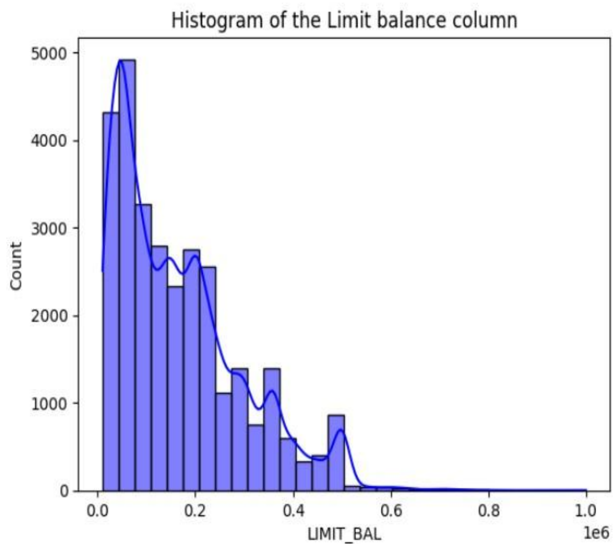


Figure 40 Histogram of limit balance column

Visualization of credit default status based on gender

From the below analysis, it tells us that female tends be on credit card default than men but in case of not default of their credit card female ratio is more, Female have higher ratio in both category of default credit card.

```
plt.figure(figsize=(10,9))
ax = sns.countplot(data=df_default_credit, x='default payment next month',hue="SEX")
handles, labels = ax.get_legend_handles_labels()
ax.legend(handles, ['Male', 'Female'])
for p in ax.patches:
    ax.annotate(str(p.get_height()), (p.get_x() * 1.01, p.get_height() * 1.01))
plt.xticks(ticks=[0, 1], labels=[' not Default', 'Default'])
plt.title("Number of credit card Default status based on gender  in the dataset")
```

```
Text(0.5, 1.0, 'Number of credit card Default status based on gender  in the dataset')
```

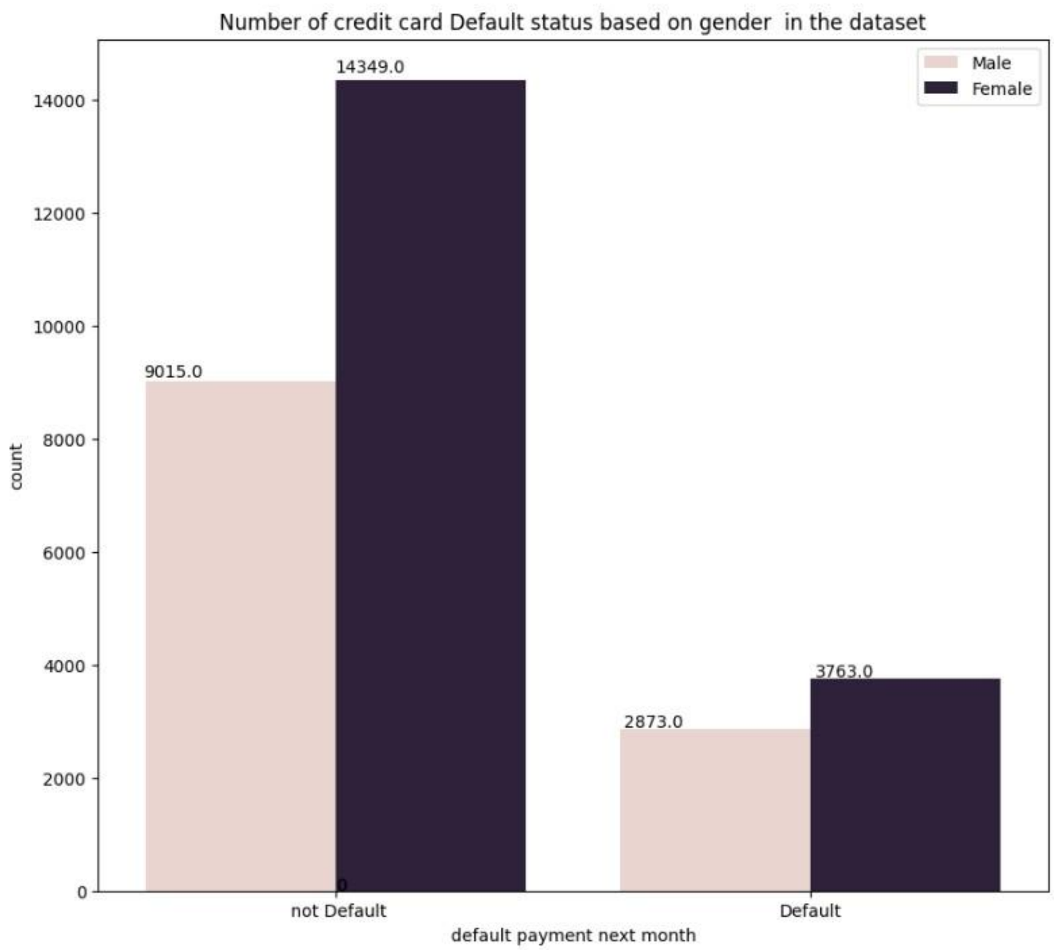


Figure 41 Visualization of default credit status based on the gender

Visualization of credit default status based on the marriage status

```
[88]: plt.figure(figsize=(10,9))
      ax = sns.countplot(data=df_default_credit, x='default payment next month',hue="MARRIAGE")
      handles, labels = ax.get_legend_handles_labels()
      ax.legend(handles, ['unknown', 'married','single','others'])
      plt.xticks(ticks=[0, 1], labels=[' not Default', 'Default'])
      plt.title("Number of credit card Default status based on marriage  in the dataset")
      for p in ax.patches:
          ax.annotate(str(p.get_height()), (p.get_x() * 1.02, p.get_height() * 1.02))
```

