ThottiyamVenkatakrishnan530Project

March 3, 2023

[41]:
```python
import pandas as pd
import matplotlib.pyplot as plt
import scipy.stats as stats
import seaborn as sns
import pandas as pd
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import␣
↪accuracy_score,classification_report,confusion_matrix,mean_squared_error
import numpy as np
import statsmodels.api as sm

# Load dataset
df = pd.read_csv('https://raw.githubusercontent.com/uberdatascientist/dsc520/
↪master/UCI_Credit_Card.csv')

# Select variables
var_list = ['LIMIT_BAL', 'SEX', 'EDUCATION', 'MARRIAGE', 'PAY_0']
df_vars = df[var_list]

# Create histograms
for var in var_list:
    plt.hist(df_vars[var], bins=20)
    plt.title(var)
    plt.show()

# Create boxplots
for var in ['LIMIT_BAL']:
    plt.boxplot(df_vars[var])
    plt.title(var)
    plt.show()

# Calculate descriptive statistics
for var in var_list:
    print(var)
    print('Mean:', df_vars[var].mean())
    print('Mode:', df_vars[var].mode().values)
```

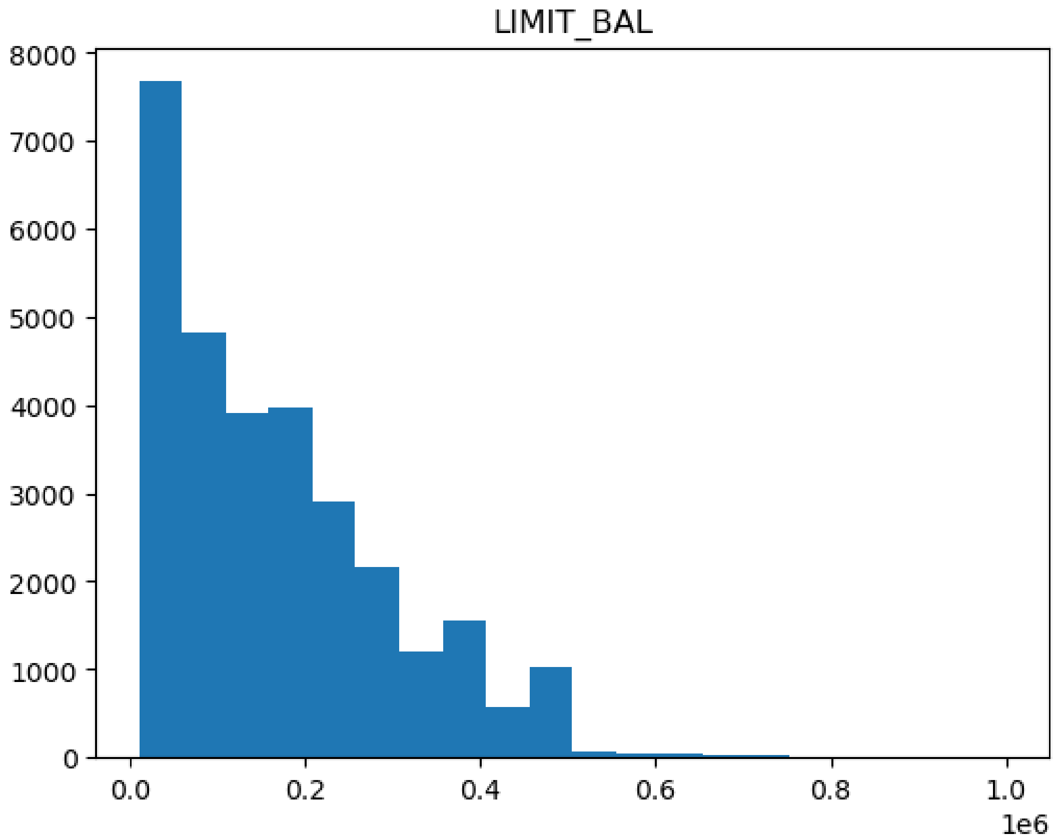
```
    print('Spread:', df_vars[var].max() - df_vars[var].min())
    print('Left tail:', len(df_vars[df_vars[var] < df_vars[var].mean()]) /
 ↪len(df_vars))
    print('Right tail:', len(df_vars[df_vars[var] > df_vars[var].mean()]) /
 ↪len(df_vars))
    print()

# identify outliers in LIMIT_BAL
q1, q3 = df['LIMIT_BAL'].quantile([0.25, 0.75])
iqr = q3 - q1
lower_bound = q1 - 1.5*iqr
upper_bound = q3 + 1.5*iqr
outliers = df[(df['LIMIT_BAL'] < lower_bound) | (df['LIMIT_BAL'] > upper_bound)]
print(outliers)

# identify outliers in BILL_AMT1
q1, q3 = df['BILL_AMT1'].quantile([0.25, 0.75])
iqr = q3 - q1
lower_bound = q1 - 1.5*iqr
upper_bound = q3 + 1.5*iqr
outliers = df[(df['BILL_AMT1'] < lower_bound) | (df['BILL_AMT1'] > upper_bound)]
print(outliers)

# identify outliers in PAY_AMT1
q1, q3 = df['PAY_AMT1'].quantile([0.25, 0.75])
iqr = q3 - q1
lower_bound = q1 - 1.5*iqr
upper_bound = q3 + 1.5*iqr
outliers = df[(df['PAY_AMT1'] < lower_bound) | (df['PAY_AMT1'] > upper_bound)]
print(outliers)


# Filter data for two scenarios
scenario1 = df.loc[df['EDUCATION'] == 1, 'default.payment.next.month']
scenario2 = df.loc[df['EDUCATION'] == 5, 'default.payment.next.month']

# Create PMFs for two scenarios
pmf1 = scenario1.value_counts(normalize=True).sort_index()
pmf2 = scenario2.value_counts(normalize=True).sort_index()


# Plot PMFs for two scenarios
fig, ax = plt.subplots()
ax.bar(pmf1.index, pmf1.values, label="Graduate School")
ax.bar(pmf2.index, pmf2.values, alpha=0.5, label="Unknown")
ax.set_xlabel("Default Payment")
ax.set_ylabel("PMF")
```

