Offline 4 on KNN

In this offline, you will implement KNN for both classification and regression.

KNN Classification [4]

1. Read the ‘iris.csv’ file and randomly partition into 3 portions [training data(70%), validation data(15%), test data(15%)]. Last column represents the discrete output classes.

2. For each data from the validation set, calculate its Euclidean distance to all the training data points. Consider the K closest training data points and consider the majority class as the predicted class.
3. Calculate the total accuracy for this K value.
   Accuracy = correctly classified data / total number of data

Perform step 2 & 3 multiple times for the following values of K and fill up the accuracy column values below:

K	Accuracy
3	
5	
7	
9	
11	
13	
15	

4. Choose the K with the maximum accuracy as your final value of K.

5. Now for each data from the test set, calculate its Euclidean distance to all the training data points. Consider the K (received from step 4) closest training data points and consider the majority class as the predicted class.
6. Calculate the accuracy for this test data and fill up the following table.

Test Data	Accuracy = … …

Note: I have partially implemented the algorithm in ‘KNN_classifier.py’ file. Your task is to understand the code and implement only the predict_each_data() method of KNN_Classifier class.

KNN Regression [6]

1. Read the ‘diabetes.csv’ file and randomly partition into 3 portions [training data(70%), validation data(15%), test data(15%)]. Last column is the continuous output label.

2. For each data from the validation set, calculate its Euclidean distance to all the training data points. Consider the K closest training data points and calculate the average value of their outputs as your predicted output.

   Squared Error for this data = (real output-predicted output)^2
3. Calculate the mean squared error.
   Mean squared error=sum of all the squared errors of step 2/number of data points.

Perform step 2 & 3 multiple times for the following values of K and fill up the mean squared error column values:

K	MSE
3	
5	
7	
9	
11	
13	
15	

4. Choose the K with the minimum error as your final value of K.

5. Now for each data from the test set, calculate its Euclidean distance to all the training data points. Consider the K (received from step 4) closest training data points and calculate the average value of their outputs.
6. Similarly, calculate the mean squared error for this test data and fill up the following table:

Test Data	MSE = … …

Note: you must implement this KNN Regression within a python class. Add necessary methods and variables on your own. Consider the KNN Classification code as your reference.

[Zero Tolerance for Plagiarism]