```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline


from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.metrics import roc_auc_score


df.set_index(['ID'],inplace=True)
df.head()
```

ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	...	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_
1	20000.0	2	2	1	24	2	2	-1	-1	-2	...	0.0	0.0	0.0	0.0	6
2	120000.0	2	2	2	26	-1	2	0	0	0	...	3272.0	3455.0	3261.0	0.0	10
3	90000.0	2	2	2	34	0	0	0	0	0	...	14331.0	14948.0	15549.0	1518.0	15
4	50000.0	2	2	1	37	0	0	0	0	0	...	28314.0	28959.0	29547.0	2000.0	20
5	50000.0	1	2	1	57	-1	0	-1	0	0	...	20940.0	19146.0	19131.0	2000.0	366

5 rows × 24 columns

Start coding or generate with AI.

#

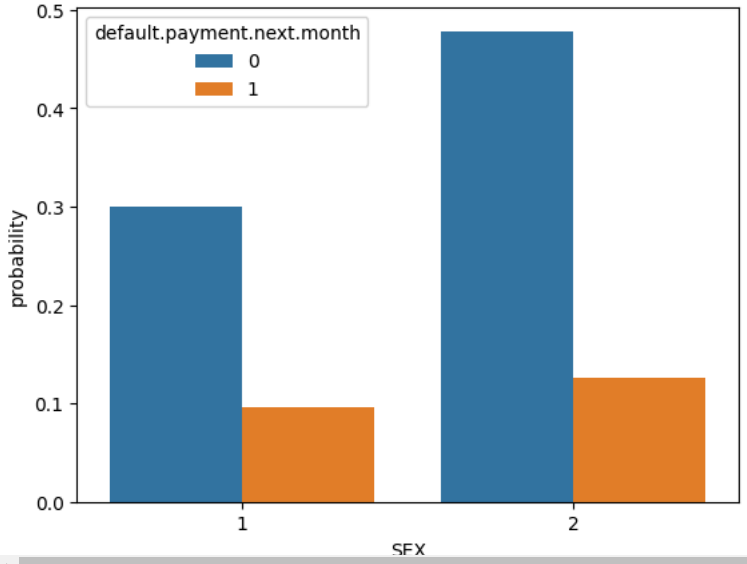
- SEX: Gender (1=male, 2=female)
- EDUCATION: (1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown)
- MARRIAGE: Marital status (1=married, 2=single, 3=others)

```
df.SEX.value_counts()
```

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

```
sns.countplot(x='SEX',data=df,stat='probability',hue='default.payment.next.month')
```

```
<Axes: xlabel='SEX', ylabel='probability'>
```



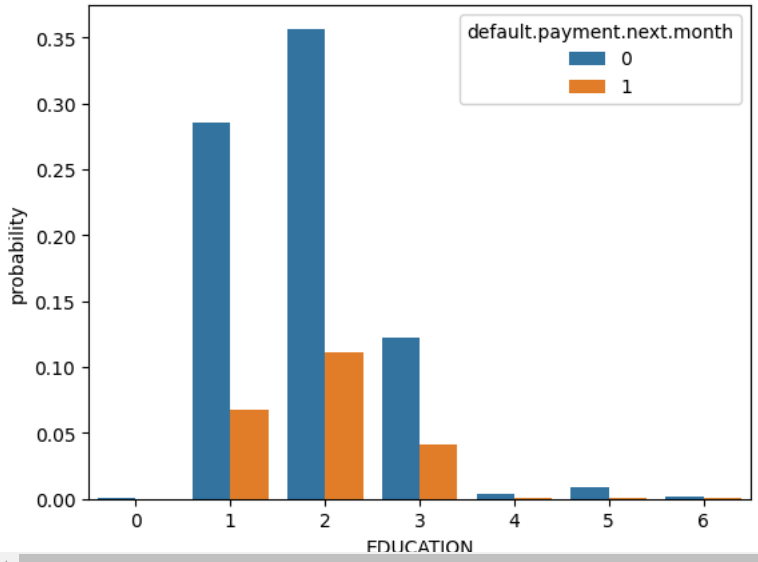
- EDUCATION: (1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown)

```
df.EDUCATION.value_counts()
```

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

```
sns.countplot(x='EDUCATION',data=df,stat='probability',hue='default.payment.next.month')
```

```
<Axes: xlabel='EDUCATION', ylabel='probability'>
```




```
df.EDUCATION=df.EDUCATION.replace({0:1,4:3,5:1,6:3})
```

```
df.head()
```

ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	...	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_
1	20000.0	2	2	1	24	2	2	-1	-1	-2	...	0.0	0.0	0.0	0.0	6
2	120000.0	2	2	2	26	-1	2	0	0	0	...	3272.0	3455.0	3261.0	0.0	10
3	90000.0	2	2	2	34	0	0	0	0	0	...	14331.0	14948.0	15549.0	1518.0	15
4	50000.0	2	2	1	37	0	0	0	0	0	...	28314.0	28959.0	29547.0	2000.0	20
5	50000.0	1	2	1	57	-1	0	-1	0	0	...	20940.0	19146.0	19131.0	2000.0	366

5 rows × 24 columns

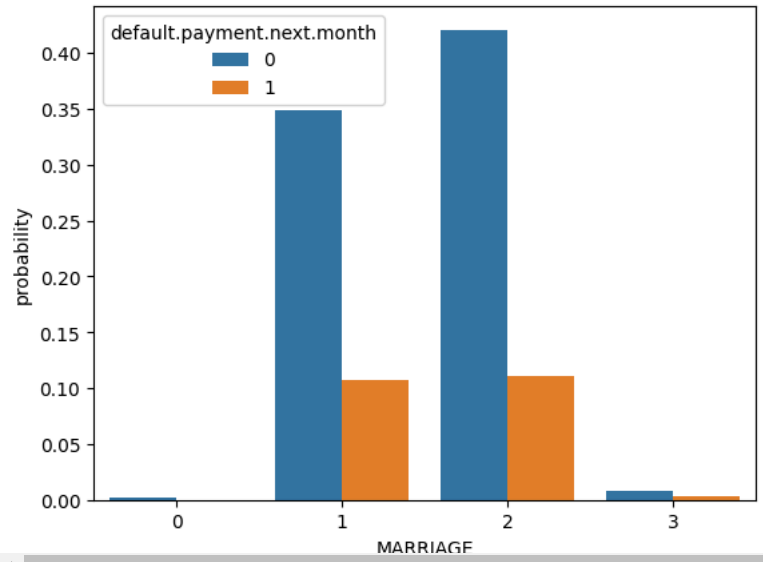
- MARRIAGE: Marital status (1=married, 2=single, 3=others)

```
df.MARRIAGE.value_counts()
```

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

```
sns.countplot(x='MARRIAGE',data=df,stat='probability',hue='default.payment.next.month')
```

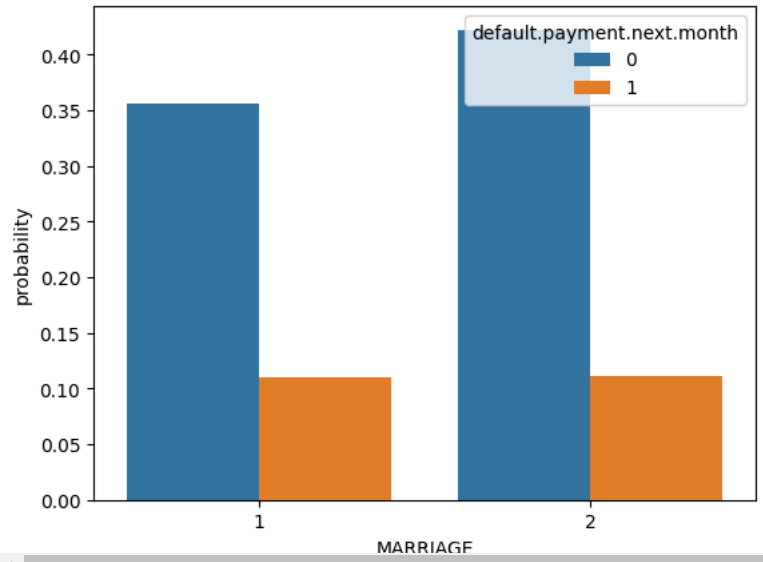
```
<Axes: xlabel='MARRIAGE', ylabel='probability'>
```




```
df.MARRIAGE=df.MARRIAGE.replace({0:2,3:1})
```

```
sns.countplot(x='MARRIAGE',data=df,stat='probability',hue='default.payment.next.month')
```

```
<Axes: xlabel='MARRIAGE', ylabel='probability'>
```



