The George Washington University
School of Engineering and Applied Science
Department of Computer Science
CSCI 4364/6364 Machine Learning

Homework 1:

Objective: In this assignment, you will construct an image classifier using the k-nearest neighbor algorithm. You should evaluate for a variety of k values and submit your best answers for the test set.

You may use existing libraries and packages. For example, if you are using Python, you might have code fragments of the following form:

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(i)
knn.fit(X_train,y_train)
knn.score(X_train,y_train)
knn.score(X_test,y_test)
```

Data Description:

The data files data_mnist.csv and test_mnist.csv contain gray-scale images of hand-drawn digits, from zero through nine.

Each image is 28 pixels in height and 28 pixels in width, for a total of 784 pixels in total. Each pixel has a single pixel-value associated with it, indicating the lightness or darkness of that pixel, with higher numbers meaning darker. This pixel-value is an integer between 0 and 255, inclusive.

The training data set, (data_mnist.csv), has 785 columns. The first column, called "label", is the digit that was drawn by the user. The rest of the columns contain the pixel-values of the associated image.

Each pixel column in the training set has a name like pixelx, where x is an integer between 0 and 783, inclusive. To locate this pixel on the image, suppose that we have decomposed x as x = i * 28 + j, where i and j are integers between 0 and 27, inclusive. Then pixelx is located on row i and column j of a 28 x 28 matrix, (indexing by zero).

For example, pixel31 indicates the pixel that is in the fourth column from the left, and the second row from the top, as in the ascii-diagram below.

The test data set, (test_mnist.csv), is the same as the training set, except that it does not contain the "label" column.

Submission:

Your submission file should be in the following format: For each of the 10k images in the test set, output a single line containing the ImageId and the digit you predict. For example, if you predict that the first image is of a 3, the second image is of a 7, and the third image is of a 8, then your submission file would look like:

```
ImageId,Label
1,3
2,7
3,8 (9997 more lines)
```

You will also submit a short 1 to 2 (max) page writeup of how you evaluated the performance of your MNIST kNN model and decided on the k value.

Note: The pixel values are between 0 and 255, you should appropriately scale those. If using Python, it contains both scaling packages as well as train-test split algorithms. For example,

```
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()

from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.25,random_state=42, stratify=y)
```

You will need to split the data_mnist.csv data appropriately to train and test your model before using your model to predict the classes for test_mnist.csv