**Project 1: Classifier code files - description and output**

**--------------------------------------------------------------------------------------------------**

**Steps to run:**

* From the Project1\_kxk5334.zip file, extract all files into the same directory as the ATNTFaceImages400.txt and HandWrittenLetters.txt files.
* Run the program to view the accuracy of the predictions.

**Files and output:**

1. knn\_faceimages.py performs kNN algorithm(k=5) on the ATNTFaceImages400.txt file.

Output:

Accuracy is : 86.25

Accuracy is : 92.5

Accuracy is : 92.5

Accuracy is : 82.5

Accuracy is : 85.0

1. knn\_handwritten.py performs kNN algorithm(k=5) on the HandWrittenLetters.txt file.

Output:

Accuracy is : 74.3842364532

Accuracy is : 79.802955665

Accuracy is : 69.9507389163

Accuracy is : 76.8472906404

Accuracy is : 75.7425742574

1. centroid\_faceimages.py performs Centroid algorithm on the ATNTFaceImages400.txt file.

Output:

Accuracy is: 95.0

Accuracy is: 97.5

Accuracy is: 95.0

Accuracy is: 86.25

Accuracy is: 91.25

1. centroid\_handwritten.py performs Centroid algorithm on the HandWrittenLetters.txt file.

Output:

Accuracy is: 68.4729064039

Accuracy is: 71.4285714286

Accuracy is: 72.4137931034

Accuracy is: 71.921182266

Accuracy is: 74.7524752475

1. svm\_faceimages.py performs SVM algorithm on the ATNTFaceImages400.txt file.

Output:

Accuracy is: 99.8102954553

Accuracy is: 93.270070154

Accuracy is: 83.9167890909

Accuracy is: 97.4590648188

Accuracy is: 87.2757038676

1. svm\_handwritten.py performs SVM algorithm on the HandWrittenLetters.txt file.

Output:

Accuracy is: 79.3103448276

Accuracy is: 73.8916256158

Accuracy is: 76.8472906404

Accuracy is: 77.3399014778

Accuracy is: 78.2178217822

1. TaskC- ATNTFaceImages400.py splits input file(ATNTFaceImages400.txt) to train and test data

Output:

y\_train: [[ 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3

3 3 3 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5 5 5 5 6 6 6

6 6 6 6 6 6 7 7 7 7 7 7 7 7 7 8 8 8 8 8 8 8 8 8

9 9 9 9 9 9 9 9 9 10 10 10 10 10 10 10 10 10 11 11 11 11 11 11

11 11 11 12 12 12 12 12 12 12 12 12 13 13 13 13 13 13 13 13 13 14 14 14

14 14 14 14 14 14 15 15 15 15 15 15 15 15 15 16 16 16 16 16 16 16 16 16

17 17 17 17 17 17 17 17 17 18 18 18 18 18 18 18 18 18 19 19 19 19 19 19

19 19 19 20 20 20 20 20 20 20 20 20 21 21 21 21 21 21 21 21 21 22 22 22

22 22 22 22 22 22 23 23 23 23 23 23 23 23 23 24 24 24 24 24 24 24 24 24

25 25 25 25 25 25 25 25 25 26 26 26 26 26 26 26 26 26 27 27 27 27 27 27

27 27 27 28 28 28 28 28 28 28 28 28 29 29 29 29 29 29 29 29 29 30 30 30

30 30 30 30 30 30 31 31 31 31 31 31 31 31 31 32 32 32 32 32 32 32 32 32

33 33 33 33 33 33 33 33 33 34 34 34 34 34 34 34 34 34 35 35 35 35 35 35

35 35 35 36 36 36 36 36 36 36 36 36 37 37 37 37 37 37 37 37 37 38 38 38

38 38 38 38 38 38 39 39 39 39 39 39 39 39 39 40 40 40 40 40 40 40 40 40]]

X\_train: [[ 47 64 50 ..., 87 88 89]

[ 47 65 58 ..., 86 88 89]

[ 49 69 55 ..., 87 88 90]

...,

[ 45 36 37 ..., 54 46 50]

[ 46 35 33 ..., 76 74 52]

[ 46 34 30 ..., 119 126 76]]

y\_test: [ 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

26 27 28 29 30 31 32 33 34 35 36 37 38 39 40]

X\_test: [33 34 28 ..., 41 40 38]

1. TaskC- HandWrittenLetters.py splits input file(HandWrittenLetters.txt) to train and test data and performs a 2-class svm classifier

Output:

X\_train: [[0 0 0 ..., 0 1 1]

[0 0 0 ..., 1 1 1]

[0 0 0 ..., 0 0 0]

...,

[0 0 0 ..., 0 0 0]

[0 0 0 ..., 0 0 0]

[0 1 1 ..., 0 0 0]]

y\_train: [3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 6 6 6 6 6 6 6

6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6]

X\_test: [[0 0 0 ..., 0 0 0]

[0 0 0 ..., 0 0 0]

[0 0 0 ..., 0 0 1]

...,

[1 0 1 ..., 0 0 0]

[1 0 1 ..., 0 0 0]

[1 0 1 ..., 0 0 0]]

y\_test: [[3 3 3 3 3 3 3 3 3 6 6 6 6 6 6 6 6 6]]

Accuracy is: 100.0