Name: Shahid Mohammed Shaikbepari

USC Id: 4491455475

Assignment: Lab 7, EE595

PART A

----------

Steps to Build and Run:

------------------------

1. First run the commands to build an virtual environment as given in the spec of Lab6
2. **1. Set up a Python 3 virtual environment**
3. Please log in the StudentVM as CS103 Student User to work on this Lab. Some packages are pre-installed but might be outdated, please use the provided *requirements.txt* to update the packages, by entering the following command in your terminal:
4. $ sudo pip install virtualenv # Install if you didn't install it
5. $ virtualenv -p python3 .env # create a virtual environment
6. $ source .env/bin/activate # activate the virtual environment
7. $ pip install -r requirements.txt # install necessary packages inside of the virtual environment
8. # Work on your lab...
9. $ deactivate

2. To run: python knn\_4491455475.py 5 100 900 cifar-10-batches-py/data\_batch\_1

Algorithm description:

----------------------

1. Read the K D N Path values from sys

2. Divide the data into train and test data

3. Convert the images into grayscale with given formula in the description

4. Perform preprocessing on the data from the sklearn PCA module

5. Finally in the function doKNN, calculate the distance of the test point from each training point and note down first N values with least distance

6. Perform voting based on the weights = 1/distance and assign the class with max votes

7. Repeat the experiment with sklearn library’s KNeighbors Classifier

8. The outputs are reported in knn\_results.txt and knn\_results\_sklearn.txt files

References:

-------------------

1. upickle() was taken from <https://www.cs.toronto.edu/~kriz/cifar.html>
2. sklearn’s PCA was taken from https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html
3. lines 60 – 64 were taken from https://stackoverflow.com/questions/7971618/python-return-first-n-keyvalue-pairs-from-dict
4. lines 84-89 were extracted from https://www.geeksforgeeks.org/python-dictionary/
5. sklearn KNeighborsClassifier is taken from https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsRegressor.html

PART B

----------

Steps to Build and Run:

------------------------

1. First run the commands to build an virtual environment as described in the part A

2. To run: python naive\_bayes\_4491455475.py

Algorithm description:

----------------------

1. extractData() method extracts the data from the .txt file
2. time.process\_time() gives the current time
3. predict() function calculates the mean and variance for each class for each feature
4. calculate the logSum of total conditional probability and return the predicted value
5. Finally calculate the accuracy
6. Repeated the with sklearn’s naïve\_bayes’ Gaussian model and SKLearn’s SVM model

Generated Results:

-------------------------

My Naive Bayes:

Training acc: 83.33% Training time: 0.0209 s

Testing acc: 88.10% Testing time: 0.0051 s

Sklearn Naive Bayes:

Training acc: 89.88% Training time: 0.4065 s

Testing acc: 95.24% Testing time: 0.0010 s

Sklearn SVM:

Training acc: 92.26% Training time: 0.0594 s

Testing acc: 92.86% Testing time: 0.0063 s

Comment on results:

---------------------------

In My Naïve Bayes, the data was shuffled using the function random.shuffle so everytime different results are shown up but the average is above ~85%

References:

----------------

1. line 10 was extracted from <https://docs.scipy.org/doc/numpy/reference/generated/numpy.fromstring.html>
2. np.mean, np.var were taken from https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.var.html
3. time.process\_time() was taken from https://stackoverflow.com/questions/14452145/how-to-measure-time-taken-between-lines-of-code-in-python
4. lines 117 – 120 were extracted from <https://machinelearningmastery.com/index-slice-reshape-numpy-arrays-machine-learning-python/> and <https://stackoverflow.com/questions/30062429/python-how-to-get-every-first-element-in-2-dimensional-list/30062458>