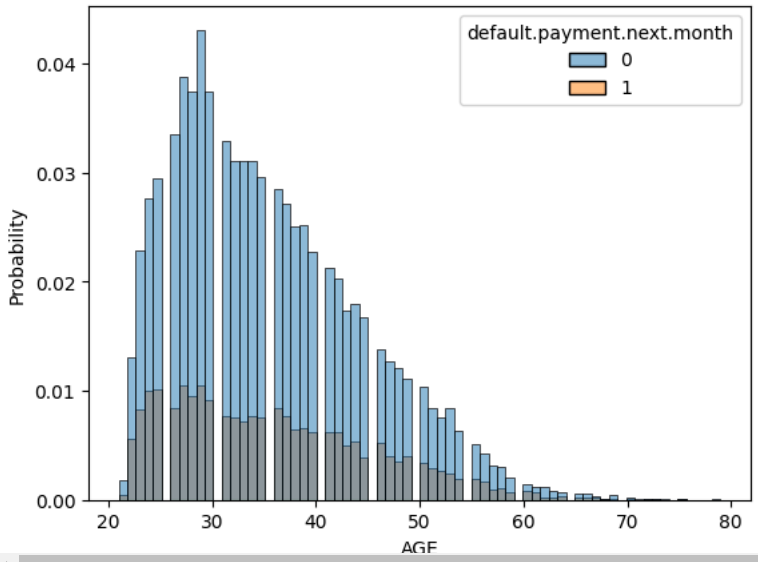

```
df.head()
```

ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	...	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_
1	20000.0	2	2	1	24	2	2	-1	-1	-2	...	0.0	0.0	0.0	0.0	6
2	120000.0	2	2	2	26	-1	2	0	0	0	...	3272.0	3455.0	3261.0	0.0	10
3	90000.0	2	2	2	34	0	0	0	0	0	...	14331.0	14948.0	15549.0	1518.0	15
4	50000.0	2	2	1	37	0	0	0	0	0	...	28314.0	28959.0	29547.0	2000.0	20
5	50000.0	1	2	1	57	-1	0	-1	0	0	...	20940.0	19146.0	19131.0	2000.0	366

5 rows × 24 columns

```
sns.histplot(x='AGE',data=df,stat='probability',hue='default.payment.next.month')
```

```
<Axes: xlabel='AGE', ylabel='Probability'>
```




```
(df['AGE']<=20).sum()
```

```
np.int64(0)
```

```
df.head()
```

ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	...	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_
1	20000.0	2	2	1	24	2	2	-1	-1	-2	...	0.0	0.0	0.0	0.0	6
2	120000.0	2	2	2	26	-1	2	0	0	0	...	3272.0	3455.0	3261.0	0.0	10
3	90000.0	2	2	2	34	0	0	0	0	0	...	14331.0	14948.0	15549.0	1518.0	15
4	50000.0	2	2	1	37	0	0	0	0	0	...	28314.0	28959.0	29547.0	2000.0	20
5	50000.0	1	2	1	57	-1	0	-1	0	0	...	20940.0	19146.0	19131.0	2000.0	366

5 rows × 24 columns

```
def rename_col(m,n):
    col1=list(df.columns[m:n])
    col2=[col1[len(col1)-1-i] for i in range(len(col1))]
    new_col={}
    for i in range(len(col1)):
        new_col[col1[i]]=col2[i]
    df.rename(columns=new_col,inplace=True)


rename_col(5,11)


rename_col(11,17),rename_col(17,23)
```

```
(None, None)
```

```
df.columns
```

