Code Report Credit Default Prediction

Student ID: 577610

Author: Marcel De Lange

Date:: 26-July-2024 - 28 July 2024

Table of Contents

1. Problem Formulation

Problem Statement

- The primary objective of this project is to predict whether a credit card holder is likely to default on their payment in the next month. Default prediction is crucial for financial institutions to manage risk and make informed decisions about credit issuance and management.

Motivation

- Predicting credit default allows financial institutions to proactively manage credit risk by identifying high-risk customers. This can lead to better financial planning, reduced losses, and targeted interventions to help prevent defaults.

2. Features/Variables in the Problem

Features Explanation

1. **LIMIT (LIMIT_BAL):** Amount of credit given to the cardholder (in dollars). This feature represents the credit limit assigned to the cardholder, which can influence their ability to default.
2. **SEX (SEX):** Gender of the credit card holder.
    - 1 = Male
    - 2 = Female
3. **EDUCATION (EDUCATION):** Educational level of the credit card holder.
    - 1 = Graduate school
    - 2 = University
    - 3 = High school
    - 4 = Other
4. **MARRIAGE (MARRIAGE):** Marital status of the credit card holder.
    - 1 = Married
    - 2 = Single
    - 3 = Others
5. **AGE (AGE):** Age of the credit card holder (in years). Age can affect the likelihood of defaulting due to varying financial responsibilities and stability.
6. **PAY_0 to PAY_6:** History of past payment. This feature shows the repayment status from the last 6 months.
    - -1 = Paid duly
    - 1 = Payment delay for one month
    - 2 = Payment delay for two months
    - ...
    - 9 = Payment delay for nine months and above

7. **BILL_AMT1 to BILL_AMT6:** Amount of bill statement for the past 6 months. Represents the credit card bill amounts for each month.
8. **PAY_AMT1 to PAY_AMT6:** Amount of previous payment for the past 6 months. Represents the amount paid towards the credit card bill each month.
9. **Target Variable (default_payment_next_month):** Indicates whether the cardholder defaulted on payment in the next month.
   - 1 = Default
   - 0 = No Default

3. Explanation of the AI/ML Algorithm Used

Logistic Regression

- Logistic Regression is a classification algorithm used to model the probability of a binary outcome based on one or more predictor variables. The key steps in Logistic Regression include:

1. **Logistic Function:** The model applies the logistic (sigmoid) function to predict probabilities: σ(z)=11+e−z\sigma(z) = \frac{1}{1 + e^{-z}}σ(z)=1+e−z1 where z=wTx+bz = \mathbf{w}^T \mathbf{x} + bz=wTx+b.
2. **Decision Boundary:** The model classifies an instance as default (1) if the predicted probability is greater than or equal to 0.5, otherwise as no default (0).
3. **Training:** The model learns optimal weights and bias by minimizing the binary cross-entropy loss function: J(w,b)=−1m∑i=1m[y(i)log(hw(x(i)))+(1−y(i))log(1−hw(x(i)))]J(\mathbf{w}, b) = -\frac{1}{m} \sum_{i=1}^{m} [y^{(i)} \log(h_{\mathbf{w}}(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_{\mathbf{w}}(x^{(i)}))]J(w,b)=−m1∑i=1m[y(i)log(hw(x(i)))+(1−y(i))log(1−hw(x(i)))]
4. **Evaluation Metrics:** The performance is assessed using accuracy, precision and other evaluation metrics.

4. Suitable Plots to Show How the AI/ML Algorithm is Applied

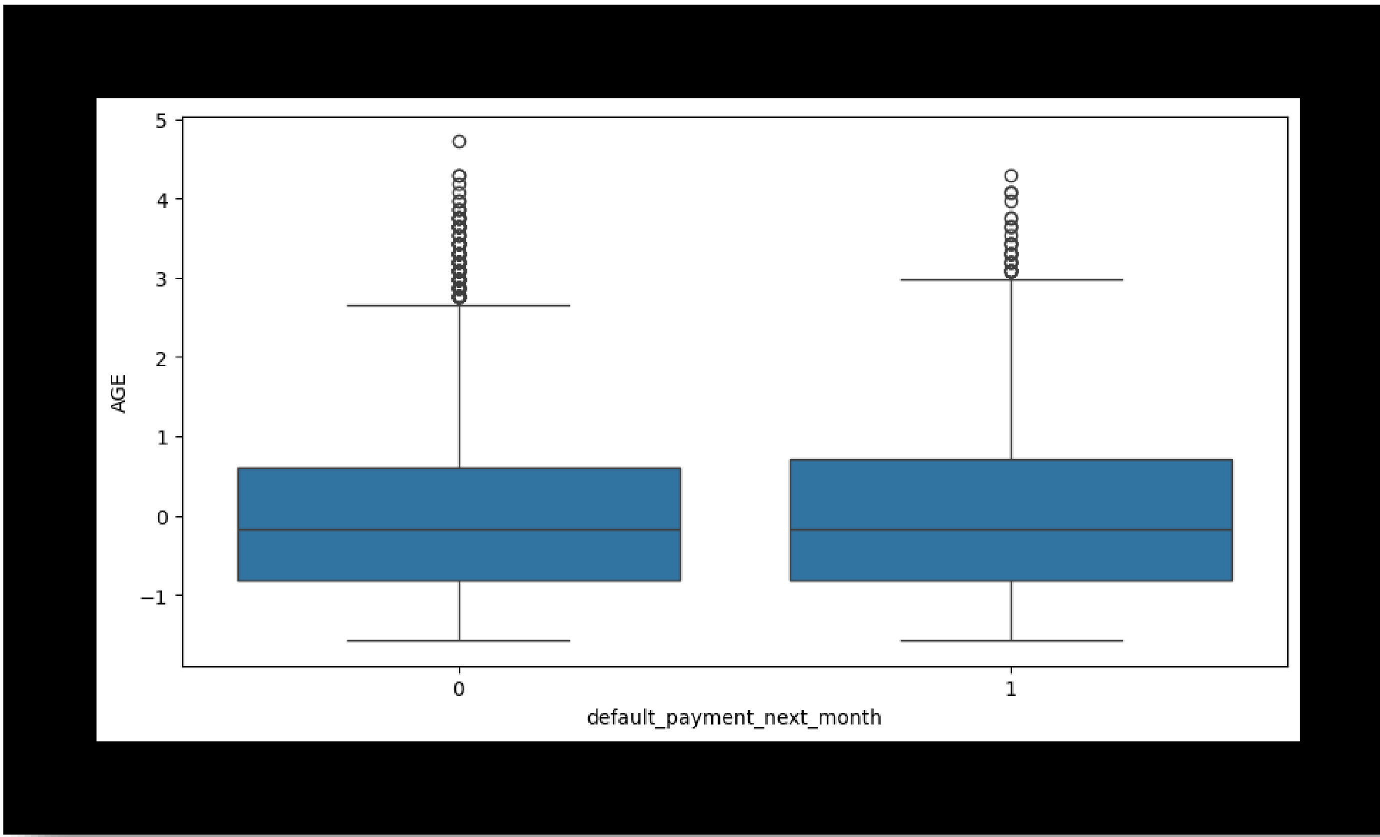
4.1. Box Plots

- Box plots can be used to visualize the distribution of numerical features across the default and no-default classes. This helps in understanding feature variation between the two classes.

