Machine Learning

Digits Data Set – Semi Supervised KNNC

Use the digits dataset available under SKLearn.
Consider the data corresponding to classes 0 and 1 only. Each pattern is a 8 × 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. Use this binary data in the following
tasks
Use 20 patterns from each class with labels and the remaining without
the labels for this subtask. Use the *KNNC* and label the patterns without labels.
Obtain the % classification accuracy. Perform this task with K values in the set
*{*1, 3, 5, 10, 20*}*

CODE:

Please find the code attached for KNNC :

- The digits dataset for class 0 and class 1 only are considered and the data values are converted to value 0 or 1 as defined in problem statement above.
- Train and Test split is done and KNNC is applied over the dataset and accuracy is found.
- KNNC is applied for different values of K like 1,3,5,10 and 20.
- Average accuracy is calculated over 10 random iterations

RESULT:

```
Average Accuracy of 10 iteration is
Average Accuracy for n_neigbours= 1  :  0.999375
Average Accuracy for n_neigbours= 3  :  0.9971875000000001
Average Accuracy for n_neigbours= 5  :  0.9956250000000001
Average Accuracy for n_neigbours= 10 :  0.99
Average Accuracy for n_neigbours= 20 :  0.98125
```

INFERENCE/ANALYSIS:

- **The average accuracy is least for 20 neighbors.** This shows that when all test data is considered in the neighbor set , then accuracy of classifying this data gets lower when compared to other values of nearest neighbors.
- With smaller smaller values of K (1,3,5) we are seeing relatively higher accuracy which is in line with our understanding of KNNC.
- **The overall KNNC gives** a high accuracy for this dataset in semi supervised learning scenario.

RESOURCES USED FOR THE ASSIGNMENT:

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