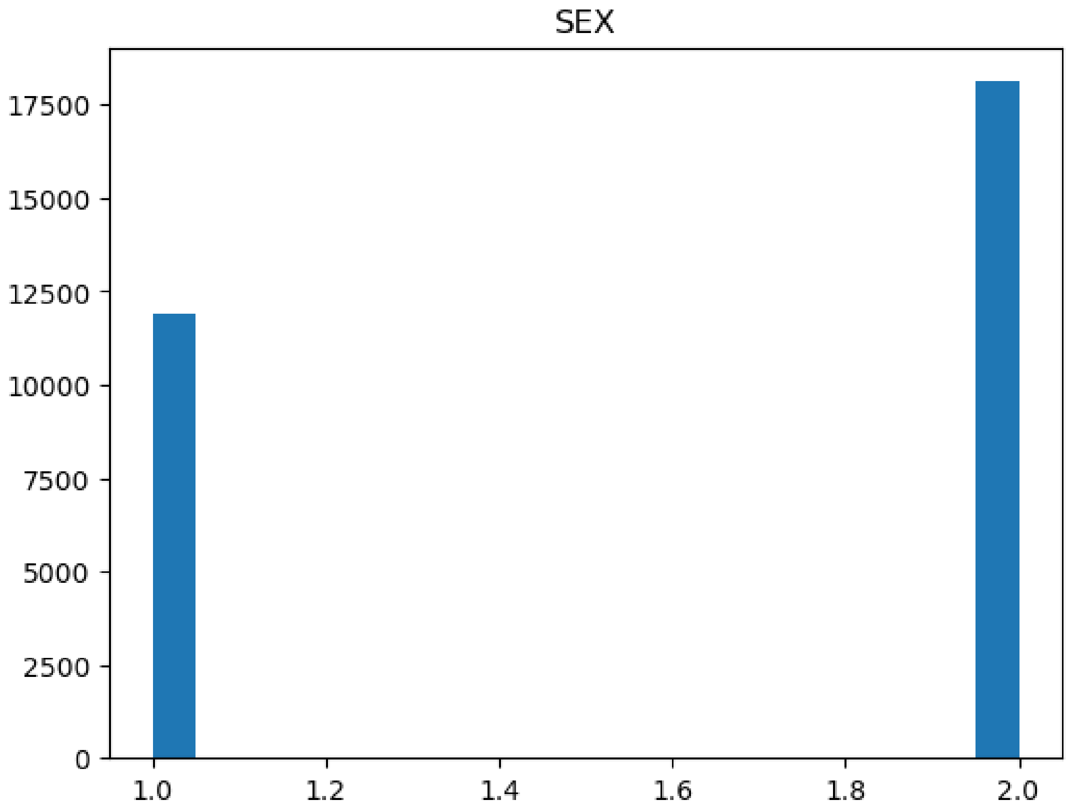
```
ax.legend()
plt.show()

# Filter data for two scenarios based on repayment status pay_0
scenario1 = df.loc[df['PAY_0'] == -1, 'default.payment.next.month']
scenario2 = df.loc[df['PAY_0'] == 9, 'default.payment.next.month']

# Create PMFs for two scenarios
pmf1 = scenario1.value_counts(normalize=True).sort_index()
pmf2 = scenario2.value_counts(normalize=True).sort_index()


# Plot PMFs for two scenarios
fig, ax = plt.subplots()
ax.bar(pmf1.index, pmf1.values, label="Duly Paid")
ax.bar(pmf2.index, pmf2.values, alpha=0.5, label="Payment delay for nine␣
 ↪months")
ax.set_xlabel("Default Payment")
ax.set_ylabel("PMF")
ax.legend()
plt.show()

# Filter data for two scenarios based on repayment status pay_0
scenario1 = df.loc[df['PAY_0'] == -1, 'default.payment.next.month']
scenario2 = df.loc[df['PAY_0'] == 1, 'default.payment.next.month']

# Create PMFs for two scenarios
pmf1 = scenario1.value_counts(normalize=True).sort_index()
pmf2 = scenario2.value_counts(normalize=True).sort_index()


# Plot PMFs for two scenarios
fig, ax = plt.subplots()
ax.bar(pmf1.index, pmf1.values, label="Duly Paid")
ax.bar(pmf2.index, pmf2.values, alpha=0.5, label="Payment delay for one month")
ax.set_xlabel("Default Payment")
ax.set_ylabel("PMF")
ax.legend()
plt.show()


# Create CDF for age variable
ages = df['AGE']
cdf = np.cumsum(np.ones_like(ages)) / len(ages)

# Plot CDF
fig, ax = plt.subplots()
```

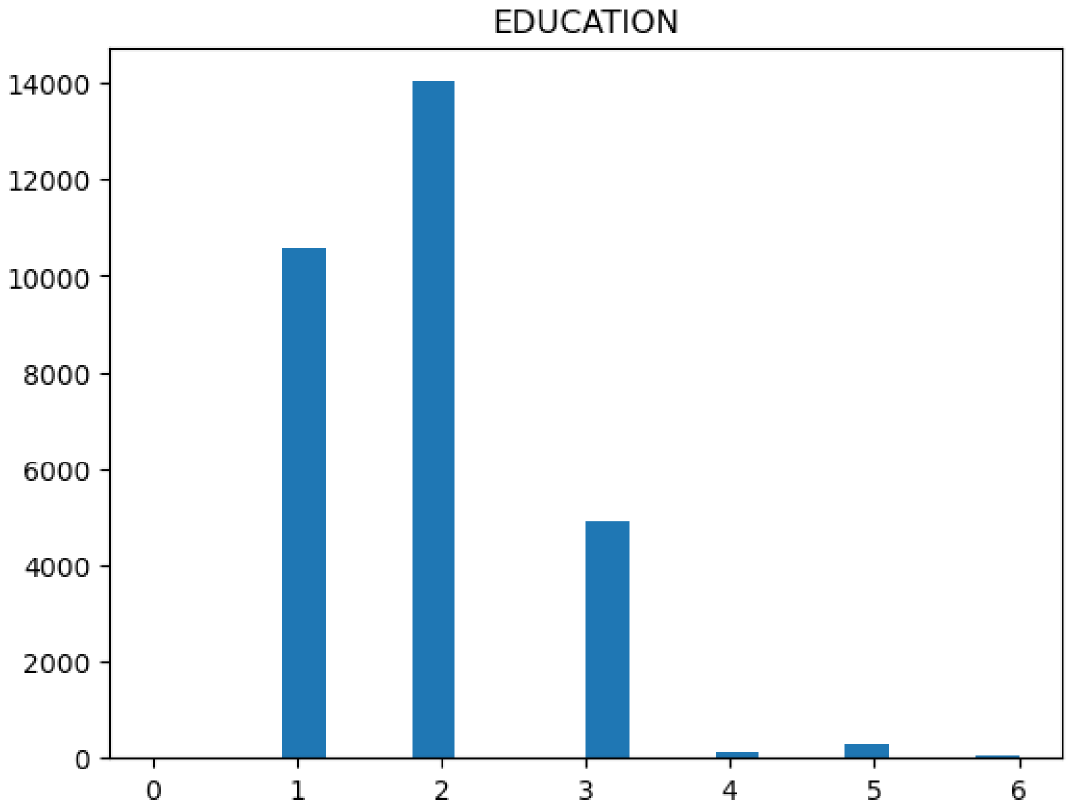
```python
ax.plot(np.sort(ages), cdf)
ax.set_xlabel("Age")
ax.set_ylabel("CDF")
plt.show()


# Create normal distribution with mean and std of BILL_AMT1 variable
bill_mean = df['BILL_AMT1'].mean()
bill_std = df['BILL_AMT1'].std()
norm_dist = stats.norm(bill_mean, bill_std)

# Plot histogram of BILL_AMT1 variable with normal distribution overlay
fig, ax = plt.subplots()
ax.hist(df['BILL_AMT1'], bins=50, density=True, label="BILL_AMT1")
ax.plot(np.linspace(bill_mean - 4 * bill_std, bill_mean + 4 * bill_std, 100),
        norm_dist.pdf(np.linspace(bill_mean - 4 * bill_std, bill_mean + 4 *␣
 ↪bill_std, 100)),
        label="Normal Distribution")
ax.set_xlabel("BILL_AMT1")
ax.set_ylabel("PDF")
ax.legend()
plt.show()

# Filter the data for males and females
male_education = df.loc[df.SEX == 1, 'EDUCATION']
female_education = df.loc[df.SEX == 2, 'EDUCATION']

# Compute PMFs for males and females
male_pmf = male_education.value_counts(normalize=True).sort_index()
female_pmf = female_education.value_counts(normalize=True).sort_index()


# Scatter plot of LIMIT_BAL and AGE
sns.scatterplot(x="AGE", y="LIMIT_BAL", data=df)
plt.title('Scatter plot of LIMIT_BAL vs AGE')
plt.show()

# Scatter plot of LIMIT_BAL and BILL_AMT1
sns.scatterplot(x="BILL_AMT1", y="LIMIT_BAL", data=df)
plt.title('Scatter plot of LIMIT_BAL vs BILL_AMT1')
plt.show()

# Check collinearity for the independent variables
pay_cols = ['PAY_0', 'PAY_2', 'PAY_3', 'PAY_4', 'PAY_5']
pay_corr = df[pay_cols].corr()

print(pay_corr)
```

