# Practical 1: Simple Linear Regression

In [52]:
```
from sklearn import linear_model
import pandas as pd
import matplotlib.pyplot as plt
```

In [53]:
```
#reading csv file
df = pd.read_csv('Salary_Data.csv')
df
```

Out[53]:

| | YearsExperience | Salary |
|---|---|---|
| 0 | 1.1 | 39343.0 |
| 1 | 1.3 | 46205.0 |
| 2 | 1.5 | 37731.0 |
| 3 | 2.0 | 43525.0 |
| 4 | 2.2 | 39891.0 |
| 5 | 2.9 | 56642.0 |
| 6 | 3.0 | 60150.0 |
| 7 | 3.2 | 54445.0 |
| 8 | 3.2 | 64445.0 |
| 9 | 3.7 | 57189.0 |
| 10 | 3.9 | 63218.0 |
| 11 | 4.0 | 55794.0 |
| 12 | 4.0 | 56957.0 |
| 13 | 4.1 | 57081.0 |
| 14 | 4.5 | 61111.0 |
| 15 | 4.9 | 67938.0 |
| 16 | 5.1 | 66029.0 |
| 17 | 5.3 | 83088.0 |
| 18 | 5.9 | 81363.0 |
| 19 | 6.0 | 93940.0 |
| 20 | 6.8 | 91738.0 |
| 21 | 7.1 | 98273.0 |
| 22 | 7.9 | 101302.0 |
| 23 | 8.2 | 113812.0 |
| 24 | 8.7 | 109431.0 |
| 25 | 9.0 | 105582.0 |
| 26 | 9.5 | 116969.0 |
| 27 | 9.6 | 112635.0 |
| 28 | 10.3 | 122391.0 |
| 29 | 10.5 | 121872.0 |

In [54]:
```
X = df.iloc[:, :-1].values
y = df.iloc[:, 1].values
```

In [55]:
```
X
```

Out[55]:
```
array([[ 1.1],
       [ 1.3],
       [ 1.5],
       [ 2. ],
       [ 2.2],
       [ 2.9],
       [ 3. ],
       [ 3.2],
       [ 3.2],
       [ 3.7],
       [ 3.9],
       [ 4. ],
       [ 4. ],
       [ 4.1],
       [ 4.5],
       [ 4.9],
       [ 5.1],
       [ 5.3],
       [ 5.9],
       [ 6. ],
       [ 6.8],
       [ 7.1],
       [ 7.9],
       [ 8.2],
       [ 8.7],
       [ 9. ],
       [ 9.5],
       [ 9.6],
       [10.3],
       [10.5]])
```

In [56]:
```
y
```

Out[56]:
```
array([ 39343.,  46205.,  37731.,  43525.,  39891.,  56642.,  60150.,
        54445.,  64445.,  57189.,  63218.,  55794.,  56957.,  57081.,
        61111.,  67938.,  66029.,  83088.,  81363.,  93940.,  91738.,
        98273., 101302., 113812., 109431., 105582., 116969., 112635.,
       122391., 121872.])
```

In [57]:
```
from sklearn.model_selection import train_test_split
```

In [58]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

In [59]:
```
X_train
```

Out[59]:
```
array([[ 4. ],
       [ 5.3],
       [10.3],
       [ 9.6],
       [ 6. ],
       [ 7.9],
       [ 1.1],
       [ 3.2],
       [ 5.1],
       [ 8.7],
       [ 9. ],
       [ 7.1],
       [ 6.8],
       [ 3. ],
       [ 2. ],
       [ 2.9],
       [ 5.9],
       [ 8.2],
       [ 9.5],
       [10.5],
       [ 4. ],
       [ 4.9],
       [ 1.3],
       [ 4.1]])
```

In [60]:
```
X_test
```

Out[60]:
```
array([[3.7],
       [4.5],
       [3.2],
       [2.2],
       [1.5],
       [3.9]])
```

In [61]:
```
y_train
```

Out[61]:
```
array([ 55794.,  83088., 122391., 112635.,  93940., 101302.,  39343.,
        54445.,  66029., 109431., 105582.,  98273.,  91738.,  60150.,
        43525.,  56642.,  81363., 113812., 116969., 121872.,  56957.,
        67938.,  46205.,  57081.])
```

In [62]:
```
y_test
```

Out[62]:
```
array([57189., 61111., 64445., 39891., 37731., 63218.])
```

In [63]:
```
# training the model
reg_model = linear_model.LinearRegression()
reg_model.fit(X_train, y_train)
```

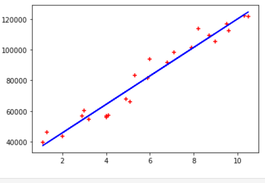
Out[63]:
```
LinearRegression()
```

In [64]:
```
y_train_predict = reg_model.predict(X_train)
y_test_predict = reg_model.predict(X_test)
```

In [65]:
```
%matplotlib inline
```

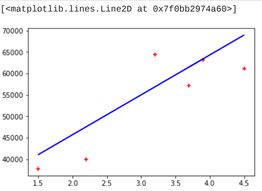
In [69]:
```
# plt.xlabel("Years ")
plt.scatter(X_train, y_train, color='red', marker='+')
plt.plot(X_train, y_train_predict, color='blue')
```

Out[69]:
```
[<matplotlib.lines.Line2D at 0x7f0bb2990d60>]
```




In [70]:
```
plt.scatter(X_test, y_test, color='red', marker='+')
plt.plot(X_test, y_test_predict, color='blue')
```

Out[70]:
```
[<matplotlib.lines.Line2D at 0x7f0bb2974a60>]
```




In [ ]:
```
```

# Practical 2: Multiple Linear Regression

In [8]:
```
from matplotlib import pyplot as plt
import pandas as pd
import numpy as np

df = pd.read_csv('50_Startups-2.csv')
df.head()
```

Out[8]:

| | R&D Spend | Administration | Marketing Spend | State | Profit |
|---|---|---|---|---|---|
| **0** | 165349.20 | 136897.80 | 471784.10 | New York | 192261.83 |
| **1** | 162597.70 | 151377.59 | 443898.53 | California | 191792.06 |
| **2** | 153441.51 | 101145.55 | 407934.54 | Florida | 191050.39 |
| **3** | 144372.41 | 118671.85 | 383199.62 | New York | 182901.99 |
| **4** | 142107.34 | 91391.77 | 366168.42 | Florida | 166187.94 |

In [9]:
```
X = df.iloc[:, :4].values # first 4 col
y = df.iloc[:, -1].values # last col
```

In [10]:
```
X
```

Out[10]:
```
array([[165349.2, 136897.8, 471784.1, 'New York'],
       [162597.7, 151377.59, 443898.53, 'California'],
       [153441.51, 101145.55, 407934.54, 'Florida'],
       [144372.41, 118671.85, 383199.62, 'New York'],
       [142107.34, 91391.77, 366168.42, 'Florida'],
       [131876.9, 99814.71, 362861.36, 'New York'],
       [134615.46, 147198.87, 127716.82, 'California'],
       [130298.13, 145530.06, 323876.68, 'Florida'],
       [120542.52, 148718.95, 311613.29, 'New York'],
       [123334.88, 108679.17, 304981.62, 'California'],
       [101913.08, 110594.11, 229160.95, 'Florida'],
       [100671.96, 91790.61, 249744.55, 'California'],
       [93863.75, 127320.38, 249839.44, 'Florida'],
       [91992.39, 135495.07, 252664.93, 'California'],
       [119943.24, 156547.42, 256512.92, 'Florida'],
       [114523.61, 122616.84, 261776.23, 'New York'],
       [78013.11, 121597.55, 264346.06, 'California'],
       [94657.16, 145077.58, 282574.31, 'New York'],
       [91749.16, 114175.79, 294919.57, 'Florida'],
       [86419.7, 153514.11, 0.0, 'New York'],
       [76253.86, 113867.3, 298664.47, 'California'],
       [78389.47, 153773.43, 299737.29, 'New York'],
       [73994.56, 122782.75, 303319.26, 'Florida'],
       [67532.53, 105751.03, 304768.73, 'Florida'],
       [77044.01, 99281.34, 140574.81, 'New York'],
       [64664.71, 139553.16, 137962.62, 'California'],
       [75328.87, 144135.98, 134050.07, 'Florida'],
       [72107.6, 127864.55, 353183.81, 'New York'],
       [66051.52, 182645.56, 118148.2, 'Florida'],
       [65605.48, 153032.06, 107138.38, 'New York'],
       [61994.48, 115641.28, 91131.24, 'Florida'],
       [61136.38, 152701.92, 88218.23, 'New York'],
       [63408.86, 129219.61, 46085.25, 'California'],
       [55493.95, 103057.49, 214634.81, 'Florida'],
       [46426.07, 157693.92, 210797.67, 'California'],
       [46014.02, 85047.44, 205517.64, 'New York'],
       [28663.76, 127056.21, 201126.82, 'Florida'],
       [44069.95, 51283.14, 197029.42, 'California'],
       [20229.59, 65947.93, 185265.1, 'New York'],
       [38558.51, 82982.09, 174999.3, 'California'],
       [28754.33, 118546.05, 172795.67, 'California'],
       [27892.92, 84710.77, 164470.71, 'Florida'],
       [23640.93, 96189.63, 148001.11, 'California'],
       [15505.73, 127382.3, 35534.17, 'New York'],
       [22177.74, 154806.14, 28334.72, 'California'],
       [1000.23, 124153.04, 1903.93, 'New York'],
       [1315.46, 115816.21, 297114.46, 'Florida'],
       [0.0, 135426.92, 0.0, 'California'],
       [542.05, 51743.15, 0.0, 'New York'],
       [0.0, 116983.8, 45173.06, 'California']], dtype=object)
```

In [11]:
```
y
```

Out[11]:
```
array([192261.83, 191792.06, 191050.39, 182901.99, 166187.94, 156991.12,
       156122.51, 155752.6 , 152211.77, 149759.96, 146121.95, 144259.4 ,
       141585.52, 134307.35, 132602.65, 129917.04, 126992.93, 125370.37,
       124266.9 , 122776.86, 118474.03, 111313.02, 110352.25, 108733.99,
       108552.04, 107404.34, 105733.54, 105008.31, 103282.38, 101004.64,
        99937.59,  97483.56,  97427.84,  96778.92,  96712.8 ,  96479.51,
        90708.19,  89949.14,  81229.06,  81005.76,  78239.91,  77798.83,
        71498.49,  69758.98,  65200.33,  64926.08,  49490.75,  42559.73,
        35673.41,  14681.4 ])
```

In [12]:
```
# preprocessinf the dataset
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [3])], remainder='passthrough')
X = np.array(ct.fit_transform(X))
X
```

Out[12]:
```
array([[0.0, 0.0, 1.0, 165349.2, 136897.8, 471784.1],
       [1.0, 0.0, 0.0, 162597.7, 151377.59, 443898.53],
       [0.0, 1.0, 0.0, 153441.51, 101145.55, 407934.54],
       [0.0, 0.0, 1.0, 144372.41, 118671.85, 383199.62],
       [0.0, 1.0, 0.0, 142107.34, 91391.77, 366168.42],
       [0.0, 0.0, 1.0, 131876.9, 99814.71, 362861.36],
       [1.0, 0.0, 0.0, 134615.46, 147198.87, 127716.82],
       [0.0, 1.0, 0.0, 130298.13, 145530.06, 323876.68],
       [0.0, 0.0, 1.0, 120542.52, 148718.95, 311613.29],
       [1.0, 0.0, 0.0, 123334.88, 108679.17, 304981.62],
       [0.0, 1.0, 0.0, 101913.08, 110594.11, 229160.95],
       [1.0, 0.0, 0.0, 100671.96, 91790.61, 249744.55],
       [0.0, 1.0, 0.0, 93863.75, 127320.38, 249839.44],
       [1.0, 0.0, 0.0, 91992.39, 135495.07, 252664.93],
       [0.0, 1.0, 0.0, 119943.24, 156547.42, 256512.92],
       [0.0, 0.0, 1.0, 114523.61, 122616.84, 261776.23],
       [1.0, 0.0, 0.0, 78013.11, 121597.55, 264346.06],
       [0.0, 0.0, 1.0, 94657.16, 145077.58, 282574.31],
       [0.0, 1.0, 0.0, 91749.16, 114175.79, 294919.57],
       [0.0, 0.0, 1.0, 86419.7, 153514.11, 0.0],
       [1.0, 0.0, 0.0, 76253.86, 113867.3, 298664.47],
       [0.0, 0.0, 1.0, 78389.47, 153773.43, 299737.29],
       [0.0, 1.0, 0.0, 73994.56, 122782.75, 303319.26],
       [0.0, 1.0, 0.0, 67532.53, 105751.03, 304768.73],
       [0.0, 0.0, 1.0, 77044.01, 99281.34, 140574.81],
       [1.0, 0.0, 0.0, 64664.71, 139553.16, 137962.62],
       [0.0, 1.0, 0.0, 75328.87, 144135.98, 134050.07],
       [0.0, 0.0, 1.0, 72107.6, 127864.55, 353183.81],
       [0.0, 1.0, 0.0, 66051.52, 182645.56, 118148.2],
       [0.0, 0.0, 1.0, 65605.48, 153032.06, 107138.38],
       [0.0, 1.0, 0.0, 61994.48, 115641.28, 91131.24],
       [0.0, 0.0, 1.0, 61136.38, 152701.92, 88218.23],
       [1.0, 0.0, 0.0, 63408.86, 129219.61, 46085.25],
       [0.0, 1.0, 0.0, 55493.95, 103057.49, 214634.81],
       [1.0, 0.0, 0.0, 46426.07, 157693.92, 210797.67],
       [0.0, 0.0, 1.0, 46014.02, 85047.44, 205517.64],
       [0.0, 1.0, 0.0, 28663.76, 127056.21, 201126.82],
       [1.0, 0.0, 0.0, 44069.95, 51283.14, 197029.42],
       [0.0, 0.0, 1.0, 20229.59, 65947.93, 185265.1],
       [1.0, 0.0, 0.0, 38558.51, 82982.09, 174999.3],
       [1.0, 0.0, 0.0, 28754.33, 118546.05, 172795.67],
       [0.0, 1.0, 0.0, 27892.92, 84710.77, 164470.71],
       [1.0, 0.0, 0.0, 23640.93, 96189.63, 148001.11],
       [0.0, 0.0, 1.0, 15505.73, 127382.3, 35534.17],
       [1.0, 0.0, 0.0, 22177.74, 154806.14, 28334.72],
       [0.0, 0.0, 1.0, 1000.23, 124153.04, 1903.93],
       [0.0, 1.0, 0.0, 1315.46, 115816.21, 297114.46],
       [1.0, 0.0, 0.0, 0.0, 135426.92, 0.0],
       [0.0, 0.0, 1.0, 542.05, 51743.15, 0.0],
       [1.0, 0.0, 0.0, 0.0, 116983.8, 45173.06]], dtype=object)
```