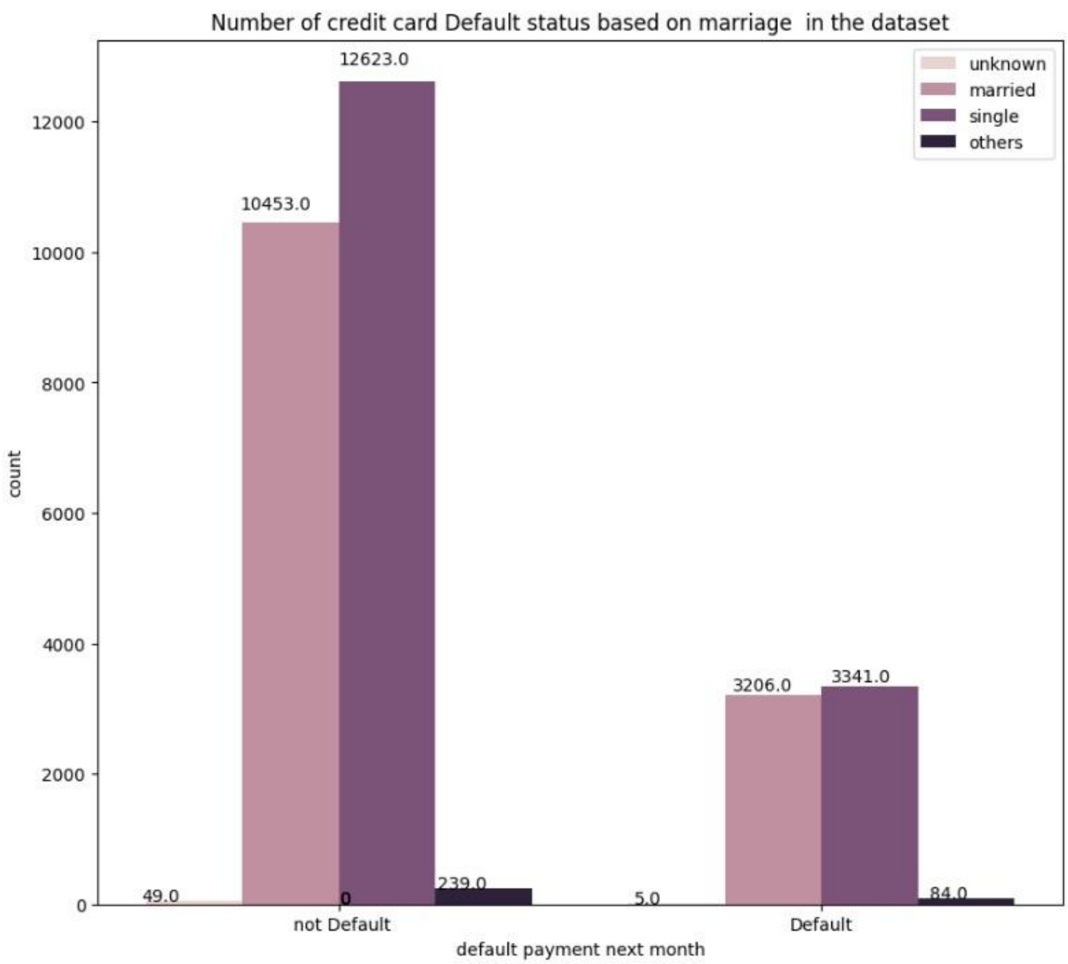


Figure 42 Visualization of default payment status based on marriage status

Heat map of all the features and target features

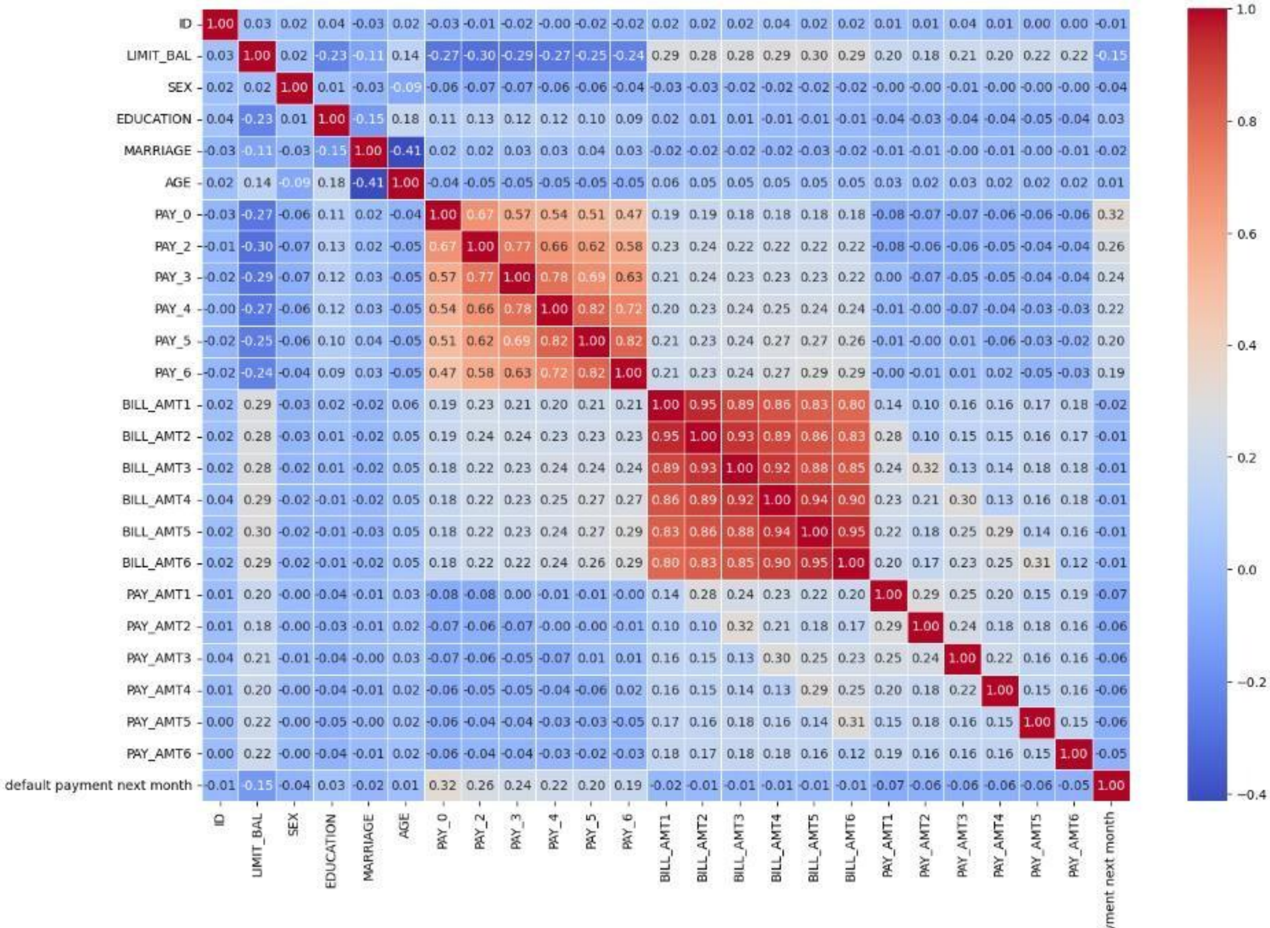


Figure 43 Heat map diagram

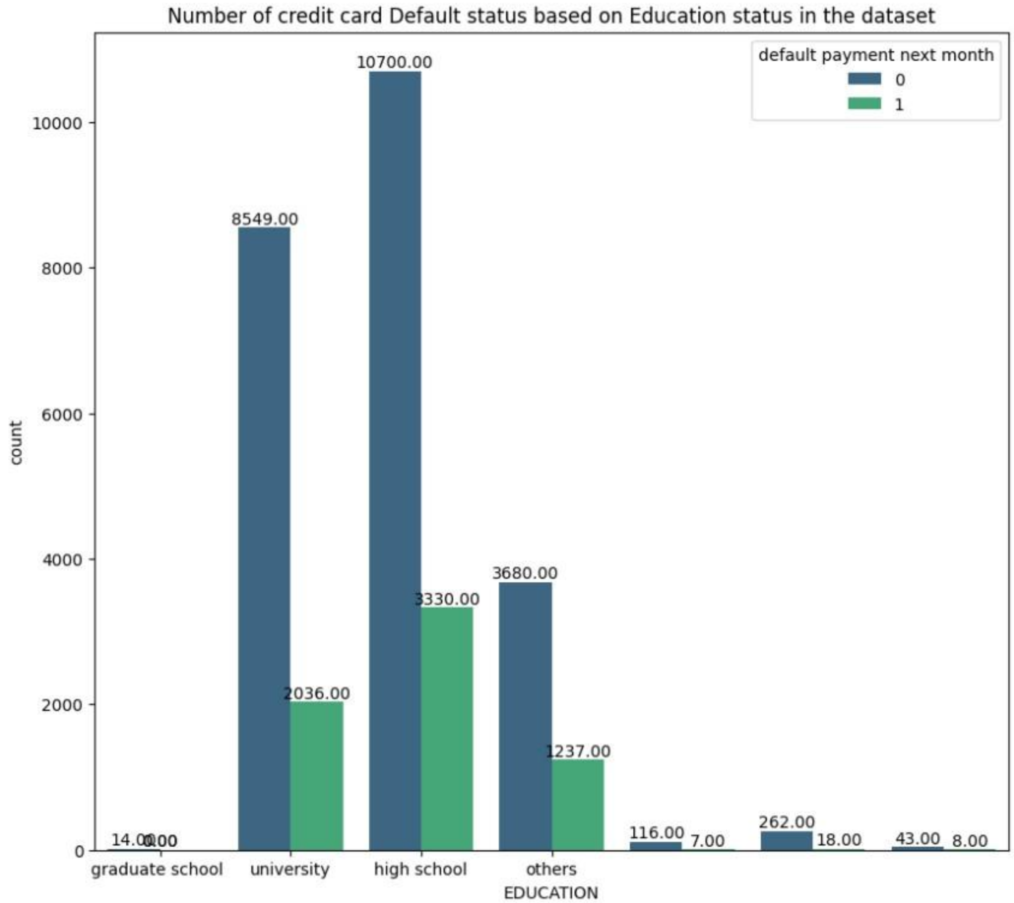


Figure 44 Number of credit card default status based on education status

4.4.4) Data Preprocessing

Replacing the '0' values of Marriage to '3' other as it is not specified and unknown The '0' values description was not specified so, replacing it to 'other' or "3" class.

5.1) Replacing the '0' values of Marriage to '3' other as it is not specified and unknown ¶

```
df_default_credit['MARRIAGE'] = df_default_credit['MARRIAGE'].replace({0:3})
```

Figure 45 Replacing values of MARRIAGE columns

Replacing the unknown values 0,5,6 of education columns to other '4'

The '0','5','6' values description was unknown so replacing it to '4' 'other 'class category.

5.2) Replacing the unknown values 0,5,6 of education columns to other '4'

```
df_default_credit['EDUCATION'] = df_default_credit['EDUCATION'].replace({0:4,5:4,6:4})
df_default_credit['EDUCATION'].value_counts()
```

```
EDUCATION
2    14030
1    10585
3     4917
4      468
Name: count, dtype: int64
```

Figure 46 Replacing the values of EDUCATION columns

Removing unwanted columns

The unnecessary columns "ID" was removed from the data frame using. drop() method where columns to remove was passes as a parameter.

5.3) Removing unwanted columns

```
df_default_credit = df_default_credit.drop(columns=['ID'])
```

Figure 47 Removing unwanted columns

Over sampling

Due to imbalance in the target value class, over sampling was performed to increase the sample of minority class to prevent model biasness toward the majority classes data.

