Credit Card Default Prediction

Cohort kaimur

Jayalaxmi Mekap

Almabetter

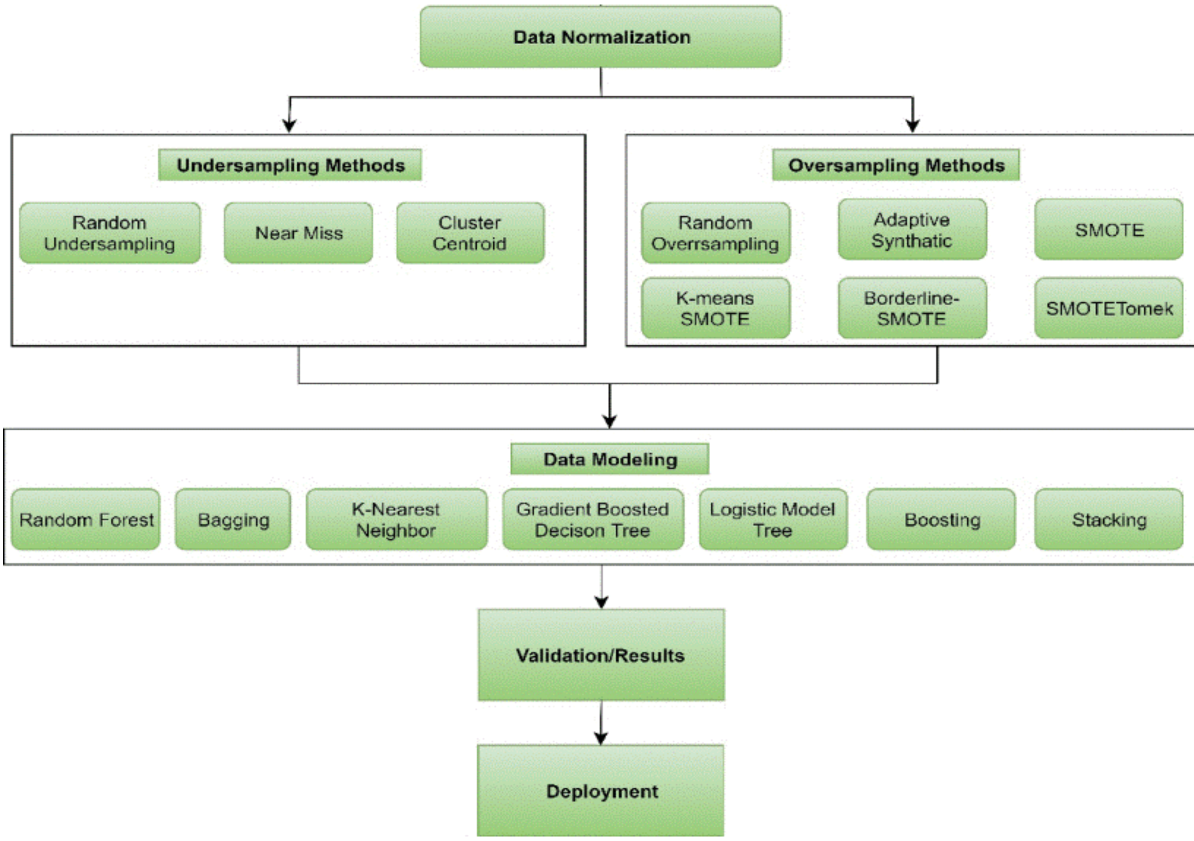
Abstract:

Credit card frauds are easy and friendly targets. E-commerce and many other online sites have increased the online payment modes, increasing the risk for online frauds. Increase in fraud rates, researchers started using different machine learning methods to detect and analyse frauds in online transactions. The main aim of the project is to design and develop a novel fraud detection method for Streaming Transaction Data, with an objective, to analyse the past transaction details of the customers and extract the behavioural patterns. Where cardholders are clustered into different groups based on their transaction amount. Then using sliding window strategy to aggregate the transaction made by the cardholders from different groups so that the behavioural pattern of the groups can be extracted respectively. Later different classifiers are trained over the groups separately. And then the classifier with better rating score can be chosen to be one of the best methods to predict frauds. Thus, followed by a feedback mechanism to solve the problem of concept drift. In this paper, we worked with European credit card fraud dataset. This data set involves various steps such as:

- Data Understanding
- Feature Analysis
- Feature Engineering
- Exploratory Data Analysis
- Implementing Logistic Regression
- Implementing Random Forest
- Hyperparameter Tuning
- Evaluating Models
- Data Visualization
- Multivariate Analysis
- SMOTE
- SVC
- ROC AUC Curve

- Research Analytics
- Technical documentation
- One Hot Encoding
- XGBoost
- Research Analytics

Data science process



1.Problem Statement

In recent years, the credit card issuers in Taiwan faced the cash and credit card debt crisis and the delinquency is expected to peak in the third quarter of 2006 (Chou,2006). In order to increase market share, card-issuing banks in Taiwan over-issued cash and credit cards to unqualified applicants. At the same time, most cardholders, irrespective of their repayment ability, overused credit card for consumption and accumulated heavy credit and cash–card debts. The crisis caused the blow to consumer finance confidence and it is a big challenge for both banks and cardholders.

2. Introduction

ID: ID of each client

LIMIT_BAL: Amount of given credit in NT dollars (includes individual and family/supplementary credit

SEX: Gender (1=male, 2=female)

EDUCATION: (1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown)

MARRIAGE: Marital status (1=married, 2=single, 3=others)

AGE: Age in years

PAY_0: Repayment status in September, 2005 (-1=pay duly, 1=payment delay for one month, 2=payment delay for two months,8=payment delay for eight months, 9=payment delay for nine months and above)

PAY_2: Repayment status in August, 2005 (scale same as above)

PAY_3: Repayment status in July, 2005 (scale same as above)

PAY_4: Repayment status in June, 2005 (scale same as above)

PAY_5: Repayment status in May, 2005 (scale same as above)

PAY_6: Repayment status in April, 2005 (scale same as above)

BILL_AMT1: Amount of bill statement in September, 2005 (NT dollar)

BILL_AMT2: Amount of bill statement in August, 2005 (NT dollar)

BILL_AMT3: Amount of bill statement in July, 2005 (NT dollar)

BILL_AMT4: Amount of bill statement in June, 2005 (NT dollar)

BILL_AMT5: Amount of bill statement in May, 2005 (NT dollar)

BILL_AMT6: Amount of bill statement in April, 2005 (NT dollar)

PAY_AMT1: Amount of previous payment in September, 2005 (NT dollar)

PAY_AMT2: Amount of previous payment in August, 2005 (NT dollar)

PAY_AMT3: Amount of previous payment in July, 2005 (NT dollar)

PAY_AMT4: Amount of previous payment in June, 2005 (NT dollar)

PAY_AMT5: Amount of previous payment in May, 2005 (NT dollar)