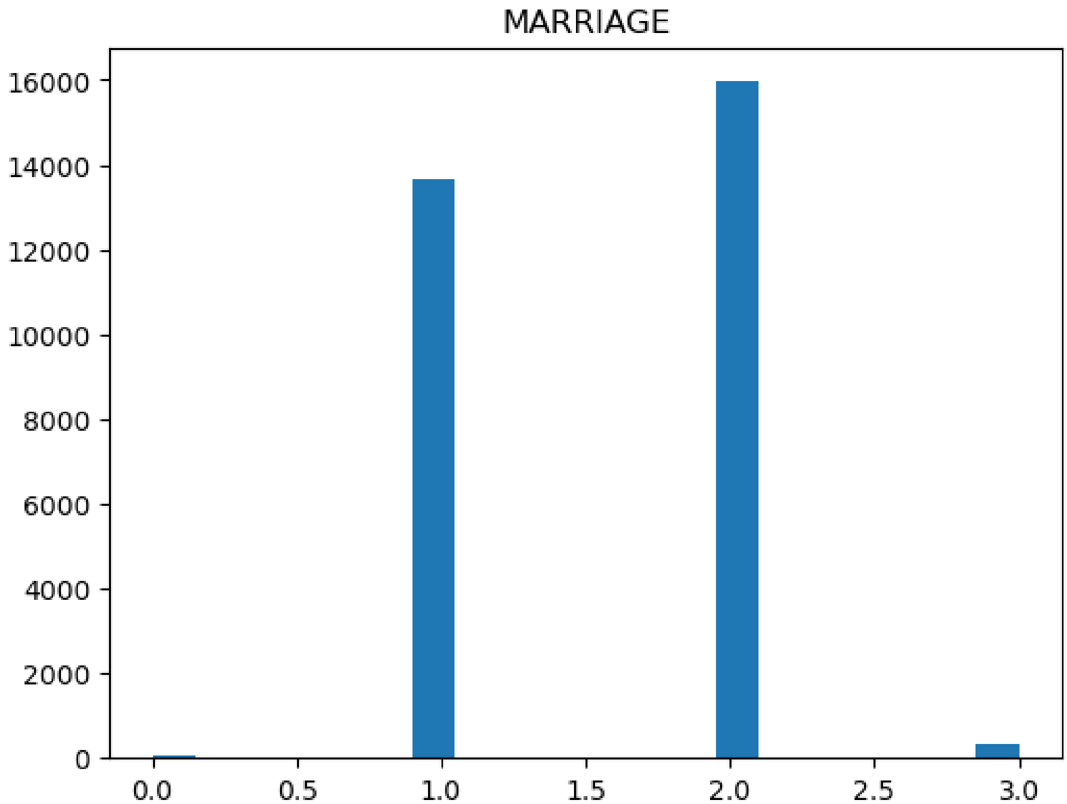
```
corr_matrix = df[['PAY_0', 'PAY_2', 'PAY_3', 'PAY_4', 'PAY_5', 'BILL_AMT1']].
 ↪corr()

# print the correlation matrix
print(corr_matrix)

# plot the heatmap
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.show()


# Select the categorical features to be converted
cat_features = ['SEX', 'EDUCATION', 'MARRIAGE', 'PAY_0', 'PAY_2', 'PAY_3',␣
 ↪'PAY_4', 'PAY_5', 'PAY_6']

# Convert the categorical features to the 'category' data type
df[cat_features] = df[cat_features].astype('category')

# Check the data types of the features after conversion
print(df.dtypes)


# Fit the regression model
X = sm.add_constant(df['AGE'])
model = sm.OLS(df['default.payment.next.month'], X).fit()

# Print the model summary
print(model.summary())

# Select the independent variables
X = df[['LIMIT_BAL', 'SEX', 'EDUCATION', 'MARRIAGE', 'AGE', 'PAY_0', 'PAY_2',␣
 ↪'PAY_3', 'PAY_4', 'PAY_5']]

# Select the dependent variable
y = df['default.payment.next.month']

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,␣
 ↪random_state=42)

# Create the logistic regression model
model = LogisticRegression()

# Train the model
model.fit(X_train, y_train)
```

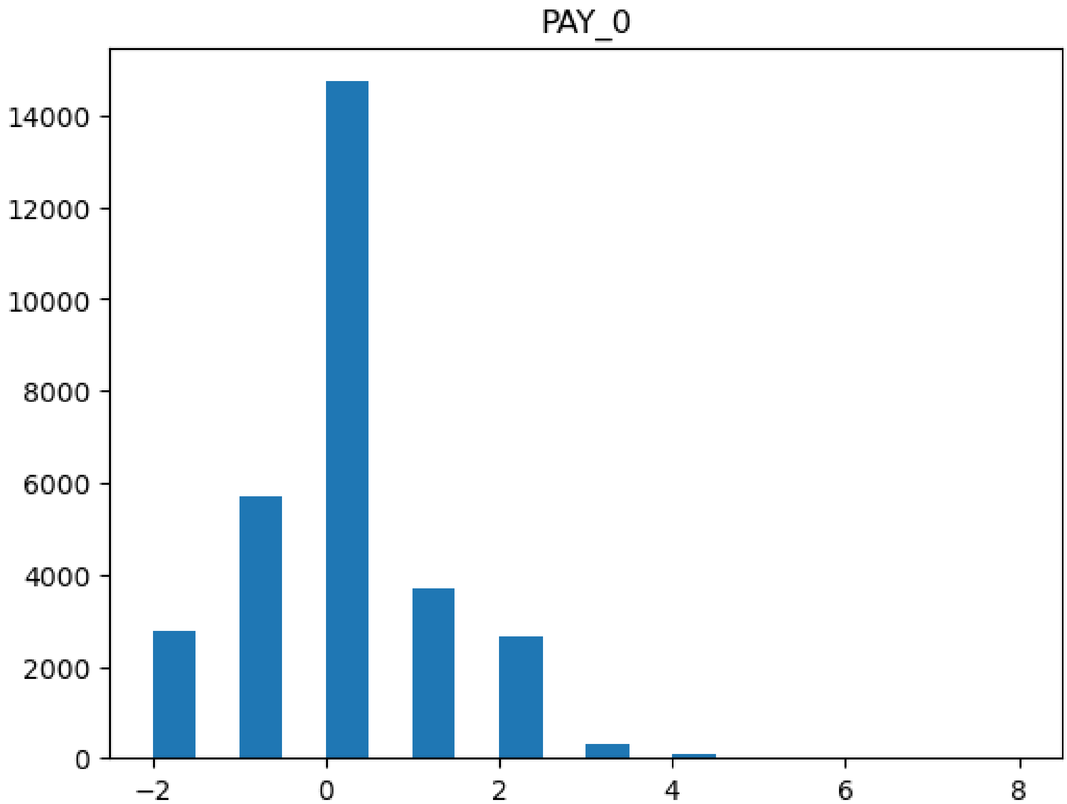
```
# Make predictions on the testing set
y_pred = model.predict(X_test)

# Calculate the accuracy score of the model
accuracy = accuracy_score(y_test, y_pred)

# Print the accuracy score
print("Accuracy:", accuracy)

# Select the categorical features to be converted
cat_features = ['SEX', 'EDUCATION', 'MARRIAGE', 'PAY_0', 'PAY_2', 'PAY_3',␣
↪'PAY_4', 'PAY_5', 'PAY_6','default.payment.next.month']

# Convert the categorical features to the 'category' data type
df[cat_features] = df[cat_features].astype('category')


# Create predictor and explanatory variable dataframes
X = df[['SEX','MARRIAGE','AGE','BILL_AMT1','EDUCATION','PAY_0']]
y = df['default.payment.next.month']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,␣
↪random_state=20)

# Create logistic regression model
logreg = LogisticRegression()
logreg.fit(X_train, y_train)

zero_division = 1

y_pred = logreg.predict(X_test)
y_train_pred = logreg.predict(X_train)

print(classification_report(y_pred, y_test, zero_division = 1))
print(confusion_matrix(y_pred, y_test))
print('\nTest Accuracy Score for Logistic Regression: ',␣
↪accuracy_score(y_pred,y_test))
print('\nTrain Accuracy Score for Logistic Regression: ',␣
↪accuracy_score(y_train_pred,y_train))

# Perform cross-validation with 5 folds
cv_scores = cross_val_score(logreg, X, y, cv=5, scoring='roc_auc')

# Print cross-validation scores
print("Cross-validated AUC: %0.2f (+/- %0.2f)" % (cv_scores.mean(), cv_scores.
↪std() * 2))
```

