ML Final

December 18, 2024

```
[135]: import pandas as pd
       df = pd.read_csv('/Users/aqsa/Desktop/ML Project/default of credit card clients.
        ↪csv', header =1)
       df
```

```
[135]:           ID  LIMIT_BAL  SEX  EDUCATION  MARRIAGE  AGE  PAY_0  PAY_2  PAY_3  \
       0          1      20000    2          2         1   24      2      2     -1
       1          2     120000    2          2         2   26     -1      2      0
       2          3      90000    2          2         2   34      0      0      0
       3          4      50000    2          2         1   37      0      0      0
       4          5      50000    1          2         1   57     -1      0     -1
       …        …         …  …        …        …  …     …     …      …
       29995  29996     220000    1          3         1   39      0      0      0
       29996  29997     150000    1          3         2   43     -1     -1     -1
       29997  29998      30000    1          2         2   37      4      3      2
       29998  29999      80000    1          3         1   41      1     -1      0
       29999  30000      50000    1          2         1   46      0      0      0

              PAY_4  …  BILL_AMT4  BILL_AMT5  BILL_AMT6  PAY_AMT1  PAY_AMT2  \
       0         -1  …          0          0          0         0       689
       1          0  …       3272       3455       3261         0      1000
       2          0  …      14331      14948      15549      1518      1500
       3          0  …      28314      28959      29547      2000      2019
       4          0  …      20940      19146      19131      2000     36681
       …        …  …        …        …          …        …       …
       29995      0  …      88004      31237      15980      8500     20000
       29996     -1  …       8979       5190          0      1837      3526
       29997     -1  …      20878      20582      19357         0         0
       29998      0  …      52774      11855      48944     85900      3409
       29999      0  …      36535      32428      15313      2078      1800

              PAY_AMT3  PAY_AMT4  PAY_AMT5  PAY_AMT6  default payment next month
       0             0         0         0         0                           1
       1          1000      1000         0      2000                           1
       2          1000      1000      1000      5000                           0
       3          1200      1100      1069      1000                           0
       4         10000      9000       689       679                           0
       …           …       …       …       …                       …
```

```
29995       5003      3047      5000      1000                            0
29996       8998       129         0         0                            0
29997      22000      4200      2000      3100                            1
29998       1178      1926     52964      1804                            1
29999       1430      1000      1000      1000                            1

[30000 rows x 25 columns]
```

[136]:

```
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
```

[137]:

```
df.head()
```

[137]:

```
   ID  LIMIT_BAL  SEX  EDUCATION  MARRIAGE  AGE  PAY_0  PAY_2  PAY_3  PAY_4  \
0   1      20000    2          2         1   24      2      2     -1     -1
1   2     120000    2          2         2   26     -1      2      0      0
```

```
2   3     90000    2          2          2   34      0      0      0      0
3   4     50000    2          2          1   37      0      0      0      0
4   5     50000    1          2          1   57     -1      0     -1      0

   …  BILL_AMT4  BILL_AMT5  BILL_AMT6  PAY_AMT1  PAY_AMT2  PAY_AMT3  \
0  …          0          0          0         0       689         0
1  …       3272       3455       3261         0      1000      1000
2  …      14331      14948      15549      1518      1500      1000
3  …      28314      28959      29547      2000      2019      1200
4  …      20940      19146      19131      2000     36681     10000

   PAY_AMT4  PAY_AMT5  PAY_AMT6  default payment next month
0         0         0         0                           1
1      1000         0      2000                           1
2      1000      1000      5000                           0
3      1100      1069      1000                           0
4      9000       689       679                           0

[5 rows x 25 columns]
```

[138]:
```
df.describe()
```

[138]:
```
                 ID        LIMIT_BAL           SEX     EDUCATION      MARRIAGE  \
count  30000.000000     30000.000000  30000.000000  30000.000000  30000.000000
mean   15000.500000    167484.322667      1.603733      1.853133      1.551867
std     8660.398374    129747.661567      0.489129      0.790349      0.521970
min        1.000000     10000.000000      1.000000      0.000000      0.000000
25%     7500.750000     50000.000000      1.000000      1.000000      1.000000
50%    15000.500000    140000.000000      2.000000      2.000000      2.000000
75%    22500.250000    240000.000000      2.000000      2.000000      2.000000
max    30000.000000   1000000.000000      2.000000      6.000000      3.000000

                AGE         PAY_0         PAY_2         PAY_3         PAY_4  \
count  30000.000000  30000.000000  30000.000000  30000.000000  30000.000000
mean      35.485500     -0.016700     -0.133767     -0.166200     -0.220667
std        9.217904      1.123802      1.197186      1.196868      1.169139
min       21.000000     -2.000000     -2.000000     -2.000000     -2.000000
25%       28.000000     -1.000000     -1.000000     -1.000000     -1.000000
50%       34.000000      0.000000      0.000000      0.000000      0.000000
75%       41.000000      0.000000      0.000000      0.000000      0.000000
max       79.000000      8.000000      8.000000      8.000000      8.000000

       …      BILL_AMT4      BILL_AMT5      BILL_AMT6      PAY_AMT1  \
count  …   30000.000000   30000.000000   30000.000000  30000.000000
mean   …   43262.948967   40311.400967   38871.760400   5663.580500
std    …   64332.856134   60797.155770   59554.107537  16563.280354
min    … -170000.000000  -81334.000000 -339603.000000      0.000000
```

```
25%    …    2326.750000    1763.000000    1256.000000    1000.000000
50%    …   19052.000000   18104.500000   17071.000000    2100.000000
75%    …   54506.000000   50190.500000   49198.250000    5006.000000
max    …  891586.000000  927171.000000  961664.000000  873552.000000

           PAY_AMT2      PAY_AMT3       PAY_AMT4       PAY_AMT5  \
count  3.000000e+04   30000.00000   30000.000000   30000.000000
mean   5.921163e+03    5225.68150    4826.076867    4799.387633
std    2.304087e+04   17606.96147   15666.159744   15278.305679
min    0.000000e+00       0.00000       0.000000       0.000000
25%    8.330000e+02     390.00000     296.000000     252.500000
50%    2.009000e+03    1800.00000    1500.000000    1500.000000
75%    5.000000e+03    4505.00000    4013.250000    4031.500000
max    1.684259e+06  896040.00000  621000.000000  426529.000000

            PAY_AMT6  default payment next month
count   30000.000000                30000.000000
mean     5215.502567                    0.221200
std     17777.465775                    0.415062
min         0.000000                    0.000000
25%       117.750000                    0.000000
50%      1500.000000                    0.000000
75%      4000.000000                    0.000000
max    528666.000000                    1.000000

[8 rows x 25 columns]
```