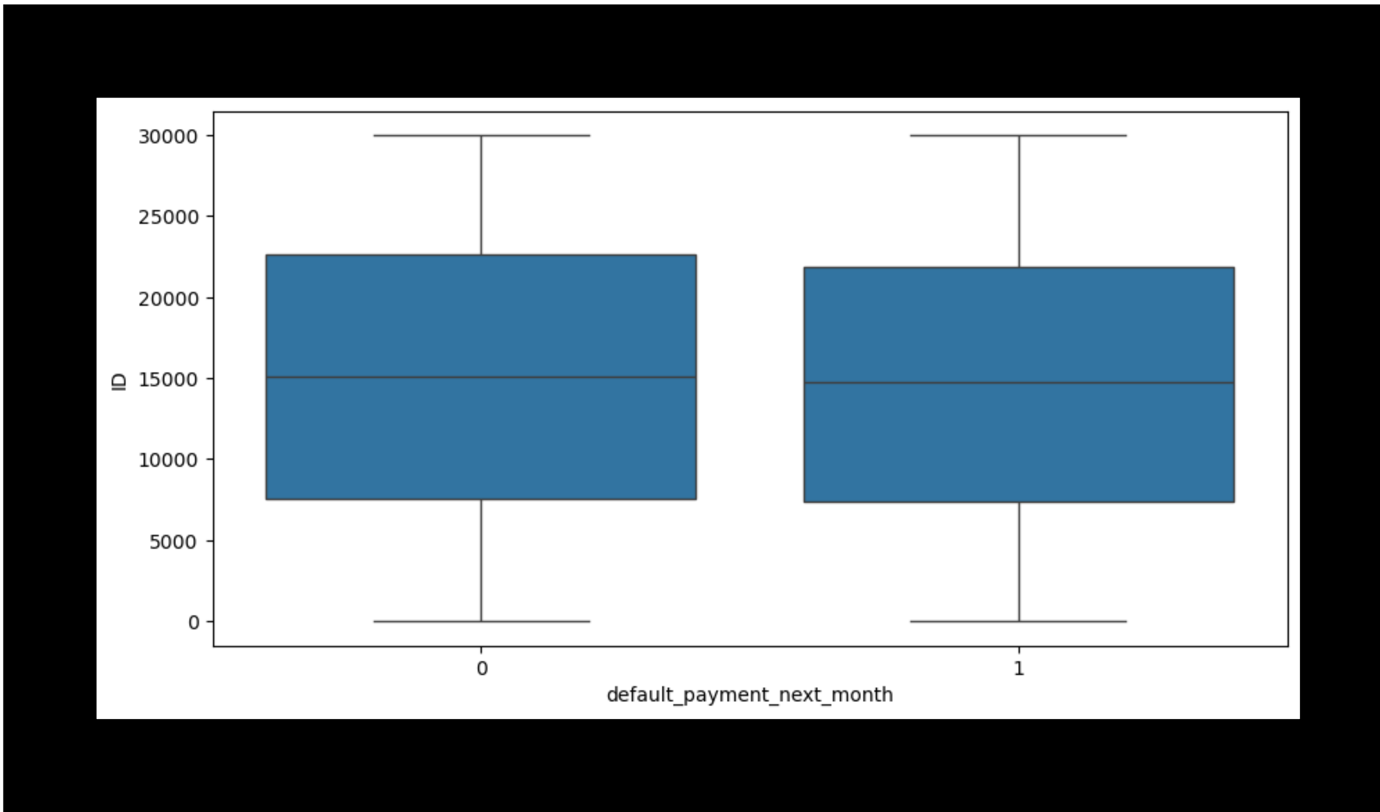
python

```
for column in X.columns:
plt.figure(figsize=(10, 5))
sns.boxplot(x=y, y=column, data=X)
plt.title(f'Box Plot of {column}')
plt.show()
```

Results:



5
4
3
2
1
0
−1
AGE
0
1
default_payment_next_month



30000
25000
20000
15000
10000
5000
0
ID
0
1
default_payment_next_month

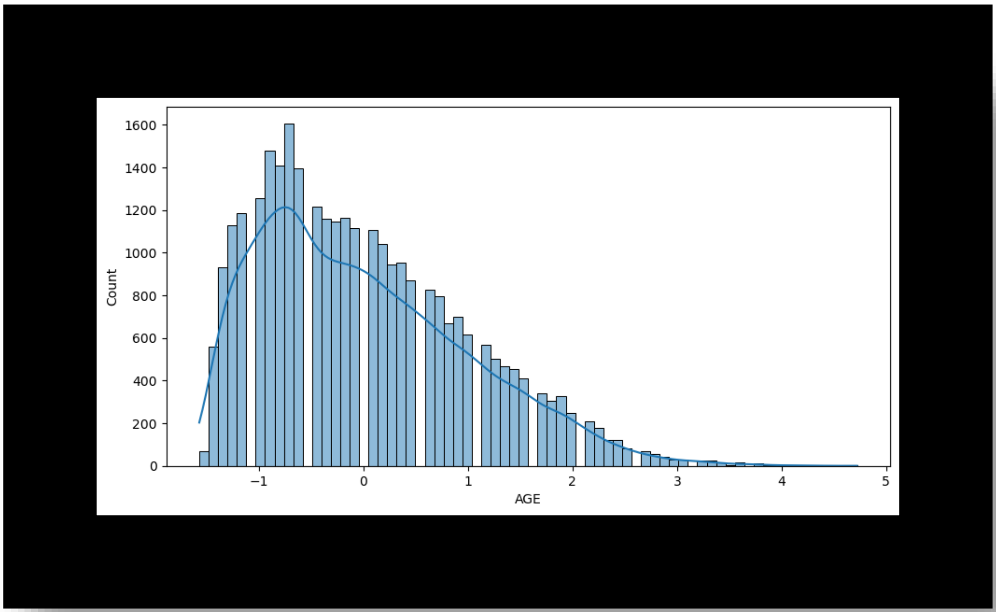
4.2. Histograms

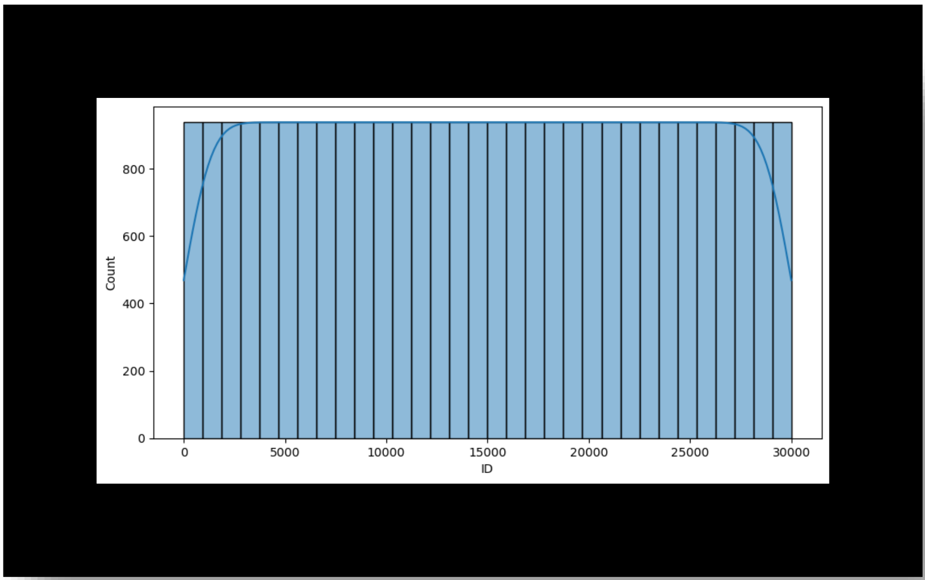
- Histograms show the distribution of individual features across the dataset, highlighting the spread and skewness of the data.

python

- `for column in X.columns:`
- `plt.figure(figsize=(10, 5))`
- `sns.histplot(X[column], kde=True)`
- `plt.show()`

Results:





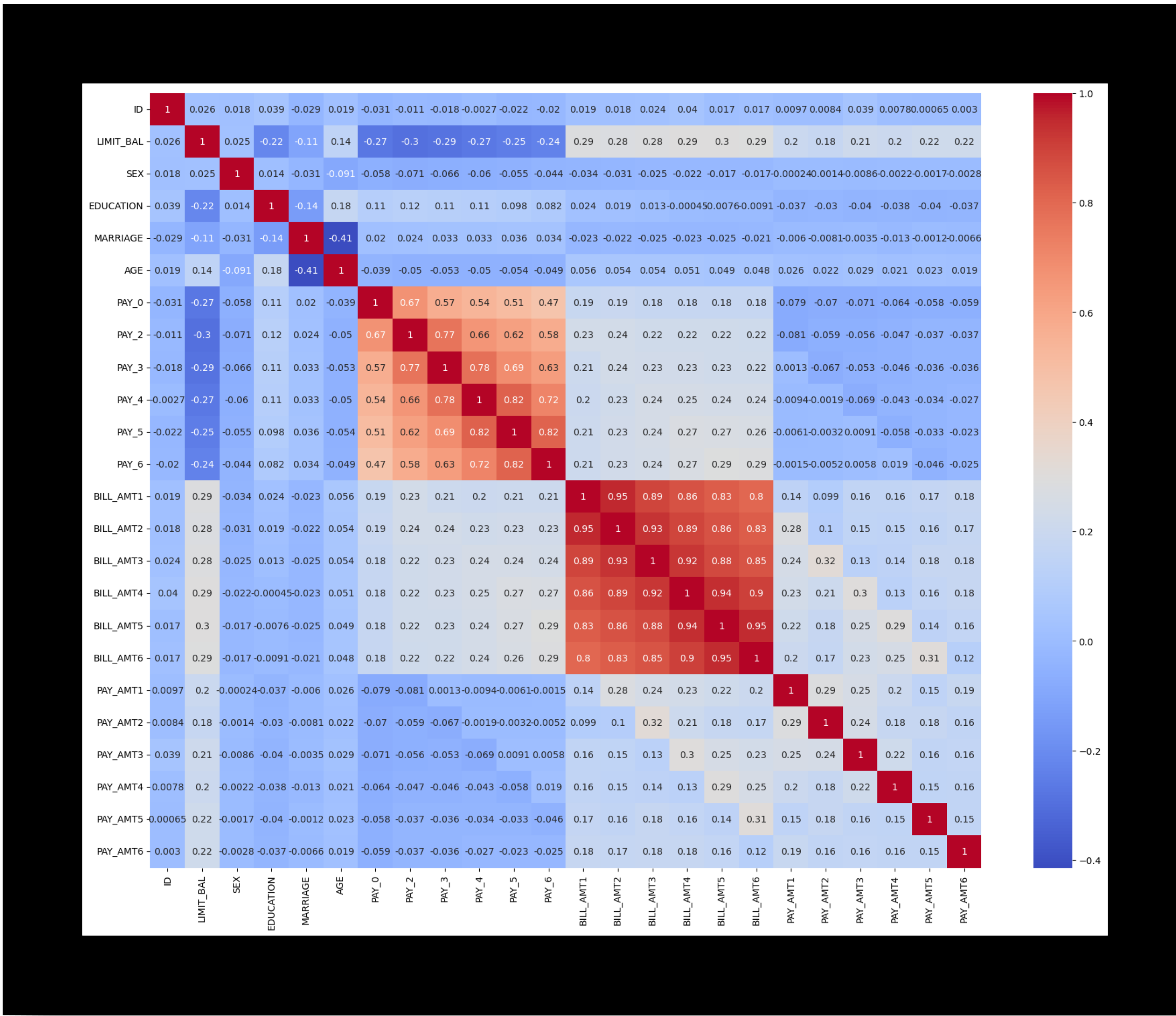
4.3. Correlation Matrix Heatmap

- A heatmap of the correlation matrix shows the relationships between numerical features, indicating how features are related to one another.

