PROJECT DESCRIPTION

The project required us to perform different classification techniques on two different datasets namely ATNTFaceImages400 and HandWrittenLetters. These classification techniques were implemented using four different classifiers which were K-Nearest, Nearest Centroid, SVM and Linear Regression to perform operations like prediction and k-fold cross-validation.

TASK A

For TASK A, using the data handlers on dataset HandWrittenLetters selected “A, B, C, D, E” classes. And for this smaller dataset, using another data handler generated test and training data: for each class using the first 30 images for training and the remaining 9 images for test. Later these test and train data were passed on to the four classifiers to predict the accuracy.

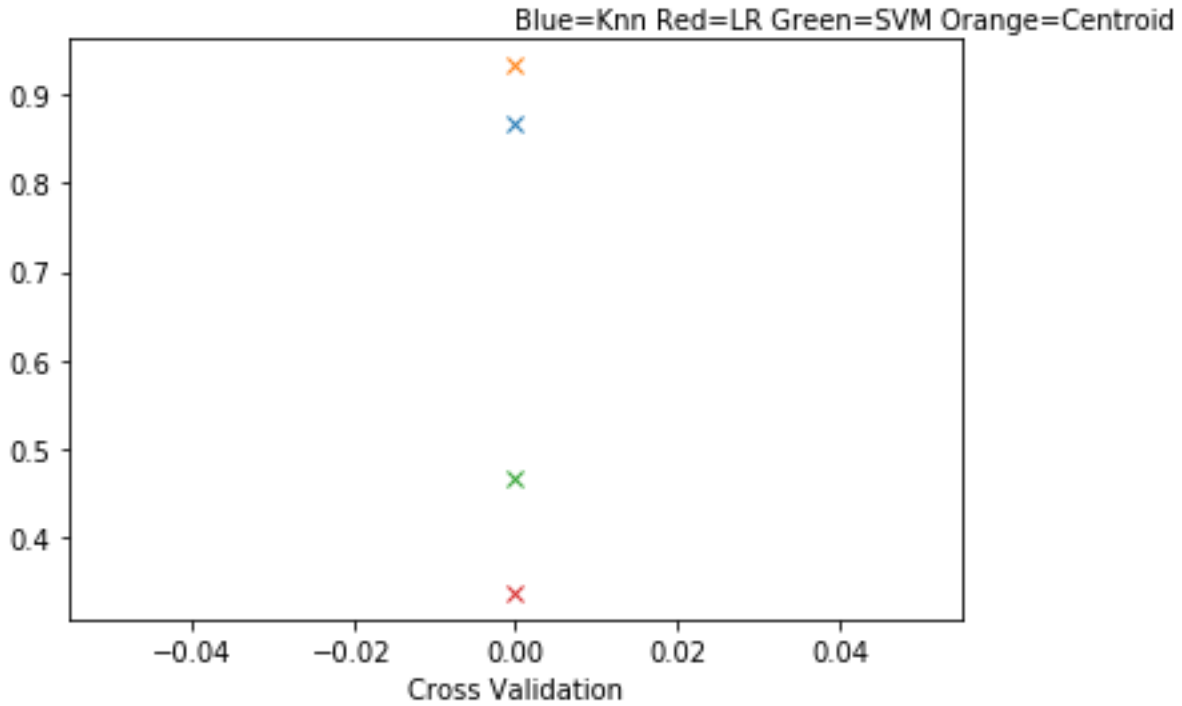
```
Enter the number of classes that you want to consider: 5
Enter the elements in order"Write in Uppercase only":

A

B

C

D

E

Enter the number of training elements: 30

Do you want to perform prediction or cross validation: 0



The Accuracy Scores are:
0.8
0.9111111111111111
0.9111111111111111
0.8954222155544589
for KNN,Centroid, SVM and LR respectively
```

TASK 2

On ATNTFaceImages400 performed 5-fold cross validation and found the accuracy of all the classifiers (KNN, Centroid and SVM) and finally plotted the average accuracy of all the classifiers.

RESULTS

```
Enter the number of classes that you want to consider: 3

Enter the 1 class you want to consider: 1

Enter the 2 class you want to consider: 2

Enter the 3 class you want to consider: 3

Enter the number of training elements: 5

Do you want to perform prediction or cross validation: 1
[0.66666667 1.         0.66666667 1.         1.        ]
[0.66666667 1.         1.         1.         1.        ]
[0.         0.72225539 0.         0.96616673 0.        ]
[0.33333333 0.66666667 0.33333333 0.66666667 0.33333333]
[[0.8666666666666666, 0.9333333333333332, 0.4666666666666667, 0.3376844243754974]]
```



TASK 3

On handwritten letter data, fix on 10 classes. Use the data handler to generate training and te0st data files. Doing this for seven different splits: (train=5 test=34), (train=10 test=29), (train=15 test=24) , (train=20 test=19), (train=25 test=24) , (train=30 test=9) , (train=35 test=4).
On these seven different cases, ran the centroid classifier to compute average test image classification accuracy and plotted these 7-average accuracies on one curve.

```
Enter the number of classes that you want to consider: 10
Enter the elements in order "Write in Uppercase only":

A

S

D

F

G

H

J

K

L

P

Enter the number of times you want to repeat the split: 7

Enter the number of training elements: 5
The number of testing elements is:  34

Enter the number of training elements: 10
The number of testing elements is:  29

Enter the number of training elements: 15
```

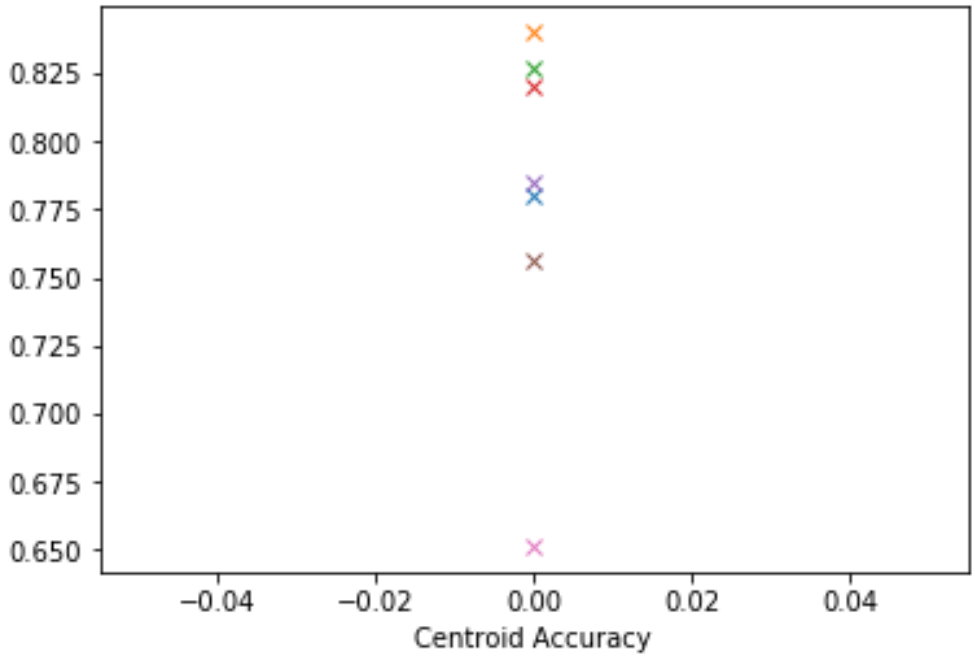
The number of testing elements is: 24

Enter the number of training elements: 20
The number of testing elements is: 19

Enter the number of training elements: 25
The number of testing elements is: 14

Enter the number of training elements: 30
The number of testing elements is: 9

Enter the number of training elements: 35
The number of testing elements is: 4
[0.78, 0.84, 0.8266666666666667, 0.82, 0.7848605577689243, 0.7566666666666667, 0.6514285714285715]
0.7799460660758327

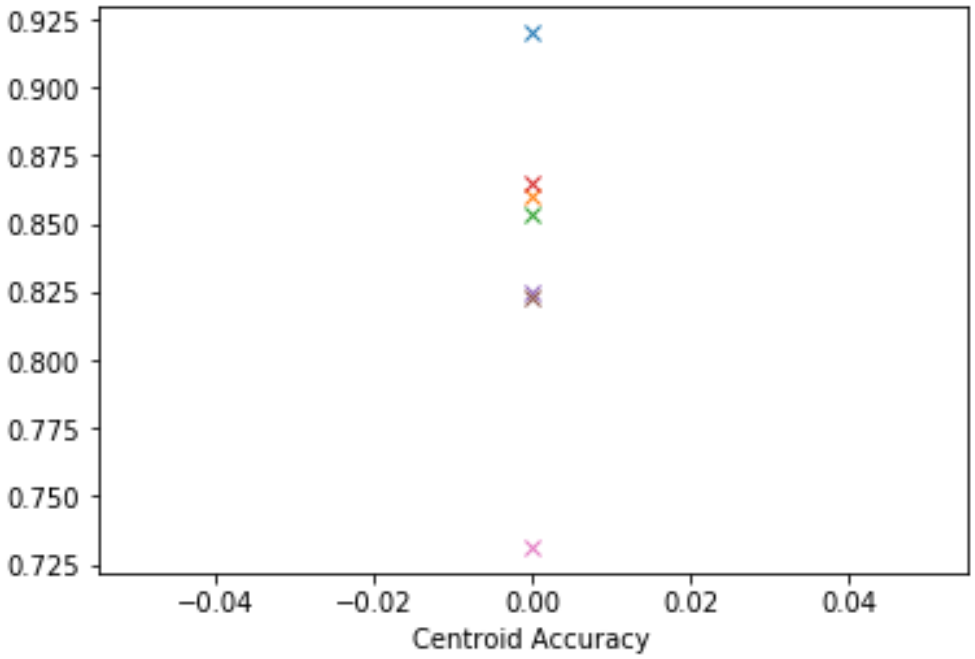


TASK 4

Task 4 is like TASK C but the only difference here is the classes that are being taken into consideration are different than that of the previous task.

```
Enter the number of classes that you want to consider: 10
Enter the elements in order "Write in Uppercase only":

Z

X

C

V

B

N

M

Q

W

E

Enter the number of times you want to repeat the split: 7

Enter the number of training elements: 5
The number of testing elements is:  34

Enter the number of training elements: 10
The number of testing elements is:  29

Enter the number of training elements: 15
The number of testing elements is:  24

Enter the number of training elements: 20
The number of testing elements is:  19

Enter the number of training elements: 25
The number of testing elements is:  14
```

```
Enter the number of training elements: 30
The number of testing elements is:  9

Enter the number of training elements: 35
The number of testing elements is:  4
[0.92, 0.86, 0.8533333333333334, 0.865, 0.8247011952191236,
0.8233333333333334, 0.7314285714285714]
0.8396852047591945
```



TASK 5

In task 5 four subroutines are used for handling data.

Subroutine1:
that picks a small portion of data from the large dataset by selecting only the given class labels and saving in a .csv file. This is done by pickDataClass(filename, class_ids), where class_ids is the classes to be picked.

Subroutine2:
splitData2TestTrain(filename, number_per_class, test_instances)
can generate a training data and test data from the ATNT or hand-writtenletter data. Then the data is split into TrainX, TestX, TrainY
and Test Y and stored in .csv files separately which is itself **subroutine 3.**

Subroutine 4:
“letters_2_digit_convert" that converts a character string to an
integer array. For example, letter_2_digit_convert('ACFG') returns array (1, 3, 6, 7).
Importantly assigned choice=0 for prediction and choice=1 for 5 fold cross validation.

REFERENCES:

1. https://machinelearningmastery.com/tutorial-to-implement-k-nearest-neighbors-in-python-from-scratch/
2. https://matplotlib.org/tutorials/index.html
3. http://scikit-learn.org/stable/modules/clustering.html#clustering