5.4) Over sampling the miniorty class using smote

```
[99]:
X_initial = df_default_credit.drop(columns=['default payment next month'])
y_initial = df_default_credit['default payment next month']
smote = SMOTE(random_state=42)
X_smote, y_smote = smote.fit_resample(X_initial, y_initial)

X_smote_df_default_credit = pd.DataFrame(X_smote, columns=X_initial.columns)
y_smote_df_default_credit = pd.DataFrame(y_smote, columns=['default payment next month'])

df_default_credit = pd.concat([X_smote_df_default_credit, y_smote_df_default_credit], axis=1)
df_default_credit.shape
```

```
[99]: (46728, 25)
```

Figure 48 Over sampling the minority class

Data imbalance

Model on imbalance data

Due to the data imbalance which causes the model to bias toward the '0' default class as, the precision, recall and f1-score for the class '1' is 0 which indicates that model is unable to predict for the class '1'.

```
Classification Report:
              precision    recall  f1-score   support

           0       0.78      1.00      0.88      4687
           1       0.00      0.00      0.00      1313

    accuracy                           0.78      6000
   macro avg       0.39      0.50      0.44      6000
weighted avg       0.61      0.78      0.69      6000
```

Figure 49 Classification report on imbalance dataset

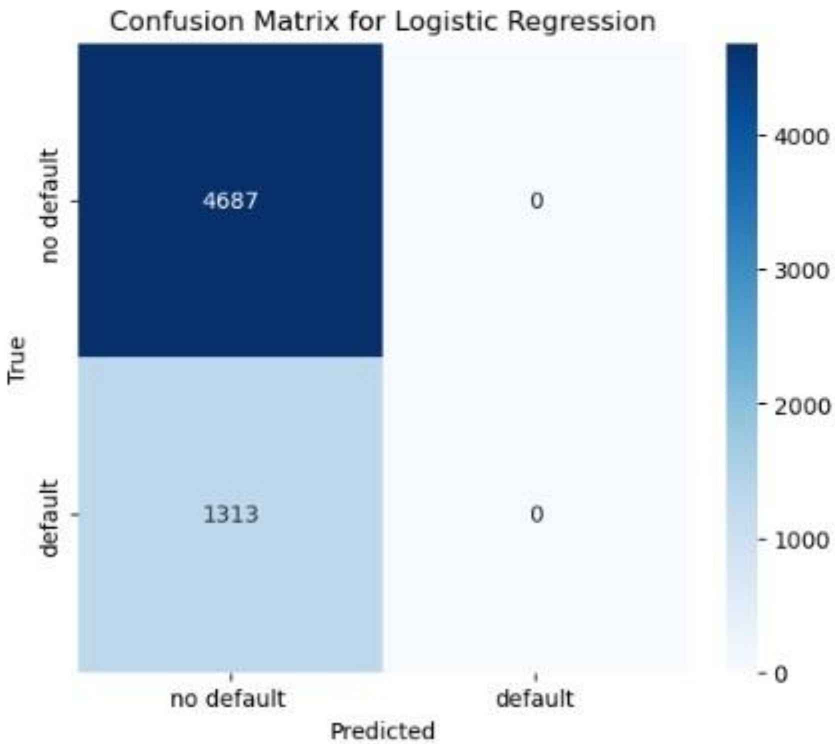


Figure 50 Confusion matrix of logistic regression on imbalance dataset

One hot encoding the categorical columns

One hot encoding is performed for categorical data as a machine learning model can misclassify the categorical data in increasing orders which can affect the prediction. The get dummies is used for one hot encoding where the required column to perform one hot encoding is passes as a parameter.

5.6) one hot encoding

```
df_default_credit_encoded = pd.get_dummies(df_default_credit, columns=['SEX','EDUCATION','MARRIAGE','PAY_0','PAY_2','PAY_3','PAY_4','PAY_5','PAY_6'])
```

Figure 51 One hot encoding

Data Scaling

Data scaling improves the machine learning model performance as the feature with larger values scale can dominate other features causing bias toward the prediction. Standardization techniques which scale the given data into mean of 0 and standard deviations of 1 for each columns.

5.7) Data scaling ¶

```
X = df_default_credit_encoded.drop(columns=['default payment next month'])  # Features
y = df_default_credit_encoded['default payment next month']  # Target

standardscaler = StandardScaler()
X_scaled = standardscaler.fit_transform(X)
X_scaled_df = pd.DataFrame(X_scaled, columns=X.columns)
df_default_credit_scale = pd.concat([X_scaled_df, y], axis=1)
```

Figure 52 Data Scaling (Standardization)

4.4.5 Feature selection

The feature and target variable are selected for training the model. X is feature variable and y is target variable.

6) Feature variable and target variable separation

```
X = df_default_credit_scale.drop(columns=['default payment next month'])
y = df_default_credit_scale['default payment next month']
```

Figure 53 Feature and target variable selection

4.4.6 Splitting the data into training and testing set

The whole data set is spitted into 80 percent for training the model and 20 percent for the model evaluation. Performing this step helps us to know the generalization of the model. i.e. underfitting and overfitting. Train_test_split() method from scikit learn module is used to perform the splitting.

7) Splitting the data into training and test set

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
y_train.value_counts()
```

```
default payment next month
0    18700
1    18682
Name: count, dtype: int64
```

Figure 54 Train test split

4.4.7 Model Initialization

In model initiation phase, the required model to train the dataset are initialized with the help of scikit learn module.

Logistic regression

```
logistic_regression_model = LogisticRegression()
```

Figure 55 Model initialization logistic regression

Random forest classifier

```
random_forest_model = RandomForestClassifier()
```

Figure 56 Model initialization Random Forest classifier

Support Vector classifier

SVC model initialization

```
svm_classify_model = SVC()
```

Figure 57 Model initialization of SVC

4.4.8 Model Training

After, the required model is initialized they are trained on training dataset, so that the model can learn pattern from the dataset.

Logistic regression

```
logistic_regression_model.fit(X_train, y_train)
```

```
▾ LogisticRegression
LogisticRegression()
```

Figure 58 Model training logistic regression

Random forest classifier