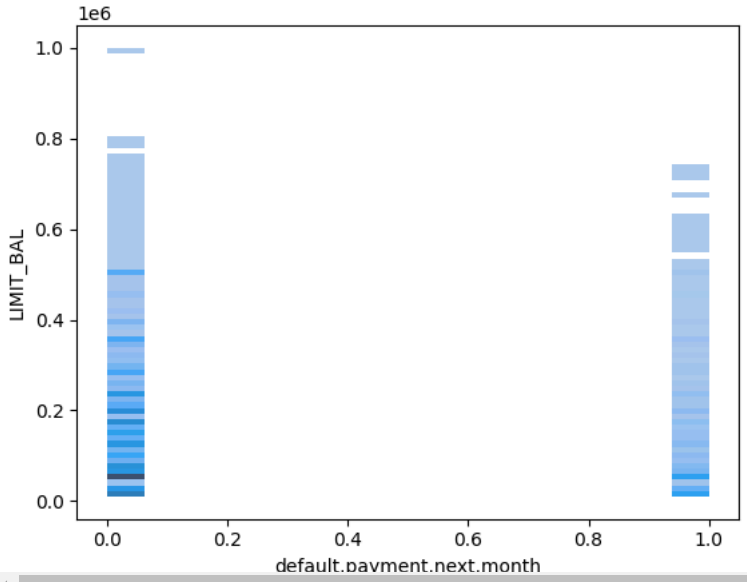
```
Index(['LIMIT_BAL', 'SEX', 'EDUCATION', 'MARRIAGE', 'AGE', 'PAY_6', 'PAY_5',
       'PAY_4', 'PAY_3', 'PAY_2', 'PAY_0', 'BILL_AMT6', 'BILL_AMT5',
       'BILL_AMT4', 'BILL_AMT3', 'BILL_AMT2', 'BILL_AMT1', 'PAY_AMT6',
       'PAY_AMT5', 'PAY_AMT4', 'PAY_AMT3', 'PAY_AMT2', 'PAY_AMT1',
       'default.payment.next.month'],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

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

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

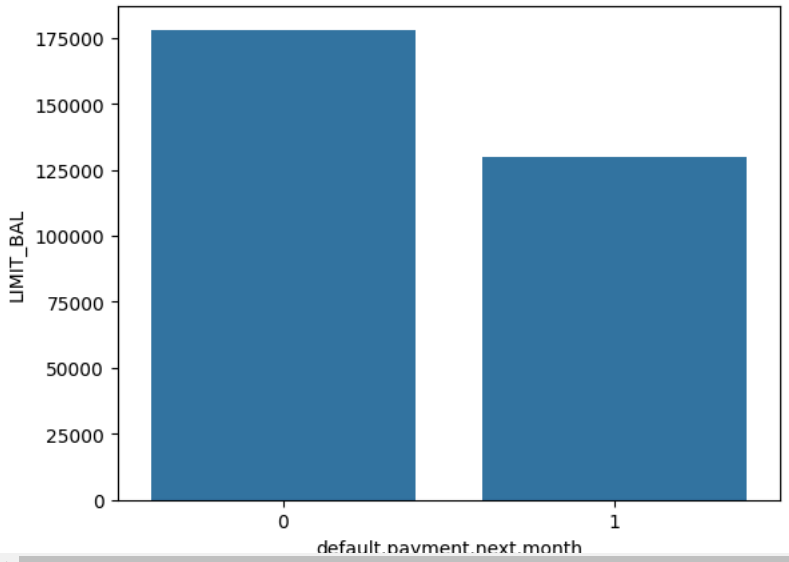
```
sns.histplot(data=df,y='LIMIT_BAL',x='default.payment.next.month')
```

```
<Axes: xlabel='default.payment.next.month', ylabel='LIMIT_BAL'>
```




```
sns.barplot(df.groupby('default.payment.next.month')['LIMIT_BAL'].mean())
```