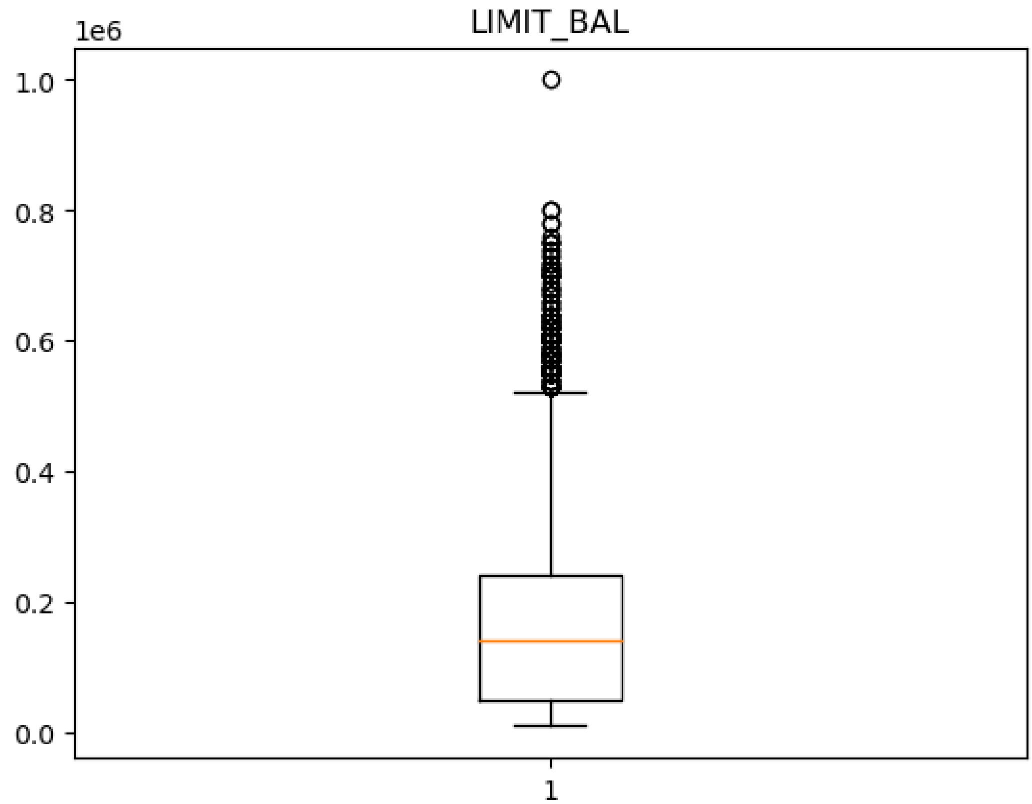

LIMIT_BAL
8000
7000
6000
5000
4000
3000
2000
1000
0
0.0
0.2
0.4
0.6
0.8
1.0
1e6


SEX
17500
15000
12500
10000
7500
5000
2500
0
1.0
1.2
1.4
1.6
1.8
2.0


EDUCATION
14000
12000
10000
8000
6000
4000
2000
0
0
1
2
3
4
5
6


MARRIAGE
16000
14000
12000
10000
8000
6000
4000
2000
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0


PAY_0
14000
12000
10000
8000
6000
4000
2000
0
−2
0
2
4
6
8

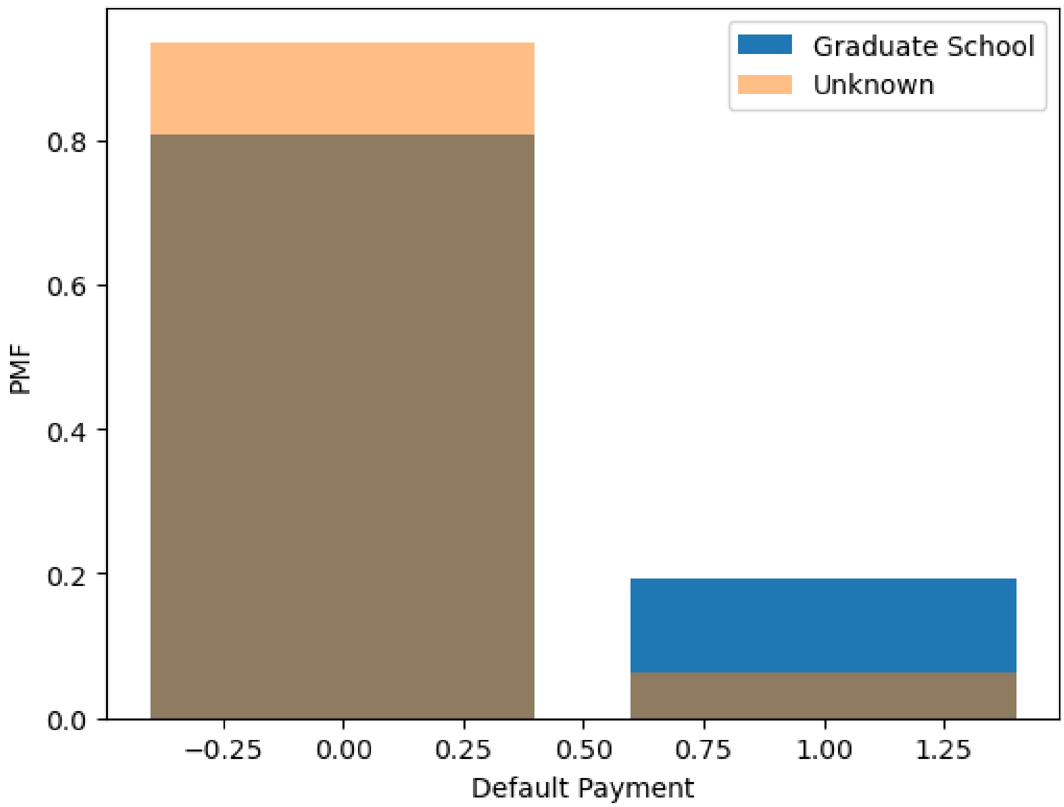


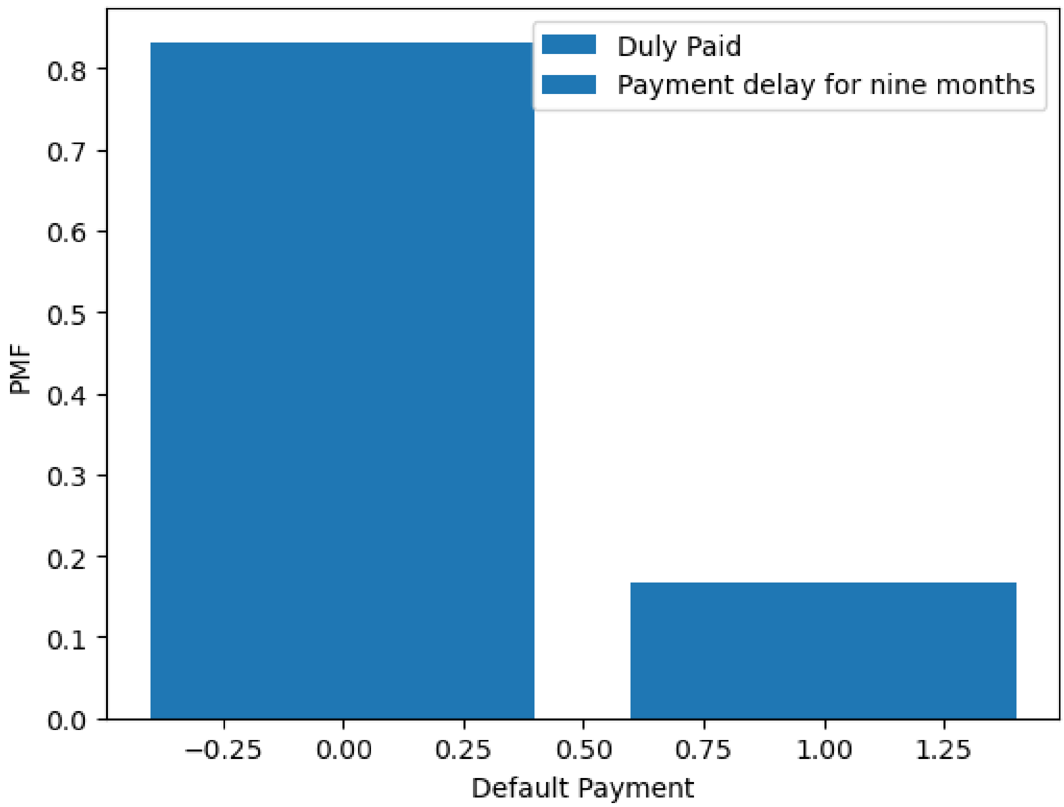

```
LIMIT_BAL
Mean: 167484.32266666667
Mode: [50000.]
Spread: 990000.0
Left tail: 0.5698
Right tail: 0.4302

SEX
Mean: 1.6037333333333332
Mode: [2]
Spread: 1
Left tail: 0.39626666666666666
Right tail: 0.6037333333333333

EDUCATION
Mean: 1.8531333333333333
Mode: [2]
Spread: 6
Left tail: 0.3533
Right tail: 0.6467
```

