**Lab Submission – 5**

**Support Vector Machine (SVM)**

**Source Code:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

cell\_df = pd.read\_csv('cell\_samples.csv')

# print(cell\_df.head(5))

""" cell\_df.head(15)

cell\_df.tail(5)

cell\_df.shape    #returns rows and columns i.e (699, 11)

cell\_df.size     #returns size in bytes.

cell\_df.count()  #returns how many values are present in each coulmns """

print(cell\_df['Class'].value\_counts())       #Returns the values of the specific coulmns and its dtype also.

malignant\_df = cell\_df[cell\_df['Class']==4][0:200]

benign\_df = cell\_df[cell\_df['Class']==2][0:200]

# help(benign\_df.plot())       #Plot is method to use different kind of plots.

axes = benign\_df.plot(kind='scatter', x='Clump', y='UnifSize', color='blue', label='Benign')

malignant\_df.plot(kind='scatter', x='Clump', y='UnifSize', color='red', label='Malignant', ax=axes)

# Note: There's a parameter i.e ax which represents that in which you want to plot that data.

# print(cell\_df.dtypes)

cell\_df = cell\_df[pd.to\_numeric(cell\_df['BareNuc'], errors='coerce').notnull()]

cell\_df['BareNuc'] = cell\_df['BareNuc'].astype('int')

print(cell\_df.dtypes)

print(cell\_df.columns)

feature\_df = cell\_df[['Clump', 'UnifSize', 'UnifShape', 'MargAdh', 'SingEpiSize',

       'BareNuc', 'BlandChrom', 'NormNucl', 'Mit']]

# cell\_df 100 rows and 11 columns,

# picked 9 columns out of 11

# Independent variable (are those who will contribute in matching the results to dependent variable)

X = np.asarray(feature\_df)

# Dependent variable (are those which we have to predict)

Y = np.asarray(cell\_df['Class'])

print(X[0:5])

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=4)

# returns 4 numpy array as o/p in the above order.

# 546 x 9

print(X\_train.shape)

# 546 x 1

print(Y\_train.shape)

# 137 x 9

print(X\_test.shape)

# 137 x 1

print(Y\_test.shape)

from sklearn import svm

classifier = svm.SVC(kernel='linear', gamma='auto', C=2)

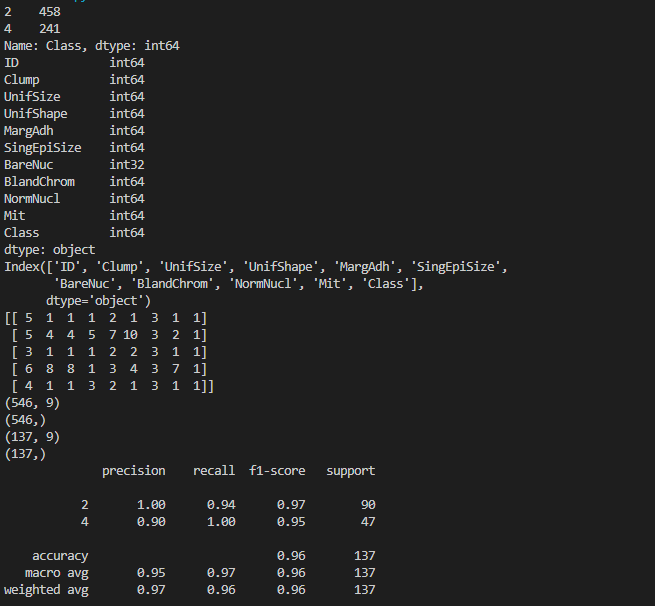
classifier.fit(X\_train, Y\_train)

Y\_predict = classifier.predict(X\_test) #returns 1D array.

from sklearn.metrics import classification\_report

print(classification\_report(Y\_test, Y\_predict))

**Output:**

****