

Importing Libraries & Loading Data

```
In [1]:  import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
In [2]:  df = pd.read_csv('CreditCardDefault.csv')
```

```
In [3]:  type(df)
```

Out[3]: pandas.core.frame.DataFrame

```
In [4]:  df.shape
```

Out[4]: (30000, 25)

```
In [5]:  df.head()
```

Out[5]:

	ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	...	BI
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	...	

5 rows × 25 columns

In [6]: `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  next_month_payment      30000 non-null  int64
dtypes: int64(25)
memory usage: 5.7 MB
```

In [7]: `df.describe()`

Out[7]:

	ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE
count	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000
mean	15000.500000	167484.322667	1.603733	1.853133	1.551867	35.485500
std	8660.398374	129747.661567	0.489129	0.790349	0.521970	9.217904
min	1.000000	10000.000000	1.000000	0.000000	0.000000	21.000000
25%	7500.750000	50000.000000	1.000000	1.000000	1.000000	28.000000
50%	15000.500000	140000.000000	2.000000	2.000000	2.000000	34.000000
75%	22500.250000	240000.000000	2.000000	2.000000	2.000000	41.000000
max	30000.000000	1000000.000000	2.000000	6.000000	3.000000	79.000000

8 rows × 25 columns

Feature Engineering

```
In [8]: X = df.iloc[:, :-1]
        y = pd.DataFrame(df.iloc[:, -1])
```

```
In [9]: X.shape, y.shape
```

```
Out[9]: ((30000, 24), (30000, 1))
```

```
In [10]: from sklearn.model_selection import train_test_split
         from sklearn.preprocessing import StandardScaler
```

```
In [11]: scaler = StandardScaler()
```

```
In [12]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, str
```

```
In [13]: X_train = scaler.fit_transform(X_train)
```

```
In [14]: X_test = scaler.transform(X_test)
```

Model Building & Evaluation

```
In [15]: from sklearn.svm import SVC
         from sklearn.naive_bayes import GaussianNB
         from sklearn.tree import DecisionTreeClassifier
         from sklearn.linear_model import LogisticRegression
         from mlxtend.classifier import StackingClassifier
```

```
In [16]: base_model1 = SVC(probability=True)
         base_model2 = GaussianNB()
         base_model3 = DecisionTreeClassifier()
         meta_model = LogisticRegression()
```

```
In [17]: stack_model = StackingClassifier(classifiers=[base_model1, base_model2, base_
```

In [18]: `stack_model.fit(X_train, y_train)`

```
C:\Users\15516\anaconda3\lib\site-packages\sklearn\utils\validation.py:760:
DataConversionWarning: A column-vector y was passed when a 1d array was exp
ected. Please change the shape of y to (n_samples, ), for example using ravel().
    y = column_or_1d(y, warn=True)
C:\Users\15516\anaconda3\lib\site-packages\sklearn\naive_bayes.py:206: Data
ConversionWarning: A column-vector y was passed when a 1d array was expecte
d. Please change the shape of y to (n_samples, ), for example using ravel
().
    y = column_or_1d(y, warn=True)
C:\Users\15516\anaconda3\lib\site-packages\sklearn\utils\validation.py:760:
DataConversionWarning: A column-vector y was passed when a 1d array was exp
ected. Please change the shape of y to (n_samples, ), for example using ravel().
    y = column_or_1d(y, warn=True)
```

```
Out[18]: StackingClassifier(average_probab=False,
                             classifiers=[SVC(C=1.0, break_ties=False, cache_size=20
0,
                                     class_weight=None, coef0=0.0,
                                     decision_function_shape='ovr', degree=
3,
                                     gamma='scale', kernel='rbf', max_iter=-
1,
                                     probability=True, random_state=None,
                                     shrinking=True, tol=0.001, verbose=False),
                             GaussianNB(priors=None, var_smoothing=1e-0
9),
                             DecisionTreeClassifier(ccp_alpha=0.001,
meta_classifier=LogisticRegression(C=1.0, class_weight=None,
dual=False,
fit_intercept=True,
intercept_scaling=1,
l1_ratio=None,
max_iter=100,
multi_class='auto',
n_jobs=None, penalty
='l2',
random_state=None,
solver='lbfgs',
tol=0.0001, verbose=
0,
warm_start=False),
store_train_meta_features=False, use_clones=True,
use_features_in_secondary=False, use_probab=False,
verbose=0)
```

In [19]: `y_pred = stack_model.predict(X_test)`

```
In [20]: ▶ from sklearn.metrics import accuracy_score, confusion_matrix
```

```
In [21]: ▶ accuracy_score(y_test, y_pred)
```

```
Out[21]: 0.7206666666666667
```

```
In [22]: ▶ confusion_matrix(y_test, y_pred)
```

```
Out[22]: array([[5706, 1303],  
               [1211,  780]], dtype=int64)
```

```
In [ ]: ▶
```