Project 1
Classification using KNN, Centroid method, Linear Regression and SVM

Data Mining(CSE-5334)
Chris H Q Ding
University of Texas, Arlington

Jenil Desai
Jinesh Modi

Task A

We selected classes "A, B, C, D, E" from the hand-written-letter data. From this smaller dataset, we generate a training and test data: for each class using the first 30 images for training and the remaining 9 images for test. Further, classification was done on the generated data using the four classifiers – Centroid, KNN, Linear Regression, SVM.

For Centroid Classifier-

Predictions on Test set:

```
[ 1.  1.  1.  1.  1.  1.  1.  2.  1.  5.  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.  3.  5.  3.  5.]
```

Accuracy of Test set: 0.91

For KNN Classifier-

Predictions on Test set:

```
[ 1.  1.  1.  1.  1.  1.  1.  2.  1.  5.  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.  3.  5.  3.  5.]
```

Accuracy of Test set: 0.91

For Linear Regression-

Predictions on Test set:

```
[-0.  1.  2.  2.  3.  3.  2.  1.  3.  3.  4.  2.  4.  2.  3.  2.  1.  2.  2.  2.  4.  3.  4.  4.  2.  4.  4.  4.  4.  5.  5.  4.  4.  5.  4.
5.  6.  5.  6.  6.  5.  4.  7.  3.  5.]
```

Accuracy of Test set: -97.77777777777777

For SVM Classifier-

Predictions on Test set:

```
[ 1.  1.  1.  1.  1.  1.  1.  2.  1.  2.  2.  2.  1.  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.  3.  5.  3.  5.]
```

Accuracy of Test set: 0.91

TREND: We observe that all the three accuracies appear to be similar except linear regression, this may appear due to a very small dataset.

Task B

On ATNT data, we ran 5-fold cross-validation (CV)using all the four classifiers: KNN, Centroid, Linear regression and SVM. Each of the 5-Fold CV gave one accuracy, below I presented all the 5 accuracies and the average of them.

```
KNN Accuracy with 5-Fold:  0.925
KNN Accuracy with 5-Fold:  0.8875
KNN Accuracy with 5-Fold:  0.9125
KNN Accuracy with 5-Fold:  0.9
KNN Accuracy with 5-Fold:  0.9125
Average accuracy of KNN with 5-fold:  0.9075

Centroid Accuracy with 5-Fold:  0.9125
Centroid Accuracy with 5-Fold:  0.95
Centroid Accuracy with 5-Fold:  0.9125
Centroid Accuracy with 5-Fold:  0.925
Centroid Accuracy with 5-Fold:  0.9125
Average accuracy of centroid with 5-fold:  0.9225

Linear Regression Accuracy with 5-Fold:  -98.75
Linear Regression Accuracy with 5-Fold:  -98.75
Linear Regression Accuracy with 5-Fold:  -98.75
Linear Regression Accuracy with 5-Fold:  -98.75
Linear Regression Accuracy with 5-Fold:  -98.75
Average accuracy of Linear Regression with 5-fold:  -19.75

SVM Accuracy with 5-Fold:  0.751479289941
SVM Accuracy with 5-Fold:  0.753694581281
SVM Accuracy with 5-Fold:  0.743842364532
SVM Accuracy with 5-Fold:  0.79802955665
SVM Accuracy with 5-Fold:  0.772277227723
Average accuracy of SVM with 5-fold:  0.761351997269
```

Task C

On handwritten letter data, I have used method pickData()routine to generate training and test data files.
This input is given for different splits and for each split we get an accuracy.
I have plotted all these 7 accuracies on a graph which is attached below.

Fixed classes or input 'abcdefghij' which will be converted to [1,2,3,4,5,6,7,8,9,10] from letter_2_digit_convert (input) routine.

Input = 'abcdefghij'

1)Split (train=5 test=34)
Accuracy: 0.738235294118

2)Split (train=10 test=29)
Accuracy: 0.824137931034

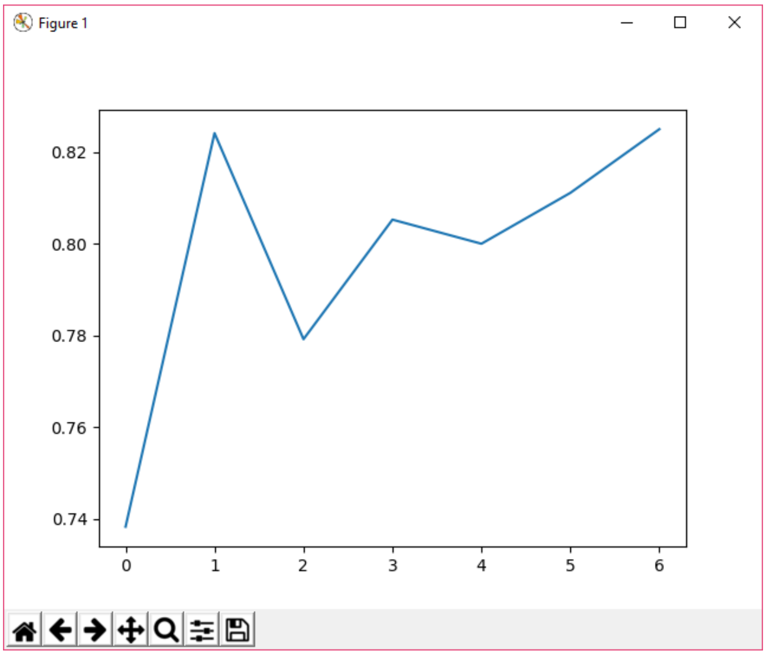
3)Split (train=15 test=24)
Accuracy: 0.779166666667

4)Split (train=20 test=19)
Accuracy: 0.805263157895

5)Split (train=25 test=14)
Accuracy: 0.8

6)Split (train=30 test=9)
Accuracy: 0.811111111111

7)Split (train=35 test=4)
Accuracy   0.825



TREND:

a) Looking at the graph we can say that accuracy is getting increased abruptly for the Split(train=5 test=34) and then decreased for the next split.
b) Alternatively increase-decrease graph is getting plotted for increasing size of training datasets.

Task D

On handwritten letter data, I have used method pickData()routine to generate training and test data files.
This input is given for different splits and for each split we get an accuracy.
I have plotted all these 7 accuracies on a graph which is attached below.

Fixed classes or input 'klmnopqrst' which will be converted to [11,12,13,14,15,16,17,18,19,20] from letter_2_digit_convert (input) routine.

Input = 'klmnopqrst'

1)Split (train=5 test=34)
Accuracy: 0.75

2)Split (train=10 test=29)
Accuracy: 0.765517241379

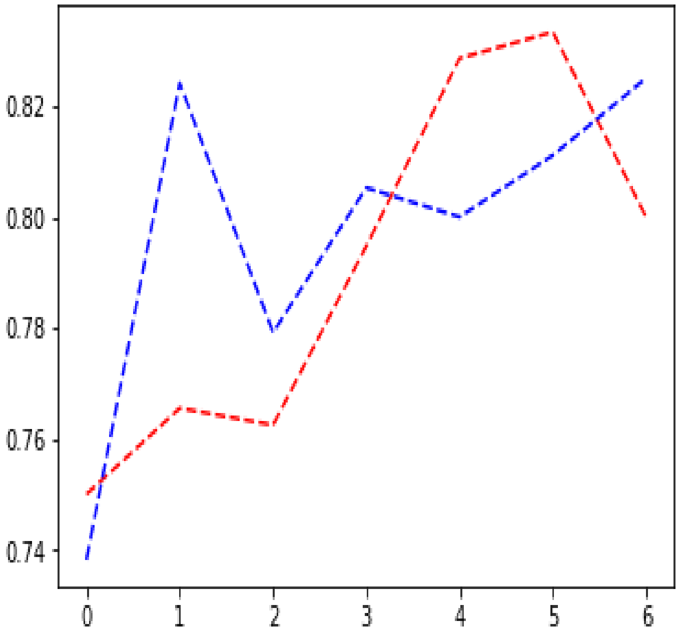
3)Split (train=15 test=24)
Accuracy: 0.7625

4)Split (train=20 test=19)
Accuracy: 0.794736842105

5)Split (train=25 test=14)
Accuracy: 0.828571428571

6)Split (train=30 test=9)
Accuracy: 0.833333333333

7)Split (train=35 test=4)
Accuracy   0.8



<u>Trend :</u>

a) The red graph here denotes data for the other 10 classes. As looking at the graph we can that when we are increasing the training dataset, the accuracy is also getting increased.

b) But for Split(train=35 test=4), the accuracy dipped for this particular input”klmnopqrst”.

c) We are getting different trend for different set of input. From A to J we are getting increase-decrease accuracy but from K to T, we are getting increased accuracy for the increase size of training dataset, the only exception is at the end of the input string.