CSC 535/635 – HW 3

**Instructions:** Students are required to work alone on this assignment.

For this assignment, you will **implement** the KNN algorithm as given on slide 38 of Module 8 PowerPoint slides on Blackboard. The only difference is that while the given algorithm uses simple unweighted voting, **you will use weighted voting in your implementation.** In simple unweighted voting, once the k nearest neighbors are found, their distances from the test sample do not matter. One neighbor is one vote. **In weighted voting**, the vote of a neighbor is inversely proportional to its distance to the test sample. **You need to use Python or IPython for your implementation.**

Use the given MNIST\_train.cvs as your training data set and MNIST\_test.csv as the test data set. There are 10 classes, with labels 0, 1, 2, …, 9, for this data set. The first attribute/column is the class label. Also notice that the first line/row in both data sets is a headers line. Do not modify the given data sets in any way because your code will be graded using them. In your code, you can just skip over the header lines. A description of the MNIST data is available at <https://www.kaggle.com/c/digit-recognizer/data>

The output from your program will display the following:

* value of K
* for each test sample, print both the desired class and the computed class, where desired class, is the class label as given in the data set, and computed class, is what your code produces as the output for the sample
* the accuracy rate
* number of misclassified test samples, and
* total number of test samples

Sample output is as follows. Notice that, this sample output does not show the best value of K:

K = 4

Desired class: 0 computed class: 0

Desired class: 0 computed class: 0

Desired class: 0 computed class: 0

Desired class: 0 computed class: 0

Desired class: 0 computed class: 0

Desired class: 1 computed class: 1

Desired class: 1 computed class: 1

….

Desired class: 2 computed class: 2

Desired class: 2 computed class: 6

Desired class: 2 computed class: 2

Desired class: 3 computed class: 7

Desired class: 3 computed class: 3

…

Desired class: 9 computed class: 9

Desired class: 9 computed class: 9

Accuracy rate: 86.0%

Number of misclassified test samples: 7

Total number of test samples: 50

# **Remarks:**

* Use Euclidean distance measure to compute distances. In your code, do not use any built-in function for this, you need to provide your own implementation of Euclidean distance. This is to make sure that students know how to calculate Euclidean distance between two vectors.
* You may use a random sample of the training data to decide on the value of K to use for the algorithm.
* Make sure to use the data sets provided with this homework assignment and not any other instance of MNIST

# **What to turn in?**

Upload to Blackboard copies of your source code and **report** describing how you choose k, the accuracy of the algorithm, and any comments about this homework assignment you may want to share.

Please make sure that your code is well organized and properly documented and commented. Please use the Homework Report Template available on Blackboard to write your report.

Name the code file hw3.py or hw3.ipynb as appropriate. Name the report hw3\_Report.docx.