```
6]: random_forest_model.fit(X_train, y_train)
```

```
6]: ▾ RandomForestClassifier
RandomForestClassifier()
```

Figure 59 Model training random forest classifier

Support vector classifier

```
: svm_classify_model.fit(X_train, y_train)
```

```
: ▾ SVC
SVC()
```

Figure 60 Model initialization support vector classifier

4.4.9 Model Evaluation on test data before and after hyper parameter tunning

The model after being trained on the training dataset, test data set is used for the prediction and further evaluation and hyperparameter tunning.

Logistic regression

```
y_pred_logistic_test = logistic_regression_model.predict(X_test)
```

```
accuracy_logistic = accuracy_score(y_test, y_pred_logistic_test)
confusion_matrix_logistic = confusion_matrix(y_test, y_pred_logistic_test)
classification_logistic = classification_report(y_test, y_pred_logistic_test)

# Print the evaluation metrics
print("Accuracy:", accuracy_logistic)
print("Confusion Matrix:\n", confusion_matrix_logistic)
print("Classification Report:\n", classification_logistic)
```

```
Accuracy: 0.7411726942007276
Confusion Matrix:
 [[3813  851]
 [1568 3114]]
Classification Report:
               precision    recall  f1-score   support

           0       0.71      0.82      0.76      4664
           1       0.79      0.67      0.72      4682

    accuracy                           0.74      9346
   macro avg       0.75      0.74      0.74      9346
weighted avg       0.75      0.74      0.74      9346
```

```
accuracy = accuracy_score(y_train, y_pred_logistic_train)
```

Figure 61 Evaluation of logistic regression

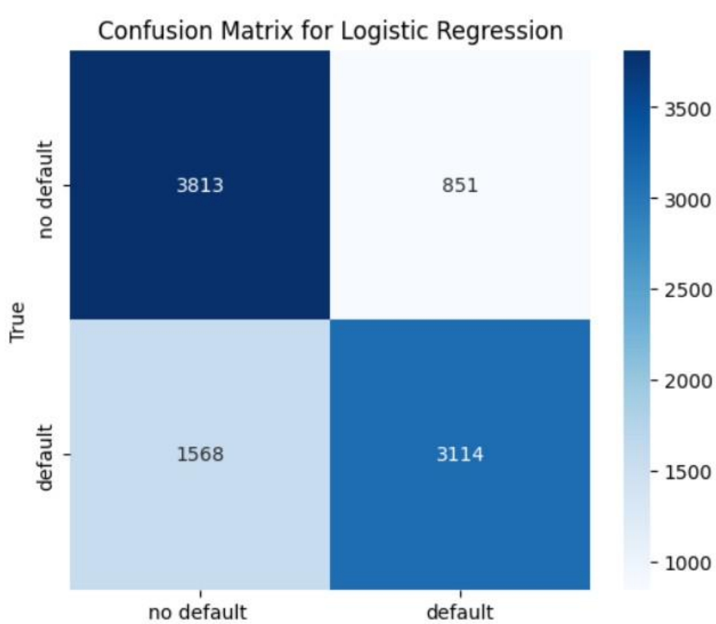


Figure 62 Confusion matrix of logistic regression

Hyper parameter tuning for logistic regression

```
param_logistic_regression = {
    'C': [0.01,0.05,0.1,0.5,1,1.5]  ,
    'max_iter': [1500, 2000],
    'penalty': ['l1', 'l2'],
    'solver': ['liblinear', 'saga']
}
random_search_logistic_regression = RandomizedSearchCV(
    estimator=logistic_regression_model,
    param_distributions=param_logistic_regression,
    scoring='f1',
    cv=5,
    n_jobs=-1,
    random_state=42,
    n_iter=10
)
random_search_logistic_regression.fit(X_train, y_train)
best_logistic_regression_model = random_search_logistic_regression.best_estimator_
y_pred_lr = best_logistic_regression_model.predict(X_test)
print("Best Hyperparameters for Logistic Regression:", random_search_logistic_regression.best_params_)
print("\nClassification Report for Logistic Regression:\n", classification_report(y_test, y_pred_lr))
```

```
Fitting 5 folds for each of 10 candidates, totalling 50 fits
Best Hyperparameters for Logistic Regression: {'solver': 'liblinear', 'penalty': 'l1', 'max_iter': 1500, 'C': 0.5}

Classification Report for Logistic Regression:
              precision    recall  f1-score   support

           0       0.71      0.82      0.76      4664
           1       0.79      0.66      0.72      4682

    accuracy                           0.74      9346
   macro avg       0.75      0.74      0.74      9346
weighted avg       0.75      0.74      0.74      9346
```

Evaluation of logistic regression

Evaluation metric	Before hyper parameter tuning	After hyper parameter tunning
Precision	0.75	0.75
Recall	0.74	0.74
F1-score	0.74	0.74
accuracy	0.74	0.74

The performance of the logistic regression didn’t change after the hyper parameter tuning was performed.

Random forest classifier

Random forest model prediction in test data

```
[58]: y_pred_random_forest_test = random_forest_model.predict(X_test)
```

Evaluation metric for Random forest classifier ¶

```
[61]: accuracy_Random_forest_classifier = accuracy_score(y_test, y_pred_random_forest_test)
confusion_matrix_Random_forest_classifier = confusion_matrix(y_test, y_pred_random_forest_test)
classification_Random_forest_classifier = classification_report(y_test, y_pred_random_forest_test)

# Print the evaluation metrics
print("Accuracy:", accuracy_Random_forest_classifier)
print("Confusion Matrix:\n", confusion_matrix_Random_forest_classifier)
print("Classification Report:\n", classification_Random_forest_classifier)
```

```
Accuracy: 0.8346886368499893
Confusion Matrix:
 [[4061  603]
 [ 942 3740]]
Classification Report:
               precision    recall  f1-score   support

