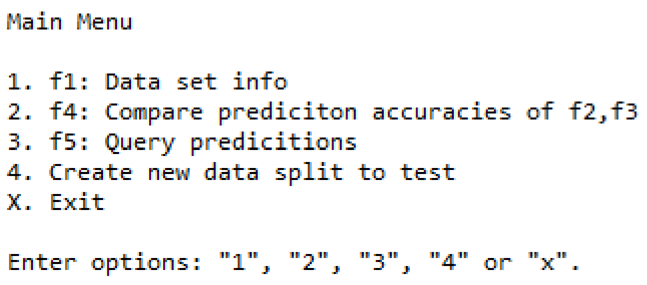
COMP219 Assignment 1

I've chosen to implement a KNN algorithm to predict Digits from images. There are no saved models for the KNN algorithm. When you load my program, it loads the digits data set and splits it into 80% training data and 20% test data. The split selects random instances from the data set. New random instances can be selected through my "createNewDataSplit" function. To change the percentage of training and test data requires manually changing the code.

How to Run the Program

I've provided a main menu to access all functionalities. Simply follow input instructions.



```
Main Menu

1. f1: Data set info
2. f4: Compare prediciton accuracies of f2,f3
3. f5: Query predicitions
4. Create new data split to test
X. Exit

Enter options: "1", "2", "3", "4" or "x".
```

Checklist:

1. f1: Covered by option 1
2. f2: Covered by option 4
3. f3: Covered by option 4
4. f4: Covered by option 4
5. f5: Covered by option 3

Functionality 4 uses functions which implement f2 and f3.

Software Dependencies

I have imported several libraries and used various methods from them:

1. Sklearn
2. Math
3. Numpy
4. Matplotlib

Functionalities and Additional Requirements

f1

Takes no parameters and returns nothing. F1 displays information about the dataset being used.

f2

I have implemented f2 as a function that takes training data and targets as well as test data and targets. It uses SciKitLearn's KNN algorithm to go through each test and make a prediction based on the training data. It then calculates an accuracy based how many predictions were correct and returns it.

There is no need to directly call f2 as f4 displays its information and you can query individual predictions and what they should have been through option 3 in the main menu. However, if you needed to call it, make sure you setup the data properly and input the parameters in the following order. Data train, data test, target train, target test. Laying out the data shouldn't be too tricky as I have included a function that splits the data for you in the correct order. I have also included some global variables with the correct layout that can be updated with option 4 from the main menu.

f3

f3 is similar to f2. It takes the same parameters in the same layout and returns the accuracy of its predictions for each test digit. However, unlike f2, I have implemented the KNN algorithm myself in a separate function for use in f3.

My “knn” function takes a k value, a 2-dimensional array combining all training data’s digit data and its target value and finally the digit data and target value for the number to be predicted. In other words, the parameters are “k”, “data” and “prediction”. It then stores the Euclidean distances between all training data and the data of the value to be predicted. Then it calculates the k nearest distances. In my code, “nearest” is the array of length k storing the nearest data to the prediction where I have chosen k to be the square root of the number of training samples. From these it works out which target number the predicting data was closest to by looking at the frequency of training targets in the “nearest” array. This becomes the prediction and is returned to the function f3.

To return the accuracy of its predictions, f3 makes a call to my “knn” function for every test instance and compares with the test targets to count how many predictions were correct. This means that for all 360 test instances, my program iterates over all 1437 training data instances twice (once to calculate distances, then to work out nearest neighbours) as well as 2 further iterations over the “nearest” array to determine the most frequent nearest neighbour. This alone contributes largely to the slow computation time of my algorithm. On my system this takes about 5 seconds but may take longer on other systems. (the number of training and test instances can be changed in the code, but I chose to do an 80%, 20% split).

There is no need to directly call f3 as f4 displays its information and you can query individual predictions and what they should have been through option 3 in the main menu. However, if you needed to call it, follow the same instructions as discussed for f2.

f4

f4 takes no inputs and returns nothing. It just displays the accuracies and errors of both KNN algorithms by calling f2 and f3.

f5

f5 takes no inputs and returns nothing. It allows a user to input an index of the test data. It then prints the target and image at this index and displays the predictions made by f2 and f3. There is no error handling for the user’s input here so if the user inputs an index outside of the test data’s range or any unexpected values there will be an error.

Other Functions

“SplitData” loads the data and returns the 80%, 20%, training, test split. This is called to be stored in global variables at the start of the program and can be changed later in the program with my “createNewDataSplit” function.

The “mainMenu” function is just a basic user interface to make it easier for users to utilise the functionalities from the spec. It’s possible to manually call each individual functionality but hopefully unnecessary.