School of Computer Science Engineering and Technology

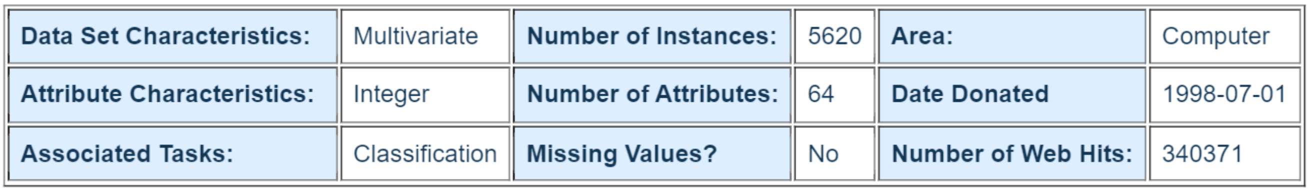
Course- BTech	Type- Core
Course Code- 301	Course Name-AIML
Year- 2022	Semester- Even
Date- 04-04-2022	Batch- 4th Sem (SPL)

10 - Lab Assignment No. 10.1

Objective: To Implement Support Vector Machine Classifier

Problem Statement: Build a SVM Classifier using Sklearn for Multiclass Classification on handwritten digit dataset.

About Dataset: This dataset summarizes a Optical recognition of handwritten digits dataset (5 M).



Data Set Characteristics:	Multivariate	**Number of Instances:**	5620	**Area:**	Computer
Attribute Characteristics:	Integer	**Number of Attributes:**	64	**Date Donated**	1998-07-01
Associated Tasks:	Classification	**Missing Values?**	No	**Number of Web Hits:**	340371

This is a copy of the test set of the UCI ML hand-written digits datasets

https://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits

The data set contains images of hand-written digits: 10 classes where each class refers to a digit. Preprocessing programs made available by NIST were used to extract normalized bitmaps of handwritten digits from a preprinted form. From a total of 43 people, 30 contributed to the training set and different 13 to the test set. 32x32 bitmaps are divided into nonoverlapping blocks of 4x4 and the number of on pixels are counted in each block. This generates an input matrix of 8x8 where each element is an integer in the range 0..16. This reduces dimensionality and gives invariance to small distortions.

Steps

1. **Dataset:** Download the dataset from the link https://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits
   Or use load_digits() to load the dataset (5)
2. Check the shape of data (5)
3. Display the first 10 images using matplotlib (10)
4. Extract the Independent and Dependent Variable (Hint: Indep Var=digits.data, Dep Var= dogits.target) (10)
5. Split the dataset into 80% for training and rest 20% for testing (sklearn.model_selection.train_test_split function) (10)
6. Build a SVM model using Sklearn with default parameters. (10)
7. Predict the target values in the testing set. (5)
8. Apply classification metrics and visualize the results as graphs. (10)
9. Playing with SVM: Change the following parameters of the SVM and analyze their performance for training and testing using the evaluation measures. (20)
   a) Kernel: {‘linear’, ‘poly’, ‘rbf’, ‘sigmoid’, ‘precomputed’}

   b) degreeint, default=3
   c) gamma{‘scale’, ‘auto’} or float, default=’scale’
   d) random_stateint, RandomState instance or None, default=None
   e) Cfloat, default=1.0

10. Compare the performance of the SVM model with other classification models (20).

Suggested Platform: Python: Azure Notebook/Google Colab Notebook, packages such as numpy, pandas, sklearn, matplotib.