In [13]:
```
# spliting dataset into training and testing
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)
```

In [14]:
```
# train the model
from sklearn.linear_model import LinearRegression

model = LinearRegression()
model.fit(X_train, y_train)
```

Out[14]:
```
LinearRegression()
```

In [16]:
```
model.predict(X_test)
```

Out[16]:
```
array([116048.74829345,  50278.44439985,  88063.34217743, 157135.69310186,
        74151.87108247,  51622.85399204,  56309.08504563, 128181.54620862,
        76679.5123215 , 149433.2234064 ])
```

In [19]:
```
model.score(X_train, y_train)
```

Out[19]:
```
0.9650877484218092
```

In [ ]:
```

```

# Practical 3: Support Vector Machine

In [2]:
```
import pandas as pd

df = pd.read_csv('Social_Network_Ads.csv')
df.head()
```

Out[2]:

| | Age | EstimatedSalary | Purchased |
|---|---|---|---|
| **0** | 19 | 19000 | 0 |
| **1** | 35 | 20000 | 0 |
| **2** | 26 | 43000 | 0 |
| **3** | 27 | 57000 | 0 |
| **4** | 19 | 76000 | 0 |

In [6]:
```
X = df.iloc[:, :2].values # first 2 columns
y = df.iloc[:, -1].values # last column
```

In [7]:
```
X
```

Out[7]:
```
array([[    19,  19000],
       [    35,  20000],
       [    26,  43000],
       [    27,  57000],
       [    19,  76000],
       [    27,  58000],
       [    27,  84000],
       [    32, 150000],
       [    25,  33000],
       [    35,  65000],
       [    26,  80000],
       [    26,  52000],
       [    20,  86000],
       [    32,  18000],
       [    18,  82000],
       [    29,  80000],
       [    47,  25000],
       [    45,  26000],
       [    46,  28000],
       [    48,  29000],
       [    45,  22000],
       [    47,  49000],
       [    48,  41000],
       [    45,  22000],
       [    46,  23000],
       [    47,  20000],
       [    49,  28000],
       [    47,  30000],
       [    29,  43000],
       [    31,  18000],
       [    31,  74000],
       [    27, 137000],
       [    21,  16000],
       [    28,  44000],
       [    27,  90000],
       [    35,  27000],
       [    33,  28000],
       [    30,  49000],
       [    26,  72000],
       [    27,  31000],
       [    27,  17000],
       [    33,  51000],
       [    35, 108000],
       [    30,  15000],
       [    28,  84000],
       [    23,  20000],
       [    25,  79000],
       [    27,  54000],
       [    30, 135000],
       [    31,  89000],
       [    24,  32000],
       [    18,  44000],
       [    29,  83000],
       [    35,  23000],
       [    27,  58000],
       [    24,  55000],
       [    23,  48000],
```

```
[    28,  79000],
[    22,  18000],
[    32, 117000],
[    27,  20000],
[    25,  87000],
[    23,  66000],
[    32, 120000],
[    59,  83000],
[    24,  58000],
[    24,  19000],
[    23,  82000],
[    22,  63000],
[    31,  68000],
[    25,  80000],
[    24,  27000],
[    20,  23000],
[    33, 113000],
[    32,  18000],
[    34, 112000],
[    18,  52000],
[    22,  27000],
[    28,  87000],
[    26,  17000],
[    30,  80000],
[    39,  42000],
[    20,  49000],
[    35,  88000],
[    30,  62000],
[    31, 118000],
[    24,  55000],
[    28,  85000],
[    26,  81000],
[    35,  50000],
[    22,  81000],
[    30, 116000],
[    26,  15000],
[    29,  28000],
[    29,  83000],
[    35,  44000],
[    35,  25000],
[    28, 123000],
[    35,  73000],
[    28,  37000],
[    27,  88000],
[    28,  59000],
[    32,  86000],
[    33, 149000],
[    19,  21000],
[    21,  72000],
[    26,  35000],
[    27,  89000],
[    26,  86000],
[    38,  80000],
[    39,  71000],
[    37,  71000],
[    38,  61000],
[    37,  55000],
[    42,  80000],
[    40,  57000],
[    35,  75000],
[    36,  52000],
[    40,  59000],
[    41,  59000],
[    36,  75000],
[    37,  72000],
[    40,  75000],
[    35,  53000],
[    41,  51000],
[    39,  61000],
[    42,  65000],
[    26,  32000],
[    30,  17000],
[    26,  84000],
[    31,  58000],
[    33,  31000],
[    30,  87000],
[    21,  68000],
[    28,  55000],
[    23,  63000],
[    20,  82000],
[    30, 107000],
[    28,  59000],
[    19,  25000],
[    19,  85000],
[    18,  68000],
[    35,  59000],
[    30,  89000],
[    34,  25000],
[    24,  89000],
```

```
[    27,  96000],
[    41,  30000],
[    29,  61000],
[    20,  74000],
[    26,  15000],
[    41,  45000],
[    31,  76000],
[    36,  50000],
[    40,  47000],
[    31,  15000],
[    46,  59000],
[    29,  75000],
[    26,  30000],
[    32, 135000],
[    32, 100000],
[    25,  90000],
[    37,  33000],
[    35,  38000],
[    33,  69000],
[    18,  86000],
[    22,  55000],
[    35,  71000],
[    29, 148000],
[    29,  47000],
[    21,  88000],
[    34, 115000],
[    26, 118000],
[    34,  43000],
[    34,  72000],
[    23,  28000],
[    35,  47000],
[    25,  22000],
[    24,  23000],
[    31,  34000],
[    26,  16000],
[    31,  71000],
[    32, 117000],
[    33,  43000],
[    33,  60000],
[    31,  66000],
[    20,  82000],
[    33,  41000],
[    35,  72000],
[    28,  32000],
[    24,  84000],
[    19,  26000],
[    29,  43000],
[    19,  70000],
[    28,  89000],
[    34,  43000],
[    30,  79000],
[    20,  36000],
[    26,  80000],
[    35,  22000],
[    35,  39000],
[    49,  74000],
[    39, 134000],
[    41,  71000],
[    58, 101000],
[    47,  47000],
[    55, 130000],
[    52, 114000],
[    40, 142000],
[    46,  22000],
[    48,  96000],
[    52, 150000],
[    59,  42000],
[    35,  58000],
[    47,  43000],
[    60, 108000],
[    49,  65000],
[    40,  78000],
[    46,  96000],
[    59, 143000],
[    41,  80000],
[    35,  91000],
[    37, 144000],
[    60, 102000],
[    35,  60000],
[    37,  53000],
[    36, 126000],
[    56, 133000],
[    40,  72000],
[    42,  80000],
[    35, 147000],
[    39,  42000],
[    40, 107000],
[    49,  86000],
[    38, 112000],
```

```
[    46,  79000],
[    40,  57000],
[    37,  80000],
[    46,  82000],
[    53, 143000],
[    42, 149000],
[    38,  59000],
[    50,  88000],
[    56, 104000],
[    41,  72000],
[    51, 146000],
[    35,  50000],
[    57, 122000],
[    41,  52000],
[    35,  97000],
[    44,  39000],
[    37,  52000],
[    48, 134000],
[    37, 146000],
[    50,  44000],
[    52,  90000],
[    41,  72000],
[    40,  57000],
[    58,  95000],
[    45, 131000],
[    35,  77000],
[    36, 144000],
[    55, 125000],
[    35,  72000],
[    48,  90000],
[    42, 108000],
[    40,  75000],
[    37,  74000],
[    47, 144000],
[    40,  61000],
[    43, 133000],
[    59,  76000],
[    60,  42000],
[    39, 106000],
[    57,  26000],
[    57,  74000],
[    38,  71000],
[    49,  88000],
[    52,  38000],
[    50,  36000],
[    59,  88000],
[    35,  61000],
[    37,  70000],
[    52,  21000],
[    48, 141000],
[    37,  93000],
[    37,  62000],
[    48, 138000],
[    41,  79000],
[    37,  78000],
[    39, 134000],
[    49,  89000],
[    55,  39000],
[    37,  77000],
[    35,  57000],
[    36,  63000],
[    42,  73000],
[    43, 112000],
[    45,  79000],
[    46, 117000],
[    58,  38000],
[    48,  74000],
[    37, 137000],
[    37,  79000],
[    40,  60000],
[    42,  54000],
[    51, 134000],
[    47, 113000],
[    36, 125000],
[    38,  50000],
[    42,  70000],
[    39,  96000],
[    38,  50000],
[    49, 141000],
[    39,  79000],
[    39,  75000],
[    54, 104000],
[    35,  55000],
[    45,  32000],
[    36,  60000],
[    52, 138000],
[    53,  82000],
[    41,  52000],
[    48,  30000],
```

```
[    48, 131000],
[    41,  60000],
[    41,  72000],
[    42,  75000],
[    36, 118000],
[    47, 107000],
[    38,  51000],
[    48, 119000],
[    42,  65000],
[    40,  65000],
[    57,  60000],
[    36,  54000],
[    58, 144000],
[    35,  79000],
[    38,  55000],
[    39, 122000],
[    53, 104000],
[    35,  75000],
[    38,  65000],
[    47,  51000],
[    47, 105000],
[    41,  63000],
[    53,  72000],
[    54, 108000],
[    39,  77000],
[    38,  61000],
[    38, 113000],
[    37,  75000],
[    42,  90000],
[    37,  57000],
[    36,  99000],
[    60,  34000],
[    54,  70000],
[    41,  72000],
[    40,  71000],
[    42,  54000],
[    43, 129000],
[    53,  34000],
[    47,  50000],
[    42,  79000],
[    42, 104000],
[    59,  29000],
[    58,  47000],
[    46,  88000],
[    38,  71000],
[    54,  26000],
[    60,  46000],
[    60,  83000],
[    39,  73000],
[    59, 130000],
[    37,  80000],
[    46,  32000],
[    46,  74000],
[    42,  53000],
[    41,  87000],
[    58,  23000],
[    42,  64000],
[    48,  33000],
[    44, 139000],
[    49,  28000],
[    57,  33000],
[    56,  60000],
[    49,  39000],
[    39,  71000],
[    47,  34000],
[    48,  35000],
[    48,  33000],
[    47,  23000],
[    45,  45000],
[    60,  42000],
[    39,  59000],
[    46,  41000],
[    51,  23000],
[    50,  20000],
[    36,  33000],
[    49,  36000]])
```

In [8]:
```
y
```

Out[8]:
```
array([0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

```
0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1,
0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0,
1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0,
1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1,
0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1,
1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1,
0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0,
1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1,
0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1,
1, 1, 0, 1])
```

In [9]:

```
# spliting training and testing dataset
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)
```

In [10]:

```
X_train
```

Out[10]:

```
array([[    33,  41000],
       [    48,  33000],
       [    33,  43000],
       [    35,  77000],
       [    26,  43000],
       [    48,  96000],
       [    38,  50000],
       [    35,  47000],
       [    29,  83000],
       [    51, 134000],
       [    42,  79000],
       [    30,  17000],
       [    37,  70000],
       [    48,  29000],
       [    36,  63000],
       [    35,  53000],
       [    53,  34000],
       [    34,  25000],
       [    37,  80000],
       [    46,  88000],
       [    23,  82000],
       [    39, 134000],
       [    42,  65000],
       [    20,  36000],
       [    24,  19000],
       [    41,  52000],
       [    47,  30000],
       [    37,  80000],
       [    48, 131000],
       [    35,  50000],
       [    35,  97000],
       [    57,  60000],
       [    35,  60000],
       [    40,  57000],
       [    60, 108000],
       [    31,  68000],
       [    25,  87000],
       [    56,  60000],
       [    36,  50000],
       [    18,  86000],
       [    20,  74000],
       [    36, 126000],
       [    26,  72000],
       [    45, 131000],
       [    58, 144000],
       [    30,  87000],
       [    29,  43000],
       [    57,  33000],
       [    27,  31000],
       [    38,  50000],
       [    48,  33000],
       [    27, 137000],
       [    29,  61000],
       [    59,  29000],
       [    28,  59000],
       [    24,  84000],
       [    37,  72000],
       [    33, 113000],
       [    36,  52000],
       [    31,  34000],
       [    41,  45000],
       [    32,  18000],
       [    41,  71000],
```

```
[    36,  54000],
[    19,  85000],
[    26,  35000],
[    42,  90000],
[    48, 138000],
[    47, 113000],
[    48, 141000],
[    24,  32000],
[    56, 104000],
[    49,  36000],
[    41,  59000],
[    47,  34000],
[    47,  23000],
[    26,  52000],
[    36, 118000],
[    59,  42000],
[    27,  57000],
[    32,  18000],
[    49,  86000],
[    49,  89000],
[    35,  61000],
[    39,  73000],
[    58,  47000],
[    35,  73000],
[    37,  75000],
[    46,  41000],
[    39,  71000],
[    49,  28000],
[    24,  55000],
[    37, 144000],
[    46,  32000],
[    45,  32000],
[    32, 150000],
[    41,  80000],
[    42,  70000],
[    29, 148000],
[    41,  72000],
[    47, 105000],
[    26,  30000],
[    37,  57000],
[    28, 123000],
[    41,  87000],
[    39,  42000],
[    21,  72000],
[    41,  72000],
[    40,  57000],
[    20,  82000],
[    60,  42000],
[    25,  79000],
[    37,  78000],
[    18,  52000],
[    54,  70000],
[    45,  45000],
[    20,  82000],
[    24,  23000],
[    50,  44000],
[    22,  81000],
[    35,  55000],
[    40,  78000],
[    35,  57000],
[    35,  75000],
[    29,  43000],
[    31,  71000],
[    41,  79000],
[    37,  77000],
[    19,  21000],
[    27,  90000],
[    21,  16000],
[    58,  38000],
[    26,  84000],
[    44,  39000],
[    60,  42000],
[    47,  51000],
[    58,  23000],
[    53,  82000],
[    19,  25000],
[    40,  47000],
[    46,  79000],
[    45,  79000],
[    28,  85000],
[    49,  28000],
[    38,  80000],
[    52,  21000],
[    29,  83000],
[    43, 129000],
[    54, 108000],
[    37,  53000],
[    35,  71000],
[    40,  57000],
```