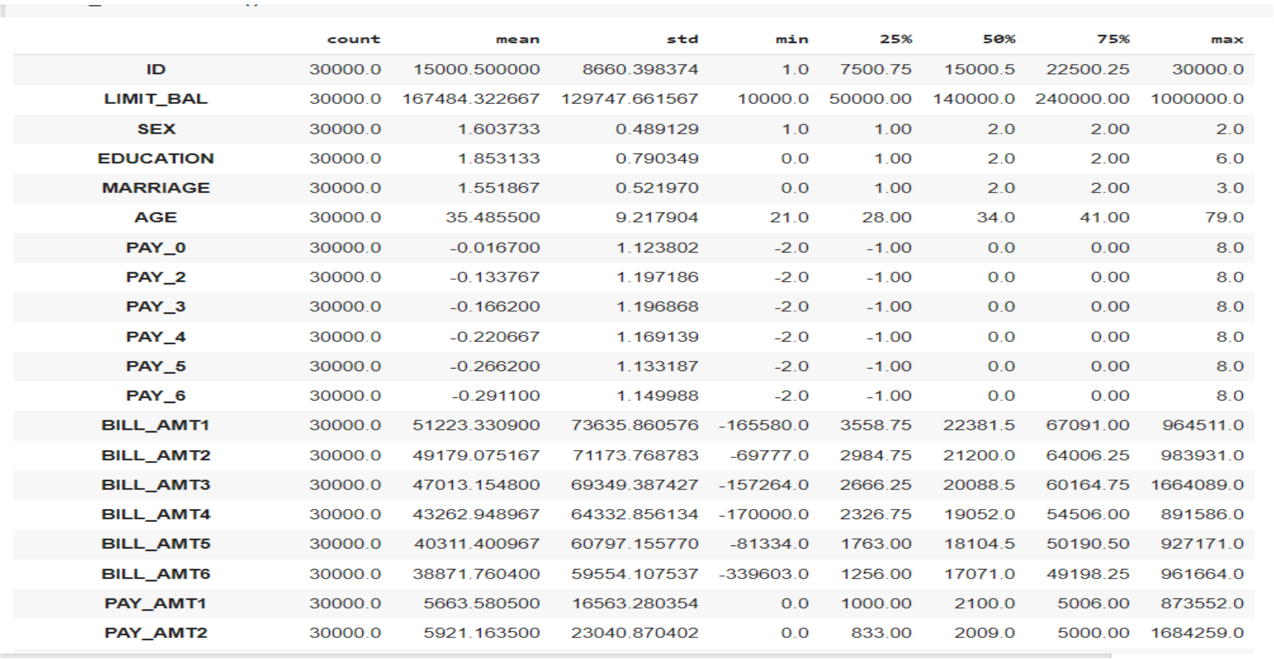
PAY_AMT6: Amount of previous payment in April, 2005 (NT dollar)

Default payment next month: Default payment (1=yes, 0=no)

3. **Exploratory Data Analysis:**

The following steps are involved in the project

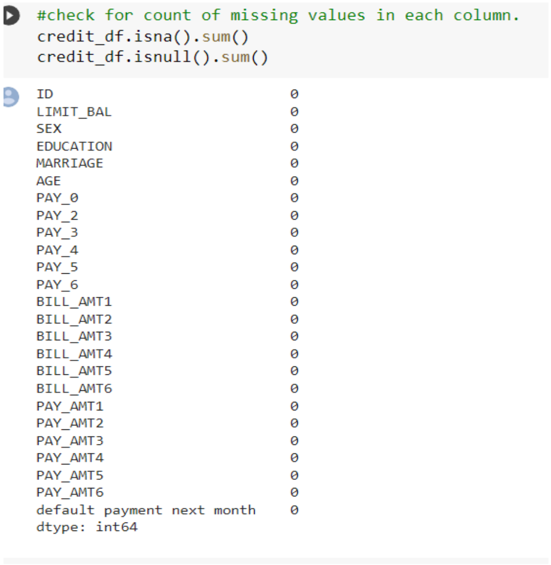
After loading and reading the dataset in a notebook, we performed exploratory data analysis. The purpose of exploratory data analysis is to identify the variables that impact payment default next month and the correlations between them. We use graphical data exploratory analysis to check every categorical variable. We plot the graph and visualize the data.



	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

Null values Treatment and Outliers:

Dataset contains no null values to disturb the accuracy.

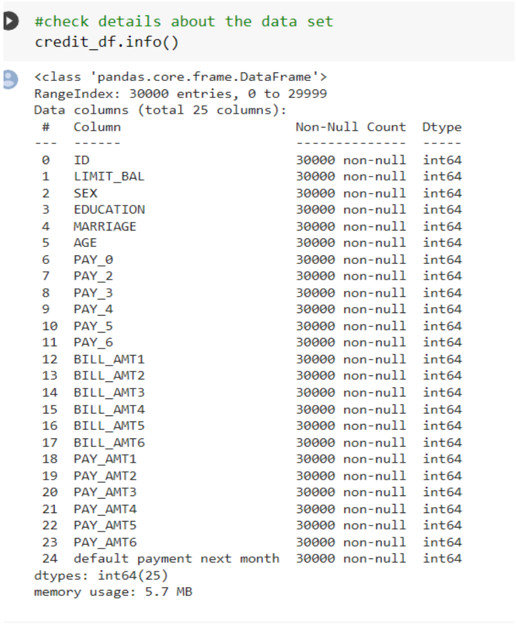


```
#check for count of missing values in each column.
credit_df.isna().sum()
credit_df.isnull().sum()
```

```
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
```

