Answer 1:

We have implement nearest neighbor classifier for digit classification
There are around 1800 images in total with 10 digit classes, and each image is 8x8 sized with single channel.

Steps followed in implementing KNN are below:

- Load the MNIST digits dataset
- Splitting the dataset into train and test sets, keeping 500 images for testing
- Viewing the image and label with matplotlib to check how it looks like
- Implemented 3 methods:
  - Function to Calculate Euclidean Distance
    - Used L2-Norm to calculate the distance between two samples
  - Function to implement K-Nearest Neighbors
    - For each testing sample, calculate the distance between all training samples and test point.
    - Store all the distance values in a distances array
    - Sort the distances
    - Select the top K predictions
    - Find the most frequent out of first k predictions as majority vote
  - Function to calculate accuracy
    - It calculates the accuracy of KNN model with specific K value
- Determining Optimal Number of Neighbors
  - To detect optimal value for 'K', let's compare the accuracy of KNN models with different 'K' values.
  - By plotting the values in a graph, we can see the performance visually, and select the best performing 'K' value
  - Values of K, and performance
    - K = 1, accuracy = 98.6
    - k = 2, accuracy = 98.6
    - k = 3, accuracy = 98.6
    - k = 4, accuracy = 98.4
    - k = 5, accuracy = 98.6
    - k = 6, accuracy = 98.2
    - k = 7, accuracy = 98.4
    - k = 8, accuracy = 98.0
    - k = 9, accuracy = 97.8

Looking at the performance scores, K = 1 might seem the most promising model as it has the highest accuracy.

But pragmatically, trying to predict a data point based on only one neighbor is definitely not a better choice to make.

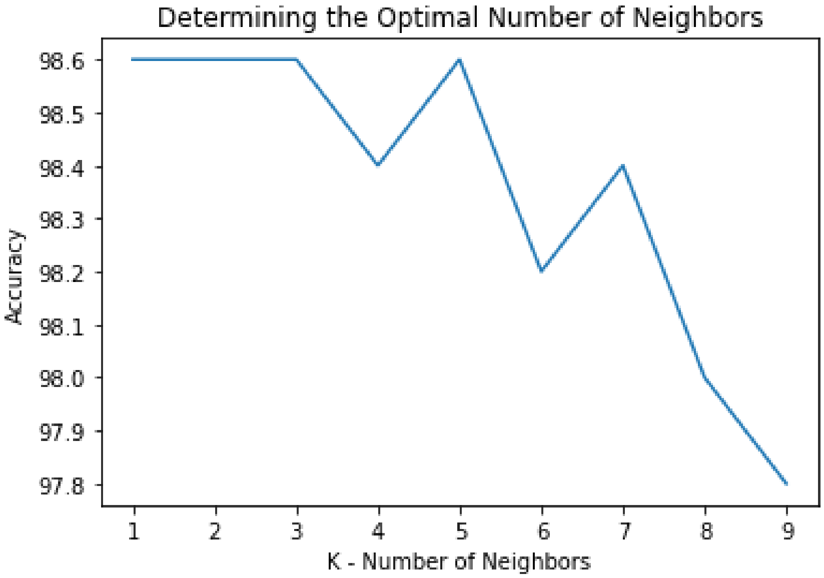
So, it's a wise choice to ignore the option.

For K = 2 and K = 3 it stays the same

For K = 4, there is a slight deep but when neighbors are increased to 5, i.e., when K = 5, it get again increase and gives the accuracy of 98.6%.

But after that from K = 6, it start decreasing the accuracy and there is a huge dip in performance for K = 8 and K = 9.

Below is the performance curve



It's up to us to choose either 3 or 5 for our requirement. Both give the same accuracy.

KNN Model with 5 neighbors performed really well with 98.6% accuracy. So, we will go with this for this model.

GitHub Link:

https://github.com/SunilDevlops/Programs/blob/master/ComputerVision/ProgramAssignment2/Question1/KNN_MNIST_Digit_Classification.ipynb