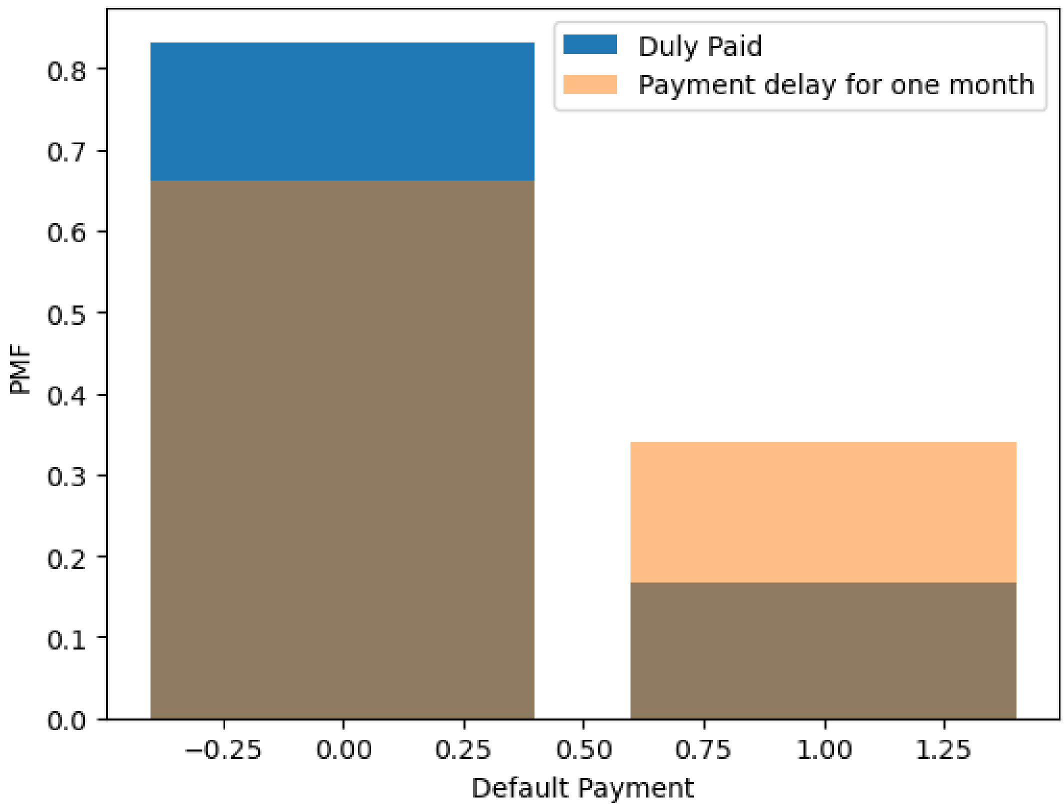
```
MARRIAGE
Mean: 1.5518666666666667
Mode: [2]
Spread: 3
Left tail: 0.4571
Right tail: 0.5429

PAY_0
Mean: -0.0167
Mode: [0]
Spread: 10
Left tail: 0.2815
Right tail: 0.7185

          ID  LIMIT_BAL  SEX  EDUCATION  MARRIAGE  AGE  PAY_0  PAY_2  PAY_3  \
12        13   630000.0    2          2         2   41     -1      0     -1
433      434   580000.0    2          1         1   36      0      0      0
451      452   600000.0    1          1         1   53      2      2      0
527      528   620000.0    2          2         1   45      2      2      0
555      556   630000.0    2          2         1   47      0      0      0
…        …          …   …          …        …  …    …     …     …
29571  29572   570000.0    1          1         2   33      0      0      0
29740  29741   620000.0    1          2         2   31     -2     -2     -2
29861  29862   650000.0    1          1         1   44     -2     -2     -2
29886  29887   630000.0    1          2         1   46      0      0      0
29963  29964   610000.0    1          1         2   31      0     -1      2

       PAY_4  …  BILL_AMT4  BILL_AMT5  BILL_AMT6  PAY_AMT1  PAY_AMT2  \
12        -1  …     6500.0     6500.0     2870.0    1000.0    6500.0
433        0  …   169365.0   168755.0   167964.0    6422.0    6565.0
451        0  …   447130.0   440982.0   434715.0       0.0   18000.0
527        0  …   163781.0   167159.0   170894.0       0.0    6200.0
555       -1  …     2632.0     8654.0        0.0   38187.0    1207.0
…        …  …        …         …         …         …         …
29571      0  …   266800.0        0.0        0.0    9083.0   11472.0
29740     -2  …    13846.0     3565.0     7076.0   11881.0   21171.0
29861     -2  …     7139.0     1034.0     2127.0    5115.0    5180.0
29886      0  …   146005.0   146207.0   106467.0    3416.0    4300.0
29963     -1  …   347303.0   248893.0   269528.0  323014.0    1605.0

       PAY_AMT3  PAY_AMT4  PAY_AMT5  PAY_AMT6  default.payment.next.month
12       6500.0    6500.0    2870.0       0.0                           0
433      5951.0    6006.0    5894.0    5946.0                           0
451     16000.0   16000.0   21000.0   20000.0                           1
527      6000.0    6000.0    6500.0    6000.0                           1
555      2632.0    8654.0       0.0    4981.0                           0
…           …        …        …        …                         …
29571   12000.0       0.0       0.0       0.0                           0
```

