CS489/689- 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 (link below), where “MNIST_training.csv” contains training data that you will find the K-nearest neighbors, whereas “MNIST_test.csv” consists of test data that you need to predict labels. The training data contains 10 classes (i.e., 0, 1, 2, …, 9), each of which has 95 samples, while there are 5 samples on each class in the test data set.

MNIST_training.csv: http://mkang.faculty.unlv.edu/teaching/CS489_689/HW2/MNIST_training.csv

MNIST_test.csv: http://mkang.faculty.unlv.edu/teaching/CS489_689/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 with the training data.
2. Find the K-nearest neighbors and decide the majority class.
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:

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

You CANNOT use any libraries or built-in functions of KNN. You have to implement it.

<u>You need to think of what is the optimal “K” in KNN.</u> Describe how you decided the optimal K in the assignment.

You must submit the followings to UNLV WebCampus:

1. MS word file
    - Describe what you did for the homework assignment.
    - <u>Accuracy by KNN (with different K) → Table or plot</u>
    - 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 R or python file (*.r or *.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:

- KNN algorithm should be correctly implemented
- Must show accuracy with different “K”
- Accuracy is correctly measured or not

Deadline:

The deadline is **11:59pm Wednesday, Feb 19, 2020**. Late assignments will not be accepted.