benhur Dabre

roll no-12

In [39]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix ,precision_score, recall_score, accuracy_
```

In [40]:
```
data = pd.read_csv("credit.csv" , skiprows=1)
data = data.sample(5000, random_state=36)
print(data.columns)
print(data.info())
print(data.describe())
print(data.head())
print(data.tail())
print(data.shape)
```

```
Index(['ID', 'LIMIT_BAL', 'GENDER', 'EDUCATION', 'MARRIAGE', 'AGE', 'PAY_0',
       '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')
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5000 entries, 17237 to 9437
Data columns (total 25 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   ID                          5000 non-null   int64
 1   LIMIT_BAL                   5000 non-null   int64
 2   GENDER                      5000 non-null   int64
 3   EDUCATION                   5000 non-null   int64
 4   MARRIAGE                    5000 non-null   int64
 5   AGE                         5000 non-null   int64
 6   PAY_0                       5000 non-null   int64
 7   PAY_2                       5000 non-null   int64
 8   PAY_3                       5000 non-null   int64
 9   PAY_4                       5000 non-null   int64
 10  PAY_5                       5000 non-null   int64
 11  PAY_6                       5000 non-null   int64
 12  BILL_AMT1                   5000 non-null   int64
 13  BILL_AMT2                   5000 non-null   int64
 14  BILL_AMT3                   5000 non-null   int64
 15  BILL_AMT4                   5000 non-null   int64
 16  BILL_AMT5                   5000 non-null   int64
 17  BILL_AMT6                   5000 non-null   int64
 18  PAY_AMT1                    5000 non-null   int64
 19  PAY_AMT2                    5000 non-null   int64
 20  PAY_AMT3                    5000 non-null   int64
 21  PAY_AMT4                    5000 non-null   int64
 22  PAY_AMT5                    5000 non-null   int64
 23  PAY_AMT6                    5000 non-null   int64
 24  default payment next month  5000 non-null   int64
dtypes: int64(25)
memory usage: 1015.6 KB
None
                 ID      LIMIT_BAL       GENDER    EDUCATION     MARRIAGE  \
count   5000.000000    5000.000000  5000.000000  5000.000000  5000.000000
mean   15104.530200  166501.536000     1.598600     1.869200     1.549600
std     8665.660087  129135.620755     0.490231     0.791844     0.526493
min        3.000000   10000.000000     1.000000     0.000000     0.000000
25%     7727.500000   50000.000000     1.000000     1.000000     1.000000
50%    15019.500000  140000.000000     2.000000     2.000000     2.000000
75%    22686.500000  240000.000000     2.000000     2.000000     2.000000
max    29995.000000  750000.000000     2.000000     6.000000     3.000000

               AGE        PAY_0        PAY_2      PAY_3        PAY_4  ...  \
count  5000.000000  5000.000000  5000.000000  5000.0000  5000.000000  ...
mean     35.460000    -0.025800    -0.149400    -0.1694    -0.224600  ...
std       9.069049     1.118653     1.186154     1.1897     1.156902  ...
min      21.000000    -2.000000    -2.000000    -2.0000    -2.000000  ...
25%      28.000000    -1.000000    -1.000000    -1.0000    -1.000000  ...
50%      34.000000     0.000000     0.000000     0.0000     0.000000  ...
75%      42.000000     0.000000     0.000000     0.0000     0.000000  ...
max      74.000000     8.000000     7.000000     7.0000     7.000000  ...

           BILL_AMT4      BILL_AMT5     BILL_AMT6      PAY_AMT1  \
```

```
count    5000.000000    5000.000000    5000.00000    5000.00000
mean    42603.721600   39867.246000   38194.15940    5736.27900
std     62896.573856   59228.587526   57984.93723   15952.80187
min    -15910.000000  -30481.000000 -339603.00000       0.00000
25%      2392.000000    2000.000000    1380.25000    1000.00000
50%     19331.500000   18668.500000   17484.00000    2100.00000
75%     53576.500000   49836.000000   48449.50000    5000.00000
max    548020.000000  530672.000000  511905.00000  405016.00000

            PAY_AMT2      PAY_AMT3       PAY_AMT4       PAY_AMT5  \
count    5000.000000    5000.00000    5000.000000    5000.000000
mean     5510.448600    5243.60140    4666.822600    4646.791000
std     16968.914645   16314.14835   13616.969163   14794.580513
min         0.000000       0.00000       0.000000       0.000000
25%       805.500000     393.50000     325.000000     246.000000
50%      2002.500000    1800.00000    1500.000000    1500.000000
75%      5000.000000    4234.75000    4000.000000    4000.000000
max    580464.000000  380478.00000  265852.000000  388071.000000

            PAY_AMT6  default payment next month
count    5000.000000                 5000.000000
mean     5141.630600                    0.215800
std     17159.241133                    0.411417
min         0.000000                    0.000000
25%        65.000000                    0.000000
50%      1500.000000                    0.000000
75%      4000.000000                    0.000000
max    345293.000000                    1.000000

[8 rows x 25 columns]
          ID  LIMIT_BAL  GENDER  EDUCATION  MARRIAGE  AGE  PAY_0  PAY_2  \
17237  17238     170000       2          3         1   40      0      0
10971  10972     340000       2          1         1   33      0      0
8579    8580      30000       2          2         1   47      0      0
792      793     410000       1          1         2   32     -1     -1
17953  17954     200000       1          3         1   46     -2     -2

       PAY_3  PAY_4  ...  BILL_AMT4  BILL_AMT5  BILL_AMT6  PAY_AMT1  PAY_AMT2  \
17237      0      0  ...      60343      61176      62092      2128      2729
10971      0     -2  ...          0       7687      21200      6018         0
8579       0      0  ...      58287      29522      30103      1505      1457
792       -1     -2  ...       -281       -281       -281      5400         0
17953     -2     -2  ...          0          0          0         0         0

       PAY_AMT3  PAY_AMT4  PAY_AMT5  PAY_AMT6  default payment next month
17237      4000      2169      2241      2258                           0
10971         0      7687     21200     10600                           0
8579       1016      1207      1600      1000                           0
792           0         0         0         0                           0
17953         0         0         0         0                           0

[5 rows x 25 columns]
          ID  LIMIT_BAL  GENDER  EDUCATION  MARRIAGE  AGE  PAY_0  PAY_2  \
25634  25635      50000       2          3         1   31      0      0
200      201     180000       2          1         1   38     -2     -2
978      979     180000       1          2         2   26      0      0
4490    4491     100000       2          2         2   23      0      0
9437    9438      20000       2          2         2   28      2      2

       PAY_3  PAY_4  ...  BILL_AMT4  BILL_AMT5  BILL_AMT6  PAY_AMT1  PAY_AMT2  \
```

```
25634      0      0  ...      27880      27935      26314      1914      1900
200       -2     -2  ...          0          0          0         0         0
978        0      0  ...       7733       6794       5487      1126      4081
4490       0      0  ...      18306      20594      26368      1258      1255
9437       0      0  ...      18859      19828      36666         0      1200

       PAY_AMT3  PAY_AMT4  PAY_AMT5  PAY_AMT6  default payment next month
25634      1478      1172      2005      1000                           0
200           0         0         0         0                           0
978         240       504       169        62                           1
4490       3000      3000      7000      1225                           0
9437       7500      1015     15000         0                           1

[5 rows x 25 columns]
(5000, 25)
```

In [41]:
```python
data.isnull().sum()
```

Out[41]:
```
ID                            0
LIMIT_BAL                     0
GENDER                        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
```

In [42]:
```python
print(data.dtypes)
```