```
<Axes: xlabel='default.payment.next.month', ylabel='LIMIT_BAL'>
```




```
df.head()
```

ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_6	PAY_5	PAY_4	PAY_3	PAY_2	...	BILL_AMT3	BILL_AMT2	BILL_AMT1	PAY_AMT6	PAY_
1	20000.0	2	2	1	24	2	2	-1	-1	-2	...	0.0	0.0	0.0	0.0	6
2	120000.0	2	2	2	26	-1	2	0	0	0	...	3272.0	3455.0	3261.0	0.0	10
3	90000.0	2	2	2	34	0	0	0	0	0	...	14331.0	14948.0	15549.0	1518.0	15
4	50000.0	2	2	1	37	0	0	0	0	0	...	28314.0	28959.0	29547.0	2000.0	20
5	50000.0	1	2	1	57	-1	0	-1	0	0	...	20940.0	19146.0	19131.0	2000.0	366

5 rows × 24 columns

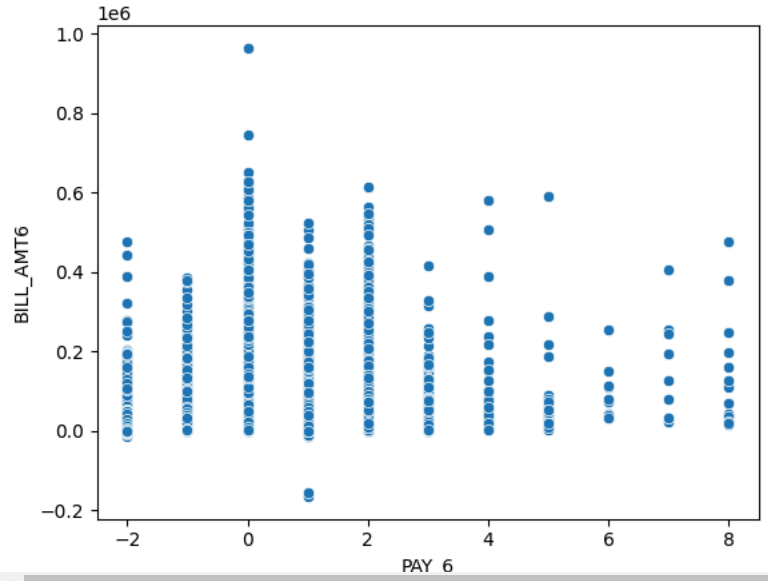
```
df.head()
```

ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_6	PAY_5	PAY_4	PAY_3	PAY_2	...	BILL_AMT3	BILL_AMT2	BILL_AMT1	PAY_AMT6	PAY_
1	20000.0	2	2	1	24	2	2	-1	-1	-2	...	0.0	0.0	0.0	0.0	6
2	120000.0	2	2	2	26	-1	2	0	0	0	...	3272.0	3455.0	3261.0	0.0	10
3	90000.0	2	2	2	34	0	0	0	0	0	...	14331.0	14948.0	15549.0	1518.0	15
4	50000.0	2	2	1	37	0	0	0	0	0	...	28314.0	28959.0	29547.0	2000.0	20
5	50000.0	1	2	1	57	-1	0	-1	0	0	...	20940.0	19146.0	19131.0	2000.0	366

5 rows × 24 columns

```
sns.scatterplot(data=df,x='PAY_6',y='BILL_AMT6')
```

```
<Axes: xlabel='PAY_6', ylabel='BILL_AMT6'>
```




```
df.head()
```

ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_6	PAY_5	PAY_4	PAY_3	PAY_2	...	BILL_AMT3	BILL_AMT2	BILL_AMT1	PAY_AMT6	PAY_
1	20000.0	2	2	1	24	2	2	-1	-1	-2	...	0.0	0.0	0.0	0.0	6
2	120000.0	2	2	2	26	-1	2	0	0	0	...	3272.0	3455.0	3261.0	0.0	10
3	90000.0	2	2	2	34	0	0	0	0	0	...	14331.0	14948.0	15549.0	1518.0	15
4	50000.0	2	2	1	37	0	0	0	0	0	...	28314.0	28959.0	29547.0	2000.0	20
5	50000.0	1	2	1	57	-1	0	-1	0	0	...	20940.0	19146.0	19131.0	2000.0	366

5 rows × 24 columns