           0       0.81      0.87      0.84      4664
           1       0.86      0.80      0.83      4682

    accuracy                           0.83      9346
   macro avg       0.84      0.83      0.83      9346
weighted avg       0.84      0.83      0.83      9346
```

Figure 63 Model Evaluation of Random Forest classifier

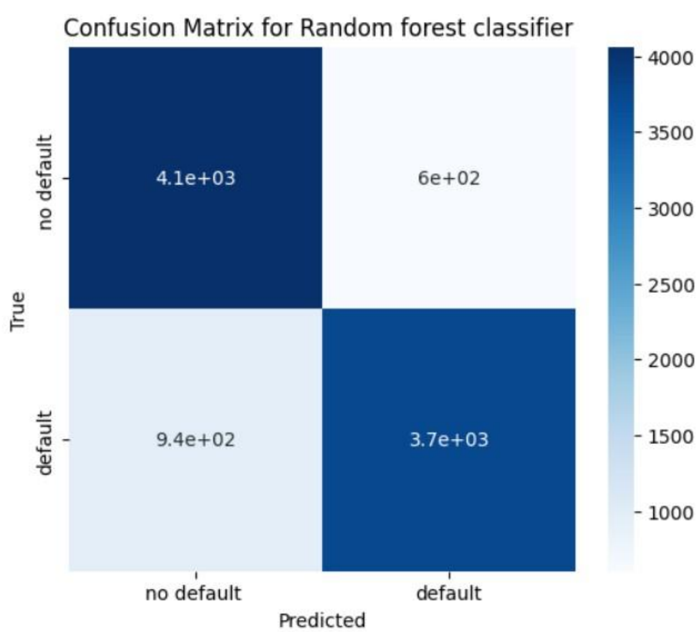


Figure 64 Confusion matrix of logistic regression

Hyper parameter tuning Random forest classifier

```python
param_random_forest_model = {
    'n_estimators': [200, 500,1000],
    'max_features': ['sqrt', 'log2'],
    'max_depth': [None, 10, 30],
    'min_samples_split': [2, 4,8],
    'min_samples_leaf': [1, 2,4],
}

random_search_random_forest = RandomizedSearchCV(
    estimator=random_forest_model,
    param_distributions=param_random_forest_model,
    scoring='f1',
    cv=5,
    n_jobs=-1,
    random_state=42,
    n_iter=10
)


random_search_random_forest.fit(X_train, y_train)
best_randomforest_model = random_search_random_forest.best_estimator_
y_pred_rf = best_randomforest_model.predict(X_test)

print("Best Hyperparameters for Random Forest Classifier:", random_search_random_forest.best_params_)
print("\nClassification Report:\n", classification_report(y_test, y_pred_rf))
```

```
Best Hyperparameters for Random Forest Classifier: {'n_estimators': 1000, 'min_samples_split': 2, 'min_samples_leaf': 1, 'max_features': 'log2', 'max_depth': None}

Classification Report:
              precision    recall  f1-score   support

           0       0.82      0.87      0.84      4664
           1       0.86      0.80      0.83      4682

    accuracy                           0.84      9346
   macro avg       0.84      0.84      0.84      9346
weighted avg       0.84      0.84      0.84      9346
```

Evaluation of random forest classifier

Evaluation metric	Before hyper parameter tuning	After hyper parameter tunning
Precision	0.83	0.84
Recall	0.83	0.84
F1-score	0.83	0.84
accuracy	0.83	0.84

After performing the hyper parameter tuning the performance improve slightly by 0.1 percent only in overall performance metric i.e. 0.84.

Support vector classifier

SVC Model prediction on test data

```
[197]: y_pred_svm = svm_classify_model.predict(X_test)
```

Model Evaluation for SVC Model

```
[199]: accuracy_svm_classifier = accuracy_score(y_test, y_pred_svm)
confusion_matrix_svm_classifier = confusion_matrix(y_test, y_pred_svm)
classification_svm_classifier = classification_report(y_test,y_pred_svm)

# Print the evaluation metrics
print("Accuracy:", accuracy_svm_classifier)
print("Confusion Matrix:\n", confusion_matrix_svm_classifier)
print("Classification Report:\n", classification_svm_classifier)
```

```
Accuracy: 0.7642841857479136
Confusion Matrix:
 [[3915  749]
 [1454 3228]]
Classification Report:
              precision    recall  f1-score   support

           0       0.73      0.84      0.78      4664
           1       0.81      0.69      0.75      4682

    accuracy                           0.76      9346
   macro avg       0.77      0.76      0.76      9346
weighted avg       0.77      0.76      0.76      9346
```

Figure 65 Model evaluation of Support vector classifier

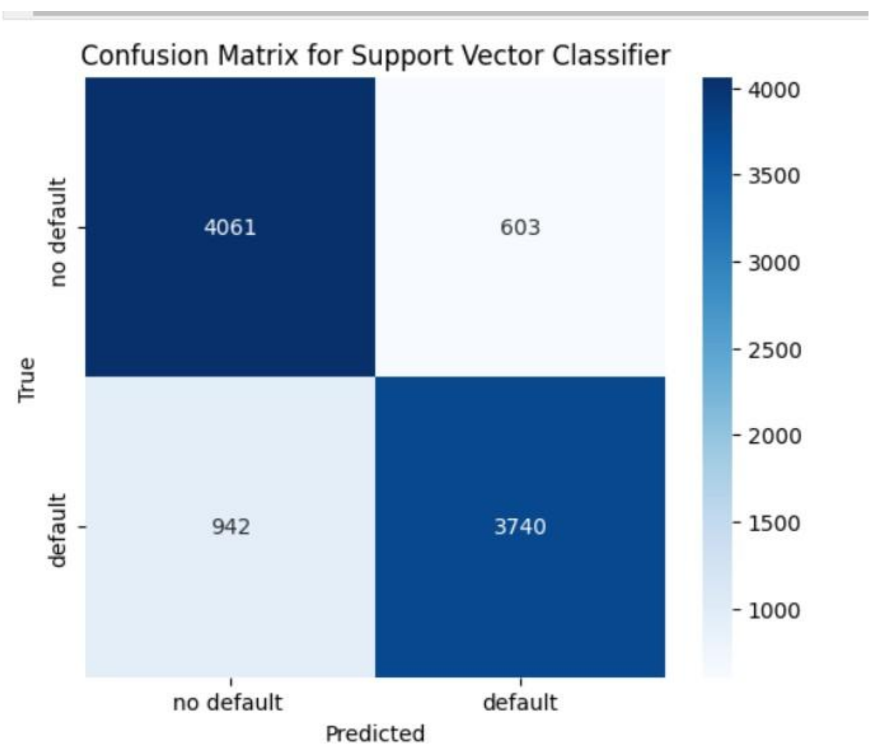


Figure 66 Confusion matrix of Support vector classifier

Voting classifier

Initialization of base learner model for voting classifier