```
ID                            int64
LIMIT_BAL                     int64
GENDER                        int64
EDUCATION                     int64
MARRIAGE                      int64
AGE                           int64
PAY_0                         int64
PAY_2                         int64
PAY_3                         int64
PAY_4                         int64
PAY_5                         int64
PAY_6                         int64
BILL_AMT1                     int64
BILL_AMT2                     int64
BILL_AMT3                     int64
BILL_AMT4                     int64
BILL_AMT5                     int64
BILL_AMT6                     int64
PAY_AMT1                      int64
PAY_AMT2                      int64
PAY_AMT3                      int64
PAY_AMT4                      int64
PAY_AMT5                      int64
PAY_AMT6                      int64
default payment next month    int64
dtype: object
```

In [43]:

```
y=data['default payment next month']
x=data[['BILL_AMT1','BILL_AMT2']]

print(data.dtypes)
print(data.head())
```

```
ID                            int64
LIMIT_BAL                     int64
GENDER                        int64
EDUCATION                     int64
MARRIAGE                      int64
AGE                           int64
PAY_0                         int64
PAY_2                         int64
PAY_3                         int64
PAY_4                         int64
PAY_5                         int64
PAY_6                         int64
BILL_AMT1                     int64
BILL_AMT2                     int64
BILL_AMT3                     int64
BILL_AMT4                     int64
BILL_AMT5                     int64
BILL_AMT6                     int64
PAY_AMT1                      int64
PAY_AMT2                      int64
PAY_AMT3                      int64
PAY_AMT4                      int64
PAY_AMT5                      int64
PAY_AMT6                      int64
default payment next month    int64
dtype: object
          ID  LIMIT_BAL  GENDER  EDUCATION  MARRIAGE  AGE  PAY_0  PAY_2  \
17237  17238     170000       2          3         1   40      0      0
10971  10972     340000       2          1         1   33      0      0
8579    8580      30000       2          2         1   47      0      0
792      793     410000       1          1         2   32     -1     -1
17953  17954     200000       1          3         1   46     -2     -2

       PAY_3  PAY_4  ...  BILL_AMT4  BILL_AMT5  BILL_AMT6  PAY_AMT1  PAY_AMT2  \
17237      0      0  ...      60343      61176      62092      2128      2729
10971      0     -2  ...          0       7687      21200      6018         0
8579       0      0  ...      58287      29522      30103      1505      1457
792       -1     -2  ...       -281       -281       -281      5400         0
17953     -2     -2  ...          0          0          0         0         0

       PAY_AMT3  PAY_AMT4  PAY_AMT5  PAY_AMT6  default payment next month
17237      4000      2169      2241      2258                           0
10971         0      7687     21200     10600                           0
8579       1016      1207      1600      1000                           0
792           0         0         0         0                           0
17953         0         0         0         0                           0

[5 rows x 25 columns]
```

In [44]:
```python
x_train, x_test, y_train, y_test = train_test_split( x,y,test_size=0.3,random_state=42
```

In [45]:
```python
scaler=StandardScaler()
x_train = scaler.fit_transform(x_train)
x_test=scaler.transform(x_test)
```

In [50]:
```python
models = {
    "Linear SVM ": SVC(kernel='linear', C= 1, class_weight='balanced'),