```
df.MARRIAGE.value_counts()
```

```
MARRIAGE
2    16018
1    13982
Name: count, dtype: int64
```

Outlier Analysis

```
df.head()
```

ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_6	PAY_5	PAY_4	PAY_3	PAY_2	...	BILL_AMT3	BILL_AMT2	BILL_AMT1	PAY_AMT6	PAY_
1	20000.0	2	2	1	24	2	2	-1	-1	-2	...	0.0	0.0	0.0	0.0	6
2	120000.0	2	2	2	26	-1	2	0	0	0	...	3272.0	3455.0	3261.0	0.0	10
3	90000.0	2	2	2	34	0	0	0	0	0	...	14331.0	14948.0	15549.0	1518.0	15
4	50000.0	2	2	1	37	0	0	0	0	0	...	28314.0	28959.0	29547.0	2000.0	20
5	50000.0	1	2	1	57	-1	0	-1	0	0	...	20940.0	19146.0	19131.0	2000.0	366

5 rows × 24 columns

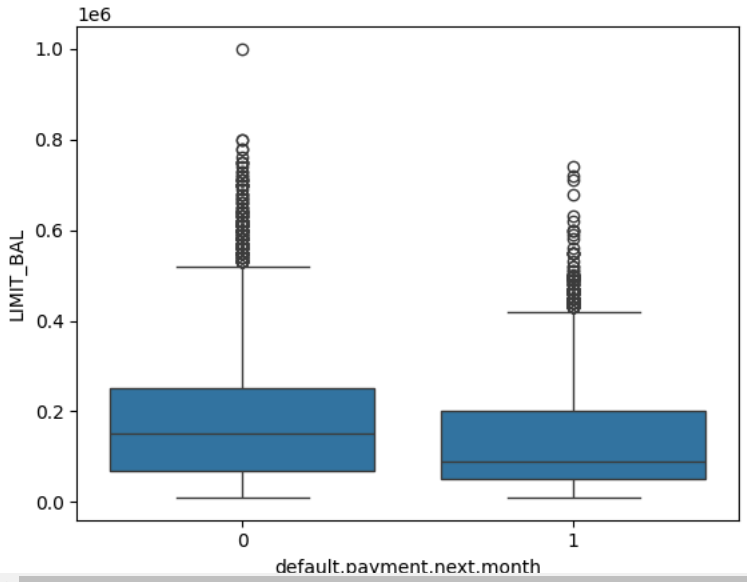
Limit_BAL

```
df['LIMIT_BAL'].describe()
```

```
count      30000.000000
mean      167484.322667
std       129747.661567
min        10000.000000
25%        50000.000000
50%       140000.000000
75%       240000.000000
max      1000000.000000
Name: LIMIT_BAL, dtype: float64
```

