

# SUPPORT VECTOR MACHINES

## PYTHON CODE :

```
import numpy as np
import pandas as pd
import quadprog as quad

svm = pd.read_csv("mystery.data", names=["X1", "X2", "X3", "X4", "Y"])
X = svm.loc[:, "X1":"X4"]
X['Xb'] = 1
X = X.values
Y = svm.loc[:, "Y"].values

## SOLVING PRIMAL PROBLEM USING FEATURE TRANSFORMATION
def featureTransformation(X):
    ft = []

    ftx31 = X[:,0].reshape(1000,1)**3
    ftx32 = X[:,1].reshape(1000,1)**3
    ftx33 = X[:,2].reshape(1000,1)**3
    ftx34 = X[:,3].reshape(1000,1)**3

    ftx21 = X[:,0].reshape(1000,1)**2
    ftx22 = X[:,1].reshape(1000,1)**2
    ftx23 = X[:,2].reshape(1000,1)**2
    ftx24 = X[:,3].reshape(1000,1)**2

    ftx212 = X[:,0].reshape(1000,1)*X[:,1].reshape(1000,1)
    ftx223 = X[:,1].reshape(1000,1)*X[:,2].reshape(1000,1)
    ftx234 = X[:,2].reshape(1000,1)*X[:,3].reshape(1000,1)
    ftx213 = X[:,0].reshape(1000,1)*X[:,2].reshape(1000,1)
    ftx214 = X[:,0].reshape(1000,1)*X[:,3].reshape(1000,1)
    ftx224 = X[:,1].reshape(1000,1)*X[:,3].reshape(1000,1)

    #ft = np.concatenate((ftx31, ftx32, ftx33, ftx34, ftx21, ftx22, ftx23, ftx24, ftx212, ftx223, ftx234,
    ftx213, ftx214, ftx224, X), axis=1)
    ft = np.concatenate((ftx21, ftx22, ftx23, ftx24, ftx212, ftx223, ftx234, ftx213, ftx214, ftx224, X),
    axis=1)
    #ft = np.concatenate((ftx31, ftx32, ftx33, ftx34, ftx21, ftx22, ftx23, ftx24, ftx212, ftx223, ftx234,
    X), axis=1)
    return ft

def solve(X,Y):
    NUM, DIM = X.shape
    P = np.eye(DIM)
    q = np.zeros(DIM)

    G = featureMap*Y*-1.
    h = np.ones(NUM)*-1.
```

```

return quad.solve_qp(.5 * P, q, -G.T, -h, 0)

def score(W, X, Y):
    pred = np.dot(X, W)
    pred[pred>0] = 1
    pred[pred<0] = -1
    return np.sum(pred-Y)

def support_vectors(lagrange):
    lagrange = lagrange > 0
    return X[lagrange.ravel()]

featureMap = featureTransformation(X)
solution = solve(featureMap,Y)
W = solution[0]

print("WEIGHT VECTORS ", W[:14])
print("BIAS ", W[14])
print("MARGIN ", (1 / np.sqrt((W[0:14] ** 2).sum()))))
print("SCORE ", score(W, featureMap, Y))
print("SUPPORT VECTORS ", support_vectors(solution[4].reshape(1000,1)))

#print("WEIGHT VECTORS ", W[:18])
#print("BIAS ", W[18])
#print("MARGIN ", (1 / np.sqrt((W[0:18] ** 2).sum()))))
#print("SCORE ", score(W, featureMap, Y))
#print("SUPPORT VECTORS ", support_vectors(solution[4].reshape(1000,1)))

#print("WEIGHT VECTORS ", W[:15])
#print("BIAS ", W[15])
#print("MARGIN ", (1 / np.sqrt((W[0:15] ** 2).sum()))))
#print("ERROR ", score(W.reshape(16,1), featureMap, Y))
#print("SUPPORT VECTORS ", support_vectors(solution[4].reshape(1000,1)))

```

## RESULT:

```

('WEIGHT VECTORS ', array([ 5.57715117e+01,  2.43938599e+00, -1.19481235e+01,
        -2.85422573e+01,  1.24810054e+02, -3.79566720e+00,
         9.83338305e+00, -8.41185412e+01,  7.14584105e-02,
         4.88019179e+00,  1.07538595e+02, -9.63224918e+00,
         1.96300790e+00, -4.76537826e+01, -7.53784124e+01]))
('BIAS ', 82.066939694760279)
('MARGIN ', 0.0046385760055291589)
('ERROR ', 0.0)
('SUPPORT VECTORS ', array([[ 0.01852987,  0.50782148,  0.71364513,  0.69774797,  1.      ],
        [ 0.67760052,  0.01731556,  0.22660751,  0.91446604,  1.      ],
        [ 0.56116788,  0.94932549,  0.95929542,  0.9899408 ,  1.      ],
        [ 0.523322 ,  0.39248601,  0.06702511,  0.77851889,  1.      ],
        [ 0.06156021,  0.65899664,  0.0993981 ,  0.60866886,  1.      ]],

```

[ 0.02269783, 0.29143716, 0.35408261, 0.64963031, 1. ],  
[ 0.19385208, 0.83338556, 0.74510427, 0.74385454, 1. ],  
[ 0.16933499, 0.62115084, 0.42461128, 0.67708347, 1. ],  
[ 0.50743509, 0.14759421, 0.92070003, 0.92946127, 1. ],  
[ 0.22847848, 0.0126919 , 0.12022385, 0.63297825, 1. ],  
[ 0.38033462, 0.37208327, 0.42427464, 0.74312689, 1. ],  
[ 0.60270148, 0.62545054, 0.01968289, 0.8197326 , 1. ]))