[139]:

```
df.rename(columns={'PAY_0': 'PAY_1'}, inplace=True)
```

[140]:

```
df[['PAY_1', 'PAY_2', 'PAY_3', 'PAY_4', 'PAY_5', 'PAY_6']].describe()
```

[140]:

```
              PAY_1         PAY_2         PAY_3         PAY_4         PAY_5  \
count  30000.000000  30000.000000  30000.000000  30000.000000  30000.000000
mean      -0.016700     -0.133767     -0.166200     -0.220667     -0.266200
std        1.123802      1.197186      1.196868      1.169139      1.133187
min       -2.000000     -2.000000     -2.000000     -2.000000     -2.000000
25%       -1.000000     -1.000000     -1.000000     -1.000000     -1.000000
50%        0.000000      0.000000      0.000000      0.000000      0.000000
75%        0.000000      0.000000      0.000000      0.000000      0.000000
max        8.000000      8.000000      8.000000      8.000000      8.000000

              PAY_6
count  30000.000000
mean      -0.291100
std        1.149988
min       -2.000000
25%       -1.000000
```

```
50%          0.000000
75%          0.000000
max          8.000000
```

[141]:

```
df['EDUCATION'].unique()
```

[141]:

```
array([2, 1, 3, 5, 4, 6, 0])
```

[142]:

```
df = df[~df['EDUCATION'].isin([0, 5, 6])]
```

[143]:

```
df = df[df['MARRIAGE'] != 0]
```

[144]:

```
df['MARRIAGE'].unique()
```

[144]:

```
array([1, 2, 3])
```

[165]:

```
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

#Ordinal
df['EDUCATION'] = le.fit_transform(df['EDUCATION'])
df['MARRIAGE'] = le.fit_transform(df['MARRIAGE'])

#binary
df['SEX'] = le.fit_transform(df['SEX'])


print(df.head(5))
```

```
   ID  LIMIT_BAL  SEX  EDUCATION  MARRIAGE  AGE  PAY_1  PAY_2  PAY_3  PAY_4  \
0   1      20000    1          1         0   24      2      2     -1     -1
1   2     120000    1          1         1   26     -1      2      0      0
2   3      90000    1          1         1   34      0      0      0      0
3   4      50000    1          1         0   37      0      0      0      0
4   5      50000    0          1         0   57     -1      0     -1      0

   …  BILL_AMT4  BILL_AMT5  BILL_AMT6  PAY_AMT1  PAY_AMT2  PAY_AMT3  \
0  …          0          0          0         0       689         0
1  …       3272       3455       3261         0      1000      1000
2  …      14331      14948      15549      1518      1500      1000
3  …      28314      28959      29547      2000      2019      1200
4  …      20940      19146      19131      2000     36681     10000

   PAY_AMT4  PAY_AMT5  PAY_AMT6  DEFAULT
0         0         0         0        1
1      1000         0      2000        1
2      1000      1000      5000        0
```

```
3      1100      1069      1000         0
4      9000       689       679         0

[5 rows x 25 columns]
```

1 After encoding I will standardize features and then prepare to split the data

[ ]:

```
from sklearn.preprocessing import StandardScaler

# Numerical columns to standardize
numerical_columns = ['LIMIT_BAL', 'AGE', 'BILL_AMT1', 'BILL_AMT2', 'BILL_AMT3',␣
 ↪'BILL_AMT4', 'BILL_AMT5', 'BILL_AMT6', 'PAY_AMT1', 'PAY_AMT2', 'PAY_AMT3',␣
 ↪'PAY_AMT4', 'PAY_AMT5', 'PAY_AMT6']


scaler = StandardScaler()

data[numerical_columns] = scaler.fit_transform(data[numerical_columns])
```