python

- `plt.figure(figsize=(20, 15))`
- `sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')`
- `plt.show()`

Result:



4.4. ROC Curve

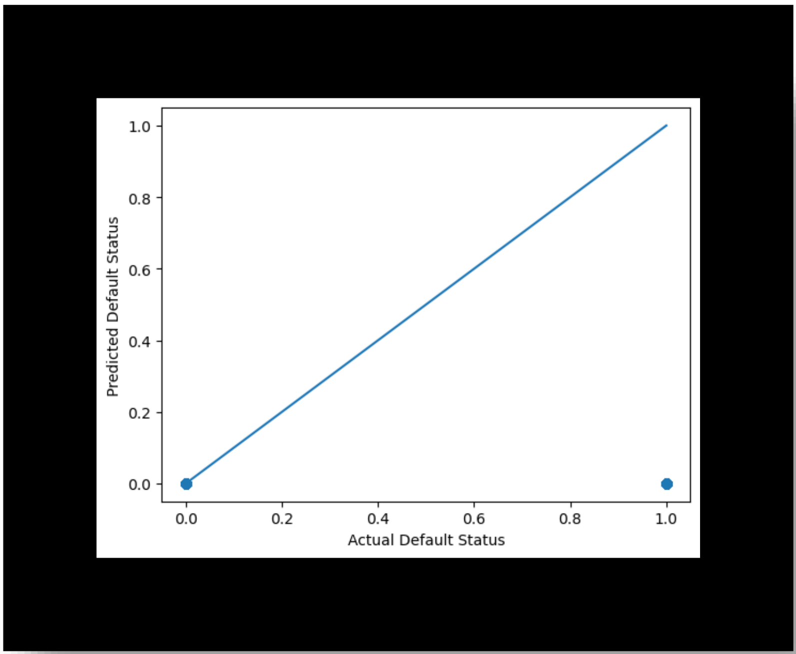
- The ROC Curve shows the evaluation of train and test data in scatterplots.

python

- # Evaluation of Training data predciton vs actual training data in a sactterplot

    - plt.scatter(y_train, y_pred_train)
    - plt.xlabel('Actual Default Status')
    - plt.ylabel('Predicted Default Status')
- # add a logistic line
    - fpr, tpr, thresholds = roc_curve(y_train, y_pred_train)
    - plt.plot(fpr, tpr)
    - plt.show()

Result:

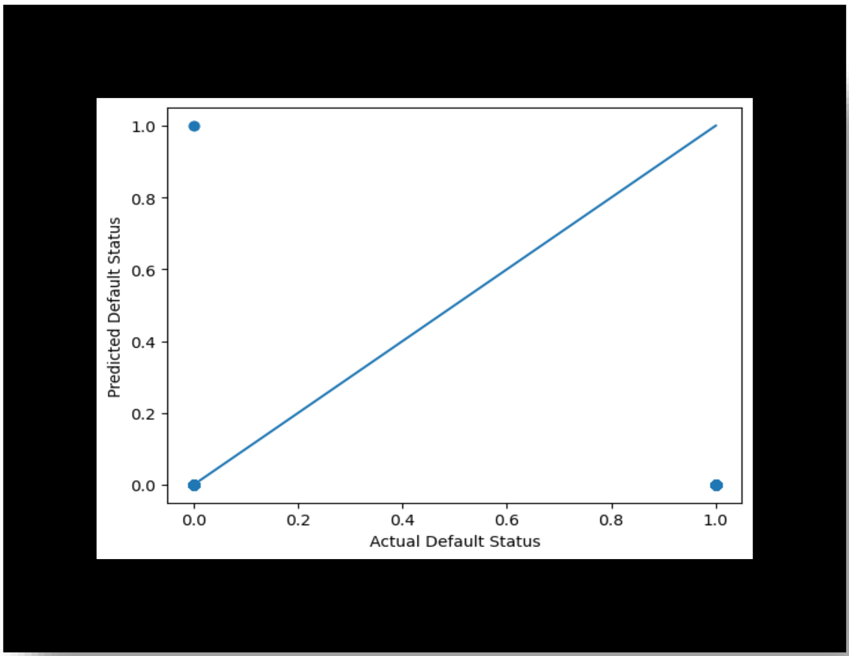


Scatter plot actual test data vs predicted data

- plt.scatter(y_test, y_pred_test)
- plt.xlabel('Actual Default Status')
- plt.ylabel('Predicted Default Status')
- # add a suitable line

- fpr, tpr, thresholds = roc_curve(y_test, y_pred_test)
- plt.plot(fpr, tpr)
- plt.show()

Result:



5. Suitable Analytics of the Data

5.1. Descriptive Statistics

python

- `df.describe().T`

Result:

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

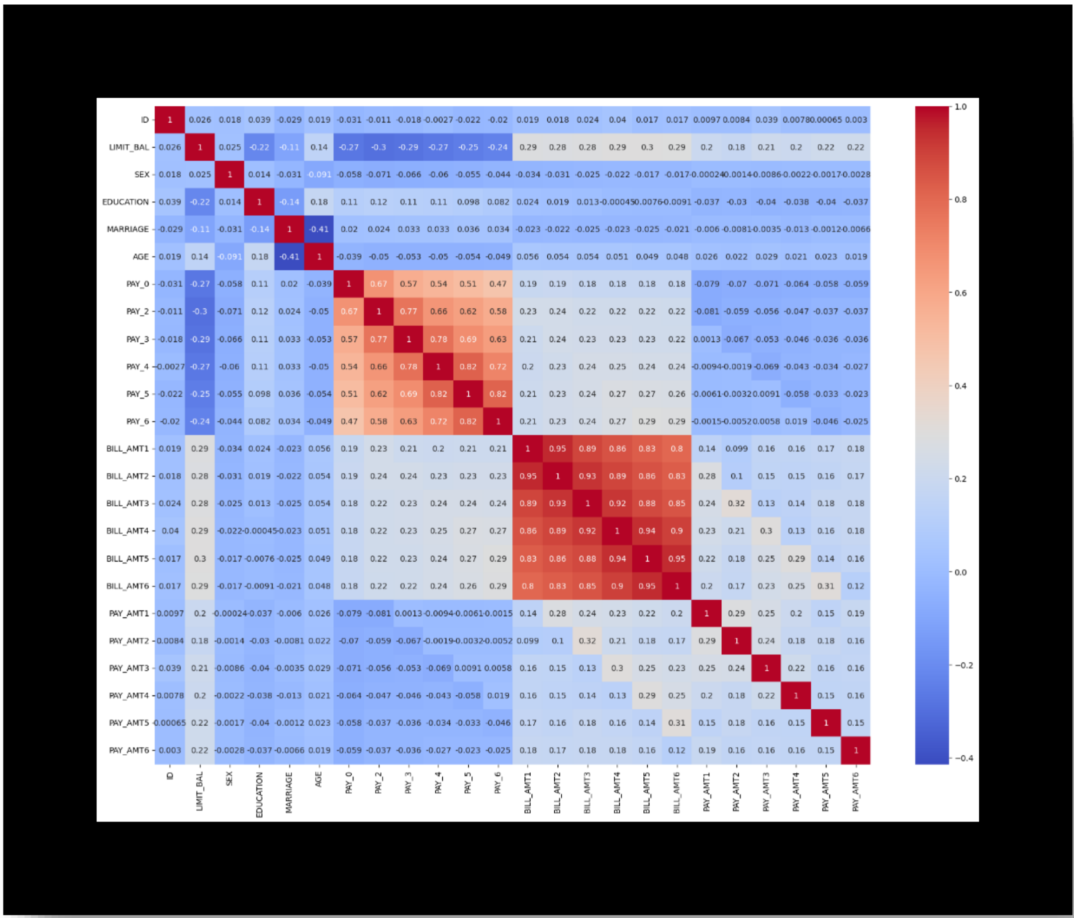
5.2. Feature Correlation

Analyze the correlation between features to identify which features are strongly related to each other.

```
python

# Correlation Matrix

    o  correlation_matrix = X.corr()
```

```
    o   correlation_matrix

# Heatmap

    o   plt.figure(figsize=(20, 15))
    o   sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
    o   plt.show()
```

Result:



6. Evaluation of the Model

6.1. Training Accuracy

Measure the accuracy of the model on the training dataset.

```
python

    o   accuracy_train = accuracy_score(y_train, y_pred_train)
    o   accuracy_train
```

Result:

- `0.778125`

6.2. Testing Accuracy

Measure the accuracy of the model on the testing dataset.

python

```
o accuracy_test = accuracy_score(y_test, y_pred_test)
o accuracy_test
```

Result:

- `0.7811666666666667`

6.3 Prediction Example

Provide an example prediction and its interpretation.

python

```
o y_pred_full_case = model.predict([full_x.values.tolist()])
o y_pred_full_case

o # Display the prediction

o if y_pred_full_case[0] == 0:
o print('The customer is not likely to default.')
o else:
o print('The customer is likely to default.')
```

Result:

- The customer is not likely to default.

7. Tools used :

- VS code for code and debugging
- Data Wrangler for viewing the dataset and gaining key insights from it
- Chat GPT for the layout used in this report
- Tabnine AI code completion for assistance in structuring the coding and assisting in debugging
- Dataset Source
- Link :
  - https://archive.ics.uci.edu/dataset/350/default+of+credit+card+clients
- Phyton Libraries used:
  - import pandas as pd
  - import matplotlib.pyplot as plt

- import seaborn as sns
- from sklearn.preprocessing import LabelEncoder
- from sklearn.model_selection import train_test_split
- from sklearn.linear_model import LogisticRegression
- from sklearn.metrics import accuracy_score
- from sklearn.preprocessing import StandardScaler
- from sklearn.metrics import roc_curve
- from sklearn.metrics import accuracy_score

8. References :

Books

1. **"Pattern Recognition and Machine Learning"** by Christopher M. Bishop
   - This book provides a thorough introduction to various machine learning algorithms, including Logistic Regression, with mathematical foundations and practical applications.
2. **"Machine Learning: A Probabilistic Perspective"** by Kevin P. Murphy
   - Murphy's book offers detailed explanations of machine learning techniques, including Logistic Regression, with emphasis on probabilistic models.
3. **"Introduction to Statistical Learning: with Applications in R"** by Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani
   - This book is a great resource for understanding statistical learning methods, including Logistic Regression, with practical examples and applications.

Online Courses & Tutorials

1. **Coursera – "Machine Learning" by Andrew Ng**
   - This popular course provides an accessible introduction to various machine learning algorithms, including Logistic Regression. It is available for free and covers both theoretical and practical aspects.
2. **Khan Academy – "Logistic Regression"**
   - Khan Academy offers a video tutorial that covers the basics of Logistic Regression and its application.
3. **edX – "Introduction to Machine Learning with Python"**
   - This course offers a practical introduction to machine learning using Python, including Logistic Regression.

Research Papers

1. **"Logistic Regression: An Overview"** by David W. Hosmer Jr., Stanley Lemeshow, and Rodney X. Sturdivant
    - A foundational paper that provides an in-depth review of Logistic Regression, including its theoretical aspects and applications.
2. **"The Use of Logistic Regression in Epidemiology: A Review"** by T. L. M. B. Croghan
    - This paper discusses the application of Logistic Regression in epidemiology, providing context on its use in different fields.