```
[    28,  44000],
[    41,  72000],
[    27,  88000],
[    37, 137000],
[    27,  89000],
[    48, 134000],
[    26,  86000],
[    37,  71000],
[    58,  95000],
[    31,  89000],
[    37,  74000],
[    33, 149000],
[    24,  89000],
[    34,  72000],
[    23,  63000],
[    35,  75000],
[    33,  31000],
[    40,  61000],
[    57,  26000],
[    32, 117000],
[    43, 112000],
[    42,  75000],
[    40,  65000],
[    39,  42000],
[    41,  60000],
[    27,  84000],
[    37,  55000],
[    35,  65000],
[    20,  49000],
[    41,  30000],
[    40, 142000],
[    60,  83000],
[    57,  74000],
[    27,  96000],
[    59,  88000],
[    34,  43000],
[    35,  91000],
[    55, 130000],
[    36,  99000],
[    38,  61000],
[    35, 108000],
[    28,  87000],
[    45,  26000],
[    48,  90000],
[    33,  69000],
[    37,  93000],
[    49,  88000],
[    35, 147000],
[    26,  15000],
[    26,  80000],
[    39,  61000],
[    19,  70000],
[    38,  65000],
[    41,  63000],
[    30,  49000],
[    59, 143000],
[    33,  51000],
[    27,  20000],
[    18,  68000],
[    41,  52000],
[    32, 117000],
[    22,  55000],
[    42,  54000],
[    47,  20000],
[    36, 125000],
[    20,  86000],
[    44, 139000],
[    50,  20000],
[    37,  62000],
[    26,  17000],
[    59,  76000],
[    38,  59000],
[    50,  36000],
[    31,  76000],
[    30,  62000],
[    46,  23000],
[    31,  18000],
[    35,  50000],
[    48, 119000],
[    48,  41000],
[    52, 138000],
[    45,  22000],
[    33,  60000],
[    25,  33000],
[    35,  27000],
[    35,  44000],
[    36,  33000],
[    47, 107000],
[    35,  72000],
```

```
       [    40,  72000],
       [    22,  18000],
       [    49,  65000],
       [    56, 133000],
       [    22,  27000],
       [    38,  71000],
       [    23,  66000],
       [    46,  22000],
       [    52, 114000],
       [    26,  81000],
       [    19,  26000],
       [    40,  60000],
       [    46, 117000],
       [    53, 104000],
       [    28,  37000],
       [    36,  75000],
       [    35,  88000],
       [    53, 143000],
       [    39,  96000],
       [    23,  28000],
       [    55,  39000],
       [    39, 122000],
       [    27,  58000],
       [    47,  50000],
       [    49,  74000],
       [    21,  68000],
       [    46,  96000],
       [    39, 134000],
       [    42,  73000],
       [    47,  49000],
       [    42,  80000],
       [    26,  15000],
       [    35,  38000],
       [    38, 113000],
       [    19,  76000],
       [    46,  82000],
       [    53,  72000],
       [    34, 112000],
       [    35,  58000],
       [    35,  72000],
       [    21,  88000],
       [    39,  59000],
       [    46,  59000],
       [    34, 115000],
       [    42,  65000],
       [    45,  22000],
       [    26, 118000],
       [    30, 135000],
       [    40, 107000],
       [    37,  52000],
       [    37,  33000],
       [    40,  59000],
       [    46,  74000],
       [    47,  25000],
       [    52,  38000],
       [    26,  16000],
       [    42,  53000],
       [    35,  23000],
       [    36, 144000],
       [    42,  64000],
       [    28,  89000],
       [    48,  74000],
       [    29,  47000],
       [    60, 102000],
       [    54, 104000],
       [    22,  63000],
       [    34,  43000],
       [    50,  88000],
       [    31,  66000],
       [    35,  79000],
       [    30,  15000],
       [    40,  75000],
       [    38,  51000],
       [    51,  23000],
       [    60,  46000],
       [    39,  71000],
       [    19,  19000],
       [    29,  28000],
       [    24,  27000]])
```

In [11]:

```
X_test
```

Out[11]:

```
array([[    42,  54000],
       [    35,  22000],
       [    25,  80000],
```

```
[    42,  80000],
[    39,  79000],
[    60,  34000],
[    33,  28000],
[    35,  59000],
[    57, 122000],
[    37,  79000],
[    30, 116000],
[    37, 146000],
[    49,  39000],
[    47, 144000],
[    35,  20000],
[    28,  79000],
[    43, 133000],
[    31,  15000],
[    28,  59000],
[    29,  80000],
[    32, 120000],
[    25,  22000],
[    23,  20000],
[    27,  58000],
[    18,  82000],
[    38,  55000],
[    47,  47000],
[    30,  80000],
[    47,  43000],
[    26,  80000],
[    30,  89000],
[    42, 104000],
[    18,  44000],
[    59,  83000],
[    25,  90000],
[    41,  72000],
[    28,  32000],
[    32, 100000],
[    29,  75000],
[    38, 112000],
[    30, 107000],
[    58, 101000],
[    49, 141000],
[    48,  30000],
[    31,  74000],
[    42, 149000],
[    28,  55000],
[    40,  71000],
[    35,  25000],
[    35,  39000],
[    32,  86000],
[    30,  79000],
[    41,  51000],
[    55, 125000],
[    39,  77000],
[    26,  32000],
[    42, 108000],
[    28,  84000],
[    27,  17000],
[    27,  54000],
[    32, 135000],
[    38,  61000],
[    20,  23000],
[    51, 146000],
[    54,  26000],
[    31, 118000],
[    59, 130000],
[    52, 150000],
[    36,  60000],
[    46,  28000],
[    40,  75000],
[    24,  55000],
[    38,  71000],
[    31,  58000],
[    23,  48000],
[    39, 106000],
[    39,  75000],
[    24,  58000],
[    48,  35000],
[    52,  90000]])
```

In [12]:
```
y_train
```

Out[12]:
```
array([0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1,
       0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1,
       1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0,
       1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0,
```

```
       1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0,
       0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1,
       1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1,
       0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0,
       1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0,
       1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0])
```

In [13]:
```
y_test
```

Out[13]:
```
array([0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1,
       0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1,
       1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1])
```

In [14]:
```
# creating SVM model object
from sklearn.svm import SVC

model = SVC()
```

In [15]:
```
#training the model
model.fit(X_train, y_train)
```

Out[15]:
```
SVC()
```

In [18]:
```
#checking the score of the model
model.score(X_test, y_test)
```

Out[18]:
```
0.759375
```

In [23]:
```
#Scaling and preprocessing our dataset
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

In [24]:
```
X_train
```

Out[24]:
```
array([[-0.46596422, -0.83191905],
       [ 0.96319122, -1.07261895],
       [-0.46596422, -0.77174407],
       [-0.27541016,  0.25123053],
       [-1.13290343, -0.77174407],
       [ 0.96319122,  0.8228928 ],
       [ 0.01042093, -0.56113165],
       [-0.27541016, -0.65139412],
       [-0.84707234,  0.43175545],
       [ 1.24902231,  1.96621735],
       [ 0.39152904,  0.3114055 ],
       [-0.75179531, -1.55401876],
       [-0.0848561 ,  0.04061811],
       [ 0.96319122, -1.1929689 ],
       [-0.18013313, -0.16999431],
       [-0.27541016, -0.47086919],
       [ 1.43957637, -1.04253146],
       [-0.37068719, -1.31331886],
       [-0.0848561 ,  0.34149299],
       [ 0.77263716,  0.58219289],
       [-1.41873452,  0.40166797],
       [ 0.10569795,  1.96621735],
       [ 0.39152904, -0.10981933],
       [-1.70456561, -0.98235649],
       [-1.32345749, -1.49384378],
       [ 0.29625201, -0.50095668],
       [ 0.86791419, -1.16288142],
       [-0.0848561 ,  0.34149299],
       [ 0.96319122,  1.87595488],
       [-0.27541016, -0.56113165],
       [-0.27541016,  0.85298029],
```

```
[ 1.82068449, -0.26025677],
[-0.27541016, -0.26025677],
[ 0.20097498, -0.35051924],
[ 2.10651558,  1.18394266],
[-0.65651828, -0.01955687],
[-1.22818046,  0.55210541],
[ 1.72540746, -0.26025677],
[-0.18013313, -0.56113165],
[-1.89511967,  0.52201792],
[-1.70456561,  0.16096806],
[-0.18013313,  1.72551744],
[-1.13290343,  0.10079309],
[ 0.67736013,  1.87595488],
[ 1.91596152,  2.26709223],
[-0.75179531,  0.55210541],
[-0.84707234, -0.77174407],
[ 1.82068449, -1.07261895],
[-1.0376264 , -1.13279393],
[ 0.01042093, -0.56113165],
[ 0.96319122, -1.07261895],
[-1.0376264 ,  2.05647981],
[-0.84707234, -0.23016928],
[ 2.01123855, -1.1929689 ],
[-0.94234937, -0.29034426],
[-1.32345749,  0.46184294],
[-0.0848561 ,  0.10079309],
[-0.46596422,  1.3343801 ],
[-0.18013313, -0.50095668],
[-0.65651828, -1.04253146],
[ 0.29625201, -0.71156909],
[-0.56124125, -1.52393127],
[ 0.29625201,  0.0707056 ],
[-0.18013313, -0.4407817 ],
[-1.79984264,  0.49193043],
[-1.13290343, -1.01244397],
[ 0.39152904,  0.64236787],
[ 0.96319122,  2.0865673 ],
[ 0.86791419,  1.3343801 ],
[ 0.96319122,  2.17682976],
[-1.32345749, -1.10270644],
[ 1.72540746,  1.0635927 ],
[ 1.05846825, -0.98235649],
[ 0.29625201, -0.29034426],
[ 0.86791419, -1.04253146],
[ 0.86791419, -1.37349383],
[-1.13290343, -0.50095668],
[-0.18013313,  1.48481754],
[ 2.01123855, -0.80183156],
[-1.0376264 , -0.35051924],
[-0.56124125, -1.52393127],
[ 1.05846825,  0.52201792],
[ 1.05846825,  0.61228038],
[-0.27541016, -0.23016928],
[ 0.10569795,  0.13088057],
[ 1.91596152, -0.65139412],
[-0.27541016,  0.13088057],
[-0.0848561 ,  0.19105555],
[ 0.77263716, -0.83191905],
[ 0.10569795,  0.0707056 ],
[ 1.05846825, -1.22305639],
[-1.32345749, -0.41069421],
[-0.0848561 ,  2.26709223],
[ 0.77263716, -1.10270644],
[ 0.67736013, -1.10270644],
[-0.56124125,  2.44761716],
[ 0.29625201,  0.34149299],
[ 0.39152904,  0.04061811],
[-0.84707234,  2.38744218],
[ 0.29625201,  0.10079309],
[ 0.86791419,  1.09368019],
[-1.13290343, -1.16288142],
[-0.0848561 , -0.35051924],
[-0.94234937,  1.63525498],
[ 0.29625201,  0.55210541],
[ 0.10569795, -0.80183156],
[-1.60928858,  0.10079309],
[ 0.29625201,  0.10079309],
[ 0.20097498, -0.35051924],
[-1.70456561,  0.40166797],
[ 2.10651558, -0.80183156],
[-1.22818046,  0.3114055 ],
[-0.0848561 ,  0.28131801],
[-1.89511967, -0.50095668],
[ 1.5348534 ,  0.04061811],
[ 0.67736013, -0.71156909],
[-1.70456561,  0.40166797],
[-1.32345749, -1.37349383],
[ 1.15374528, -0.74165658],
[-1.51401155,  0.37158048],
```

```
[-0.27541016, -0.41069421],
[ 0.20097498,  0.28131801],
[-0.27541016, -0.35051924],
[-0.27541016,  0.19105555],
[-0.84707234, -0.77174407],
[-0.65651828,  0.0707056 ],
[ 0.29625201,  0.3114055 ],
[-0.0848561 ,  0.25123053],
[-1.79984264, -1.43366881],
[-1.0376264 ,  0.64236787],
[-1.60928858, -1.58410625],
[ 1.91596152, -0.92218151],
[-1.13290343,  0.46184294],
[ 0.5820831 , -0.89209402],
[ 2.10651558, -0.80183156],
[ 0.86791419, -0.53104416],
[ 1.91596152, -1.37349383],
[ 1.43957637,  0.40166797],
[-1.79984264, -1.31331886],
[ 0.20097498, -0.65139412],
[ 0.77263716,  0.3114055 ],
[ 0.67736013,  0.3114055 ],
[-0.94234937,  0.49193043],
[ 1.05846825, -1.22305639],
[ 0.01042093,  0.34149299],
[ 1.34429934, -1.43366881],
[-0.84707234,  0.43175545],
[ 0.48680607,  1.81577991],
[ 1.5348534 ,  1.18394266],
[-0.0848561 , -0.47086919],
[-0.27541016,  0.0707056 ],
[ 0.20097498, -0.35051924],
[-0.94234937, -0.74165658],
[ 0.29625201,  0.10079309],
[-1.0376264 ,  0.58219289],
[-0.0848561 ,  2.05647981],
[-1.0376264 ,  0.61228038],
[ 0.96319122,  1.96621735],
[-1.13290343,  0.52201792],
[-0.0848561 ,  0.0707056 ],
[ 1.91596152,  0.79280531],
[-0.65651828,  0.61228038],
[-0.0848561 ,  0.16096806],
[-0.46596422,  2.41752967],
[-1.32345749,  0.61228038],
[-0.37068719,  0.10079309],
[-1.41873452, -0.16999431],
[-0.27541016,  0.19105555],
[-0.46596422, -1.13279393],
[ 0.20097498, -0.23016928],
[ 1.82068449, -1.28323137],
[-0.56124125,  1.45473005],
[ 0.48680607,  1.30429261],
[ 0.39152904,  0.19105555],
[ 0.20097498, -0.10981933],
[ 0.10569795, -0.80183156],
[ 0.29625201, -0.26025677],
[-1.0376264 ,  0.46184294],
[-0.0848561 , -0.41069421],
[-0.27541016, -0.10981933],
[-1.70456561, -0.59121914],
[ 0.29625201, -1.16288142],
[ 0.20097498,  2.20691725],
[ 2.10651558,  0.43175545],
[ 1.82068449,  0.16096806],
[-1.0376264 ,  0.8228928 ],
[ 2.01123855,  0.58219289],
[-0.37068719, -0.77174407],
[-0.27541016,  0.67245536],
[ 1.63013043,  1.84586739],
[-0.18013313,  0.91315526],
[ 0.01042093, -0.23016928],
[-0.27541016,  1.18394266],
[-0.94234937,  0.55210541],
[ 0.67736013, -1.28323137],
[ 0.96319122,  0.64236787],
[-0.46596422,  0.01053062],
[-0.0848561 ,  0.73263034],
[ 1.05846825,  0.58219289],
[-0.27541016,  2.35735469],
[-1.13290343, -1.61419374],
[-1.13290343,  0.34149299],
[ 0.10569795, -0.23016928],
[-1.79984264,  0.04061811],
[ 0.01042093, -0.10981933],
[ 0.29625201, -0.16999431],
[-0.75179531, -0.59121914],
[ 2.01123855,  2.23700474],
[-0.46596422, -0.53104416],
```

```
[-1.0376264 , -1.4637563 ],
[-1.89511967, -0.01955687],
[ 0.29625201, -0.50095668],
[-0.56124125,  1.45473005],
[-1.51401155, -0.41069421],
[ 0.39152904, -0.4407817 ],
[ 0.86791419, -1.4637563 ],
[-0.18013313,  1.69542995],
[-1.70456561,  0.52201792],
[ 0.5820831 ,  2.11665479],
[ 1.15374528, -1.4637563 ],
[-0.0848561 , -0.2000818 ],
[-1.13290343, -1.55401876],
[ 2.01123855,  0.22114304],
[ 0.01042093, -0.29034426],
[ 1.15374528, -0.98235649],
[-0.65651828,  0.22114304],
[-0.75179531, -0.2000818 ],
[ 0.77263716, -1.37349383],
[-0.65651828, -1.52393127],
[-0.27541016, -0.56113165],
[ 0.96319122,  1.51490503],
[ 0.96319122, -0.83191905],
[ 1.34429934,  2.0865673 ],
[ 0.67736013, -1.40358132],
[-0.46596422, -0.26025677],
[-1.22818046, -1.07261895],
[-0.27541016, -1.25314388],
[-0.27541016, -0.74165658],
[-0.18013313, -1.07261895],
[ 0.86791419,  1.15385517],
[-0.27541016,  0.10079309],
[ 0.20097498,  0.10079309],
[-1.51401155, -1.52393127],
[ 1.05846825, -0.10981933],
[ 1.72540746,  1.93612986],
[-1.51401155, -1.25314388],
[ 0.01042093,  0.0707056 ],
[-1.41873452, -0.07973184],
[ 0.77263716, -1.40358132],
[ 1.34429934,  1.36446759],
[-1.13290343,  0.37158048],
[-1.79984264, -1.28323137],
[ 0.20097498, -0.26025677],
[ 0.77263716,  1.45473005],
[ 1.43957637,  1.0635927 ],
[-0.94234937, -0.952269  ],
[-0.18013313,  0.19105555],
[-0.27541016,  0.58219289],
[ 1.43957637,  2.23700474],
[ 0.10569795,  0.8228928 ],
[-1.41873452, -1.22305639],
[ 1.63013043, -0.89209402],
[ 0.10569795,  1.60516749],
[-1.0376264 , -0.32043175],
[ 0.86791419, -0.56113165],
[ 1.05846825,  0.16096806],
[-1.60928858, -0.01955687],
[ 0.77263716,  0.8228928 ],
[ 0.10569795,  1.96621735],
[ 0.39152904,  0.13088057],
[ 0.86791419, -0.59121914],
[ 0.39152904,  0.34149299],
[-1.13290343, -1.61419374],
[-0.27541016, -0.92218151],
[ 0.01042093,  1.3343801 ],
[-1.79984264,  0.22114304],
[ 0.77263716,  0.40166797],
[ 1.43957637,  0.10079309],
[-0.37068719,  1.30429261],
[-0.27541016, -0.32043175],
[-0.27541016,  0.10079309],
[-1.60928858,  0.58219289],
[ 0.10569795, -0.29034426],
[ 0.77263716, -0.29034426],
[-0.37068719,  1.39455507],
[ 0.39152904, -0.10981933],
[ 0.67736013, -1.40358132],
[-1.13290343,  1.48481754],
[-0.75179531,  1.99630484],
[ 0.20097498,  1.15385517],
[-0.0848561 , -0.50095668],
[-0.0848561 , -1.07261895],
[ 0.20097498, -0.29034426],
[ 0.77263716,  0.16096806],
[ 0.86791419, -1.31331886],
[ 1.34429934, -0.92218151],
[-1.13290343, -1.58410625],
[ 0.39152904, -0.47086919],
```

```
       [-0.27541016, -1.37349383],
       [-0.18013313,  2.26709223],
       [ 0.39152904, -0.13990682],
       [-0.94234937,  0.61228038],
       [ 0.96319122,  0.16096806],
       [-0.84707234, -0.65139412],
       [ 2.10651558,  1.00341773],
       [ 1.5348534 ,  1.0635927 ],
       [-1.51401155, -0.16999431],
       [-0.37068719, -0.77174407],
       [ 1.15374528,  0.58219289],
       [-0.65651828, -0.07973184],
       [-0.27541016,  0.3114055 ],
       [-0.75179531, -1.61419374],
       [ 0.20097498,  0.19105555],
       [ 0.01042093, -0.53104416],
       [ 1.24902231, -1.37349383],
       [ 2.10651558, -0.68148161],
       [ 0.10569795,  0.0707056 ],
       [-1.79984264, -1.49384378],
       [-0.84707234, -1.22305639],
       [-1.32345749, -1.25314388]])
```

In [25]:
```
X_test
```

Out[25]:
```
array([[ 0.39152904, -0.4407817 ],
       [-0.27541016, -1.40358132],
       [-1.22818046,  0.34149299],
       [ 0.39152904,  0.34149299],
       [ 0.10569795,  0.3114055 ],
       [ 2.10651558, -1.04253146],
       [-0.46596422, -1.22305639],
       [-0.27541016, -0.29034426],
       [ 1.82068449,  1.60516749],
       [-0.0848561 ,  0.3114055 ],
       [-0.75179531,  1.42464256],
       [-0.0848561 ,  2.3272672 ],
       [ 1.05846825, -0.89209402],
       [ 0.86791419,  2.26709223],
       [-0.27541016, -1.4637563 ],
       [-0.94234937,  0.3114055 ],
       [ 0.48680607,  1.93612986],
       [-0.65651828, -1.61419374],
       [-0.94234937, -0.29034426],
       [-0.84707234,  0.34149299],
       [-0.56124125,  1.54499251],
       [-1.22818046, -1.40358132],
       [-1.41873452, -1.4637563 ],
       [-1.0376264 , -0.32043175],
       [-1.89511967,  0.40166797],
       [ 0.01042093, -0.41069421],
       [ 0.86791419, -0.65139412],
       [-0.75179531,  0.34149299],
       [ 0.86791419, -0.77174407],
       [-1.13290343,  0.34149299],
       [-0.75179531,  0.61228038],
       [ 0.39152904,  1.0635927 ],
       [-1.89511967, -0.74165658],
       [ 2.01123855,  0.43175545],
       [-1.22818046,  0.64236787],
       [ 0.29625201,  0.10079309],
       [-0.94234937, -1.10270644],
       [-0.56124125,  0.94324275],
       [-0.84707234,  0.19105555],
       [ 0.01042093,  1.30429261],
       [-0.75179531,  1.15385517],
       [ 1.91596152,  0.97333024],
       [ 1.05846825,  2.17682976],
       [ 0.96319122, -1.16288142],
       [-0.65651828,  0.16096806],
       [ 0.39152904,  2.41752967],
       [-0.94234937, -0.41069421],
       [ 0.20097498,  0.0707056 ],
       [-0.27541016, -1.31331886],
       [-0.27541016, -0.89209402],
       [-0.56124125,  0.52201792],
       [-0.75179531,  0.3114055 ],
       [ 0.29625201, -0.53104416],
       [ 1.63013043,  1.69542995],
       [ 0.10569795,  0.25123053],
       [-1.13290343, -1.10270644],
       [ 0.39152904,  1.18394266],
       [-0.94234937,  0.46184294],
       [-1.0376264 , -1.55401876],
       [-1.0376264 , -0.4407817 ],
```

```
       [-0.56124125,  1.99630484],
       [ 0.01042093, -0.23016928],
       [-1.70456561, -1.37349383],
       [ 1.24902231,  2.3272672 ],
       [ 1.5348534 , -1.28323137],
       [-0.65651828,  1.48481754],
       [ 2.01123855,  1.84586739],
       [ 1.34429934,  2.44761716],
       [-0.18013313, -0.26025677],
       [ 0.77263716, -1.22305639],
       [ 0.20097498,  0.19105555],
       [-1.32345749, -0.41069421],
       [ 0.01042093,  0.0707056 ],
       [-0.65651828, -0.32043175],
       [-1.41873452, -0.62130663],
       [ 0.10569795,  1.12376768],
       [ 0.10569795,  0.19105555],
       [-1.32345749, -0.32043175],
       [ 0.96319122, -1.01244397],
       [ 1.34429934,  0.64236787]])
```

In [27]:
```
#again training our model
new_model = SVC()
new_model.fit(X_train, y_train)
```

Out[27]:
```
SVC()
```

In [31]:
```
new_model.score(X_train, y_train)
```

Out[31]:
```
0.91875
```

In [41]:
```
# find confusion matrix
from sklearn.metrics import confusion_matrix

y_pred = new_model.predict(X_test)
cm = confusion_matrix(y_test, y_pred)
cm
```

Out[41]:
```
array([[46,  6],
       [ 4, 24]])
```

In [ ]:
```

```

# Practical 4: K-nearrest neightbor

In [1]:
```
import pandas as pd

df = pd.read_csv('Social_Network_Ads.csv')
df.head()
```

Out[1]:

| | Age | EstimatedSalary | Purchased |
|---|---|---|---|
| **0** | 19 | 19000 | 0 |
| **1** | 35 | 20000 | 0 |
| **2** | 26 | 43000 | 0 |
| **3** | 27 | 57000 | 0 |
| **4** | 19 | 76000 | 0 |

In [4]:
```
X = df.iloc[:, :2].values # first 2 col
y = df.iloc[:, -1].values # last col
```

In [5]:
```
X
```

Out[5]:
```
array([[    19,  19000],
       [    35,  20000],
       [    26,  43000],
       [    27,  57000],
       [    19,  76000],
       [    27,  58000],
       [    27,  84000],
       [    32, 150000],
       [    25,  33000],
       [    35,  65000],
       [    26,  80000],
       [    26,  52000],
       [    20,  86000],
       [    32,  18000],
       [    18,  82000],
       [    29,  80000],
       [    47,  25000],
       [    45,  26000],
       [    46,  28000],
       [    48,  29000],
       [    45,  22000],
       [    47,  49000],
       [    48,  41000],
       [    45,  22000],
       [    46,  23000],
       [    47,  20000],
       [    49,  28000],
       [    47,  30000],
       [    29,  43000],
       [    31,  18000],
       [    31,  74000],
       [    27, 137000],
       [    21,  16000],
       [    28,  44000],
       [    27,  90000],
       [    35,  27000],
       [    33,  28000],
       [    30,  49000],
       [    26,  72000],
       [    27,  31000],
       [    27,  17000],
       [    33,  51000],
       [    35, 108000],
       [    30,  15000],
       [    28,  84000],
       [    23,  20000],
       [    25,  79000],
       [    27,  54000],
       [    30, 135000],
       [    31,  89000],
       [    24,  32000],
       [    18,  44000],
       [    29,  83000],
       [    35,  23000],
       [    27,  58000],
       [    24,  55000],
       [    23,  48000],
```

```
[    28,  79000],
[    22,  18000],
[    32, 117000],
[    27,  20000],
[    25,  87000],
[    23,  66000],
[    32, 120000],
[    59,  83000],
[    24,  58000],
[    24,  19000],
[    23,  82000],
[    22,  63000],
[    31,  68000],
[    25,  80000],
[    24,  27000],
[    20,  23000],
[    33, 113000],
[    32,  18000],
[    34, 112000],
[    18,  52000],
[    22,  27000],
[    28,  87000],
[    26,  17000],
[    30,  80000],
[    39,  42000],
[    20,  49000],
[    35,  88000],
[    30,  62000],
[    31, 118000],
[    24,  55000],
[    28,  85000],
[    26,  81000],
[    35,  50000],
[    22,  81000],
[    30, 116000],
[    26,  15000],
[    29,  28000],
[    29,  83000],
[    35,  44000],
[    35,  25000],
[    28, 123000],
[    35,  73000],
[    28,  37000],
[    27,  88000],
[    28,  59000],
[    32,  86000],
[    33, 149000],
[    19,  21000],
[    21,  72000],
[    26,  35000],
[    27,  89000],
[    26,  86000],
[    38,  80000],
[    39,  71000],
[    37,  71000],
[    38,  61000],
[    37,  55000],
[    42,  80000],
[    40,  57000],
[    35,  75000],
[    36,  52000],
[    40,  59000],
[    41,  59000],
[    36,  75000],
[    37,  72000],
[    40,  75000],
[    35,  53000],
[    41,  51000],
[    39,  61000],
[    42,  65000],
[    26,  32000],
[    30,  17000],
[    26,  84000],
[    31,  58000],
[    33,  31000],
[    30,  87000],
[    21,  68000],
[    28,  55000],
[    23,  63000],
[    20,  82000],
[    30, 107000],
[    28,  59000],
[    19,  25000],
[    19,  85000],
[    18,  68000],
[    35,  59000],
[    30,  89000],
[    34,  25000],
[    24,  89000],
```

```
[    27,  96000],
[    41,  30000],
[    29,  61000],
[    20,  74000],
[    26,  15000],
[    41,  45000],
[    31,  76000],
[    36,  50000],
[    40,  47000],
[    31,  15000],
[    46,  59000],
[    29,  75000],
[    26,  30000],
[    32, 135000],
[    32, 100000],
[    25,  90000],
[    37,  33000],
[    35,  38000],
[    33,  69000],
[    18,  86000],
[    22,  55000],
[    35,  71000],
[    29, 148000],
[    29,  47000],
[    21,  88000],
[    34, 115000],
[    26, 118000],
[    34,  43000],
[    34,  72000],
[    23,  28000],
[    35,  47000],
[    25,  22000],
[    24,  23000],
[    31,  34000],
[    26,  16000],
[    31,  71000],
[    32, 117000],
[    33,  43000],
[    33,  60000],
[    31,  66000],
[    20,  82000],
[    33,  41000],
[    35,  72000],
[    28,  32000],
[    24,  84000],
[    19,  26000],
[    29,  43000],
[    19,  70000],
[    28,  89000],
[    34,  43000],
[    30,  79000],
[    20,  36000],
[    26,  80000],
[    35,  22000],
[    35,  39000],
[    49,  74000],
[    39, 134000],
[    41,  71000],
[    58, 101000],
[    47,  47000],
[    55, 130000],
[    52, 114000],
[    40, 142000],
[    46,  22000],
[    48,  96000],
[    52, 150000],
[    59,  42000],
[    35,  58000],
[    47,  43000],
[    60, 108000],
[    49,  65000],
[    40,  78000],
[    46,  96000],
[    59, 143000],
[    41,  80000],
[    35,  91000],
[    37, 144000],
[    60, 102000],
[    35,  60000],
[    37,  53000],
[    36, 126000],
[    56, 133000],
[    40,  72000],
[    42,  80000],
[    35, 147000],
[    39,  42000],
[    40, 107000],
[    49,  86000],
[    38, 112000],
```

```
[    46,  79000],
[    40,  57000],
[    37,  80000],
[    46,  82000],
[    53, 143000],
[    42, 149000],
[    38,  59000],
[    50,  88000],
[    56, 104000],
[    41,  72000],
[    51, 146000],
[    35,  50000],
[    57, 122000],
[    41,  52000],
[    35,  97000],
[    44,  39000],
[    37,  52000],
[    48, 134000],
[    37, 146000],
[    50,  44000],
[    52,  90000],
[    41,  72000],
[    40,  57000],
[    58,  95000],
[    45, 131000],
[    35,  77000],
[    36, 144000],
[    55, 125000],
[    35,  72000],
[    48,  90000],
[    42, 108000],
[    40,  75000],
[    37,  74000],
[    47, 144000],
[    40,  61000],
[    43, 133000],
[    59,  76000],
[    60,  42000],
[    39, 106000],
[    57,  26000],
[    57,  74000],
[    38,  71000],
[    49,  88000],
[    52,  38000],
[    50,  36000],
[    59,  88000],
[    35,  61000],
[    37,  70000],
[    52,  21000],
[    48, 141000],
[    37,  93000],
[    37,  62000],
[    48, 138000],
[    41,  79000],
[    37,  78000],
[    39, 134000],
[    49,  89000],
[    55,  39000],
[    37,  77000],
[    35,  57000],
[    36,  63000],
[    42,  73000],
[    43, 112000],
[    45,  79000],
[    46, 117000],
[    58,  38000],
[    48,  74000],
[    37, 137000],
[    37,  79000],
[    40,  60000],
[    42,  54000],
[    51, 134000],
[    47, 113000],
[    36, 125000],
[    38,  50000],
[    42,  70000],
[    39,  96000],
[    38,  50000],
[    49, 141000],
[    39,  79000],
[    39,  75000],
[    54, 104000],
[    35,  55000],
[    45,  32000],
[    36,  60000],
[    52, 138000],
[    53,  82000],
[    41,  52000],
[    48,  30000],
```

```
[    48, 131000],
[    41,  60000],
[    41,  72000],
[    42,  75000],
[    36, 118000],
[    47, 107000],
[    38,  51000],
[    48, 119000],
[    42,  65000],
[    40,  65000],
[    57,  60000],
[    36,  54000],
[    58, 144000],
[    35,  79000],
[    38,  55000],
[    39, 122000],
[    53, 104000],
[    35,  75000],
[    38,  65000],
[    47,  51000],
[    47, 105000],
[    41,  63000],
[    53,  72000],
[    54, 108000],
[    39,  77000],
[    38,  61000],
[    38, 113000],
[    37,  75000],
[    42,  90000],
[    37,  57000],
[    36,  99000],
[    60,  34000],
[    54,  70000],
[    41,  72000],
[    40,  71000],
[    42,  54000],
[    43, 129000],
[    53,  34000],
[    47,  50000],
[    42,  79000],
[    42, 104000],
[    59,  29000],
[    58,  47000],
[    46,  88000],
[    38,  71000],
[    54,  26000],
[    60,  46000],
[    60,  83000],
[    39,  73000],
[    59, 130000],
[    37,  80000],
[    46,  32000],
[    46,  74000],
[    42,  53000],
[    41,  87000],
[    58,  23000],
[    42,  64000],
[    48,  33000],
[    44, 139000],
[    49,  28000],
[    57,  33000],
[    56,  60000],
[    49,  39000],
[    39,  71000],
[    47,  34000],
[    48,  35000],
[    48,  33000],
[    47,  23000],
[    45,  45000],
[    60,  42000],
[    39,  59000],
[    46,  41000],
[    51,  23000],
[    50,  20000],
[    36,  33000],
[    49,  36000]])
```

In [6]:
```
y
```

Out[6]:
```
array([0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

```
0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1,
0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0,
1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0,
1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1,
0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1,
1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1,
0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0,
1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1,
0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1,
1, 1, 0, 1])
```

In [7]:

```
# spliting dataset into training and testing
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)
```

In [8]:

```
X_train
```

Out[8]:

```
array([[    58,  95000],
       [    60, 102000],
       [    49,  74000],
       [    56,  60000],
       [    30, 116000],
       [    40,  60000],
       [    40,  75000],
       [    47,  23000],
       [    26,  86000],
       [    26,  32000],
       [    34,  72000],
       [    41,  60000],
       [    36, 126000],
       [    29,  28000],
       [    48,  41000],
       [    58,  47000],
       [    43, 133000],
       [    35,  55000],
       [    50,  20000],
       [    58,  38000],
       [    33,  31000],
       [    35,  60000],
       [    19,  76000],
       [    40,  57000],
       [    31, 118000],
       [    52, 114000],
       [    29,  83000],
       [    53, 104000],
       [    59,  29000],
       [    18,  82000],
       [    25,  33000],
       [    24,  84000],
       [    46,  22000],
       [    37,  62000],
       [    38,  55000],
       [    27,  20000],
       [    58, 144000],
       [    23,  20000],
       [    38,  61000],
       [    52,  38000],
       [    42,  75000],
       [    47, 113000],
       [    29, 148000],
       [    47,  49000],
       [    35,  97000],
       [    19,  70000],
       [    59,  83000],
       [    49, 141000],
       [    52, 138000],
       [    35,  23000],
       [    40,  57000],
       [    44,  39000],
       [    49,  28000],
       [    26,  17000],
       [    52, 150000],
       [    48, 141000],
       [    30, 135000],
       [    27,  89000],
       [    36, 118000],
       [    35,  75000],
       [    28,  32000],
       [    45,  32000],
       [    25,  79000],
```

```
[    18,  68000],
[    24,  32000],
[    29,  47000],
[    29,  43000],
[    24,  55000],
[    32, 120000],
[    51, 134000],
[    35,  27000],
[    39,  61000],
[    57, 122000],
[    48,  30000],
[    42,  54000],
[    22,  63000],
[    38,  80000],
[    37,  72000],
[    53,  34000],
[    51,  23000],
[    35,  47000],
[    35, 147000],
[    22,  55000],
[    35,  91000],
[    31,  89000],
[    45,  26000],
[    26, 118000],
[    59,  76000],
[    38,  71000],
[    47, 107000],
[    42,  65000],
[    48,  33000],
[    42,  53000],
[    42,  80000],
[    27,  88000],
[    55,  39000],
[    49,  28000],
[    35,  72000],
[    21,  88000],
[    39,  77000],
[    35,  20000],
[    24,  55000],
[    31,  66000],
[    37, 137000],
[    19,  25000],
[    19,  85000],
[    48,  33000],
[    27,  96000],
[    35,  59000],
[    48,  96000],
[    40,  78000],
[    37,  79000],
[    48, 119000],
[    47,  47000],
[    50,  88000],
[    49,  65000],
[    26,  35000],
[    40, 107000],
[    36,  99000],
[    40,  65000],
[    28,  44000],
[    45,  79000],
[    23,  28000],
[    40,  75000],
[    36, 144000],
[    29,  83000],
[    42,  70000],
[    42, 104000],
[    38,  50000],
[    41,  72000],
[    38,  50000],
[    41,  45000],
[    53,  72000],
[    34, 115000],
[    26,  43000],
[    39, 106000],
[    22,  81000],
[    37,  71000],
[    57,  33000],
[    37, 144000],
[    35,  65000],
[    26,  72000],
[    47, 144000],
[    26,  84000],
[    33, 149000],
[    30,  89000],
[    35,  71000],
[    33, 113000],
[    47,  51000],
[    33,  43000],
[    25,  87000],
[    20,  74000],
```

```
[    47,  34000],
[    54, 108000],
[    31,  68000],
[    20,  82000],
[    59,  88000],
[    38,  65000],
[    42,  64000],
[    42,  73000],
[    44, 139000],
[    32, 100000],
[    26,  80000],
[    32, 150000],
[    40,  71000],
[    27, 137000],
[    60, 108000],
[    49,  89000],
[    23,  82000],
[    35,  50000],
[    27,  90000],
[    45,  45000],
[    35,  72000],
[    50,  36000],
[    33,  51000],
[    47,  25000],
[    35, 108000],
[    40,  72000],
[    38,  59000],
[    32, 117000],
[    19,  21000],
[    48, 131000],
[    46,  32000],
[    56, 104000],
[    37,  53000],
[    47,  43000],
[    35,  53000],
[    45,  22000],
[    42,  80000],
[    59, 143000],
[    31,  34000],
[    32,  18000],
[    18,  44000],
[    32, 117000],
[    20,  82000],
[    60,  34000],
[    37,  78000],
[    58,  23000],
[    27,  31000],
[    30,  79000],
[    53, 143000],
[    41,  72000],
[    39,  71000],
[    39, 134000],
[    23,  48000],
[    20,  49000],
[    57,  60000],
[    27,  84000],
[    40, 142000],
[    39,  71000],
[    19,  19000],
[    24,  58000],
[    29,  75000],
[    43, 112000],
[    41,  51000],
[    28,  79000],
[    46,  74000],
[    49,  86000],
[    30,  62000],
[    47,  20000],
[    47, 105000],
[    36,  52000],
[    46,  96000],
[    30,  49000],
[    39,  96000],
[    49,  88000],
[    36,  63000],
[    34, 112000],
[    40,  47000],
[    30,  15000],
[    28,  89000],
[    60,  42000],
[    28, 123000],
[    42,  79000],
[    26,  52000],
[    33,  41000],
[    53,  82000],
[    30,  87000],
[    54,  70000],
[    26,  30000],
[    51, 146000],
```

```
       [    54, 104000],
       [    24,  27000],
       [    28,  59000],
       [    35,  57000],
       [    38,  71000],
       [    31,  74000],
       [    41,  72000],
       [    28,  84000],
       [    56, 133000],
       [    36,  75000],
       [    37,  77000],
       [    47,  50000],
       [    23,  66000],
       [    18,  52000],
       [    46,  82000],
       [    35,  88000],
       [    54,  26000],
       [    22,  27000],
       [    41,  59000],
       [    29,  61000],
       [    41,  63000],
       [    55, 125000],
       [    60,  46000],
       [    37,  57000],
       [    50,  44000],
       [    41,  80000],
       [    60,  83000],
       [    28,  87000],
       [    48,  29000],
       [    20,  36000],
       [    32, 135000],
       [    39,  42000],
       [    39,  73000],
       [    34,  43000],
       [    26,  81000],
       [    27,  17000],
       [    45,  22000],
       [    29,  80000],
       [    42, 108000],
       [    37,  55000],
       [    39, 134000],
       [    28,  55000],
       [    31,  76000],
       [    38, 113000],
       [    20,  23000],
       [    45, 131000],
       [    41,  52000],
       [    30,  80000],
       [    36,  60000],
       [    41,  30000],
       [    32,  18000],
       [    30,  17000],
       [    39,  79000],
       [    34,  43000],
       [    39,  75000],
       [    35,  77000],
       [    35,  79000],
       [    24,  89000],
       [    46,  41000],
       [    35,  58000],
       [    52,  90000],
       [    27,  58000],
       [    48,  90000],
       [    31,  15000],
       [    23,  63000],
       [    27,  58000],
       [    47,  30000],
       [    37, 146000],
       [    37,  80000],
       [    33,  69000],
       [    41,  72000],
       [    43, 129000],
       [    33,  60000],
       [    21,  72000],
       [    46, 117000],
       [    39,  42000],
       [    59,  42000],
       [    25,  22000],
       [    36,  50000]])
```

In [9]:
```
X_test
```

Out[9]:
```
array([[    40,  61000],
       [    31,  71000],
       [    41,  71000],
```

```
[    38,  51000],
[    34,  25000],
[    41,  52000],
[    38, 112000],
[    31,  18000],
[    20,  86000],
[    35,  75000],
[    21,  16000],
[    42, 149000],
[    28,  59000],
[    38,  61000],
[    26,  15000],
[    52,  21000],
[    24,  19000],
[    48,  74000],
[    39,  59000],
[    40,  59000],
[    40,  57000],
[    35,  44000],
[    36,  33000],
[    49,  36000],
[    42,  90000],
[    35,  73000],
[    35,  39000],
[    37,  70000],
[    37,  93000],
[    57,  74000],
[    26,  80000],
[    37,  52000],
[    59, 130000],
[    27,  57000],
[    25,  90000],
[    19,  26000],
[    46,  28000],
[    57,  26000],
[    18,  86000],
[    36, 125000],
[    31,  58000],
[    26,  15000],
[    42,  65000],
[    35,  25000],
[    30, 107000],
[    35,  22000],
[    55, 130000],
[    21,  68000],
[    42,  54000],
[    37,  74000],
[    26,  16000],
[    37,  80000],
[    25,  80000],
[    58, 101000],
[    48,  35000],
[    46,  59000],
[    48, 134000],
[    24,  23000],
[    46,  23000],
[    27,  54000],
[    35,  61000],
[    41,  79000],
[    46,  88000],
[    60,  42000],
[    46,  79000],
[    39, 122000],
[    49,  39000],
[    28,  37000],
[    37,  33000],
[    35,  38000],
[    37,  75000],
[    48, 138000],
[    28,  85000],
[    32,  86000],
[    29,  43000],
[    35,  50000],
[    41,  87000],
[    22,  18000],
[    36,  54000],
[    33,  28000]])
```

In [10]:
```
y_train
```

Out[10]:
```
array([1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0,
       0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1,
       1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0,
       0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1,
       0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1,
```

```
       0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0,
       1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1,
       0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1,
       0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1,
       0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1,
       1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1,
       0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1,
       0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1,
       1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0])
```

In [11]:
```
y_test
```

Out[11]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0,
       0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0,
       1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1,
       1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0])
```

In [13]:
```
# scaling and preprocessing our dataset
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

In [14]:
```
X_train
```

Out[14]:
```
array([[ 1.90044906,  0.68445743],
       [ 2.08890388,  0.88967556],
       [ 1.05240238,  0.06880304],
       [ 1.71199424, -0.34163321],
       [-0.7379184 ,  1.30011181],
       [ 0.20435569, -0.34163321],
       [ 0.20435569,  0.09811992],
       [ 0.86394756, -1.42635761],
       [-1.11482804,  0.42060555],
       [-1.11482804, -1.16250573],
       [-0.36100876,  0.01016929],
       [ 0.2985831 , -0.34163321],
       [-0.17255394,  1.59328057],
       [-0.83214581, -1.27977323],
       [ 0.95817497, -0.89865385],
       [ 1.90044906, -0.7227526 ],
       [ 0.48703792,  1.7984987 ],
       [-0.26678135, -0.48821759],
       [ 1.14662979, -1.51430824],
       [ 1.90044906, -0.98660448],
       [-0.45523617, -1.1918226 ],
       [-0.26678135, -0.34163321],
       [-1.7744199 ,  0.12743679],
       [ 0.20435569, -0.42958384],
       [-0.64369099,  1.35874556],
       [ 1.33508461,  1.24147806],
       [-0.83214581,  0.33265492],
       [ 1.42931201,  0.94830931],
       [ 1.99467647, -1.25045636],
       [-1.86864731,  0.30333805],
       [-1.20905545, -1.13318885],
       [-1.30328285,  0.3619718 ],
       [ 0.76972015, -1.45567448],
       [-0.07832653, -0.28299946],
       [ 0.01590088, -0.48821759],
       [-1.02060063, -1.51430824],
       [ 1.90044906,  2.12098433],
       [-1.39751026, -1.51430824],
       [ 0.01590088, -0.31231634],
       [ 1.33508461, -0.98660448],
       [ 0.39281051,  0.09811992],
       [ 0.86394756,  1.21216119],
       [-0.83214581,  2.23825183],
       [ 0.86394756, -0.66411885],
       [-0.26678135,  0.74309118],
       [-1.7744199 , -0.04846446],
       [ 1.99467647,  0.33265492],
       [ 1.05240238,  2.0330337 ],
       [ 1.33508461,  1.94508307],
       [-0.26678135, -1.42635761],
       [ 0.20435569, -0.42958384],
       [ 0.58126533, -0.9572876 ],
       [ 1.05240238, -1.27977323],
       [-1.11482804, -1.60225886],
```

```
[ 1.33508461,  2.29688558],
[ 0.95817497,  2.0330337 ],
[-0.7379184 ,  1.85713245],
[-1.02060063,  0.50855617],
[-0.17255394,  1.35874556],
[-0.26678135,  0.09811992],
[-0.92637322, -1.16250573],
[ 0.67549274, -1.16250573],
[-1.20905545,  0.21538742],
[-1.86864731, -0.10709821],
[-1.30328285, -1.16250573],
[-0.83214581, -0.7227526 ],
[-0.83214581, -0.8400201 ],
[-1.30328285, -0.48821759],
[-0.54946358,  1.41737932],
[ 1.2408572 ,  1.82781557],
[-0.26678135, -1.30909011],
[ 0.11012828, -0.31231634],
[ 1.80622165,  1.47601307],
[ 0.95817497, -1.22113948],
[ 0.39281051, -0.51753447],
[-1.49173767, -0.25368259],
[ 0.01590088,  0.2447043 ],
[-0.07832653,  0.01016929],
[ 1.42931201, -1.10387198],
[ 1.2408572 , -1.42635761],
[-0.26678135, -0.7227526 ],
[-0.26678135,  2.20893495],
[-1.49173767, -0.48821759],
[-0.26678135,  0.56718993],
[-0.64369099,  0.50855617],
[ 0.67549274, -1.33840698],
[-1.11482804,  1.35874556],
[ 1.99467647,  0.12743679],
[ 0.01590088, -0.01914758],
[ 0.86394756,  1.03625993],
[ 0.39281051, -0.19504884],
[ 0.95817497, -1.13318885],
[ 0.39281051, -0.54685134],
[ 0.39281051,  0.2447043 ],
[-1.02060063,  0.4792393 ],
[ 1.61776683, -0.9572876 ],
[ 1.05240238, -1.27977323],
[-0.26678135,  0.01016929],
[-1.58596508,  0.4792393 ],
[ 0.11012828,  0.15675367],
[-0.26678135, -1.51430824],
[-1.30328285, -0.48821759],
[-0.64369099, -0.16573196],
[-0.07832653,  1.9157662 ],
[-1.7744199 , -1.36772386],
[-1.7744199 ,  0.39128867],
[ 0.95817497, -1.13318885],
[-1.02060063,  0.7137743 ],
[-0.26678135, -0.37095009],
[ 0.95817497,  0.7137743 ],
[ 0.20435569,  0.18607054],
[-0.07832653,  0.21538742],
[ 0.95817497,  1.38806244],
[ 0.86394756, -0.7227526 ],
[ 1.14662979,  0.4792393 ],
[ 1.05240238, -0.19504884],
[-1.11482804, -1.0745551 ],
[ 0.20435569,  1.03625993],
[-0.17255394,  0.80172493],
[ 0.20435569, -0.19504884],
[-0.92637322, -0.81070322],
[ 0.67549274,  0.21538742],
[-1.39751026, -1.27977323],
[ 0.20435569,  0.09811992],
[-0.17255394,  2.12098433],
[-0.83214581,  0.33265492],
[ 0.39281051, -0.04846446],
[ 0.39281051,  0.94830931],
[ 0.01590088, -0.63480197],
[ 0.2985831 ,  0.01016929],
[ 0.01590088, -0.63480197],
[ 0.2985831 , -0.78138635],
[ 1.42931201,  0.01016929],
[-0.36100876,  1.27079494],
[-1.11482804, -0.8400201 ],
[ 0.11012828,  1.00694306],
[-1.49173767,  0.27402117],
[-0.07832653, -0.01914758],
[ 1.80622165, -1.13318885],
[-0.07832653,  2.12098433],
[-0.26678135, -0.19504884],
[-1.11482804,  0.01016929],
[ 0.86394756,  2.12098433],
```

```
[-1.11482804,  0.3619718 ],
[-0.45523617,  2.26756871],
[-0.7379184 ,  0.50855617],
[-0.26678135, -0.01914758],
[-0.45523617,  1.21216119],
[ 0.86394756, -0.60548509],
[-0.45523617, -0.8400201 ],
[-1.20905545,  0.44992242],
[-1.68019249,  0.06880304],
[ 0.86394756, -1.10387198],
[ 1.52353942,  1.06557681],
[-0.64369099, -0.10709821],
[-1.68019249,  0.30333805],
[ 1.99467647,  0.4792393 ],
[ 0.01590088, -0.19504884],
[ 0.39281051, -0.22436571],
[ 0.39281051,  0.03948617],
[ 0.58126533,  1.97439995],
[-0.54946358,  0.83104181],
[-1.11482804,  0.2447043 ],
[-0.54946358,  2.29688558],
[ 0.20435569, -0.01914758],
[-1.02060063,  1.9157662 ],
[ 2.08890388,  1.06557681],
[ 1.05240238,  0.50855617],
[-1.39751026,  0.30333805],
[-0.26678135, -0.63480197],
[-1.02060063,  0.53787305],
[ 0.67549274, -0.78138635],
[-0.26678135,  0.01016929],
[ 1.14662979, -1.04523823],
[-0.45523617, -0.60548509],
[ 0.86394756, -1.36772386],
[-0.26678135,  1.06557681],
[ 0.20435569,  0.01016929],
[ 0.01590088, -0.37095009],
[-0.54946358,  1.32942869],
[-1.7744199 , -1.48499136],
[ 0.95817497,  1.73986495],
[ 0.76972015, -1.16250573],
[ 1.71199424,  0.94830931],
[-0.07832653, -0.54685134],
[ 0.86394756, -0.8400201 ],
[-0.26678135, -0.54685134],
[ 0.67549274, -1.45567448],
[ 0.39281051,  0.2447043 ],
[ 1.99467647,  2.09166745],
[-0.64369099, -1.10387198],
[-0.54946358, -1.57294199],
[-1.86864731, -0.81070322],
[-0.54946358,  1.32942869],
[-1.68019249,  0.30333805],
[ 2.08890388, -1.10387198],
[-0.07832653,  0.18607054],
[ 1.90044906, -1.42635761],
[-1.02060063, -1.1918226 ],
[-0.7379184 ,  0.21538742],
[ 1.42931201,  2.09166745],
[ 0.2985831 ,  0.01016929],
[ 0.11012828, -0.01914758],
[ 0.11012828,  1.82781557],
[-1.39751026, -0.69343572],
[-1.68019249, -0.66411885],
[ 1.80622165, -0.34163321],
[-1.02060063,  0.3619718 ],
[ 0.20435569,  2.06235058],
[ 0.11012828, -0.01914758],
[-1.7744199 , -1.54362511],
[-1.30328285, -0.40026697],
[-0.83214581,  0.09811992],
[ 0.48703792,  1.18284431],
[ 0.2985831 , -0.60548509],
[-0.92637322,  0.21538742],
[ 0.76972015,  0.06880304],
[ 1.05240238,  0.42060555],
[-0.7379184 , -0.28299946],
[ 0.86394756, -1.51430824],
[ 0.86394756,  0.97762618],
[-0.17255394, -0.57616822],
[ 0.76972015,  0.7137743 ],
[-0.7379184 , -0.66411885],
[ 0.11012828,  0.7137743 ],
[ 1.05240238,  0.4792393 ],
[-0.17255394, -0.25368259],
[-0.36100876,  1.18284431],
[ 0.20435569, -0.7227526 ],
[-0.7379184 , -1.66089261],
[-0.92637322,  0.50855617],
[ 2.08890388, -0.86933697],
```

```
[-0.92637322,  1.50532994],
[ 0.39281051,  0.21538742],
[-1.11482804, -0.57616822],
[-0.45523617, -0.89865385],
[ 1.42931201,  0.30333805],
[-0.7379184 ,  0.44992242],
[ 1.52353942, -0.04846446],
[-1.11482804, -1.22113948],
[ 1.2408572 ,  2.17961808],
[ 1.52353942,  0.94830931],
[-1.30328285, -1.30909011],
[-0.92637322, -0.37095009],
[-0.26678135, -0.42958384],
[ 0.01590088, -0.01914758],
[-0.64369099,  0.06880304],
[ 0.2985831 ,  0.01016929],
[-0.92637322,  0.3619718 ],
[ 1.71199424,  1.7984987 ],
[-0.17255394,  0.09811992],
[-0.07832653,  0.15675367],
[ 0.86394756, -0.63480197],
[-1.39751026, -0.16573196],
[-1.86864731, -0.57616822],
[ 0.76972015,  0.30333805],
[-0.26678135,  0.4792393 ],
[ 1.52353942, -1.33840698],
[-1.49173767, -1.30909011],
[ 0.2985831 , -0.37095009],
[-0.83214581, -0.31231634],
[ 0.2985831 , -0.25368259],
[ 1.61776683,  1.56396369],
[ 2.08890388, -0.75206947],
[-0.07832653, -0.42958384],
[ 1.14662979, -0.81070322],
[ 0.2985831 ,  0.2447043 ],
[ 2.08890388,  0.33265492],
[-0.92637322,  0.44992242],
[ 0.95817497, -1.25045636],
[-1.68019249, -1.04523823],
[-0.54946358,  1.85713245],
[ 0.11012828, -0.86933697],
[ 0.11012828,  0.03948617],
[-0.36100876, -0.8400201 ],
[-1.11482804,  0.27402117],
[-1.02060063, -1.60225886],
[ 0.67549274, -1.45567448],
[-0.83214581,  0.2447043 ],
[ 0.39281051,  1.06557681],
[-0.07832653, -0.48821759],
[ 0.11012828,  1.82781557],
[-0.92637322, -0.48821759],
[-0.64369099,  0.12743679],
[ 0.01590088,  1.21216119],
[-1.68019249, -1.42635761],
[ 0.67549274,  1.73986495],
[ 0.2985831 , -0.57616822],
[-0.7379184 ,  0.2447043 ],
[-0.17255394, -0.34163321],
[ 0.2985831 , -1.22113948],
[-0.54946358, -1.57294199],
[-0.7379184 , -1.60225886],
[ 0.11012828,  0.21538742],
[-0.36100876, -0.8400201 ],
[ 0.11012828,  0.09811992],
[-0.26678135,  0.15675367],
[-0.26678135,  0.21538742],
[-1.30328285,  0.50855617],
[ 0.76972015, -0.89865385],
[-0.26678135, -0.40026697],
[ 1.33508461,  0.53787305],
[-1.02060063, -0.40026697],
[ 0.95817497,  0.53787305],
[-0.64369099, -1.66089261],
[-1.39751026, -0.25368259],
[-1.02060063, -0.40026697],
[ 0.86394756, -1.22113948],
[-0.07832653,  2.17961808],
[-0.07832653,  0.2447043 ],
[-0.45523617, -0.07778134],
[ 0.2985831 ,  0.01016929],
[ 0.48703792,  1.6812312 ],
[-0.45523617, -0.34163321],
[-1.58596508,  0.01016929],
[ 0.76972015,  1.32942869],
[ 0.11012828, -0.86933697],
[ 1.99467647, -0.86933697],
[-1.20905545, -1.45567448],
[-0.17255394, -0.63480197]])
```

In [15]:
```
X_test
```

Out[15]:
```
array([[ 0.20435569, -0.31231634],
       [-0.64369099, -0.01914758],
       [ 0.2985831 , -0.01914758],
       [ 0.01590088, -0.60548509],
       [-0.36100876, -1.36772386],
       [ 0.2985831 , -0.57616822],
       [ 0.01590088,  1.18284431],
       [-0.64369099, -1.57294199],
       [-1.68019249,  0.42060555],
       [-0.26678135,  0.09811992],
       [-1.58596508, -1.63157574],
       [ 0.39281051,  2.26756871],
       [-0.92637322, -0.37095009],
       [ 0.01590088, -0.31231634],
       [-1.11482804, -1.66089261],
       [ 1.33508461, -1.48499136],
       [-1.30328285, -1.54362511],
       [ 0.95817497,  0.06880304],
       [ 0.11012828, -0.37095009],
       [ 0.20435569, -0.37095009],
       [ 0.20435569, -0.42958384],
       [-0.26678135, -0.81070322],
       [-0.17255394, -1.13318885],
       [ 1.05240238, -1.04523823],
       [ 0.39281051,  0.53787305],
       [-0.26678135,  0.03948617],
       [-0.26678135, -0.9572876 ],
       [-0.07832653, -0.04846446],
       [-0.07832653,  0.62582368],
       [ 1.80622165,  0.06880304],
       [-1.11482804,  0.2447043 ],
       [-0.07832653, -0.57616822],
       [ 1.99467647,  1.71054807],
       [-1.02060063, -0.42958384],
       [-1.20905545,  0.53787305],
       [-1.7744199 , -1.33840698],
       [ 0.76972015, -1.27977323],
       [ 1.80622165, -1.33840698],
       [-1.86864731,  0.42060555],
       [-0.17255394,  1.56396369],
       [-0.64369099, -0.40026697],
       [-1.11482804, -1.66089261],
       [ 0.39281051, -0.19504884],
       [-0.26678135, -1.36772386],
       [-0.7379184 ,  1.03625993],
       [-0.26678135, -1.45567448],
       [ 1.61776683,  1.71054807],
       [-1.58596508, -0.10709821],
       [ 0.39281051, -0.51753447],
       [-0.07832653,  0.06880304],
       [-1.11482804, -1.63157574],
       [-0.07832653,  0.2447043 ],
       [-1.20905545,  0.2447043 ],
       [ 1.90044906,  0.86035868],
       [ 0.95817497, -1.0745551 ],
       [ 0.76972015, -0.37095009],
       [ 0.95817497,  1.82781557],
       [-1.30328285, -1.42635761],
       [ 0.76972015, -1.42635761],
       [-1.02060063, -0.51753447],
       [-0.26678135, -0.31231634],
       [ 0.2985831 ,  0.21538742],
       [ 0.76972015,  0.4792393 ],
       [ 2.08890388, -0.86933697],
       [ 0.76972015,  0.21538742],
       [ 0.11012828,  1.47601307],
       [ 1.05240238, -0.9572876 ],
       [-0.92637322, -1.01592135],
       [-0.07832653, -1.13318885],
       [-0.26678135, -0.98660448],
       [-0.07832653,  0.09811992],
       [ 0.95817497,  1.94508307],
       [-0.92637322,  0.39128867],
       [-0.54946358,  0.42060555],
       [-0.83214581, -0.8400201 ],
       [-0.26678135, -0.63480197],
       [ 0.2985831 ,  0.44992242],
       [-1.49173767, -1.57294199],
       [-0.17255394, -0.51753447],
       [-0.45523617, -1.27977323]])
```

In [18]:
```
# traing our model
from sklearn.neighbors import KNeighborsClassifier

model = KNeighborsClassifier(n_neighbors=3)
model.fit(X_train, y_train)
```

Out[18]:
```
KNeighborsClassifier(n_neighbors=3)
```

In [19]:
```
model.score(X_test, y_test)
```

Out[19]:
```
0.875
```

In [22]:
```
# printing confusion matrix
from sklearn.metrics import confusion_matrix, accuracy_score

y_pred = model.predict(X_test)
cm = confusion_matrix(y_test, y_pred)
cm
```

Out[22]:
```
array([[52,  3],
       [ 7, 18]])
```

In [23]:
```
accuracy_score(y_test, y_test)
```

Out[23]:
```
1.0
```

In [ ]:

# Practical 5: Hierarchical Clustering

In [22]:
```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('Mall_Customers.csv')
df.head()
```

Out[22]:

| | CustomerID | Genre | Age | Annual Income (k$) | Spending Score (1-100) |
|---|---|---|---|---|---|
| **0** | 1 | Male | 19 | 15 | 39 |
| **1** | 2 | Male | 21 | 15 | 81 |
| **2** | 3 | Female | 20 | 16 | 6 |
| **3** | 4 | Female | 23 | 16 | 77 |
| **4** | 5 | Female | 31 | 17 | 40 |

In [23]:
```
df.rename(columns={'Annual Income (k$)':'income', 'Spending Score (1-100)':'score'}, inplace=True)
df.head()
```

Out[23]:

| | CustomerID | Genre | Age | income | score |
|---|---|---|---|---|---|
| **0** | 1 | Male | 19 | 15 | 39 |
| **1** | 2 | Male | 21 | 15 | 81 |
| **2** | 3 | Female | 20 | 16 | 6 |
| **3** | 4 | Female | 23 | 16 | 77 |
| **4** | 5 | Female | 31 | 17 | 40 |

In [24]:
```
from sklearn.cluster import AgglomerativeClustering
model = AgglomerativeClustering(n_clusters=5)
y_pred = model.fit_predict(df[['income', 'score']])
y_pred
```

Out[24]:
```
array([4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3,
       4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 1,
       4, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 1, 2, 0, 2, 0, 2,
       1, 2, 0, 2, 0, 2, 0, 2, 0, 2, 1, 2, 0, 2, 1, 2, 0, 2, 0, 2, 0, 2,
       0, 2, 0, 2, 0, 2, 1, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2,
       0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2,
       0, 2])
```

In [25]:
```
df['clusters'] = y_pred
df.head()
```

Out[25]:

| | CustomerID | Genre | Age | income | score | clusters |
|---|---|---|---|---|---|---|
| **0** | 1 | Male | 19 | 15 | 39 | 4 |
| **1** | 2 | Male | 21 | 15 | 81 | 3 |
| **2** | 3 | Female | 20 | 16 | 6 | 4 |
| **3** | 4 | Female | 23 | 16 | 77 | 3 |
| **4** | 5 | Female | 31 | 17 | 40 | 4 |

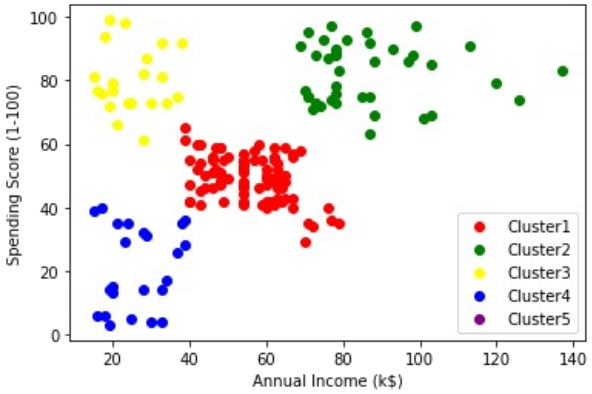
In [26]:
```
#plotting scatter plot
df1 = df[df.clusters==1]
df2 = df[df.clusters==2]
df3 = df[df.clusters==3]
df4 = df[df.clusters==4]
df5 = df[df.clusters==5]

plt.scatter(df1.income, df1.score, color='red', label='Cluster1')
plt.scatter(df2.income, df2.score, color='green', label='Cluster2')
```

```
plt.scatter(df3.income, df3.score, color='yellow', label='Cluster3')
plt.scatter(df4.income, df4.score, color='blue', label='Cluster4')
plt.scatter(df5.income, df5.score, color='purple', label='Cluster5')

plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
```

Out[26]: <matplotlib.legend.Legend at 0x7fd85617cd90>




In [ ]:

# Practical 6: K-mean Clustering

In [37]:
```
import pandas as pd
from matplotlib import pyplot as plt
%matplotlib inline

df = pd.read_csv('Mall_Customers.csv')
df.head()
```

Out[37]:

| | CustomerID | Genre | Age | Annual Income (k$) | Spending Score (1-100) |
|---|---|---|---|---|---|
| **0** | 1 | Male | 19 | 15 | 39 |
| **1** | 2 | Male | 21 | 15 | 81 |
| **2** | 3 | Female | 20 | 16 | 6 |
| **3** | 4 | Female | 23 | 16 | 77 |
| **4** | 5 | Female | 31 | 17 | 40 |

In [45]:
```
df.rename(columns={'Annual Income (k$)':'income', 'Spending Score (1-100)':'score'}, inplace=True)
df.head()
```

Out[45]:

| | CustomerID | Genre | Age | income | score | cluster |
|---|---|---|---|---|---|---|
| **0** | 1 | Male | 19 | 15 | 39 | 3 |
| **1** | 2 | Male | 21 | 15 | 81 | 0 |
| **2** | 3 | Female | 20 | 16 | 6 | 3 |
| **3** | 4 | Female | 23 | 16 | 77 | 0 |
| **4** | 5 | Female | 31 | 17 | 40 | 3 |

In [46]:
```
from sklearn.cluster import KMeans

model = KMeans(n_clusters=4)
y_pred = model.fit_predict(df[['income', 'score']])
```

In [47]:
```
y_pred
```

Out[47]:
```
array([3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0,
       3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0,
       3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2,
       1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2,
       1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2,
       1, 2], dtype=int32)
```

In [48]:
```
df['cluster'] = y_pred
df.head()
```

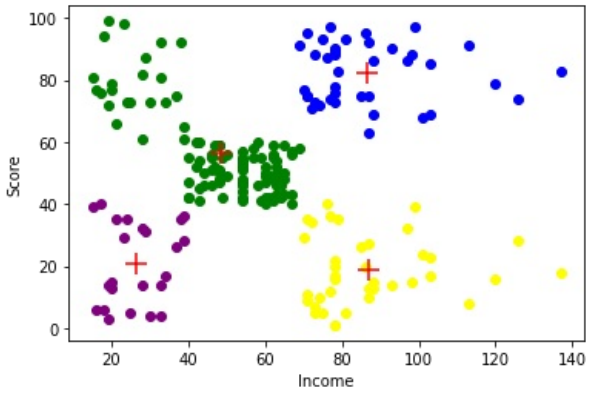
Out[48]:

| | CustomerID | Genre | Age | income | score | cluster |
|---|---|---|---|---|---|---|
| **0** | 1 | Male | 19 | 15 | 39 | 3 |
| **1** | 2 | Male | 21 | 15 | 81 | 0 |
| **2** | 3 | Female | 20 | 16 | 6 | 3 |
| **3** | 4 | Female | 23 | 16 | 77 | 0 |
| **4** | 5 | Female | 31 | 17 | 40 | 3 |

In [53]:
```
df1 = df[df.cluster==0]
df2 = df[df.cluster==1]
df3 = df[df.cluster==2]
df4 = df[df.cluster==3]
```

```
plt.scatter(df1.income, df1.score, color='green')
plt.scatter(df2.income, df2.score, color='yellow')
plt.scatter(df3.income, df3.score, color='blue')
plt.scatter(df4.income, df4.score, color='purple')
plt.scatter(model.cluster_centers_[:, 0], model.cluster_centers_[:, 1],s=200, color='red', marker='+', label='Cer
plt.xlabel('Income')
plt.ylabel('Score')
plt.legend
```

Out[53]:
```
<function matplotlib.pyplot.legend(*args, **kwargs)>
```

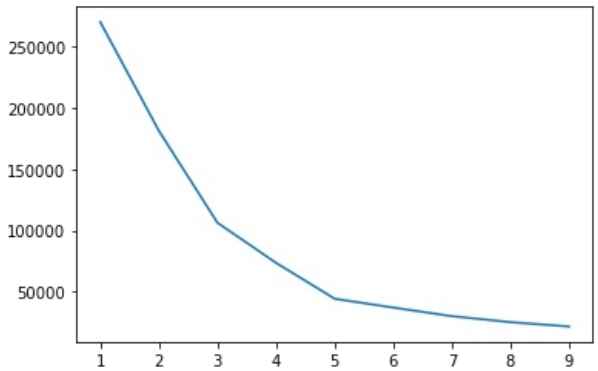



In [54]:
```
k_rng = range(1, 10)
sse = []
for k in k_rng:
    km = KMeans(n_clusters=k)
    km.fit(df[['income', 'score']])
    sse.append(km.inertia_)
sse
```

Out[54]:
```
[269981.28000000014,
 181363.59595959607,
 106348.37306211119,
 73679.78903948837,
 44448.45544793369,
 37233.81451071002,
 30241.34361793659,
 25338.024582200735,
 21829.135638779822]
```

In [55]:
```
plt.plot(k_rng, sse)
```

Out[55]:
```
[<matplotlib.lines.Line2D at 0x7f16e0c91e50>]
```




In [ ]:

# Practical 7: Artifical Neural Network

In [1]:
```
import numpy as np
import pandas as pd
import tensorflow as tf
```

In [3]:
```
dataset = pd.read_csv('Churn_Modelling.csv')
X = dataset.iloc[:, 3:-1].values
y = dataset.iloc[:, -1].values
```

In [4]:
```
print(X)
```

```
[[619 'France' 'Female' ... 1 1 101348.88]
 [608 'Spain' 'Female' ... 0 1 112542.58]
 [502 'France' 'Female' ... 1 0 113931.57]
 ...
 [709 'France' 'Female' ... 0 1 42085.58]
 [772 'Germany' 'Male' ... 1 0 92888.52]
 [792 'France' 'Female' ... 1 0 38190.78]]
```

In [5]:
```
print(y)
```

```
[1 0 1 ... 1 1 0]
```

In [6]:
```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
X[:, 2] = le.fit_transform(X[:, 2])
```

In [7]:
```
print(X)
```

```
[[619 'France' 0 ... 1 1 101348.88]
 [608 'Spain' 0 ... 0 1 112542.58]
 [502 'France' 0 ... 1 0 113931.57]
 ...
 [709 'France' 0 ... 0 1 42085.58]
 [772 'Germany' 1 ... 1 0 92888.52]
 [792 'France' 0 ... 1 0 38190.78]]
```

In [8]:
```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [1])], remainder='passthrough')
X = np.array(ct.fit_transform(X))
```

In [9]:
```
print(X)
```

```
[[1.0 0.0 0.0 ... 1 1 101348.88]
 [0.0 0.0 1.0 ... 0 1 112542.58]
 [1.0 0.0 0.0 ... 1 0 113931.57]
 ...
 [1.0 0.0 0.0 ... 0 1 42085.58]
 [0.0 1.0 0.0 ... 1 0 92888.52]
 [1.0 0.0 0.0 ... 1 0 38190.78]]
```

In [10]:
```
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, random_state = 0)
```

In [11]:
```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

In [12]:
```
ann = tf.keras.models.Sequential()
```

In [13]:
```
ann.add(tf.keras.layers.Dense(units=6, activation='relu'))
```

In [14]:
```
ann.add(tf.keras.layers.Dense(units=6, activation='relu'))
```

In [15]:
```
ann.add(tf.keras.layers.Dense(units=1, activation='sigmoid'))
```

In [16]:
```
ann.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])
```

In [17]:
```
ann.fit(X_train, y_train, batch_size = 32, epochs = 100)
```

```
Epoch 1/100
250/250 [==============================] - 1s 1ms/step - loss: 0.5750 - accuracy: 0.7490
Epoch 2/100
250/250 [==============================] - 0s 1ms/step - loss: 0.4712 - accuracy: 0.7960
Epoch 3/100
250/250 [==============================] - 0s 2ms/step - loss: 0.4428 - accuracy: 0.7986
Epoch 4/100
250/250 [==============================] - 0s 2ms/step - loss: 0.4296 - accuracy: 0.8075
Epoch 5/100
250/250 [==============================] - 0s 2ms/step - loss: 0.4212 - accuracy: 0.8149
Epoch 6/100
250/250 [==============================] - 0s 2ms/step - loss: 0.4138 - accuracy: 0.8220
Epoch 7/100
250/250 [==============================] - 1s 2ms/step - loss: 0.4062 - accuracy: 0.8217
Epoch 8/100
250/250 [==============================] - 1s 3ms/step - loss: 0.3995 - accuracy: 0.8255
Epoch 9/100
250/250 [==============================] - 1s 2ms/step - loss: 0.3932 - accuracy: 0.8257
Epoch 10/100
250/250 [==============================] - 0s 2ms/step - loss: 0.3868 - accuracy: 0.8276
Epoch 11/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3812 - accuracy: 0.8281
Epoch 12/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3766 - accuracy: 0.8288
Epoch 13/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3734 - accuracy: 0.8292
Epoch 14/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3703 - accuracy: 0.8298
Epoch 15/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3680 - accuracy: 0.8305
Epoch 16/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3658 - accuracy: 0.8298
Epoch 17/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3640 - accuracy: 0.8322
Epoch 18/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3623 - accuracy: 0.8499
Epoch 19/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3608 - accuracy: 0.8511
Epoch 20/100
250/250 [==============================] - 0s 2ms/step - loss: 0.3593 - accuracy: 0.8524
Epoch 21/100
250/250 [==============================] - 0s 2ms/step - loss: 0.3579 - accuracy: 0.8534
Epoch 22/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3570 - accuracy: 0.8545
Epoch 23/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3559 - accuracy: 0.8568
Epoch 24/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3548 - accuracy: 0.8580
Epoch 25/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3538 - accuracy: 0.8574
Epoch 26/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3528 - accuracy: 0.8585
Epoch 27/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3520 - accuracy: 0.8587
Epoch 28/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3511 - accuracy: 0.8600
Epoch 29/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3505 - accuracy: 0.8585
Epoch 30/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3494 - accuracy: 0.8606
Epoch 31/100
250/250 [==============================] - 0s 2ms/step - loss: 0.3486 - accuracy: 0.8606
Epoch 32/100
250/250 [==============================] - 0s 2ms/step - loss: 0.3479 - accuracy: 0.8596
Epoch 33/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3473 - accuracy: 0.8606
Epoch 34/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3467 - accuracy: 0.8605
Epoch 35/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3463 - accuracy: 0.8593
Epoch 36/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3458 - accuracy: 0.8610
```

```
Epoch 37/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3454 - accuracy: 0.8604
Epoch 38/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3449 - accuracy: 0.8611
Epoch 39/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3448 - accuracy: 0.8605
Epoch 40/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3442 - accuracy: 0.8629
Epoch 41/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3438 - accuracy: 0.8610
Epoch 42/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3439 - accuracy: 0.8606
Epoch 43/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3432 - accuracy: 0.8601
Epoch 44/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3431 - accuracy: 0.8597
Epoch 45/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3426 - accuracy: 0.8614
Epoch 46/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3425 - accuracy: 0.8610
Epoch 47/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3421 - accuracy: 0.8606
Epoch 48/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3420 - accuracy: 0.8608
Epoch 49/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3418 - accuracy: 0.8615
Epoch 50/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3411 - accuracy: 0.8611
Epoch 51/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3412 - accuracy: 0.8612
Epoch 52/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3408 - accuracy: 0.8616
Epoch 53/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3405 - accuracy: 0.8609
Epoch 54/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3402 - accuracy: 0.8612
Epoch 55/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3404 - accuracy: 0.8618
Epoch 56/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3400 - accuracy: 0.8609
Epoch 57/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3396 - accuracy: 0.8608
Epoch 58/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3395 - accuracy: 0.8606
Epoch 59/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3392 - accuracy: 0.8614
Epoch 60/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3390 - accuracy: 0.8618
Epoch 61/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3382 - accuracy: 0.8620
Epoch 62/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3389 - accuracy: 0.8627
Epoch 63/100
250/250 [==============================] - 0s 2ms/step - loss: 0.3384 - accuracy: 0.8631
Epoch 64/100
250/250 [==============================] - 1s 3ms/step - loss: 0.3387 - accuracy: 0.8608
Epoch 65/100
250/250 [==============================] - 1s 2ms/step - loss: 0.3386 - accuracy: 0.8611
Epoch 66/100
250/250 [==============================] - 1s 3ms/step - loss: 0.3383 - accuracy: 0.8621
Epoch 67/100
250/250 [==============================] - 1s 2ms/step - loss: 0.3377 - accuracy: 0.8626
Epoch 68/100
250/250 [==============================] - 1s 2ms/step - loss: 0.3380 - accuracy: 0.8627
Epoch 69/100
250/250 [==============================] - 1s 2ms/step - loss: 0.3373 - accuracy: 0.8630
Epoch 70/100
250/250 [==============================] - 0s 2ms/step - loss: 0.3380 - accuracy: 0.8624
Epoch 71/100
250/250 [==============================] - 0s 2ms/step - loss: 0.3374 - accuracy: 0.8634
Epoch 72/100
250/250 [==============================] - 0s 2ms/step - loss: 0.3377 - accuracy: 0.8622
Epoch 73/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3371 - accuracy: 0.8643
Epoch 74/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3373 - accuracy: 0.8626
Epoch 75/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3370 - accuracy: 0.8610
Epoch 76/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3369 - accuracy: 0.8618
Epoch 77/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3368 - accuracy: 0.8611
Epoch 78/100
250/250 [==============================] - 0s 2ms/step - loss: 0.3367 - accuracy: 0.8614
Epoch 79/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3367 - accuracy: 0.8643
Epoch 80/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3368 - accuracy: 0.8618
Epoch 81/100
```

```
250/250 [==============================] - 0s 1ms/step - loss: 0.3366 - accuracy: 0.8619
Epoch 82/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3367 - accuracy: 0.8601
Epoch 83/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3367 - accuracy: 0.8609
Epoch 84/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3362 - accuracy: 0.8608
Epoch 85/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3361 - accuracy: 0.8626
Epoch 86/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3365 - accuracy: 0.8620
Epoch 87/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3359 - accuracy: 0.8619
Epoch 88/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3363 - accuracy: 0.8629
Epoch 89/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3359 - accuracy: 0.8627
Epoch 90/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3363 - accuracy: 0.8640
Epoch 91/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3362 - accuracy: 0.8622
Epoch 92/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3357 - accuracy: 0.8626
Epoch 93/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3356 - accuracy: 0.8629
Epoch 94/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3355 - accuracy: 0.8660
Epoch 95/100
250/250 [==============================] - 0s 2ms/step - loss: 0.3360 - accuracy: 0.8608
Epoch 96/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3356 - accuracy: 0.8640
Epoch 97/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3356 - accuracy: 0.8608
Epoch 98/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3356 - accuracy: 0.8610
Epoch 99/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3357 - accuracy: 0.8635
Epoch 100/100
250/250 [==============================] - 0s 1ms/step - loss: 0.3353 - accuracy: 0.8629
```

Out[17]:
```
<keras.callbacks.History at 0x7fe7867e1510>
```

In [22]:
```
print(ann.predict(sc.transform([[1, 0, 0, 600, 1, 40, 3, 60000, 2, 1, 1, 50000]])) >0.5)
```

```
[[False]]
```

In [19]:
```
y_pred = ann.predict(X_test)
y_pred = (y_pred > 0.5)
print(np.concatenate((y_pred.reshape(len(y_pred),1), y_test.reshape(len(y_test),1)),1))
```

```
[[0 0]
 [0 1]
 [0 0]
 ...
 [0 0]
 [0 0]
 [0 0]]
```

In [20]:
```
from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
print(cm)
accuracy_score(y_test, y_pred)
```

```
[[1499   96]
 [ 186  219]]
```

Out[20]:
```
0.859
```

# Practical 8: Convolutional Neural Network

In [23]:
```
import tensorflow as tf
from keras.preprocessing.image import ImageDataGenerator
```

In [46]:
```
train_datagen = ImageDataGenerator(rescale=1./255, shear_range=0.2, zoom_range=0.2, horizontal_flip=True)
training_set = train_datagen.flow_from_directory('small_dataset/training_set', target_size=(64,64), batch_size=32
```

```
Found 10 images belonging to 2 classes.
```

In [48]:
```
train_datagen = ImageDataGenerator(rescale=1./255, shear_range=0.2, zoom_range=0.2, horizontal_flip=True)
test_set = train_datagen.flow_from_directory('small_dataset/test_set', target_size=(64,64), batch_size=32, class_
```

```
Found 10 images belonging to 2 classes.
```

In [49]:
```
cnn = tf.keras.models.Sequential()
```

In [50]:
```
cnn.add(tf.keras.layers.Conv2D(filters=32, kernel_size=3, activation='relu', input_shape=[64,64,3]))
```

In [51]:
```
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2, strides=2))
```

In [52]:
```
cnn.add(tf.keras.layers.Conv2D(filters=32, kernel_size=3, activation='relu'))
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2, strides=2))
```

In [53]:
```
cnn.add(tf.keras.layers.Flatten())
```

In [54]:
```
cnn.add(tf.keras.layers.Dense(units=128, activation='relu'))
```

In [55]:
```
cnn.add(tf.keras.layers.Dense(units=1, activation='sigmoid'))
```

In [56]:
```
cnn.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

In [57]:
```
cnn.fit(x=training_set, validation_data=test_set, epochs=25)
```

```
Epoch 1/25
1/1 [==============================] - 1s 897ms/step - loss: 0.7015 - accuracy: 0.5000 - val_loss: 0.7160 - val_a
ccuracy: 0.5000
Epoch 2/25
1/1 [==============================] - 0s 227ms/step - loss: 0.6286 - accuracy: 0.9000 - val_loss: 0.7793 - val_a
ccuracy: 0.5000
Epoch 3/25
1/1 [==============================] - 0s 224ms/step - loss: 0.6135 - accuracy: 0.5000 - val_loss: 0.7770 - val_a
ccuracy: 0.5000
Epoch 4/25
1/1 [==============================] - 0s 230ms/step - loss: 0.6256 - accuracy: 0.4000 - val_loss: 0.7575 - val_a
ccuracy: 0.4000
Epoch 5/25
1/1 [==============================] - 0s 211ms/step - loss: 0.5565 - accuracy: 0.9000 - val_loss: 0.7845 - val_a
ccuracy: 0.4000
Epoch 6/25
1/1 [==============================] - 0s 227ms/step - loss: 0.5275 - accuracy: 0.7000 - val_loss: 0.7606 - val_a
ccuracy: 0.4000
Epoch 7/25
1/1 [==============================] - 0s 224ms/step - loss: 0.4220 - accuracy: 0.9000 - val_loss: 0.7868 - val_a
ccuracy: 0.6000
Epoch 8/25
1/1 [==============================] - 0s 220ms/step - loss: 0.3937 - accuracy: 0.9000 - val_loss: 0.8448 - val_a
ccuracy: 0.4000
Epoch 9/25
1/1 [==============================] - 0s 225ms/step - loss: 0.3387 - accuracy: 0.9000 - val_loss: 0.8835 - val_a
ccuracy: 0.5000
Epoch 10/25
1/1 [==============================] - 0s 235ms/step - loss: 0.6409 - accuracy: 0.7000 - val_loss: 0.8624 - val_a
ccuracy: 0.5000
Epoch 11/25
```

```
1/1 [==============================] - 0s 230ms/step - loss: 0.2896 - accuracy: 0.9000 - val_loss: 0.9250 - val_a
ccuracy: 0.3000
Epoch 12/25
1/1 [==============================] - 0s 221ms/step - loss: 0.2462 - accuracy: 1.0000 - val_loss: 0.9689 - val_a
ccuracy: 0.3000
Epoch 13/25
1/1 [==============================] - 0s 233ms/step - loss: 0.3876 - accuracy: 0.8000 - val_loss: 1.0227 - val_a
ccuracy: 0.4000
Epoch 14/25
1/1 [==============================] - 0s 246ms/step - loss: 0.3234 - accuracy: 0.9000 - val_loss: 0.9538 - val_a
ccuracy: 0.4000
Epoch 15/25
1/1 [==============================] - 0s 228ms/step - loss: 0.2728 - accuracy: 1.0000 - val_loss: 1.0767 - val_a
ccuracy: 0.4000
Epoch 16/25
1/1 [==============================] - 0s 227ms/step - loss: 0.2297 - accuracy: 1.0000 - val_loss: 1.2349 - val_a
ccuracy: 0.3000
Epoch 17/25
1/1 [==============================] - 0s 242ms/step - loss: 0.2275 - accuracy: 0.9000 - val_loss: 1.2494 - val_a
ccuracy: 0.3000
Epoch 18/25
1/1 [==============================] - 0s 229ms/step - loss: 0.1389 - accuracy: 1.0000 - val_loss: 1.2898 - val_a
ccuracy: 0.1000
Epoch 19/25
1/1 [==============================] - 0s 240ms/step - loss: 0.1304 - accuracy: 1.0000 - val_loss: 1.3273 - val_a
ccuracy: 0.1000
Epoch 20/25
1/1 [==============================] - 0s 228ms/step - loss: 0.1246 - accuracy: 1.0000 - val_loss: 1.4391 - val_a
ccuracy: 0.1000
Epoch 21/25
1/1 [==============================] - 0s 234ms/step - loss: 0.0923 - accuracy: 1.0000 - val_loss: 1.4823 - val_a
ccuracy: 0.1000
Epoch 22/25
1/1 [==============================] - 0s 235ms/step - loss: 0.0688 - accuracy: 1.0000 - val_loss: 1.6860 - val_a
ccuracy: 0.2000
Epoch 23/25
1/1 [==============================] - 0s 228ms/step - loss: 0.0876 - accuracy: 1.0000 - val_loss: 1.6947 - val_a
ccuracy: 0.1000
Epoch 24/25
1/1 [==============================] - 0s 222ms/step - loss: 0.0465 - accuracy: 1.0000 - val_loss: 1.6785 - val_a
ccuracy: 0.1000
Epoch 25/25
1/1 [==============================] - 0s 240ms/step - loss: 0.0378 - accuracy: 1.0000 - val_loss: 1.8124 - val_a
ccuracy: 0.2000
```

Out[57]:
```
<keras.callbacks.History at 0x7f51dbd279d0>
```

In [62]:
```
import numpy as np
from keras.preprocessing import image
test_image=image.load_img('small_dataset/single_prediction/cat_or_dog_1.jpg', target_size=(64,64))
test_image=image.img_to_array(test_image)
test_image=np.expand_dims(test_image, axis=0)
result=cnn.predict(test_image)
training_set.class_indices
if result[0][0]==1:
  prediction='dog'
else:
  prediction='cat'
```

In [63]:
```
print(prediction)
```

```
dog
```