```
base_learners = [
    ('svc', SVC(kernel='linear', probability=True)),
    ('LR', LogisticRegression(solver= 'liblinear', penalty= 'l1', max_iter=1500, C=0.5))  ,
    ('RF',RandomForestClassifier(n_estimators= 1000, min_samples_split= 2, min_samples_leaf= 1, max_features= 'log2', max_depth= None))
]
```

Figure 67 Voting classifier base model

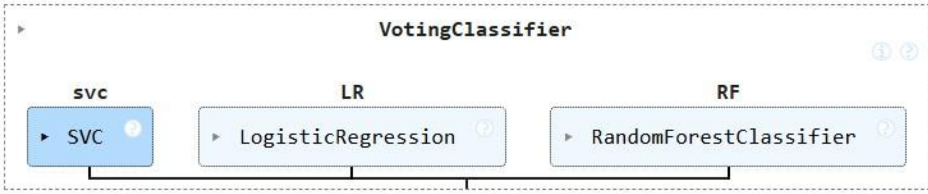
Votting classifier model

```
voting_clf = VotingClassifier(
    estimators=base_learners,
    voting='soft',
    n_jobs=-1
)
```

Figure 68 Voting classifier initialization

Fit the voting classifier model

```
voting_clf.fit(X_train, y_train)
```




```
y_pred = voting_clf.predict(X_test)
print("Classification Report:", classification_report(y_test, y_pred))
```

```
Classification Report:
              precision    recall  f1-score   support

           0       0.76      0.84      0.80      4664
           1       0.82      0.73      0.77      4682