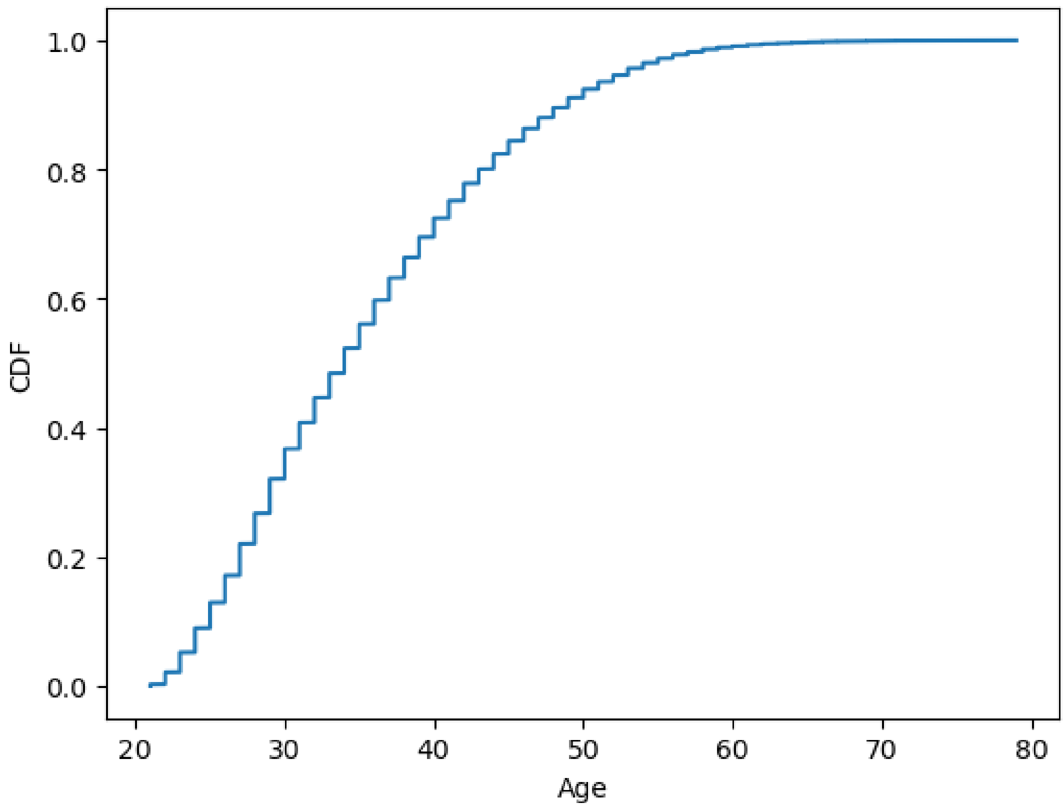
```
29740   13915.0    3583.0    7111.0    1971.0                          0
29861    7201.0    1035.0    2139.0    3463.0                          0
29886   84700.0    4211.0    4470.0    3600.0                          0
29963  349395.0  250144.0  271099.0  220076.0                          0

[167 rows x 25 columns]
          ID  LIMIT_BAL  SEX  EDUCATION  MARRIAGE  AGE  PAY_0  PAY_2  PAY_3  \
6          7   500000.0    1          1         2   29      0      0      0
17        18   320000.0    1          1         1   49      0      0      0
36        37   280000.0    1          2         1   40      0      0      0
40        41   360000.0    1          1         2   33      0      0      0
57        58   180000.0    2          2         1   34      0      0      0
…        …          …  …        …        …  …    …    …    …
29921  29922   410000.0    1          1         2   34      0      0      0
29963  29964   610000.0    1          1         2   31      0     -1      2
29978  29979   310000.0    1          2         1   39      0      0      0
29988  29989   250000.0    1          1         1   34      0      0      0
29995  29996   220000.0    1          3         1   39      0      0      0

       PAY_4  …  BILL_AMT4  BILL_AMT5  BILL_AMT6  PAY_AMT1  PAY_AMT2  \
6          0  …   542653.0   483003.0   473944.0   55000.0   40000.0
17        -1  …    70074.0     5856.0   195599.0   10358.0   10000.0
36         0  …   170410.0   173901.0   177413.0    8026.0    8060.0
40         0  …   628699.0   195969.0   179224.0   10000.0    7000.0
57         0  …   168608.0   132202.0   129918.0    8083.0    7296.0
…        …  …        …        …        …        …       …
29921     -1  …     1467.0     1421.0      -15.0   17259.0   18600.0
29963     -1  …   347303.0   248893.0   269528.0  323014.0    1605.0
29978      0  …   219409.0   216540.0   210675.0   10029.0    9218.0
29988      0  …   245750.0   175005.0   179687.0   65000.0    8800.0
29995      0  …    88004.0    31237.0    15980.0    8500.0   20000.0

       PAY_AMT3  PAY_AMT4  PAY_AMT5  PAY_AMT6  default.payment.next.month
6       38000.0   20239.0   13750.0   13770.0                          0
17      75940.0   20000.0  195599.0   50000.0                          0
36       6300.0    6400.0    6400.0    6737.0                          0
40       6000.0  188840.0   28000.0    4000.0                          0
57       5253.0    4814.0    4816.0    3800.0                          0
…          …        …        …        …                        …
29921    1474.0    1428.0       0.0       0.0                          1
29963  349395.0  250144.0  271099.0  220076.0                          0
29978   10029.0    8049.0    8040.0   10059.0                          0
29988    9011.0    6000.0    7000.0    6009.0                          0
29995    5003.0    3047.0    5000.0    1000.0                          0

[2400 rows x 25 columns]
          ID  LIMIT_BAL  SEX  EDUCATION  MARRIAGE  AGE  PAY_0  PAY_2  PAY_3  \
6          7   500000.0    1          1         2   29      0      0      0
```

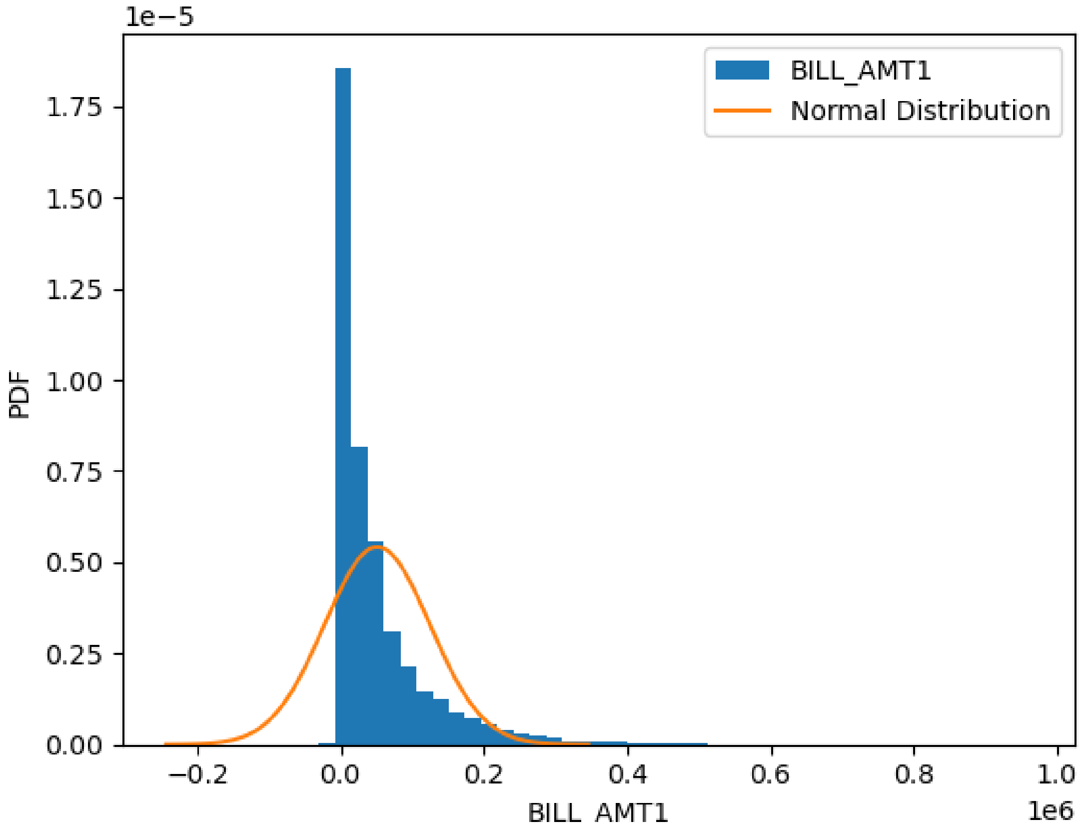
```
11        12   260000.0    2          1         2   51      -1      -1      -1
23        24   450000.0    2          1         1   40      -2      -2      -2
30        31   230000.0    2          1         2   27      -1      -1      -1
48        49   380000.0    1          2         2   32      -1      -1      -1
…      …        …  …        …       …  …     …    …    …
29948  29949   290000.0    1          1         1   32      -1      -1      -1
29963  29964   610000.0    1          1         2   31       0      -1       2
29970  29971   360000.0    1          1         1   34      -1      -1      -1
29988  29989   250000.0    1          1         1   34       0       0       0
29998  29999    80000.0    1          3         1   41       1      -1       0

       PAY_4  …  BILL_AMT4  BILL_AMT5  BILL_AMT6  PAY_AMT1  PAY_AMT2  \
6          0  …   542653.0   483003.0   473944.0   55000.0   40000.0
11        -1  …     8517.0    22287.0    13668.0   21818.0    9966.0
23        -2  …      560.0        0.0        0.0   19428.0    1473.0
30        -1  …    15339.0    14307.0    36923.0   17270.0   13281.0
48        -1  …    32018.0    11849.0    11873.0   21540.0   15138.0
…      …  …        …        …        …          …        …
29948     -1  …     1956.0      910.0        0.0   39032.0     201.0
29963     -1  …   347303.0   248893.0   269528.0  323014.0    1605.0
29970      0  …    49005.0     8676.0    19487.0   52951.0   64535.0
29988      0  …   245750.0   175005.0   179687.0   65000.0    8800.0
29998      0  …    52774.0    11855.0    48944.0   85900.0    3409.0

       PAY_AMT3  PAY_AMT4  PAY_AMT5  PAY_AMT6  default.payment.next.month
6       38000.0   20239.0   13750.0   13770.0                           0
11       8583.0   22301.0       0.0    3640.0                           0
23        560.0       0.0       0.0    1128.0                           1
30      15339.0   14307.0   37292.0       0.0                           0
48      24677.0   11851.0   11875.0    8251.0                           0
…         …        …       …       …                         …
29948    1961.0       4.0       0.0       0.0                           0
29963  349395.0  250144.0  271099.0  220076.0                           0
29970    8907.0      53.0   19584.0   16080.0                           0
29988    9011.0    6000.0    7000.0    6009.0                           0
29998    1178.0    1926.0   52964.0    1804.0                           1

[2745 rows x 25 columns]
```