    "Polynomial SVM" : SVC (kernel='poly',
```

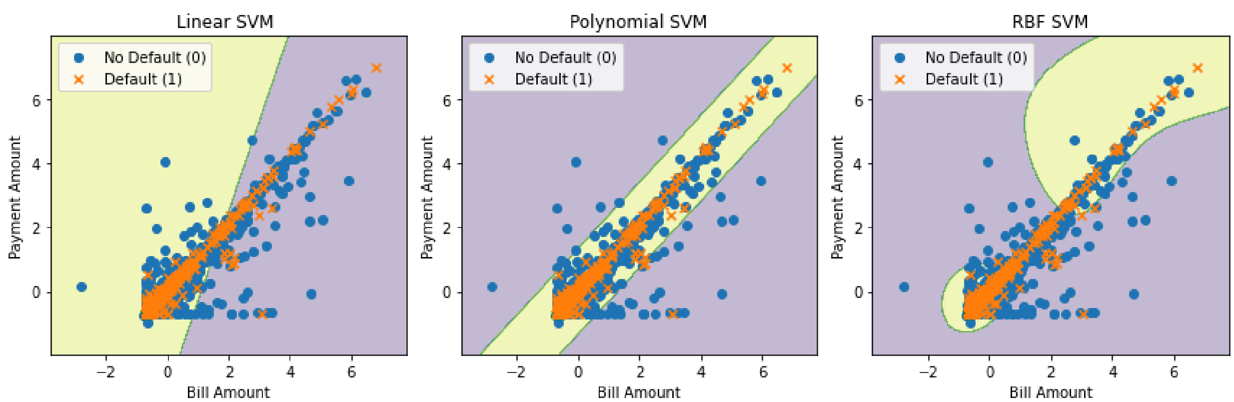
```python
                                degree=2,
                                C=1,
                                gamma='scale',
                                class_weight='balanced' ),
    "RBF SVM" : SVC(kernel = 'rbf',
                    C=5,
                    gamma = 0.1,
                    class_weight = 'balanced' )}

for name, model in models.items():
    model.fit(x_train,y_train)
    y_pred = model.predict(x_test)


    print("\n",name)
    print("Confusion Matrix;",confusion_matrix(y_test,y_pred))
    print("precision score:",precision_score(y_test,y_pred, zero_division=0))
    print("recall score:",recall_score(y_test,y_pred))
    print("f1 score:",f1_score(y_test,y_pred))
    print("Accuracy train score:",accuracy_score(y_test,y_pred))
    print("Accuracy test score:",accuracy_score(y_train,y_pred_train))
```

```
 Linear SVM
Confusion Matrix; [[  92 1093]
 [  13  302]]
precision score: 0.2164874551971326
recall score: 0.9587301587301588
f1 score: 0.3532163742690059
Accuracy train score: 0.26266666666666666
Accuracy test score: 0.3345714285714286

 Polynomial SVM
Confusion Matrix; [[  16 1169]
 [   6  309]]
precision score: 0.2090663058186739
recall score: 0.9809523809523809
f1 score: 0.3446737311767987
Accuracy train score: 0.21666666666666667
Accuracy test score: 0.3345714285714286

 RBF SVM
Confusion Matrix; [[216 969]
 [ 38 277]]
precision score: 0.22231139646869985
recall score: 0.8793650793650793
f1 score: 0.3549007046764895
Accuracy train score: 0.32866666666666666
Accuracy test score: 0.3345714285714286
```

In [51]:

```python
def plot_boundary(model, title):
    h = 0.02
    x_min, x_max = x_train[:, 0].min() - 1, x_train[:, 0].max() + 1
    y_min, y_max = x_train[:, 1].min() - 1, x_train[:, 1].max() + 1

    xx, yy = np.meshgrid(
        np.arange(x_min, x_max, h),
        np.arange(y_min, y_max, h)
    )

    Z = model.predict(np.c_[xx.ravel(), yy.ravel()])
```

```python
    Z = Z.reshape(xx.shape)

    plt.contourf(xx, yy, Z, alpha=0.3)

    plt.scatter(x_train[y_train == 0, 0],
                x_train[y_train == 0, 1],
                label='No Default (0)',
                marker='o')

    plt.scatter(x_train[y_train == 1, 0],
                x_train[y_train == 1, 1],
                label='Default (1)',
                marker='x')

    plt.legend()
    plt.title(title)
    plt.xlabel("Bill Amount")
    plt.ylabel("Payment Amount")

plt.figure(figsize=(12, 4))

for i, (name, model) in enumerate(models.items()):
    plt.subplot(1,3,i+1)
    model.fit(x_train, y_train)
    plot_boundary(model, name)

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



In [ ]: