Exp No 4

Support Vector Classifier

```
In [1]: # importing libraries
         import numpy as nm
         import matplotlib.pyplot as mtp
         import pandas as pd
In [2]: #importing datasets
         data_set= pd.read_csv('User_Data.csv')
In [3]: data_set
Out[3]:
                User ID Gender Age EstimatedSalary Purchased
           0 15624510
                                                           0
                          Male
                                            19000
           1 15810944
                                            20000
                         Male
                                35
                                                           0
           2 15668575 Female
                                26
                                            43000
                                                           0
           3 15603246 Female
                                27
                                            57000
             15804002
                                                           0
                          Male
                                19
                                            76000
          395 15691863 Female
                                46
                                            41000
                                                           1
          396 15706071
                          Male
                                51
                                            23000
          397 15654296 Female
                                50
                                            20000
          398 15755018
                                            33000
                                                           0
                          Male
                                36
          399 15594041 Female
                                            36000
                                49
         400 rows × 5 columns
In [4]: #Extracting Independent and dependent Variable
         x= data_set.iloc[:, [2,3]].values
         y= data_set.iloc[:, 4].values
```

```
In [27]: x
Out[27]: array([[
                    19,
                        19000],
                    35,
                        20000],
                    26,
                        43000],
                    27,
                        570001,
                    19,
                        760001,
                    27,
                        58000],
                        84000],
                    27,
                    32, 150000],
                    25,
                        33000],
                    35,
                        65000],
                    26,
                        800001.
                    26,
                        52000],
                    20,
                        86000],
                    32,
                        18000],
                    18,
                        82000],
                    29,
                        80000],
                    47,
                        250001,
                    45,
                        26000],
                        28000],
                    46,
In [28]: y
Out[28]: array([0, 0, 0, 0, 0, 0, 1, 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, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1,
               0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
               0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 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, 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], dtype=int64)
In [26]: # Splitting the dataset into training and test set.
        from sklearn.model selection import train test split
        x_train, x_test, y_train, y_test= train_test_split(x, y, test_size= 0.25, rand
```

```
In [31]: |print("x_train :", x_train.shape)
         print("y_train :", y_train.shape)
         print("x_test :", x_test.shape)
         print("y_test :",y_test.shape)
         x_{train} : (300, 2)
         y train : (300,)
         x_test : (100, 2)
         y test : (100,)
In [32]: #feature Scaling
         from sklearn.preprocessing import StandardScaler
         st x= StandardScaler()
         x_train= st_x.fit_transform(x_train)
         x test= st x.transform(x test)
In [33]: x_train
Out[33]: array([[ 0.58164944, -0.88670699],
                [-0.60673761, 1.46173768],
                [-0.01254409, -0.5677824],
                [-0.60673761, 1.89663484],
                [ 1.37390747, -1.40858358],
                [ 1.47293972, 0.99784738],
                [0.08648817, -0.79972756],
                [-0.01254409, -0.24885782],
                [-0.21060859, -0.5677824],
                [-0.21060859, -0.19087153],
                [-0.30964085, -1.29261101],
                [-0.30964085, -0.5677824],
                [ 0.38358493, 0.09905991],
                [0.8787462, -0.59677555],
                [ 2.06713324, -1.17663843],
                [ 1.07681071, -0.13288524],
                [ 0.68068169, 1.78066227],
                [-0.70576986, 0.56295021],
                [ 0.77971394,
                               0.35999821],
In [34]: |y_train
Out[34]: array([0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1,
                0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0,
                0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0,
                0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0,
                1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1,
                0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1,
                1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0,
                0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0,
                1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1,
                0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0,
                0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0,
                1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1,
                1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0,
                0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, dtype=int64)
```

```
In [11]: from sklearn.svm import SVC # "Support vector classifier"
         classifier = SVC(kernel='linear', random_state=0)
         classifier.fit(x_train, y_train)
Out[11]: SVC(kernel='linear', random_state=0)
In [12]: #Predicting the test set result
         y_pred= classifier.predict(x_test)
In [23]: y_pred
Out[23]: array([0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1,
                0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
                1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1,
                0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1,
                0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1], dtype=int64)
In [40]: #Prediction for random sample
         sample st = st x.transform([[19,19000]])
         print(classifier.predict(sample st))
         [0]
In [36]: #Creating the Confusion matrix
         from sklearn.metrics import confusion matrix
         cm= confusion_matrix(y_test, y_pred)
         print(cm)
         [[66 2]
          [ 8 24]]
In [35]: | classifier.score(x_test,y_test)
Out[35]: 0.9
```