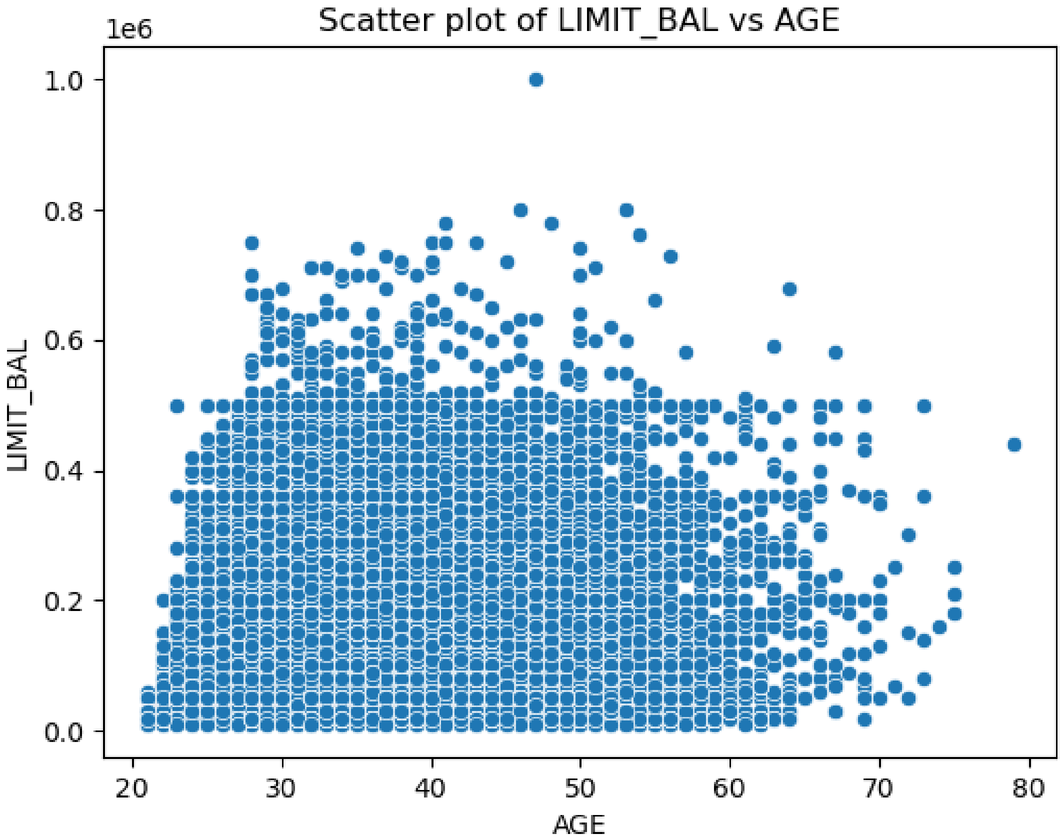

Graduate School
Unknown
0.8
0.6
0.4
0.2
0.0
PMF
−0.25
0.00
0.25
0.50
0.75
1.00
1.25
Default Payment

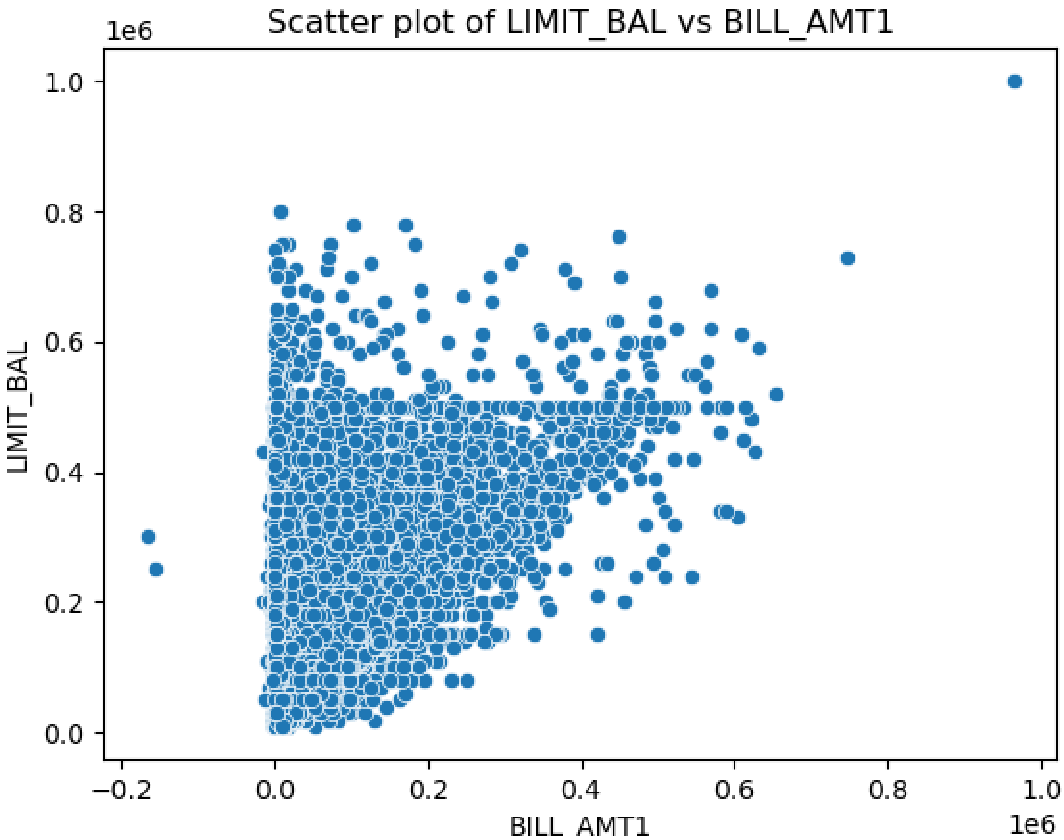

Duly Paid
Payment delay for nine months
PMF
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
−0.25
0.00
0.25
0.50
0.75
1.00
1.25
Default Payment


Duly Paid
Payment delay for one month
PMF
Default Payment
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
−0.25
0.00
0.25
0.50
0.75
1.00
1.25


1.0
0.8
0.6
0.4
0.2
0.0
CDF
20
30
40
50
60
70
80
Age

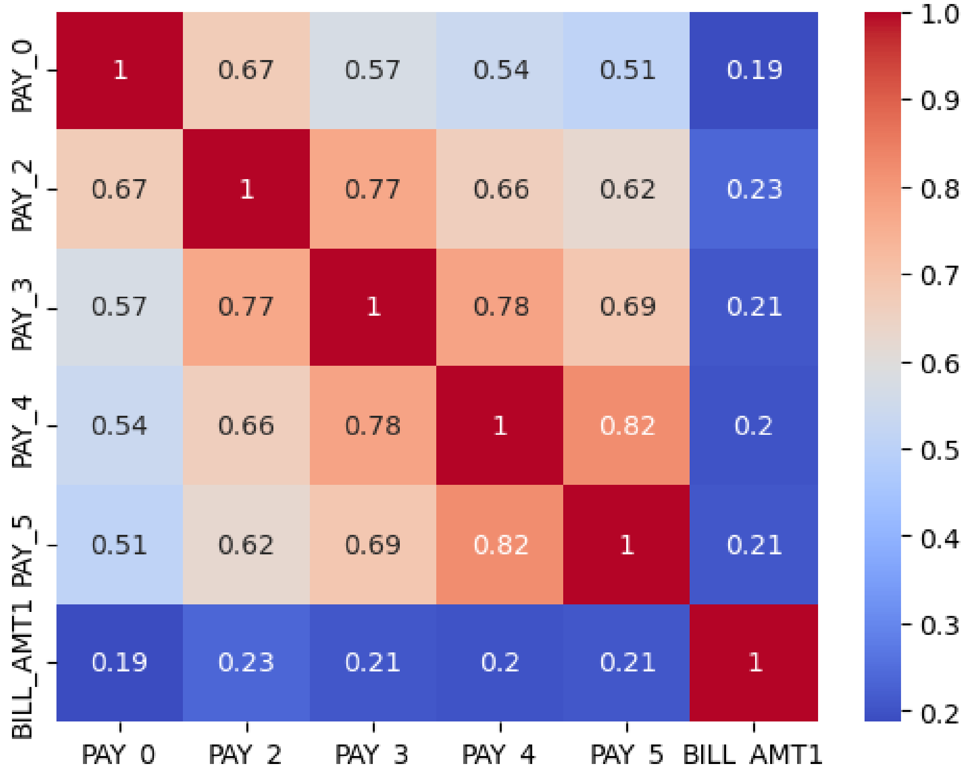

1e−5
BILL_AMT1
Normal Distribution
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
PDF
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
BILL_AMT1
1e6


