DATA MINING: PROJECT 1- REPORT

NAME: VRUSHALI KADAM ID: 1001514762

AIM: The purpose of this project is to study the detailed working of 4 different data classifiers – KNN, Centroid, SVM, and Linear Regression. This report includes detailed accuracy measures of the classifiers used for different Test, Train data sets and K fold cross validations on them.

The classification program includes the data handler code as well. Upon execution, the programs ask the user input for the specific task to be implemented (Task A/B/C/D) and outputs accordingly.

The Project is done using Python (3.6) using numpy, sys, matplotlib, sklearn libraries.

Following is the detailed description of each task:

A. Use the data-handler to select "A,B,C,D,E" classes from the hand-written-letter data. From this smaller dataset, Generate a training and test data: for each class, using the first 30 images for training and the remaining 9 images for test. Do classification on the generated data using the four classifiers.

- Used letter_to_digit_convert to get class id of string ”ABCDE” 1,2,3,4,5.
- SplitData2TestTrain returns [testVector, testLabel, trainVector, trainLabel]for 0-29 as training and 30-38 as test.
- Accuracy of the Label of testVector is predicted using SVM, Centroid, 5NN, and Linear regression.
- Task A outputs the classification accuracy of all the four classifiers.

OUTPUT:

```
(base) Mohanas-MacBook-Pro:~ mohana$ python data_mining_p1.py
Please enter which task to be performed(Task A, Task B, Task C or Task D):A
        TASK A :

                                Use the data-handler to select "A,B,C,D,E" classes from the hand-written-letter data.
                                From this smaller dataset, Generate a training and test data: for each class.
                                using the first 30 images for training and the remaining 9 images for test.
                                Do classification on the generated data using the four classifers.


Accuracy of SVM is 91.11

Accuracy of Centroid is 86.67

Accuracy of Linear is 88.89

Accuracy of 5-NN is 86.67

(base) Mohanas-MacBook-Pro:~ mohana$
```

DATA MINING: PROJECT 1- REPORT

B. On ATNT data, run 5-fold cross-validation (CV) using each of the four classifiers: KNN, centroid, Linear Regression and SVM. If you don't know how to partition the data for CV, you can use the data-handler to do that. Report the classification accuracy on each classifier. Remember, each of the 5-fold CV gives one accuracy. You need to present all 5 accuracy numbers for each classifier. Also, the average of these 5 accuracy numbers.

• Divide complete date set into 5 sets for 5 fold cross validation.
• Use splitData2TestTrain to get these data sets in following order: – Fold 1 : Test(0,1) Train ({0,1,2,3,4,5,6,7,8,9} - {0,1})
– Fold 2 : Test(2,3) Train({0,1,2,3,4,5,6,7,8,9} - {2,3})
– Fold 3 : Test(4,5) Train({0,1,2,3,4,5,6,7,8,9} - {4,5})
– Fold 4 : Test(6,7) Train({0,1,2,3,4,5,6,7,8,9} - {5,7})
– Fold 5 : Test(8,9) Train ({0,1,2,3,4,5,6,7,8,9} - {8,9})
• The Accuracy of the Labels of testVector is predicted using SVM, Centroid, 5NN, and Linear regression for each fold.
• Output of the accuracy average, with individual accuracy is printed.

OUTPUT:

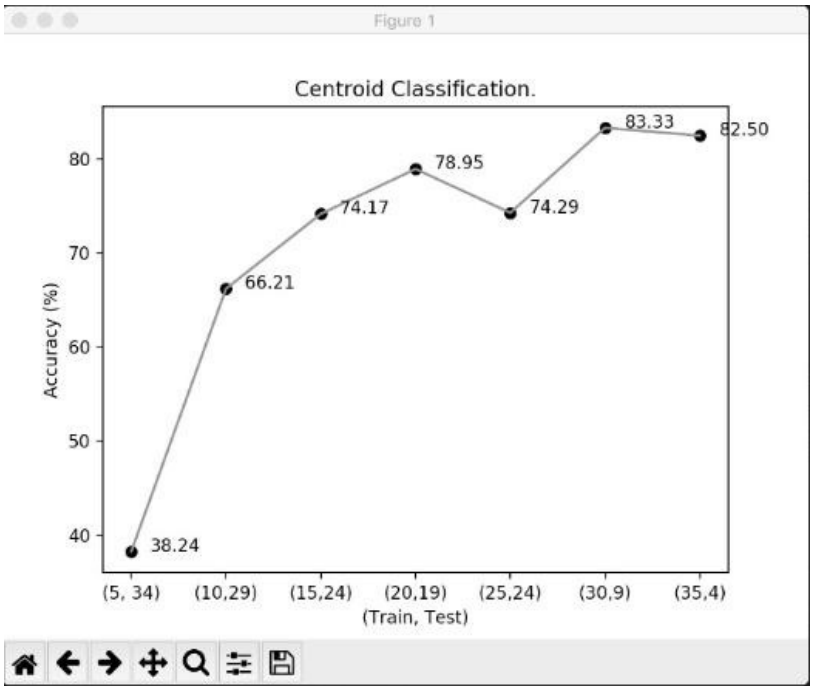
```
(base) Mohanas-MacBook-Pro:~ mohana$ python data_mining_p1.py
Please enter which task to be performed(Task A, Task B, Task C or Task D):B
        TASK B :

                            On ATNT data, run 5-fold cross-validation (CV) using  each of the
                            four classifiers: KNN, centroid, Linear Regression and SVM.
                            If you don't know how to partition the data for CV, you can use the data-handler to do that.
                            Report the classification accuracy on each classifier.
                            Remember, each of the 5-fold CV gives one accuracy. You need to present all 5 accuracy numbers
                            for each classifier. Also, the average of these 5 accuracy numbers.
Calculating..................................................

Average accuracy of 5-NN after 5-Fold is 92.00
[91.25, 90.0, 92.5, 93.75, 92.5]

Average accuracy of Centroid after 5-Fold is 92.50
[93.75, 95.0, 93.75, 92.5, 87.5]

Average accuracy of SVM after 5-Fold is 69.75
[75.0, 78.75, 71.25, 58.75, 65.0]

Average accuracy of Linear after 5-Fold is 91.25
[96.25, 91.25, 96.25, 85.0, 87.5]
(base) Mohanas-MacBook-Pro:~ mohana$
```

DATA MINING: PROJECT 1- REPORT

C. On handwritten letter data, fix on 10 classes. Use the data handler to generate training and test data files. Do 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, run the centroid classifier to compute average test image classification accuracy. Plot these 7 average accuracy on one curve in a figure. What trend can you observe? When do this task, the training data and test data do not need be written into files.

- letter_to_digit_convert returns the class ids for letters 'ABCDEFXYZG' 10 alphabets.
- splitData2TestTrain returns the [testVector, testLabel, trainVector, trainLabel]for following sets:
  – Set 1 : Test(5:38) Train (0:4)
  – Set 2 : Test(10:38) Train (0:9)
  – Set 3 : Test(15:38) Train(0:14)
  – Set 4 : Test(20:38) Train(0:19)
  – Set 5 : Test(25:38) Train (0:24)
  – Set 6 : Test(30:38) Train (0:29)
  – Set 7 : Test(35:38) Train (0:34)
- Accuracy of the Labels of testVector is predicted using Centroid method for each set.
- Output of the individual accuracy of each case is plotted on graph.

OUTPUT:

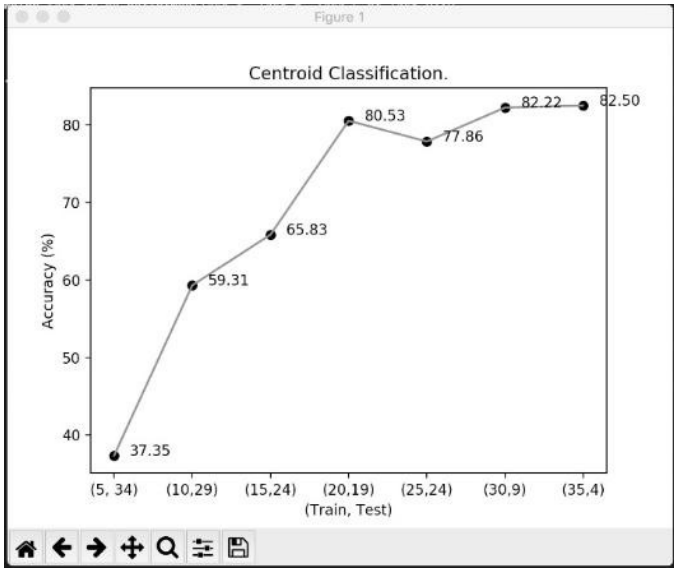


DATA MINING: PROJECT 1- REPORT

D. Repeat task (D) for another different 10 classes. You get another 7 average accuracy. Plot them on one curve in the same figure as in task (D). Do you see some trend?

- letter_to_digit_convert returns the class ids for letters “MNOPQRSTUV” 10 alphabets.
- splitData2TestTrain returns the [testVector, testLabel, trainVector, trainLabel]for following sets:
  – Set 1 : Test(5:38) Train (0:4)
  – Set 2 : Test(10:38) Train (0:9)
  – Set 3 : Test(15:38) Train(0:14)
  – Set 4 : Test(20:38) Train(0:19)
  – Set 5 : Test(25:38) Train (0:24)
  – Set 6 : Test(30:38) Train (0:29)
  – Set 7 : Test(35:38) Train (0:34)
- Accuracy of the Labels of testVector is predicted using Centroid method for each set.
- Output of the individual accuracy of each case is plotted on graph.

OUTPUT:



DATA MINING: PROJECT 1- REPORT

E. Write data handler.

(a)Subroutine1: pickDataClass(filename, class_ids)

Splits the data in the file passed as an argument, based on the class ids given by the letter_to_digit_convert function. It stores the output into TEMP_FILE_NAME.

It uses the numpy‘s genfromtxt function to get the data from the file and identifies the unique classes and their locations in the file. Stores the starting location of each class in an array.

```
35  def pickDataClass(filename, class_ids):
36      data = np.genfromtxt(filename, delimiter=',')
37      #print(data)
38      list_ClassifierCol = []
39      for i in class_ids:
40          a= np.where(data[0] == i)  # returns index locations of the perticular class
41          #print(a)
42          list_ClassifierCol.extend(np.array(a).tolist())  # appending columns into a string
43          #print(listOfClassifierColumn)
44      list_ClassifierCol = [item for sublist in list_ClassifierCol for item in sublist]  # forming a array
45      np.savetxt(TEMP_FILE_NAME, data[:, list_ClassifierCol], fmt="%i", delimiter=',')
46      #fh = open(TEMP_FILE_NAME,"r")
47      #print (fh.read())
```

(b)Subroutine2: splitData2TestTrain(filename, number_per_class, test_instances)

The splitData2TestTrain method takes the filename, the number of instances per class label and the test instances as parameters. The test_instances is a ratio that is received which is then split using the “:” delimiter to split the instances as training and testing. The method returns 4 data sets namely; test labels, test data, train labels and train data.

```
58  def splitData2TestTrain(filename, number_per_class, test_instances):
59      start, end = test_instances.split(":")
60      listTest = list(range(int(start), int(end) + 1))
61      listTrain = list((set(list(range(0, number_per_class))) - set(listTest)))
62      Training = []
63      Test = []
64      data = np.genfromtxt(filename, delimiter=',')
65      #print("x val",data[1].size)
66      for i in range(0, data[0].size, number_per_class):
67          templistTest = [x + i for x in listTest]
68          templistTrain = [x + i for x in listTrain]
69          templistTest.sort()
70          templistTrain.sort()
71          if len(Test) == 0:
72              Test = data[:, templistTest]
73          else:
74              Test = np.concatenate((Test, data[:, templistTest]), axis=1)
75          if len(Training) == 0:
76              Training = data[:, templistTrain]
77          else:
78              Training = np.concatenate((Training, data[:, templistTrain]), axis=1)
79      return Test[1:, ], Test[0], Training[1:, ], Training[0]
80
```

DATA MINING: PROJECT 1- REPORT

(c)Subroutine3: store(trainX,trainY)
This function accepts the training data and its class labels in the form of the numpy array. The labels and data are stacked together and then put in the file filename.

```
87  def store(trainX, trainY, fileName):
88      np.savetxt(fileName, np.vstack((trainY, trainX)), fmt="%i", delimiter=',')
89
```

(d)Subroutine4: letter_to_digit_convert(mystr)
Takes a string in Argument and converts this string into corresponding integers using ACSII values.
Formula: ASCII(<LETTER>) – 64 Eg.64=ASCIIVALUE('A')-1

```
19  def letter_to_digit_convert(mystr):
20      m_list = []
21      mystr = mystr.upper()
22      for i in mystr:
23          if i.isalpha():
24              m_list.append(ord(i) - 64)
25      return m_list
26
```

REFERENCES:

(1) https://www.youtube.com/watch?v=ooQtUaCExa8
(2) https://www.youtube.com/watch?v=iybATqk6LNI
(3) https://www.youtube.com/watch?v=V59bYfIomVk
(4) https://www.youtube.com/watch?v=0bMik28a5IU
(5) http://scikit-learn.org/stable/modules/svm.html
(6) Introduction to Data Mining by PANG-NING TAN Michigan State University MICHAEL STEINBACH University of Minnesota VIPIN KUMAR University of Minnesota