```
sns.boxplot(data=df,y='LIMIT_BAL',x='default.payment.next.month')
```

```
<Axes: xlabel='default.payment.next.month', ylabel='LIMIT_BAL'>
```




```
df=df[df['LIMIT_BAL']<=df['LIMIT_BAL'].quantile(0.95)]
```

```
df.shape
```

```
(28525, 24)
```

BILL_AMT

```
df[df.columns[11:17]].describe()
```

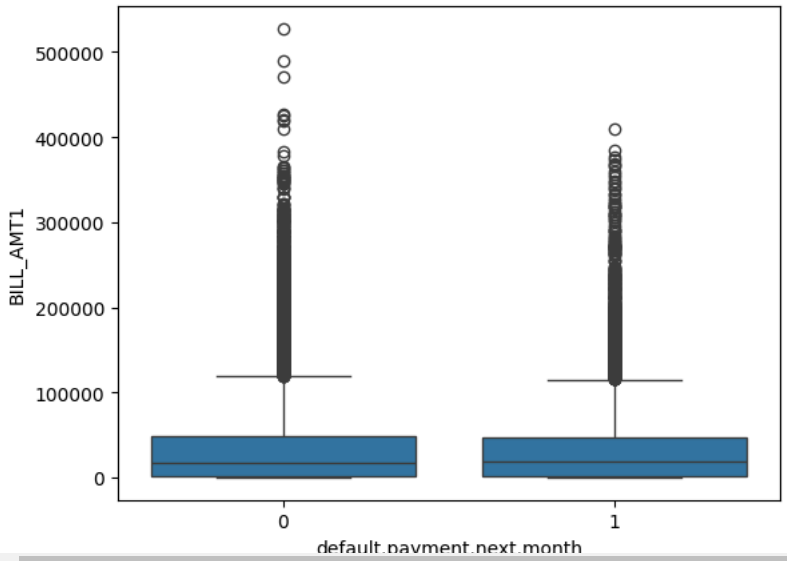
	BILL_AMT6	BILL_AMT5	BILL_AMT4	BILL_AMT3	BILL_AMT2	BILL_AMT1
count	28525.000000	28525.000000	28525.000000	28525.000000	28525.000000	28525.000000
mean	48450.145767	46610.273550	44364.406661	40691.959404	37896.754286	36528.553094
std	66121.557874	64065.059294	61879.194502	57576.881087	54391.475715	53313.536565
min	-165580.000000	-69777.000000	-157264.000000	-170000.000000	-81334.000000	-339603.000000
25%	3526.000000	2980.000000	2636.000000	2264.000000	1713.000000	1190.000000
50%	22285.000000	21120.000000	19993.000000	18917.000000	17990.000000	16893.000000
75%	64992.000000	61874.000000	58567.000000	51958.000000	49047.000000	48222.000000
max	626648.000000	605943.000000	855086.000000	628699.000000	547880.000000	527566.000000

```
for col in df.columns[11:17]:
    df[col+'_POS']=df[col].apply(lambda x:1 if x>=0 else 0)
```

```
for col in df.columns[11:17]:
    df[col]=abs(df[col])
```

```
sns.boxplot(data=df,y='BILL_AMT1',x='default.payment.next.month')
```

```
<Axes: xlabel='default.payment.next.month', ylabel='BILL_AMT1'>
```




```
def remove_outlier(col,df,per):
    return df[df[col]<=df[col].quantile(per)]
```

```
for col in df.columns[11:17]:
    df=remove_outlier(col,df)
```

```
df.head()
```

ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_6	PAY_5	PAY_4	PAY_3	PAY_2	...	PAY_AMT3	PAY_AMT2	PAY_AMT1	default.payment.n
1	20000.0	2	2	1	24	2	2	-1	-1	-2	...	0.0	0.0	0.0	
2	120000.0	2	2	2	26	-1	2	0	0	0	...	1000.0	0.0	2000.0	
3	90000.0	2	2	2	34	0	0	0	0	0	...	1000.0	1000.0	5000.0	
4	50000.0	2	2	1	37	0	0	0	0	0	...	1100.0	1069.0	1000.0	
5	50000.0	1	2	1	57	-1	0	-1	0	0	...	9000.0	689.0	679.0	

5 rows × 30 columns