[ ]:

```

```

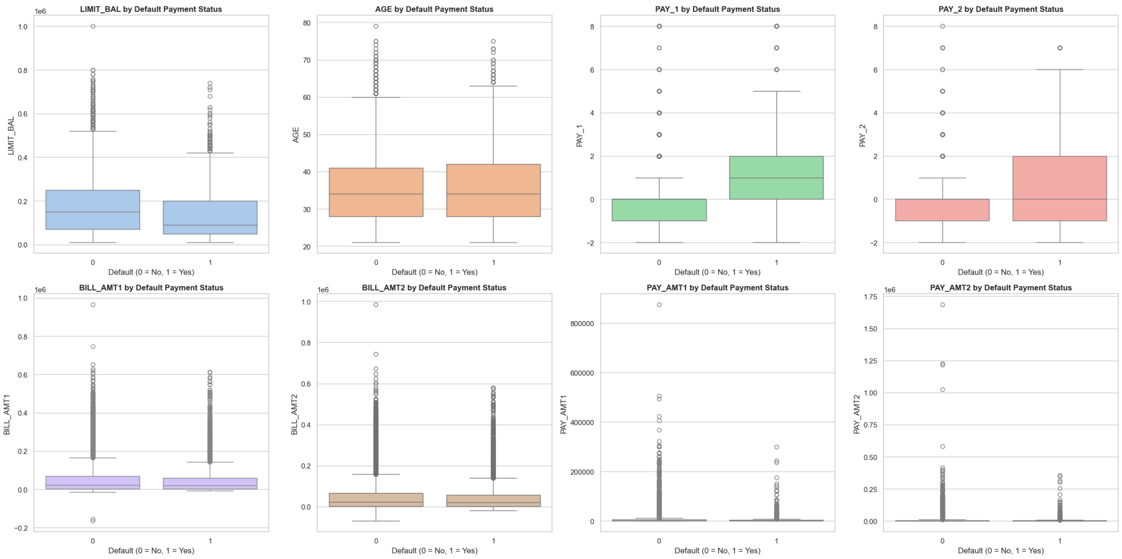
[176]:

```
import seaborn as sns
import matplotlib.pyplot as plt


box_plot_features = [
    'LIMIT_BAL', 'AGE', 'PAY_1', 'PAY_2', 'BILL_AMT1', 'BILL_AMT2', 'PAY_AMT1',␣
 ↪'PAY_AMT2'
]
colors = sns.color_palette('pastel', len(box_plot_features))


fig, axes = plt.subplots(nrows=2, ncols=4, figsize=(24, 12))
axes = axes.flatten()


for i, feature in enumerate(box_plot_features):
    sns.boxplot(
        x='DEFAULT',
        y=feature,
        data=df,
        ax=axes[i],
        palette=[colors[i]]
    )
    axes[i].set_title(f'{feature} by Default Payment Status', fontsize=12,␣
 ↪fontweight='bold')
    axes[i].set_xlabel('Default (0 = No, 1 = Yes)')
    axes[i].set_ylabel(feature)


plt.tight_layout()


plt.show()
```

```
/var/folders/f3/9rjb9s3d5095pp6w66p1r_kc0000gn/T/ipykernel_59025/3787058677.py:1
7: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.boxplot(
/var/folders/f3/9rjb9s3d5095pp6w66p1r_kc0000gn/T/ipykernel_59025/3787058677.py:1
7: UserWarning:
The palette list has fewer values (1) than needed (2) and will cycle, which may
produce an uninterpretable plot.
  sns.boxplot(
/var/folders/f3/9rjb9s3d5095pp6w66p1r_kc0000gn/T/ipykernel_59025/3787058677.py:1
7: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.boxplot(
/var/folders/f3/9rjb9s3d5095pp6w66p1r_kc0000gn/T/ipykernel_59025/3787058677.py:1
7: UserWarning:
The palette list has fewer values (1) than needed (2) and will cycle, which may
produce an uninterpretable plot.
  sns.boxplot(
/var/folders/f3/9rjb9s3d5095pp6w66p1r_kc0000gn/T/ipykernel_59025/3787058677.py:1
7: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.boxplot(
/var/folders/f3/9rjb9s3d5095pp6w66p1r_kc0000gn/T/ipykernel_59025/3787058677.py:1
7: UserWarning:
The palette list has fewer values (1) than needed (2) and will cycle, which may
```

```
produce an uninterpretable plot.
  sns.boxplot(
/var/folders/f3/9rjb9s3d5095pp6w66p1r_kc0000gn/T/ipykernel_59025/3787058677.py:1
7: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.boxplot(
/var/folders/f3/9rjb9s3d5095pp6w66p1r_kc0000gn/T/ipykernel_59025/3787058677.py:1
7: UserWarning:
The palette list has fewer values (1) than needed (2) and will cycle, which may
produce an uninterpretable plot.
  sns.boxplot(
/var/folders/f3/9rjb9s3d5095pp6w66p1r_kc0000gn/T/ipykernel_59025/3787058677.py:1
7: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.boxplot(
/var/folders/f3/9rjb9s3d5095pp6w66p1r_kc0000gn/T/ipykernel_59025/3787058677.py:1
7: UserWarning:
The palette list has fewer values (1) than needed (2) and will cycle, which may
produce an uninterpretable plot.
  sns.boxplot(
/var/folders/f3/9rjb9s3d5095pp6w66p1r_kc0000gn/T/ipykernel_59025/3787058677.py:1
7: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.boxplot(
/var/folders/f3/9rjb9s3d5095pp6w66p1r_kc0000gn/T/ipykernel_59025/3787058677.py:1
7: UserWarning:
The palette list has fewer values (1) than needed (2) and will cycle, which may
produce an uninterpretable plot.
  sns.boxplot(
/var/folders/f3/9rjb9s3d5095pp6w66p1r_kc0000gn/T/ipykernel_59025/3787058677.py:1
7: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.
```

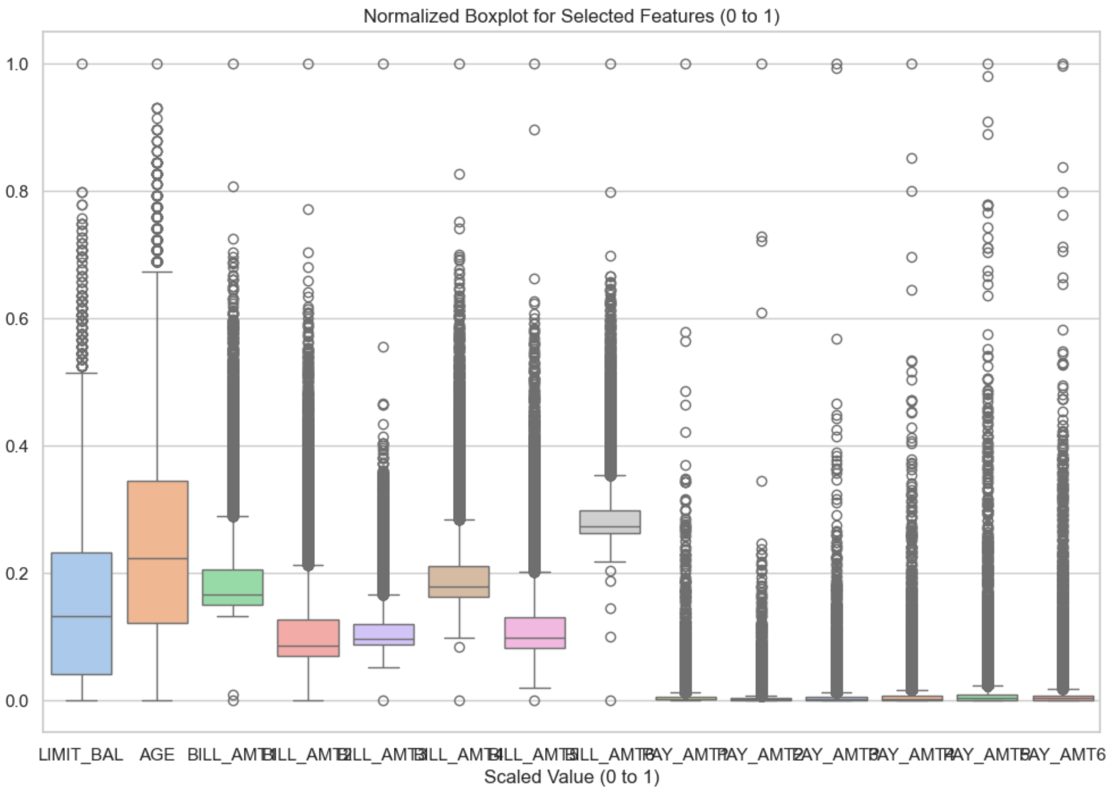
```
  sns.boxplot(
/var/folders/f3/9rjb9s3d5095pp6w66p1r_kc0000gn/T/ipykernel_59025/3787058677.py:1
7: UserWarning:
The palette list has fewer values (1) than needed (2) and will cycle, which may
produce an uninterpretable plot.
  sns.boxplot(
/var/folders/f3/9rjb9s3d5095pp6w66p1r_kc0000gn/T/ipykernel_59025/3787058677.py:1
7: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.boxplot(
/var/folders/f3/9rjb9s3d5095pp6w66p1r_kc0000gn/T/ipykernel_59025/3787058677.py:1
7: UserWarning:
The palette list has fewer values (1) than needed (2) and will cycle, which may
produce an uninterpretable plot.
  sns.boxplot(
```



[175]:

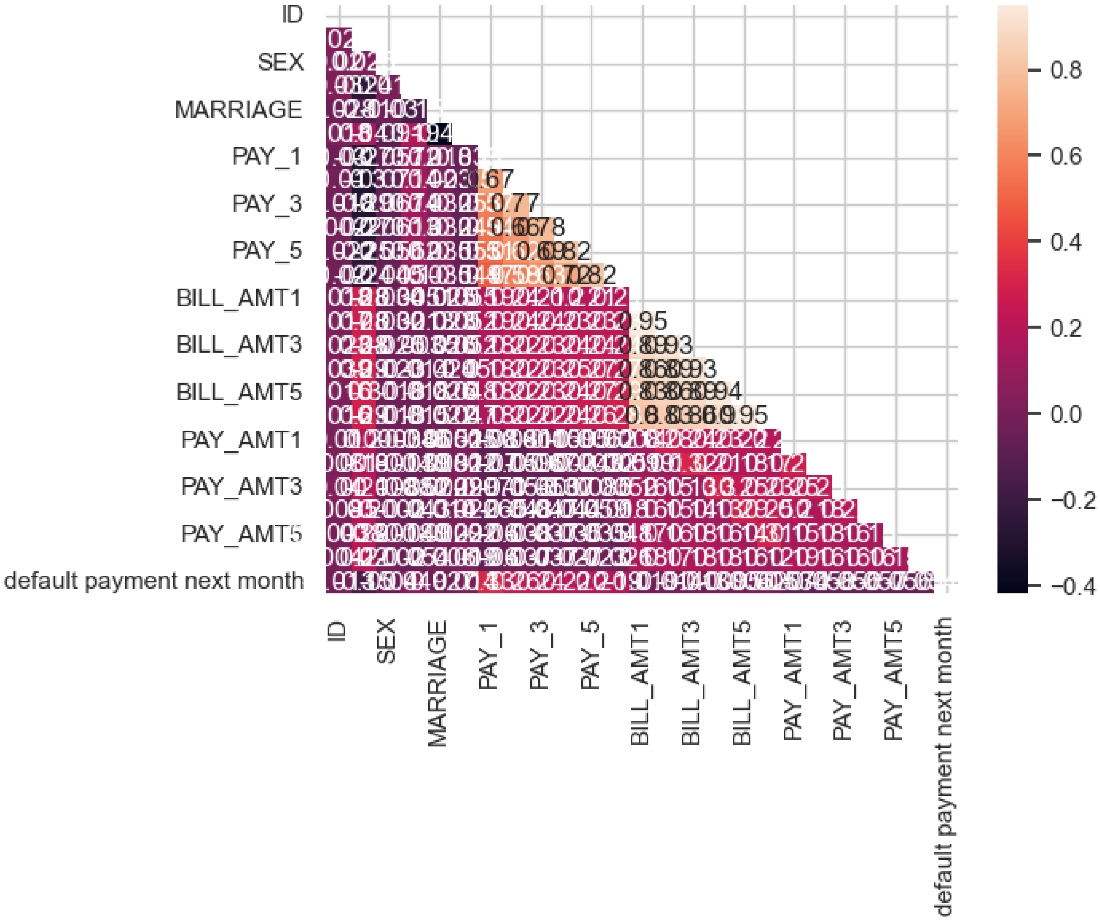
```
from sklearn.preprocessing import MinMaxScaler

df_selected = df[['LIMIT_BAL', 'AGE', 'BILL_AMT1', 'BILL_AMT2',␣
 ↪'BILL_AMT3','BILL_AMT4','BILL_AMT5', 'BILL_AMT6' , 'PAY_AMT1',␣
 ↪'PAY_AMT2','PAY_AMT3','PAY_AMT4','PAY_AMT5','PAY_AMT6']]



scaler = MinMaxScaler()
```

```
df_scaled = pd.DataFrame(scaler.fit_transform(df_selected), columns=df_selected.
 ↪columns)


plt.figure(figsize=(12, 8))
sns.boxplot(data=df_scaled, orient='v', palette="pastel")
plt.title('Normalized Boxplot for Selected Features (0 to 1)')
plt.xlabel('Scaled Value (0 to 1)')
plt.show()
```



[145]:
```
corr= df.corr()
matrix = np.triu(corr)

sns.heatmap(corr, annot=True, mask=matrix)
```

[145]:
```
<Axes: >
```



[146]:
```
df.columns
```

[146]:
```
Index(['ID', 'LIMIT_BAL', 'SEX', 'EDUCATION', 'MARRIAGE', 'AGE', 'PAY_1',
       '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')
```

[147]:
```
count_zeros = (df["default payment next month"] == 0).sum()
print(count_zeros)
```

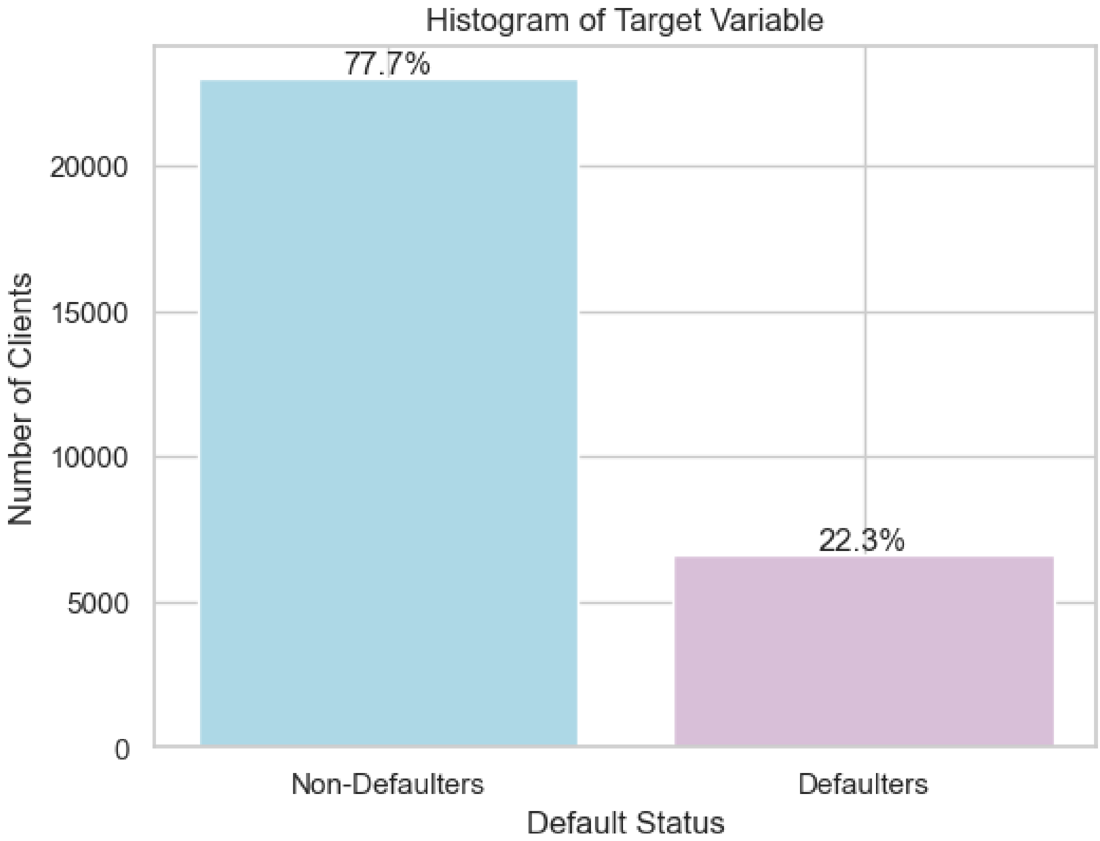
```
22996
```

[148]:
```
count_ones = (df["default payment next month"] == 1).sum()
print(count_ones)
```

```
6605
```

[149]:

```
counts = [count_zeros, count_ones]
labels = ['Non-Defaulters', 'Defaulters']

total = sum(counts)
percentages = [count / total * 100 for count in counts]

colors = ['lightblue', 'thistle']
plt.bar(labels, counts, color=colors)


plt.title('Histogram of Target Variable')
plt.xlabel('Default Status')
plt.ylabel('Number of Clients')


for i, count in enumerate(counts):
    plt.text(i, count + 10, f'{percentages[i]:.1f}%', ha='center', va='bottom',␣
 ↪fontsize=12)


plt.show()
```



[150]:
```
df['MARRIAGE'].value_counts()
```

[150]:
```
MARRIAGE
2    15806
1    13477
3      318
Name: count, dtype: int64
```

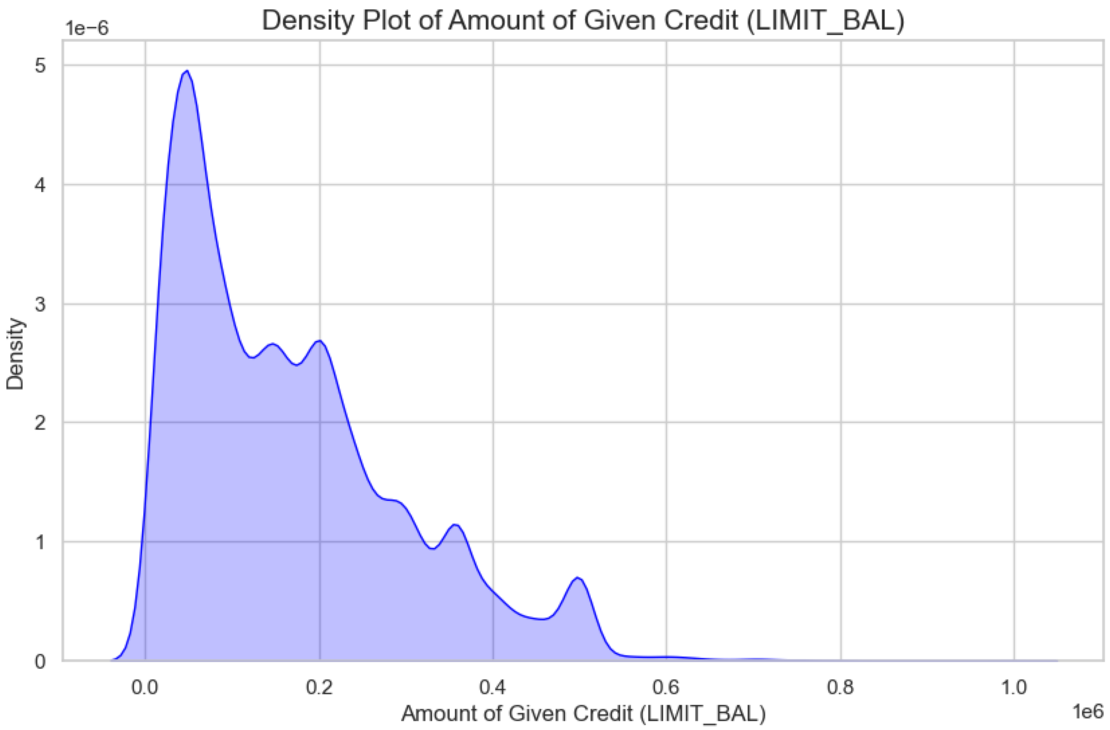
[151]:
```
df[['PAY_1', 'PAY_2', 'PAY_3', 'PAY_4', 'PAY_5', 'PAY_6']].describe()
```

[151]:
```
              PAY_1         PAY_2         PAY_3         PAY_4         PAY_5  \
count  29601.000000  29601.000000  29601.000000  29601.000000  29601.000000
mean      -0.014932     -0.131313     -0.163440     -0.218303     -0.263978
std        1.124503      1.199642      1.199793      1.172220      1.136217
min       -2.000000     -2.000000     -2.000000     -2.000000     -2.000000
25%       -1.000000     -1.000000     -1.000000     -1.000000     -1.000000
50%        0.000000      0.000000      0.000000      0.000000      0.000000
75%        0.000000      0.000000      0.000000      0.000000      0.000000
max        8.000000      8.000000      8.000000      8.000000      8.000000

              PAY_6
count  29601.000000
mean      -0.287558
std        1.152206
min       -2.000000
25%       -1.000000
50%        0.000000
75%        0.000000
max        8.000000
```

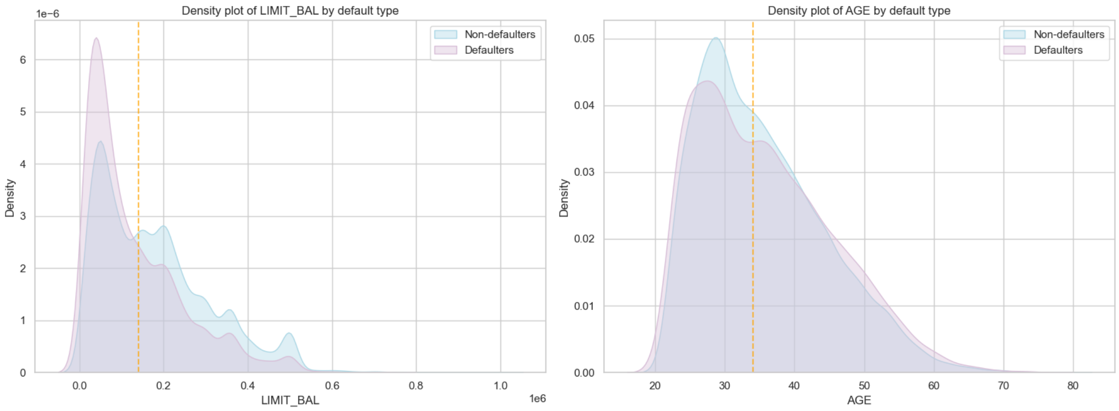
[152]:
```
sns.set(style="whitegrid")
plt.figure(figsize=(10,6))
sns.kdeplot(df['LIMIT_BAL'], shade=True, color='blue')


plt.title('Density Plot of Amount of Given Credit (LIMIT_BAL)', fontsize=15)
plt.xlabel('Amount of Given Credit (LIMIT_BAL)', fontsize=12)
plt.ylabel('Density', fontsize=12)


plt.show()
```

```
/var/folders/f3/9rjb9s3d5095pp6w66p1r_kc0000gn/T/ipykernel_59025/2209840784.py:8
: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
```

```
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(df['LIMIT_BAL'], shade=True, color='blue')
```



[153]:
```
fig, axes = plt.subplots(1, 2, figsize=(16, 6), sharey=False)

# First plot: Density plot for LIMIT_BAL
sns.kdeplot(
    data=df[df["default payment next month"] == 0],
    x='LIMIT_BAL',
    fill=True,
    color='lightblue',
    alpha=0.4,
    label='Non-defaulters',
    ax=axes[0]
)
sns.kdeplot(
    data=df[df["default payment next month"] == 1],
    x='LIMIT_BAL',
    fill=True,
    color='thistle',
    alpha=0.4,
    label='Defaulters',
    ax=axes[0]
)
axes[0].set_title("Density plot of LIMIT_BAL by default type")
axes[0].axvline(df['LIMIT_BAL'].median(), color='orange', linestyle='--', alpha=0.7)
axes[0].legend()

# Second plot: Density plot for AGE
sns.kdeplot(
    data=df[df["default payment next month"] == 0],
    x='AGE',
    fill=True,
    color='lightblue',
    alpha=0.4,
    label='Non-defaulters',
    ax=axes[1]
)
sns.kdeplot(
    data=df[df["default payment next month"] == 1],
    x='AGE',
    fill=True,
    color='thistle',
    alpha=0.4,
    label='Defaulters',
    ax=axes[1]
)
axes[1].set_title("Density plot of AGE by default type")
axes[1].axvline(df['AGE'].median(), color='orange', linestyle='--', alpha=0.7)
axes[1].legend()


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



[154]:

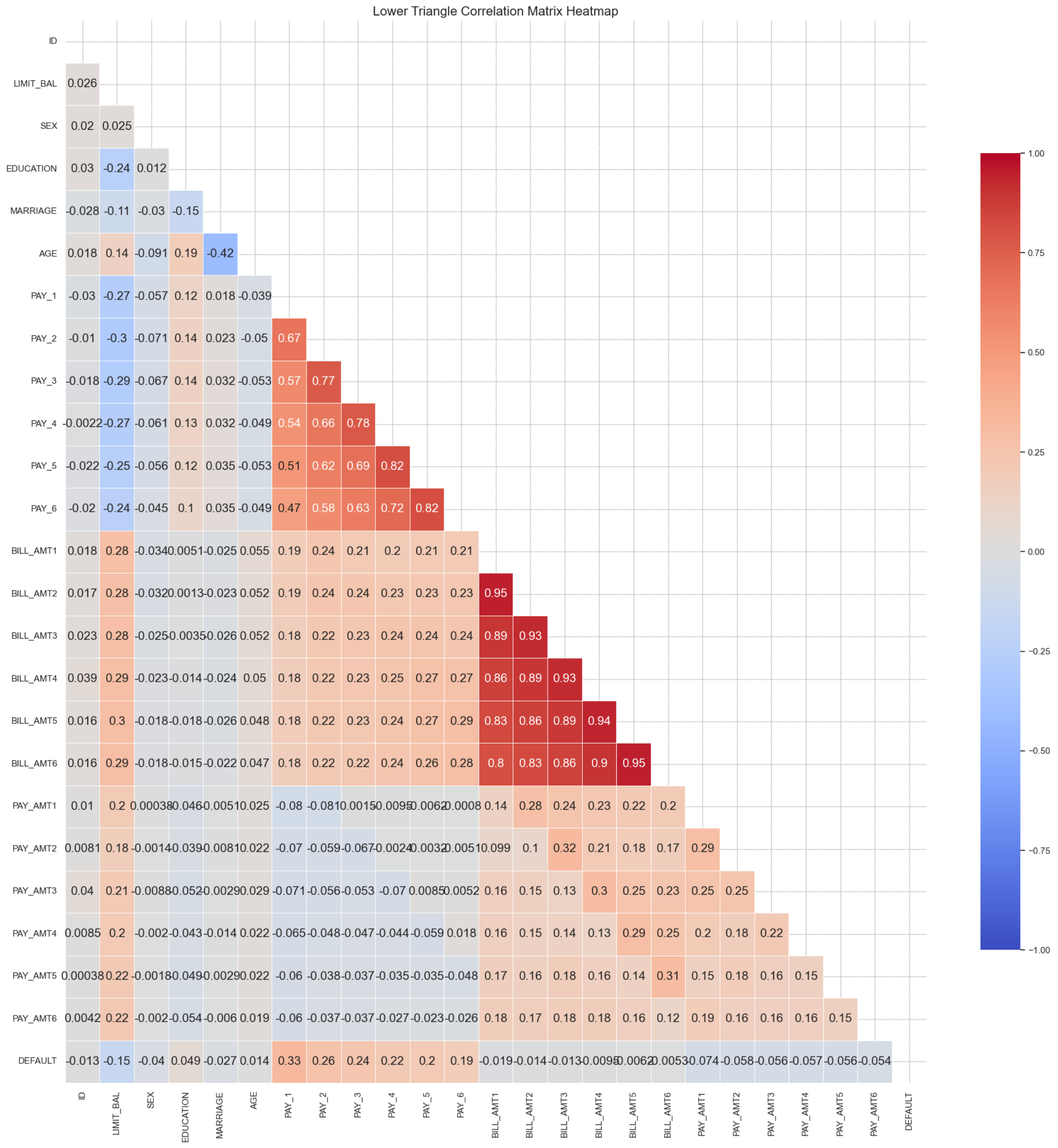
```
df.rename(columns= {'default payment next month': 'DEFAULT'}, inplace=True)
df.head(2)
```

[154]:

```
   ID  LIMIT_BAL  SEX  EDUCATION  MARRIAGE  AGE  PAY_1  PAY_2  PAY_3  PAY_4  \
0   1      20000    2          2         1   24      2      2     -1     -1
1   2     120000    2          2         2   26     -1      2      0      0

   …  BILL_AMT4  BILL_AMT5  BILL_AMT6  PAY_AMT1  PAY_AMT2  PAY_AMT3  \
0  …          0          0          0         0       689         0
1  …       3272       3455       3261         0      1000      1000

   PAY_AMT4  PAY_AMT5  PAY_AMT6  DEFAULT
0         0         0         0        1
1      1000         0      2000        1

[2 rows x 25 columns]
```

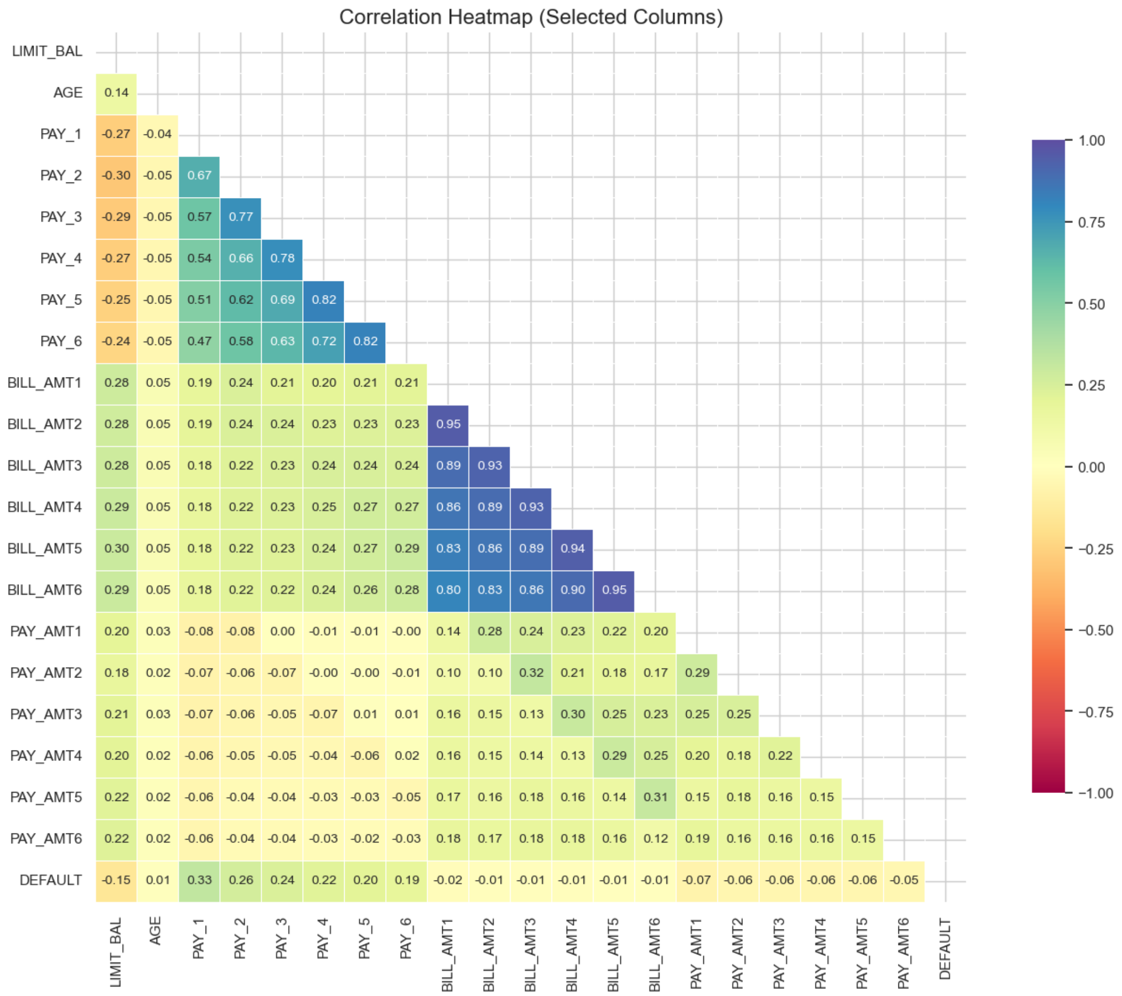
[155]:

```
correlation_matrix = df.corr()

mask = np.triu(np.ones_like(correlation_matrix, dtype=bool))

plt.figure(figsize=(20, 20))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2', vmin=-1,␣
  ↪vmax=1,
            mask=mask, linewidths=0.5, annot_kws={"size": 15},␣
  ↪cbar_kws={"shrink": 0.75})

plt.title("Lower Triangle Correlation Matrix Heatmap", fontsize=16)

plt.tight_layout()

plt.show()
```



[156]:
```
columns_to_include = [
    'LIMIT_BAL', 'AGE', 'PAY_1',
       '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'
]



filtered_df = df[columns_to_include]
```

```
correlation_matrix = filtered_df.corr()
mask = np.triu(np.ones_like(correlation_matrix, dtype=bool))

plt.figure(figsize=(18, 12))


sns.heatmap(correlation_matrix, annot=True, cmap='Spectral', fmt='.2f',␣
 ↪vmin=-1, vmax=1,
             mask=mask, linewidths=0.5, annot_kws={"size": 10},␣
 ↪cbar_kws={"shrink": 0.75}, square=True)

plt.title("Correlation Heatmap (Selected Columns)", fontsize=16)
plt.show()
```



[157]:

```
X = dataset.drop(['def_pay'],axis=1)
X.corrwith(dataset['def_pay']).plot.bar(figsize = (20, 10), title =␣
 ↪"Correlation with Default",
                                        fontsize = 20,rot = 90, grid = True)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[157], line 1
----> 1 X = dataset.drop(['def_pay'],axis=1)
      2 X.corrwith(dataset['def_pay']).plot.bar(figsize = (20, 10), title =␣
 ↪"Correlation with Default",
      3                                         fontsize = 20,rot = 90, grid =␣
 ↪True)

NameError: name 'dataset' is not defined
```

[160]:

```

```

```
<class 'pandas.core.frame.DataFrame'>
Index: 29601 entries, 0 to 29999
Data columns (total 25 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   ID         29601 non-null  int64
 1   LIMIT_BAL  29601 non-null  int64
 2   SEX        29601 non-null  int64
 3   EDUCATION  29601 non-null  int64
 4   MARRIAGE   29601 non-null  int64
 5   AGE        29601 non-null  int64
 6   PAY_1      29601 non-null  int64
 7   PAY_2      29601 non-null  int64
 8   PAY_3      29601 non-null  int64
 9   PAY_4      29601 non-null  int64
 10  PAY_5      29601 non-null  int64
 11  PAY_6      29601 non-null  int64
 12  BILL_AMT1  29601 non-null  int64
 13  BILL_AMT2  29601 non-null  int64
 14  BILL_AMT3  29601 non-null  int64
 15  BILL_AMT4  29601 non-null  int64
 16  BILL_AMT5  29601 non-null  int64
 17  BILL_AMT6  29601 non-null  int64
 18  PAY_AMT1   29601 non-null  int64
 19  PAY_AMT2   29601 non-null  int64
 20  PAY_AMT3   29601 non-null  int64
 21  PAY_AMT4   29601 non-null  int64
 22  PAY_AMT5   29601 non-null  int64
 23  PAY_AMT6   29601 non-null  int64
 24  DEFAULT    29601 non-null  int64
dtypes: int64(25)
memory usage: 5.9 MB
```

[ ]:

[ ]:

[ ]:

[ ]: