

Machine Learning

Digits Data Set – Linear SVM Classifier

Use the digits dataset available under SKLearn.

Consider the data corresponding to classes 0 and 1 only.

Each pattern is a 8 X 8 sized character where each value is an integer in the range 0 to 16.

Convert it into a binary form by replacing a value below 8 by 0 and other values (≥ 8) by 1.

split the dataset into train and test parts.

Do this splitting randomly 10 times and report the average accuracy using Linear SVM classifiers.

CODE:

Please find the code committed for SVM as [LinearSVM_DigitsDataSet_Impl.py](#)

- SVM classifier from sklearn linear model is used to classify the data in this subtask.
- The average accuracy is computed for 10 trials.

RESULT:

Average Accuracy for SVM Classifier for test_size= 0.1 : 0.9986111111111111
Average Accuracy for SVM Classifier for test_size= 0.2 : 0.9986111111111111
Average Accuracy for SVM Classifier for test_size= 0.3 : 0.9986111111111111
Average Accuracy for SVM Classifier for test_size= 0.4 : 1.0
Average Accuracy for SVM Classifier for test_size= 0.5 : 0.9986111111111111

INFERENCE/ANALYSIS:

- SVM works well in the given digit dataset with 2 classes that has 64 features and not a very high range of samples.
- For different test sizes, the accuracy has remained high when averaged over 10 trials
- Since there is a clear margin of separation between class 0 and class 1 , SVM seems to be performing well here.

RESOURCES USED FOR THE ASSIGNMENT:

- **Environment:**
Anaconda, Jupyter notebook
- **Software :**
Python
Python libraries/modules: Pandas, Numpy, SkLearn etc