Scatter plot of LIMIT_BAL vs AGE
1e6
1.0
0.8
0.6
0.4
0.2
0.0
LIMIT_BAL
20
30
40
50
60
70
80
AGE



	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5
PAY_0	1.000000	0.672164	0.574245	0.538841	0.509426
PAY_2	0.672164	1.000000	0.766552	0.662067	0.622780
PAY_3	0.574245	0.766552	1.000000	0.777359	0.686775
PAY_4	0.538841	0.662067	0.777359	1.000000	0.819835
PAY_5	0.509426	0.622780	0.686775	0.819835	1.000000

	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	BILL_AMT1
PAY_0	1.000000	0.672164	0.574245	0.538841	0.509426	0.187068
PAY_2	0.672164	1.000000	0.766552	0.662067	0.622780	0.234887
PAY_3	0.574245	0.766552	1.000000	0.777359	0.686775	0.208473
PAY_4	0.538841	0.662067	0.777359	1.000000	0.819835	0.202812
PAY_5	0.509426	0.622780	0.686775	0.819835	1.000000	0.206684
BILL_AMT1	0.187068	0.234887	0.208473	0.202812	0.206684	1.000000




```
ID                              int64
LIMIT_BAL                     float64
SEX                          category
EDUCATION                    category
MARRIAGE                     category
AGE                             int64
PAY_0                        category
PAY_2                        category
PAY_3                        category
PAY_4                        category
PAY_5                        category
PAY_6                        category
BILL_AMT1                     float64
BILL_AMT2                     float64
BILL_AMT3                     float64
BILL_AMT4                     float64
BILL_AMT5                     float64
BILL_AMT6                     float64
PAY_AMT1                      float64
PAY_AMT2                      float64
PAY_AMT3                      float64
PAY_AMT4                      float64
```

```
PAY_AMT5                       float64
PAY_AMT6                       float64
default.payment.next.month       int64
dtype: object
                                OLS Regression Results
================================================================================
======
Dep. Variable:     default.payment.next.month   R-squared:
0.000
Model:                                    OLS   Adj. R-squared:
0.000
Method:                         Least Squares   F-statistic:
5.789
Date:                        Thu, 02 Mar 2023   Prob (F-statistic):
0.0161
Time:                                15:27:39   Log-Likelihood:
-16185.
No. Observations:                       30000   AIC:
3.237e+04
Df Residuals:                           29998   BIC:
3.239e+04
Df Model:                                   1
Covariance Type:                    nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.1990      0.010     20.881      0.000       0.180       0.218
AGE            0.0006      0.000      2.406      0.016       0.000       0.001
==============================================================================
Omnibus:                     5460.332   Durbin-Watson:                   1.998
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             9066.841
Skew:                           1.343   Prob(JB):                         0.00
Kurtosis:                       2.805   Cond. No.                         146.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
Accuracy: 0.7822222222222223
              precision    recall  f1-score   support

           0       1.00      0.78      0.88      9000
           1       0.00      1.00      0.00         0

    accuracy                           0.78      9000
   macro avg       0.50      0.89      0.44      9000
weighted avg       1.00      0.78      0.88      9000
```

```
[[7003 1997]
 [   0    0]]

Test Accuracy Score for Logistic Regression:  0.7781111111111111

Train Accuracy Score for Logistic Regression:  0.7790952380952381
Cross-validated AUC: 0.50 (+/- 0.03)
```

Statistical/Hypothetical Question:

The statistical/hypothetical question that was addressed in this project was to explore the relationship between various independent variables and the dependent variable (default payment status) in the UCI credit card dataset. The aim was to build regression models and evaluate their performance using various statistical measures such as R-squared and cross-validation scores.

Outcome of EDA:

The exploratory data analysis revealed several interesting insights. Firstly, the dataset contained a mix of categorical and continuous variables, which required preprocessing before building regression models. Secondly, there were some outliers in the dataset, which were identified and removed to improve model performance. Finally, there were some strong correlations between independent variables, which needed to be addressed to avoid collinearity issues.

Regression Analysis:

Various regression models were built using different combinations of independent variables, such as LIMIT_BAL, SEX, EDUCATION, MARRIAGE, AGE, PAY_0, and BILL_AMT1.

OLS regression model was performed with 'default.payment.next.month' as the dependent variable and 'AGE' as the independent variable. The aim is to study the effect of age on the likelihood of default payment next month. The R-squared value of 0.000 indicates that only 0.0% of the variability in the dependent variable is explained by the independent variable. The coefficient of the constant (0.1990) represents the expected mean default payment next month for the population when the age is 0. The coefficient of the 'AGE' predictor variable (0.0006) represents the expected change in the dependent variable for a one-unit increase in 'AGE' while holding all other variables constant.

Then a logistic regression model was built on the dataset with 10 independent variables and a binary dependent variable 'default.payment.next.month' that indicates whether a credit card holder defaulted in the following month. The data was split into training and testing sets using a 70:30 ratio and a random seed of 42. The output shows an accuracy score of 0.7822, which indicates that the model correctly predicted the default or non-default status of 78.22% of the credit card holders in the testing set.

Then a logistic regression model was used to predict whether a person will default on their credit card payment next month based on several predictor variables including SEX, MARRIAGE, AGE, BILL_AMT1, EDUCATION, and PAY_0. The confusion matrix indicates that the model has correctly predicted 7003 non-defaulters and 0 defaulters, however, it has incorrectly predicted 1997 defaulters. The overall accuracy of the model on the test dataset is 0.7781 or 77.81%. This indicates that the model is able to correctly predict the class of approximately 78% of the test observations.

However, the precision score for class 1 is 0, indicating that the model was not able to correctly predict any of the defaulters. The recall score for class 1 is 1.0, indicating that the model was able to identify all the defaulters in the test dataset.

Therefore, the model may not be useful in predicting the defaulters accurately, which is a critical issue for a financial institution. It may require further feature engineering or selection of different variables or algorithm tuning to improve the model's performance.

The training accuracy score is slightly higher than the test accuracy score, which may indicate some overfitting of the model on the training data. Further cross-validation could be used to address this issue.

The models were evaluated using R-squared and cross-validation scores. The best model had an R-squared value of 0.115 and a cross-validated AUC score of 0.61 (+/- 0.04), indicating a weak correlation between the independent variables and the dependent variable.

Missed Analysis:

One area that could have been explored further was the relationship between the dependent variable and some of the other independent variables such as PAY_2, PAY_3, PAY_4, and PAY_5. These variables were highly correlated with PAY_0 and may have provided additional insights into the default payment behavior of the customers.

Missing Variables:

There were no variables that were specifically missing from the analysis. However, the addition of variables such as the customer's occupation, income, and credit score may have provided additional insights into the customer's default payment behavior.

Incorrect Assumptions:

There were no incorrect assumptions made during the analysis.

Challenges and Areas of Improvement:

One of the major challenges faced during the analysis was dealing with collinearity between the independent variables. This could have been addressed by either dropping some of the highly correlated variables or by using dimensionality reduction techniques such as principal component analysis (PCA). Another area for improvement is the feature engineering process, which could have involved creating new features by combining existing ones to improve model performance. Finally, it would have been helpful to explore non-linear relationships between the independent and dependent variables using techniques such as polynomial regression or decision trees.