

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 xgboost import XGBClassifier
```

```
In [16]: xgb = XGBClassifier(max_depth = 2,
                             learning_rate = 0.2,
                             objective = "multi:softmax",
                             num_class = 2,
                             booster = "gbtree",
                             n_estimators = 10,
                             random_state = 123)
```

In [17]: `xgb.fit(X_train, y_train)`

```
C:\Users\15516\anaconda3\lib\site-packages\sklearn\preprocessing\_label.py:
235: DataConversionWarning: A column-vector y was passed when a 1d array was
expected. 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\preprocessing\_label.py:
268: DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
```

```
y = column_or_1d(y, warn=True)
```

```
[11:53:56] WARNING: C:\Users\Administrator\workspace\xgboost-win64_release_
1.1.0\src\learner.cc:480:
```

```
Parameters: { n_estimators } might not be used.
```

This may not be accurate due to some parameters are only used in language bindings but

passed down to XGBoost core. Or some parameters are not used but slip through this

verification. Please open an issue if you find above cases.

```
Out[17]: XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
                      colsample_bynode=1, colsample_bytree=1, gamma=0, gpu_id=-1,
                      importance_type='gain', interaction_constraints='',
                      learning_rate=0.2, max_delta_step=0, max_depth=2,
                      min_child_weight=1, missing=nan, monotone_constraints=('',),
                      n_estimators=10, n_estimators=100, n_jobs=0, num_class=2,
                      num_parallel_tree=1, objective='multi:softmax', random_state=
123,
                      reg_alpha=0, reg_lambda=1, scale_pos_weight=None, subsample=
1,
                      tree_method='exact', validate_parameters=1, verbosity=None)
```

In [18]: `y_pred = xgb.predict(X_test)`

In [19]: `from sklearn.metrics import accuracy_score, confusion_matrix`

In [20]: `accuracy_score(y_test, y_pred)`

```
Out[20]: 0.8181111111111111
```

In [21]: `confusion_matrix(y_test, y_pred)`

```
Out[21]: array([[6665,  344],
                [1293,  698]], dtype=int64)
```

In []:

