Objectives

1. Loading data.

2. Checking for missing values.

3. Splitting the data in to train set and test set.

In [1]:

```
# Import pandas and load the dataset into a dataframe
import pandas as pd

df = pd.read_csv("Churn-Modelling.csv")
print(df.head())
```

```
   CreditScore  Age  Tenure    Balance  HasCrCard     Salary  Exited
0          619   42       2       0.00          1  101348.88       1
1          608   41       1   83807.86          0  112542.58       0
2          502   42       8  159660.80          1  113931.57       1
3          699   39       1       0.00          0   93826.63       0
4          850   43       2  125510.82          1   79084.10       0
```

pandas.isna(object)

Detects missing values.

Returns a boolen object of same size.

None and np.NaN are mapped True values. Everything else gets mapped to False values.

In [2]:

```
# Check for missing values
pd.isna(df).sum()
```

Out[2]:

```
CreditScore    0
Age            0
Tenure         0
Balance        0
HasCrCard      0
Salary         0
Exited         0
dtype: int64
```

In [3]:

```
print(df.describe())
```

```
        CreditScore           Age        Tenure        Balance    HasCrCard  \
count  10000.000000  10000.000000  10000.000000   10000.000000  10000.00000
mean     650.528800     38.921800      5.012800   76485.889288      0.70550
std       96.653299     10.487806      2.892174   62397.405202      0.45584
min      350.000000     18.000000      0.000000       0.000000      0.00000
25%      584.000000     32.000000      3.000000       0.000000      0.00000
50%      652.000000     37.000000      5.000000   97198.540000      1.00000
75%      718.000000     44.000000      7.000000  127644.240000      1.00000
max      850.000000     92.000000     10.000000  250898.090000      1.00000

              Salary        Exited
count   10000.000000  10000.000000
mean   100090.239881      0.203700
std     57510.492818      0.402769
min        11.580000      0.000000
25%     51002.110000      0.000000
50%    100193.915000      0.000000
75%    149388.247500      0.000000
max    199992.480000      1.000000
```

In [4]:
```
# Seperate the input features and target values.
x = df.iloc[:, 0:-1]
y = df.iloc[:, -1]
print(x.head())
print(y.head())
```

```
   CreditScore  Age  Tenure    Balance  HasCrCard     Salary
0          619   42       2       0.00          1  101348.88
1          608   41       1   83807.86          0  112542.58
2          502   42       8  159660.80          1  113931.57
3          699   39       1       0.00          0   93826.63
4          850   43       2  125510.82          1   79084.10
0    1
1    0
2    1
3    0
4    0
Name: Exited, dtype: int64
```

x_train, x_test, y_tran, y_test = train_test_split(x, y, test_size)

Defined in sklearn.model_selection

test_size attribute decides the proportion of split

In [5]:
```
# Split the data

from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2)

print(x_train.describe())
print(x_test.describe())
```

```
       CreditScore          Age       Tenure        Balance    HasCrCard  \
count  8000.000000  8000.000000  8000.000000    8000.000000  8000.000000
mean    650.789625    38.913625     5.008500   76682.011179     0.705000
std      97.421530    10.475919     2.883915   62382.992654     0.456071
min     350.000000    18.000000     0.000000       0.000000     0.000000
25%     584.000000    32.000000     3.000000       0.000000     0.000000
50%     652.000000    37.000000     5.000000   97267.100000     1.000000
75%     718.000000    44.000000     7.000000  127843.105000     1.000000
max     850.000000    92.000000    10.000000  250898.090000     1.000000

              Salary
count    8000.000000
mean    99750.823051
std     57497.976117
min        11.580000
25%     50606.752500
50%     99462.905000
75%    149068.075000
max    199992.480000
       CreditScore          Age       Tenure        Balance    HasCrCard  \
count  2000.000000  2000.000000  2000.000000    2000.000000  2000.000000
mean    649.485500    38.954500     5.030000   75701.401725     0.707500
std      93.533796    10.537787     2.925642   62464.474066     0.455024
min     350.000000    18.000000     0.000000       0.000000     0.000000
25%     584.000000    32.000000     2.000000       0.000000     0.000000
50%     651.000000    37.000000     5.000000   96932.590000     1.000000
75%     714.000000    44.000000     8.000000  126564.682500     1.000000
max     850.000000    83.000000    10.000000  211774.310000     1.000000

             Salary
count    2000.00000
mean   101447.90720
std     57554.89882
min       142.81000
25%     52387.84500
50%    102079.90500
75%    150811.05250
max    199970.74000
```

In [ ]:
```

```