```
df.shape
```

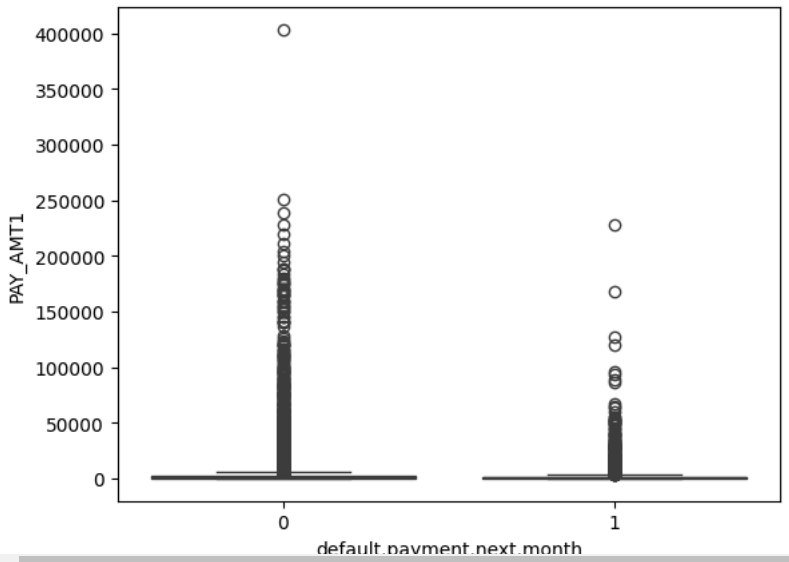
```
(20966, 30)
```

```
df[df.columns[17:23]].describe()
```

	PAY_AMT6	PAY_AMT5	PAY_AMT4	PAY_AMT3	PAY_AMT2	PAY_AMT1
count	20966.000000	20966.000000	20966.000000	20966.000000	20966.000000	20966.000000
mean	3459.566441	3357.832538	2888.183249	2594.256844	2462.851331	3354.394544
std	7833.611853	7921.655835	6614.282208	6394.292118	5741.754833	12104.207872
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	326.000000	291.000000	1.000000	0.000000	0.000000	0.000000
50%	1661.000000	1506.000000	1150.500000	1000.000000	1000.000000	953.000000
75%	3000.000000	3000.000000	2303.750000	2000.000000	2000.000000	2000.000000
max	201153.000000	344467.000000	184133.000000	200000.000000	231133.000000	403500.000000

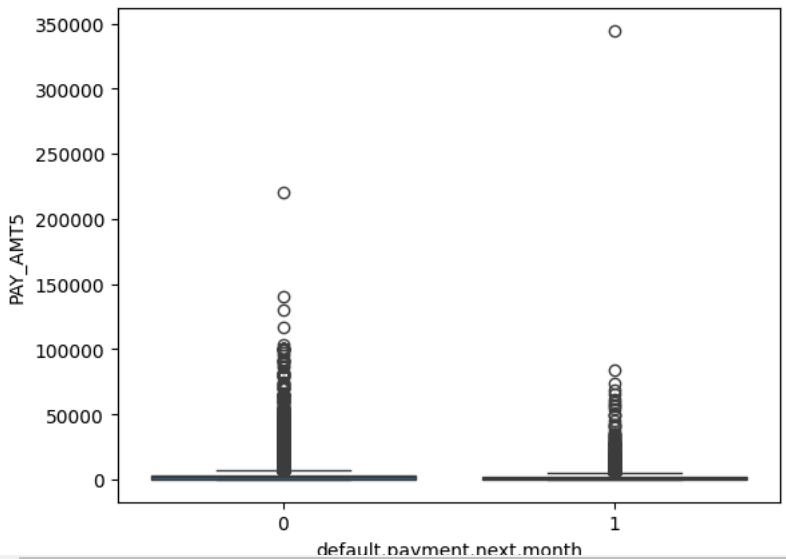
```
sns.boxplot(data=df,y='PAY_AMT1',x='default.payment.next.month')
```

```
<Axes: xlabel='default.payment.next.month', ylabel='PAY_AMT1'>
```




```
sns.boxplot(data=df,y='PAY_AMT5',x='default.payment.next.month')
```

```
<Axes: xlabel='default.payment.next.month', ylabel='PAY_AMT5'>
```




```
for col in df.columns[17:23]:
    df=remove_outlier(col,df,0.90)
```

```
df.head()
```

ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_6	PAY_5	PAY_4	PAY_3	PAY_2	...	PAY_AMT3	PAY_AMT2	PAY_AMT1	default.payment.n
1	20000.0	2	2	1	24	2	2	-1	-1	-2	...	0.0	0.0	0.0	
2	120000.0	2	2	2	26	-1	2	0	0	0	...	1000.0	0.0	2000.0	
3	90000.0	2	2	2	34	0	0	0	0	0	...	1000.0	1000.0	5000.0	