Numerical and categorical Features:

With the help of exploratory data analysis, we analysed the categorical as well as numerical features in the dataset.

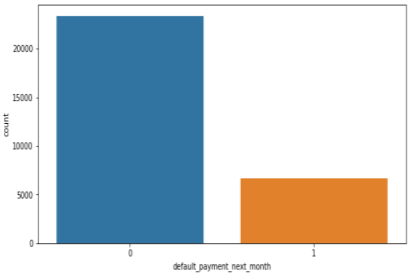


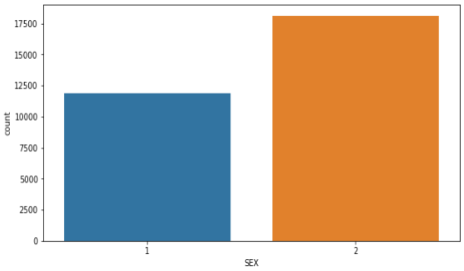
```
#check details about the data set
credit_df.info()
```

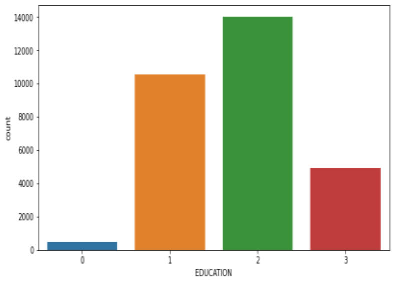
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30000 entries, 0 to 29999
Data columns (total 25 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   ID                          30000 non-null  int64
 1   LIMIT_BAL                   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
```

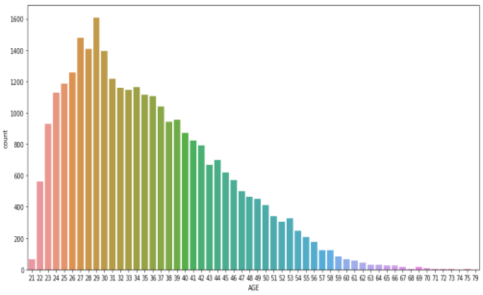
Correlation Analysis:

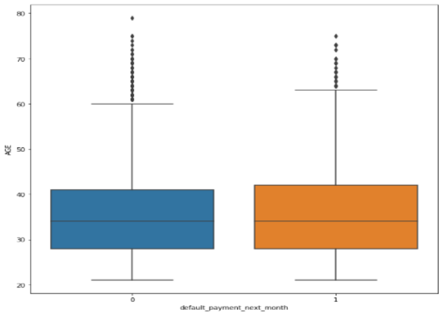
We plot the heatmap to find the correlation between both the dependent variable and independent variables.

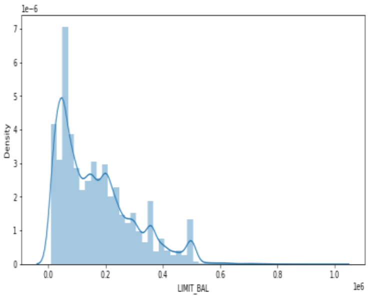

count
0
5000
10000
15000
20000
0
1
default_payment_next_month

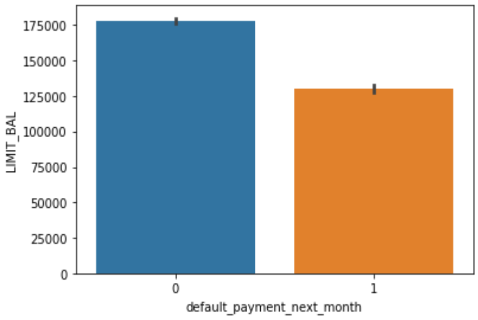

count
0
2500
5000
7500
10000
12500
15000
17500
1
2
SEX


count
0
2000
4000
6000
8000
10000
12000
14000
0
1
2
3
EDUCATION


count
0
200
400
600
800
1000
1200
1400
1600
21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 79
AGE


AGE
20
30
40
50
60
70
80
0
1
default_payment_next_month

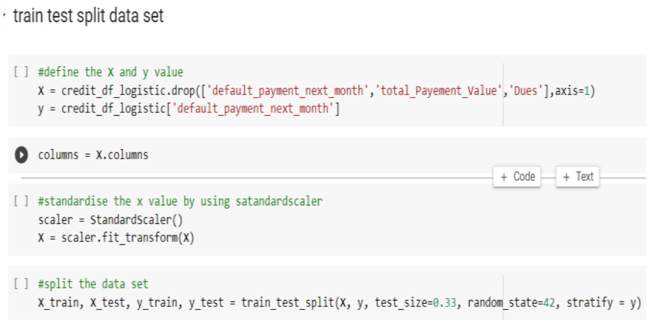

1e−6
Density
0
1
2
3
4
5
6
7
0.0
0.2
0.4
0.6
0.8
1.0
1e6
LIMIT_BAL



default_payment_next_month
0
1

Train test Split:

In the train test split, we take x as dependent variables and y take as an independent variable then train the model.



· train test split data set

```
[ ] #define the X and y value
    X = credit_df_logistic.drop(['default_payment_next_month','total_Payement_Value','Dues'],axis=1)
    y = credit_df_logistic['default_payment_next_month']
```

```
columns = X.columns
```

```
[ ] #standardise the x value by using satandardscaler
    scaler = StandardScaler()
    X = scaler.fit_transform(X)
```

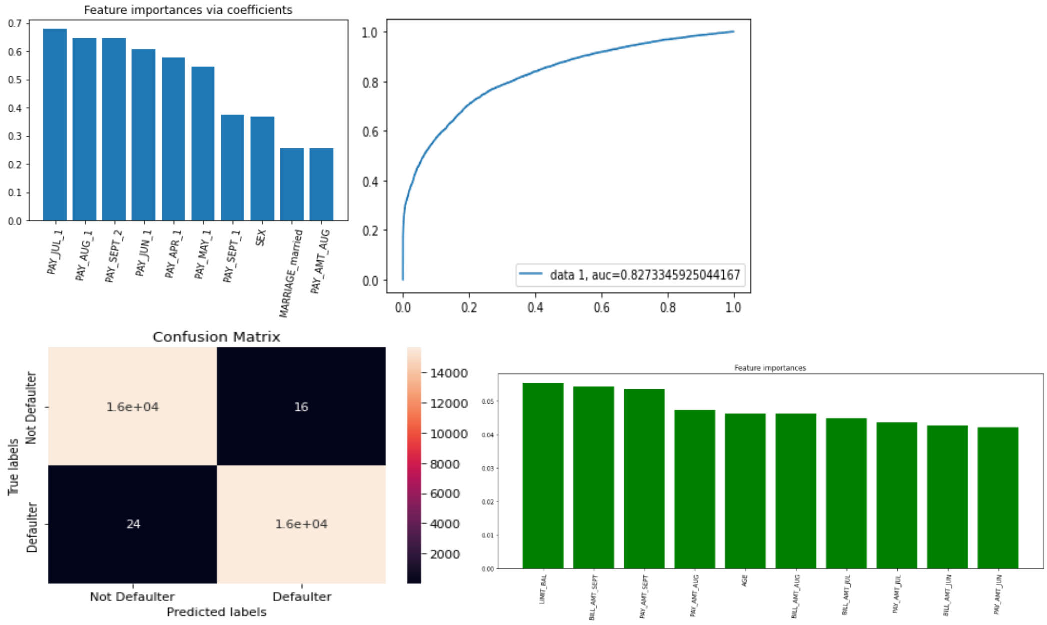
```
[ ] #split the data set
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=42, stratify = y)
```

Models:

We use 3 models to train the data and for predicting the accuracy.

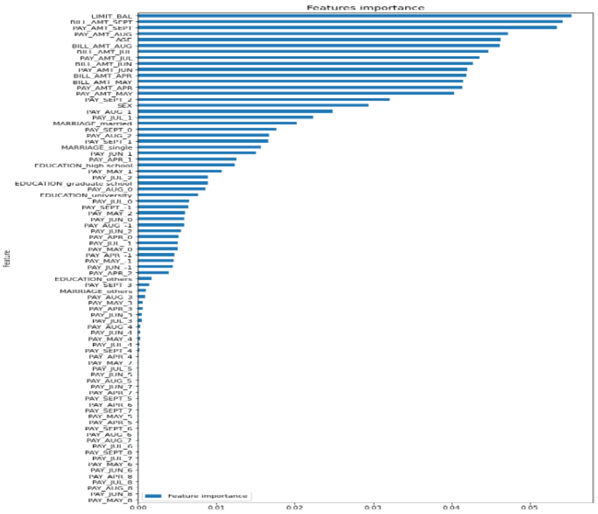
1. Logistic regression
2. Random forest
3. SVC

4. XG boost



Feature Selection

There are 25 columns in this dataset and the target variable is the column is default payment next month. We drop the column 'ID' and target variable and save the rest 23 as predictor features. These predictor variables include categorical variables such as sex, age, education marital status along with numerical variables, such as payment status, credit limit, bill amount, etc.



Imbalance data:

An imbalanced dataset will mislead machine learning algorithms and affect their performances so then we apply train test split to balance data.

Split Training and Test Data:

In the train test split, we take two variables ie X and Y where X contains all the independent variables and Y contains the dependent variable. Here the independent variable is default payment next month and dependent variables are affecting the default payment next month like age, gender, credit limit, education, bill payment, etc. For the model, we use the ratio for training and test data split by 80% for training, 20% for the test to ensure consistency. After splitting the data, we set the test data aside and leave it for the very end, which is the final testing after hyperparameter tuning

Performance Metrics:

Since this is a classification problem with imbalanced classes, we use performance metrics i.e. accuracy precision and recall is a better choice. However, there is a known trade-off between precision and recall. We can raise recall to arbitrarily high, but the precision will decrease. We use the below metrics to measure model performances.

a. Confusion matrix b. ROC_AUC c. Precision recall curve

SMOTE Oversampling:

In the initial model fitting, we start by using all models' default parameters. To compensate for the rare classes in the imbalance dataset, we use SMOTE(Synthetic Minority Over-Sampling Technique) method to oversample the minority class and ensure the sampling is not biased. What this technique does under the hood is simply duplicate examples from the minority class in the training dataset before fitting a model. after SMOTE sampling, the dataset has an equal size of 0s and 1s. To verify if SMOTE improves models' performance, all 3 models are trained with SMOTE and without SMOTE. The below table shows the ROC_ AUC scores on training data 9 improved significantly with all models after oversampling with SMOTE. This proves SMOTE is an effective method in sampling imbalanced datasets. Models Training AUC Without SMOTE Training AUC With SMOTE Logistic Regression 0.726 0.797 Random Forest 0.764 0.916 XGBoost 0.762 0.899 Table 1: Model ROC_AUC score on training data with default parameters

Conclusion

More credit card default for limit balance about 10000. It might mean that credit card might be too easy to be issued for people who have low credit scores. The variance of the default rate for limit balance over 500,000 NTD is higher than other range of limit balance. It is lower default rate for cardholders have higher education level. Moreover, the default rate for clients whose age over 60 was higher than mid age and young people. The best fit algorithm for predicting limit balance is bagging approach. The best fit algorithm for predicting whether a client default next month is classification tree.