**Department of Computing**

**CS370: Artificial Intelligence**

**Class: BSCS-9AB**

**Lab 08: Support Vector Machines**

**Date: 31-03-2022**

**Time: 10:00-13:00 (A) and 14:00 to 17:00 (B)**

**Instructor: Dr. Hashir Kiani**

**Lab Engineer: Ms Shakeela Bibi**

# Lab 08: Support Vector Machines

**Lab Task: Multi-Class Classification through SVMs**

In this lab, you are supposed to perform multi-class classification through Support Vector Machines. Data in excel files (both the training and test sets) are uploaded on LMS. In the said training and test data files, each row contains data about one instance of a plant category where four predictors/attributes are recorded for each plant (namely, leaf length, leaf width, flower length, and flower width), while “plant” is the target class which could be any one of the following at a time: “Arctica” or “Harlequin” or “Caroliniana”.

You are supposed to perform multi-class classification using SVMs. Use the training data to train your SVM Classifier. Then use the test data to check the accuracy of your classifier. Please submit your results in the form of an excel file which should contain the prediction for each example in test data.

**Note:** You are expected to perform the above-mentioned task using traditional SVMs. Off the shelf libraries like *Scikit-learn and LIBSVM* might be a useful resource during implementation. Feel free to use these-or any other suitable library you prefer.

**Code**

**import sklearn as svm**

**import pandas as pd**

**import numpy as np**

**# Read the excel file as a pandas DataFrame**

**df = pd.read\_excel(r'/content/TrainingSet.xlsx')**

**# Convert the DataFrame to a numpy array and transpose it to have the features as columns**

**arr=np.array(df)**

**arr=np.transpose(arr)**

**# Print the number of rows in the array**

**print("Number of Rows:",len(arr))**

**# Get the input features (x\_train) by selecting the first 4 columns of the transposed array**

**x\_train=arr[0:4]**

**# Transpose x\_train to have the features as columns again**

**x\_train=np.transpose(x\_train)**

**print("")**

**print(x\_train)**

**# Get the output variable (y\_train) by selecting the last column of the transposed array**

**y\_train=arr[4:5]**

**# Transpose y\_train to have a single column of output values**

**y\_train=np.transpose(y\_train)**

**print("")**

**print(y\_train)**

**# Import train\_test\_split function**

**from sklearn.model\_selection import train\_test\_split**

**# Split dataset into training set and test set**

**X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(x\_train, y\_train, test\_size=0.3,random\_state=109) # 70% training and 30% test**

**#Import svm model**

**from sklearn import svm**

**#Create a svm Classifier**

**clf = svm.SVC(kernel="linear") # Linear Kernel**

**#Train the model using the training sets**

**clf.fit(X\_train,Y\_train)**

**#Predict the response for test dataset**

**PlantName = clf.predict(X\_test)**

**# Model Accuracy: how often is the classifier correct?**

**print("Accuracy:",round(metrics.accuracy\_score(Y\_test, PlantName)\*100,2),"%")**

**# Import necessary modules**

**from sklearn import metrics**

**import sklearn as svm**

**import pandas as pd**

**import numpy as np**

**# Import module for writing to Excel**

**import xlwt**

**from xlwt import Workbook**

**# Read the test dataset from Excel file and convert to numpy array**

**df = pd.read\_excel(r'/content/TestingSet.xlsx')**

**arr=np.array(df)**

**arr=np.transpose(arr)**

**# Print the number of rows in the array**

**print("Number of Rows:",len(arr))**

**# Get the input features for prediction (X\_Test\_Predict) by selecting the first 4 columns of the transposed array**

**X\_Test\_Predict=arr[0:4]**

**X\_Test\_Predict=np.transpose(X\_Test\_Predict)**

**# Predict the response for the test dataset using a trained classifier (clf)**

**PlantName = clf.predict(X\_Test\_Predict)**

**# Create a list to store the predicted output values along with the input features as a string**

**result=[]**

**for row in range(len(PlantName)):**

**word=str(X\_Test\_Predict[row][0])+','+str(X\_Test\_Predict[row][1])+','+str(X\_Test\_Predict[row][2])+','+str(X\_Test\_Predict[row][3])+','+str(PlantName[row])**

**result.append(word)**

**print(result)**

**# Create a new Excel workbook**

**wb = Workbook()**

**# Add a new sheet to the workbook**

**sheet1 = wb.add\_sheet('Data Final')**

**# Write the header row to the sheet**

**sheet1.write(0, 0, 'leaf.length')**

**sheet1.write(0, 1, 'leaf.width')**

**sheet1.write(0, 2, 'flower.length')**

**sheet1.write(0, 3, 'flower.width')**

**sheet1.write(0, 4, 'plant')**

**# Loop through the predicted output values and input features and write them to the sheet**

**for row in range(len(PlantName)):**

**sheet1.write(row+1, 0, str(X\_Test\_Predict[row][0]))**

**sheet1.write(row+1, 1, str(X\_Test\_Predict[row][1]))**

**sheet1.write(row+1, 2, str(X\_Test\_Predict[row][2]))**

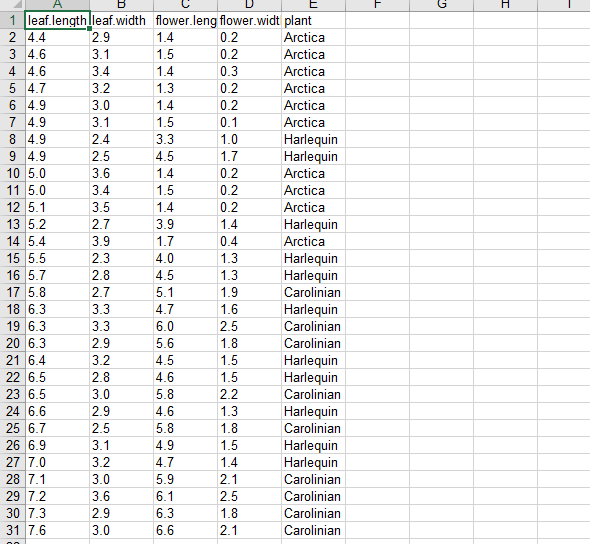
**sheet1.write(row+1, 3, str(X\_Test\_Predict[row][3]))**

**sheet1.write(row+1, 4, str(PlantName[row]))**

**# Save the workbook to a new Excel file**

**wb.save('Results.xlsx')**

**ScreenShot**

****

**Findings:**

* The code uses a support vector machine (SVM) algorithm from the sci-kit-learn library to predict the plant species based on input features such as leaf length, leaf width, flower length, and flower width.
* The code first reads the training data from an Excel file, converts it into a NumPy array, and then splits it into training and testing sets. The SVM classifier is trained on the training set, and its accuracy is measured on the testing set.
* The trained model is then used to predict the plant species for a new set of input features from another Excel file.
* Finally, the predicted plant species and the input features are written to a new Excel file using the xlwt library.