CS422/622 - HW 2

In HW2, we will implement KNN from scratch using Python. You are given two data sets: MNIST_training.csv and MNIST_test.csv (links below), where "MNIST_training.csv" contains a training data that you will find the K-nearest neighbors, whereas "MNIST_test.csv" consists of a test data that you need to predict. The training data contains 10 classes (i.e., 0, 1, 2, …, 9), each of which has 95 samples as training samples, while there are 5 samples on each class in the test data set.

MNIST_training.csv:
http://mkang.faculty.unlv.edu/teaching/CS422_622/HW2/MNIST_training.csv

MNIST_test.csv: http://mkang.faculty.unlv.edu/teaching/CS422_622/HW2/MNIST_test.csv

You can find the description of the MNIST data at https://www.kaggle.com/c/digit-recognizer/data, but have to use the given simplified data sets.

For this homework assignment, please follow the steps:

1. For each test data in "MNIST_test.csv", compute distances or similarity (Euclidean, Manhattan, or Cosine similarity) with the training data.
2. Find the K-nearest neighbors and decide the majority class. You can empirically specify the value of "K".
3. Compare the prediction with the ground truth in the test data
    a. Correctly classified if the predicted label and the ground truth is identical.
    b. Incorrectly classified if the predicted label and ground truth is NOT identical.
4. Repeat Step 1-4 for all data in the test data
5. Then, you can count how many test data are correctly classified and incorrectly classified.
6. Show the accuracy of your KNN. Compute accuracy by:

$$accuracy = \frac{\#\, of\ your\ predictions\ correctly\ classified}{\#\, of\ total\ test\ data}$$

<u>For graduate students in CS622:</u>

Merge the given training and test data into a file. Then, perform 5-fold cross validation with KNN, where the dataset is split into training and test and computes accuracy five times. Please add a table in the experimental results.

Experiment	Accuracy
Experiment 1	
Experiment 2	
Experiment 3	
Experiment 4	
Experiment 5	
Average	

You CANNOT use any libraries or built-in functions of KNN. You can use any non-KNN specific library such as numpy or pandas. You have to implement from scratch.

You must submit the followings to UNLV WebCampus:

1. MS word file
   - Describe what you did for the homework assignment.
   - Clearly show how to execute your python code (e.g., python version and command)
2. Source code file(s)
   - Must be well organized (comments, indentation, …)
   - **You need to upload the original python file (*.py). Don't upload jupyter notebook files**

You must submit the files SEPERATELY. DO NOT compress into a ZIP file. If you fail to provide all required information or files, you may be given zero score without grading.

Grading guideline:

- If implemented by using built-in KNN library, zero will be given.
- KNN algorithm should be correctly implemented
- Accuracy must be computed by using the test data, not training data.
- Accuracy is correctly measured or not

Bonus:

A couple of best codes will be selected with 20 extra points. The best codes will be shared as a reference.

Deadline:

The deadline is **11:59pm Friday, February 16, 2024**. Late assignments will not be accepted.