    accuracy                           0.79      9346
   macro avg       0.79      0.79      0.79      9346
weighted avg       0.79      0.79      0.79      9346
```

Overall Model comparison

Evaluation metric	Logistic regression	Random forest classifier	Voting classifier	Support vector classifier
Precision	0.75	0.84	0.79	0.77
Accuracy	0.74	0.84	0.79	0.76
Recall	0.74	0.84	0.79	0.76
F1 score	0.74	0.84	0.79	0.76

In comparison, of logistic regression, random forest classifier and support vector classifier model the best performing model was random forests classifier with an accuracy, precision, recall and f1 score of 0.84 as it can handle the nonlinear relationship effectively.

4. Conclusion

4.1 Analysis of the work done and application solution addresses real world problems

Credit card default is increasing on a yearly basis due to the change in the lifestyle of the people, job scarcity, and lack of awareness regarding it. Using machine learning model, the customer credit card default can be predicted with good accuracy which can benefit the banking institution as they can prevent the financial loss by using ml model to predict whether approving the customer with the credit card application will be worth it or not. Different research methodology was studied regarding the solution of credit card default using machine learning model and found out that ML algorithms can be used to predict the customer credit card default status. The data description of the dataset of the UCI Default credit card was conducted and studied in detail along with some data analysis. The Exploratory data analysis and data preprocessing was conducted to make the data ready for the model training and finding out the relationship, insight and summary about each column of the data.

The data set was imbalanced with affected the performance of the machine learning model, using smote oversampling method, the minority class was increased to balance out the target feature to prevent the model biasness toward the majority class. The three-model logistic regression, random forest classifier and support vector machine was trained on training dataset, and their performance was evaluated and compared after performing the hyper parameter tuning. The random forest classifier performed well compared to other model with an overall accuracy of 0.84. An ensemble learning model voting classifier was also used which combines the prediction of the three model averages it as a final prediction, but was not performing better than random forest as there was huge difference between the performance of random forest and other model.

4.2 Further work

- Advanced machine learning model like Neural network, Boost, LightGBM could be used to compare and evaluate the model performance for better accuracy.
- PCA for dimension reduction could be implemented for better generalization of the model as the unwanted, noisy and redundant features is removed.
- The model could be further deployed to the production using flask web framework where the best performing model is saved as a pickle file and loaded to the web environment, with proper user interface and input field. Using flask, and web API can be created to handle user input request, loading the ml model file and passing the user input into the model to generate the prediction and passing it as a response to the User.

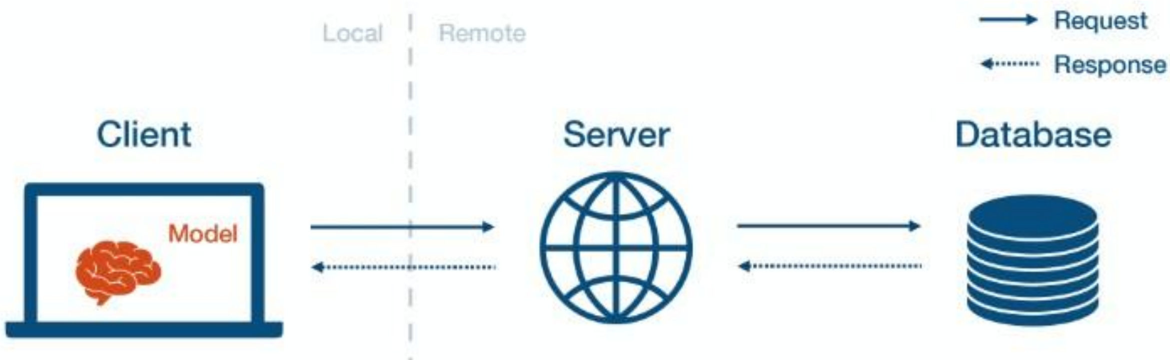


Figure 69 Further work on Flask API

5. References

AWS, 2024. *What is AI?.* [Online]
Available at: https://aws.amazon.com/what-is/artificial-intelligence/ [Accessed 24 December 2024].

Brown, S., 2021. *Machine learning, explained.* [Online]
Available at: https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained [Accessed 23 December 2024].

Bucci, S., 2023. *Bankrate.* [Online]
Available at: https://www.bankrate.com/credit-cards/advice/credit-card-default/ [Accessed 23 December 2024].

Coursera, 2024. *What Is Artificial Intelligence? Definition, Uses, and Types.* [Online]
Available at: https://www.coursera.org/articles/what-is-artificial-intelligence [Accessed 24 December 2024].

Craig, L., 2024. *What is AI? Artificial Intelligence explained.* [Online]

Available at: https://www.techtarget.com/searchenterpriseai/definition/AI-Artificial-Intelligence [Accessed 24 December 2024].

Crowther, D., 2022. *Maxing out: Credit card debts are rising in the US.* [Online]
Available at: https://sherwood.news/world/credit-card-debt-are-rising/ [Accessed 23 December 2024].

datascience@berkeley, 2022. *what-is-machine-learning.* [Online]
Available at: https://ischoolonline.berkeley.edu/blog/what-is-machine-learning/ [Accessed 23 December 2024].

Florentin Butaru, Q. C. B. C. S. D. A. W. L. a. A. S., 2016. Risk and risk management in the credit card industry. *Journal of Banking and Finance,* p. 219.

geeksforgeeks, 2024. *Logistic Regression in Machine Learning.* [Online]
Available at: https://www.geeksforgeeks.org/understanding-logistic-regression/#how-doeslogistic-regression-work [Accessed 24 December 2024].

geeksforgeeks, 2024. *Support Vector Machine (SVM) Algorithm.* [Online] Available at: https://www.geeksforgeeks.org/support-vector-machine-algorithm/ [Accessed 23 December 2024].

Gunay, D., 2023. *Random Forest.* [Online]
Available at: https://medium.com/@denizgunay/random-forest-af5bde5d7e1e [Accessed 24 December 2024].

Hritwiz Yash, A. S. D. S., 2023. *Credit card default prediction using machine.* [Online]
Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10351316 [Accessed 23 December 2024].

IBM, 2024. *What is supervised learning?.* [Online]
Available at: https://www.ibm.com/think/topics/supervised-learning [Accessed 23 December 2024].

Jainvidip, 2024. *Understanding Support Vector Machines (SVMs).* [Online]
Available at: https://medium.com/@jainvidip/understanding-support-vector-machines-svms-1f7c78bad934
[Accessed 24 December 2024].

Jaleel, T., 2024 . *Credit card defaults on the rise; what is the punishment for credit card default.* [Online]

Available at: https://economictimes.indiatimes.com/wealth/spend/credit-card-defaults-on-therise-what-is-the-punishment-for-credit-card-default/articleshow/113690164.cms?from=mdr [Accessed 23 December 2024].

Junhong Li, J. K. J. W. H. W. X. Y., 2024. *Research on credit card default repayment prediction model.* [Online]
Available at: https://pdf.sciencedirectassets.com/313360/1-s2.0-S2405918824X00021/1-s2.0-S2405918824000217/main.pdf?X-Amz-Security-Token=IQoJb3JpZ2luX2VjECMaCXVzLWVhc3QtMSJHMEUCIQDmDwWRC0dwo5bPbk0tRwrxMWDatrTxSMrDj1VlmInSjAIgAnDk1Q9Xs2cLkocaA9A4CPLp6lQjJJ2DKltcbIonhRsq
[Accessed 23 December 2024].

Kulkarni, V., 2023. *Classification Metrics.* [Online]
Available at: https://medium.com/@vyankatesh.kulkarni20/classification-metrics-63c635852995
[Accessed 20 January 2025].

Narkhede, S., 2018. *Understanding Confusion Matrix.* [Online]
Available at: https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62
[Accessed 20 January 2025].

NB, H., 2019. *Confusion Matrix, Accuracy, Precision, Recall, F1 Score.* [Online]
Available at: https://medium.com/analytics-vidhya/confusion-matrix-accuracy-precision-recallf1-score-ade299cf63cd
[Accessed 20 January 2025].

NewsDrum, 2023. *Credit card default rises to Rs 4,072 crore in FY24.* [Online]
Available at: https://www.newsdrum.in/business/credit-card-default-rises-to-rs-4072-crore-infy24
[Accessed 23 December 2024].

SALAH, B., 2024. *A-Z Guide to the Types of Machine Learning Problems.* [Online]
Available at: https://www.projectpro.io/article/types-of-machine-learning/623 [Accessed 23 December 2024].

School, P. B., 2024. *How random forest works in data science ?.* [Online]
Available at: https://www.edcparis.edu/en/school-news/how-random-forest-works-data-science
[Accessed 24 December 2024].

Shmilovici, A., 2024. *Support Vector Machines.* [Online]
Available at: https://link.springer.com/chapter/10.1007/0-387-25465-X_12 [Accessed 24 December 2024].

shung , K. P., 2018 . *Accuracy, Precision, Recall or F1?.* [Online]

Available at: https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9
[Accessed 20 January 2025].

Streaks, J., 2024. *What is a credit card default? A complete guide.* [Online]
Available at: https://www.businessinsider.com/personal-finance/credit-score/what-is-a-creditcard-default#:~:text=Credit%20card%20defaults%20occur%20when,to%20establish%20a%20payment%20plan.
[Accessed 23 December 2024].

Ying Chen, d. R. Z., 2021. Research on Credit Card Default Prediction Based on k-Means SMOTE and BP Neural Network. Volume 2021, pp. 1-13.