Handwritten Digit Recognition

OVERVIEW

In this assignment, you will apply Machine Learning skills. You are asked to design and develop one of its popular classification algorithm called kNN (k-Nearest Neighbors) to programmatically predict handwritten digits in its digitized format. kNN is supervised based algorithm by majority vote of its neighbors. It is supervised in a sense that a sufficient large sample dataset of known target values is carefully selected. This sample dataset will be split into 2 groups. One group will be used to formula a kNN model to be used for prediction. It's called Training. The other group is used to verify the accuracy of the established model, called Testing. The Training-Testing cycle might result in several iterations of tuning and adjustment until a satisfactory model is achieved with high accuracy of positive predictions, higher better. Once such a model has been implemented, it's ready to make prediction. To make a prediction, simply feed an unknown value of same format to the program and then perform the required computation in accordance to the established kNN model.

SCOPE

Note: refer to the sample output for references.

1. Construct 2 kNN models based on 2 files called digit-training.txt & digit-testing.txt for training & testing respectively (cross-validation is not required), files are text based.
2. Refer to the lab materials for more information on kNN implementation, file format and ideas.
3. Use ONLY vector-based function developed from previous lab class to determine nearest neighbors
    a. Download the latest version of vector.py from moodle
4. Implement majority vote algorithms to do the best guess prediction
5. IN THE DESIGN DOC, FOR EACH kNN model: explain (a) how the closest neighbors are chosen and (b) the rule(s) used in making the prediction
6. Based on each kNN model implementation:
    a. show training and testing info (see Output sections)
    b. show prediction outcome using file digit-predict.txt (see Output sections).
7. Note for output:
    a. Group each set of outputs (training, testing & prediction) in accordance to your kNN models implemented
8. Files to download from moodle:
    a. digit-training.txt, digit-testing.txt, digit-predict.txt & vector.py

YOU ARE ENCOURAGED TO BUILD A MORE EFFICIENT kNN MODEL AS SUGGESTED IN CLASS!!!

SKILLS

In this assignment, you will be trained on the use of the followings:

- Machine Learning life cycle – dataset, data mining, kNN construction, training & testing
- Python objects & modules (file IO, string, string formatting, sorting, dictionary, list, list comprehensions)
- Controls – if, while, for to control program flow
- Variable Scope
- Functions to breakdown the logic

DELIVERABLES

1. Design documentation (A3_School_StudentID_Design.doc)
2. Program source code (A3_School_StudentID_Source.py)
3. Output (A3_School_StudentID_Output.txt)

Zip all files above in a single file (A3_School_StudentID.zip) and submit the zip file by due date to the corresponding assignment folder under "Assignment (submission)", School is 'SSE', 'SME', or 'HSS'.

OUTPUT X 2 SETS

1. Training Info (see sample output)
2. Testing Info (see sample output)
3. Prediction Outcome (see sample output)

DUE DATE

May 12th, 2017

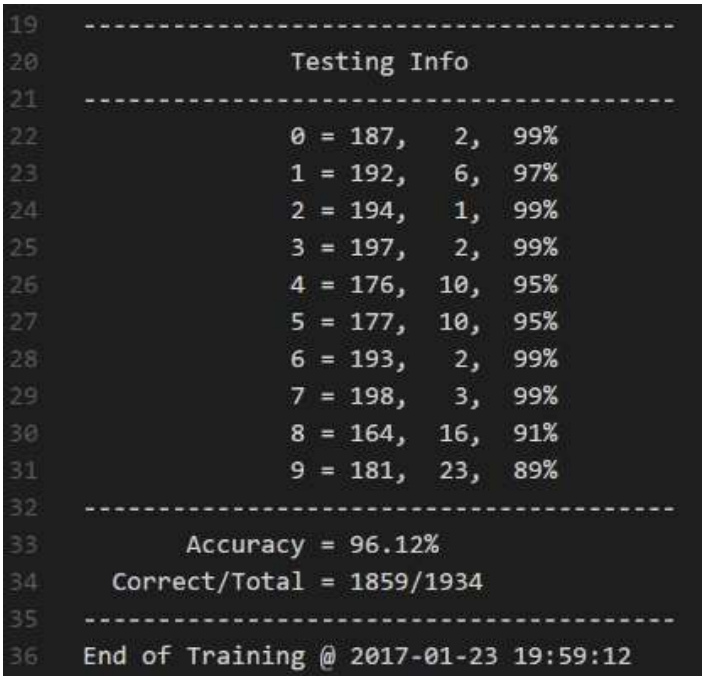
TIPS & HINTS

- Use Dictionary to keep list of digit-vectors (during training) and to track accuracy rate (during testing)
- Use Counter() and most_common() from module “collections” to return the closest neighbors
- Use zip + list comprehension for vector sum, subtract, sum, or, average, sum and so on
- Use String Formatting for training and testing info
- Use reduce() from module “functools” to combine multiple vectors into a single OR-vector or AND-vector

SAMPLE OUTPUT – TRAINING X 2

```
Beginning of Training @ 2017-01-23 19:54:44
---------------------------------------
             Training Info
---------------------------------------
              0 = 100
              1 =  94
              2 =  93
              3 = 105
              4 =  87
              5 =  81
              6 =  95
              7 =  90
              8 = 109
              9 =  89
---------------------------------------
  Total Samples = 943
---------------------------------------
```

SAMPLE OUTPUT – TESTING X 2



```
----------------------------------------
              Testing Info
----------------------------------------
              0 = 187,   2,  99%
              1 = 192,   6,  97%
              2 = 194,   1,  99%
              3 = 197,   2,  99%
              4 = 176,  10,  95%
              5 = 177,  10,  95%
              6 = 193,   2,  99%
              7 = 198,   3,  99%
              8 = 164,  16,  91%
              9 = 181,  23,  89%
----------------------------------------
       Accuracy = 96.12%
  Correct/Total = 1859/1934
----------------------------------------
End of Training @ 2017-01-23 19:59:12
```

SAMPLE OUTPUT – PREDICTION X 2

Simply output the predicted value, one number per line, such as:

9
8
6
6
8
9
6

MARKING CRETERIA

Scope + Deliverables = 80%

Program style & structure = 20%