ENGR 421 - Machine Learning

Mehmet Ustek

HW2 Report

Completeness of the project:
All requirements are done with correct outputs regarding the homework description.

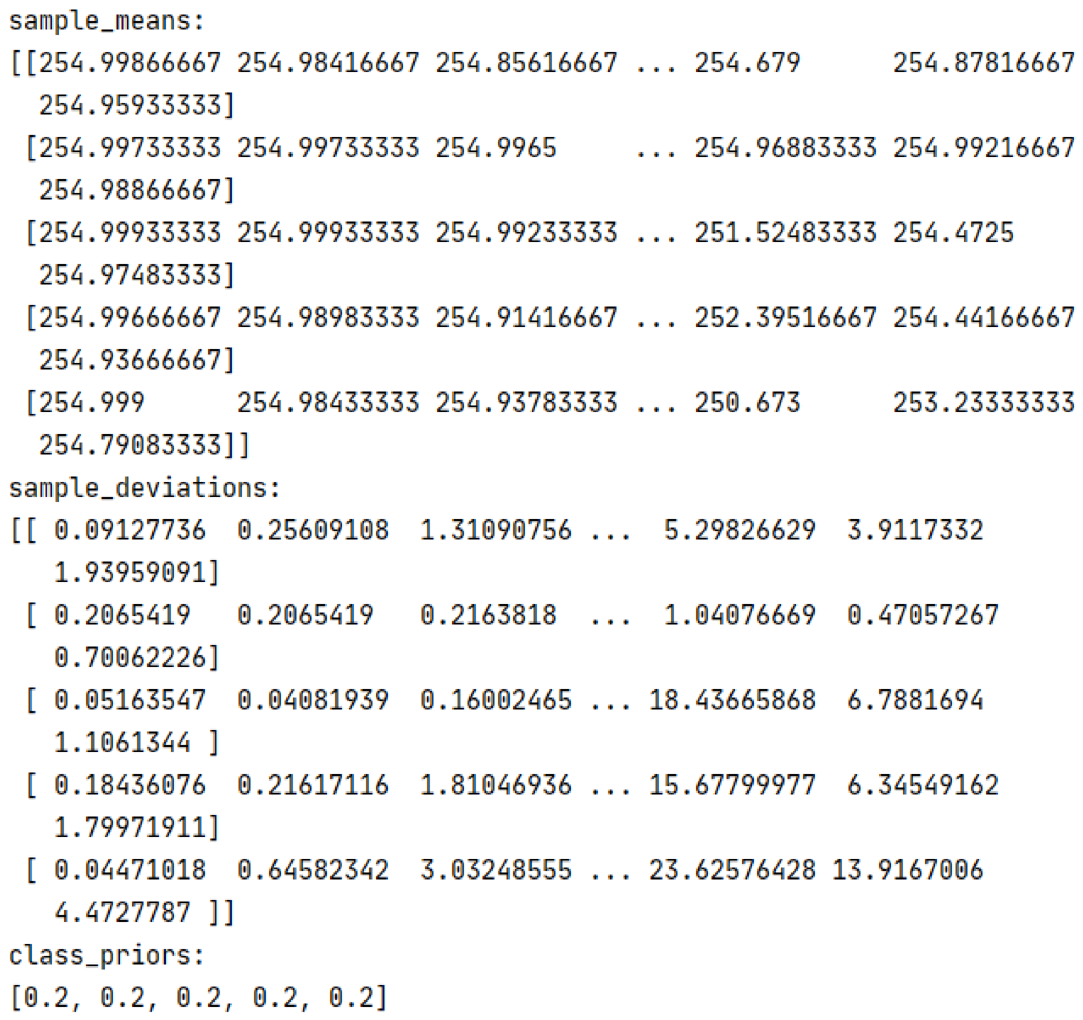
This project was pretty similar to hw01 and lab01.
First, I divided the images and labels into 2 separate sets, train and test, with train containing 30000 records, test containing 5000 records.
Since we have only 5 distinct items to distinguish, I created a K value which is equal to 5.
Then I calculated sample_means, sample_deviations and class_priors values the same way we did in lab01.
I also calculated the score values similar to lab01, with the differences below:

- The lab had data_interval, which was a set of arbitrary numbers, to calculate score values. Instead I used X values which refers to (30000,754) items, with each image identifier to its corresponding image pixels.
- Since the x values shape is (30000,754), everytime I calculate the score values, it should be the sum of all 754 pixels, so that we will have (30000,5) shape. In other words, we will have 30000 images and for every image we will have a prediction of this image being one of the 5 clusters. Thus every image has 5 values that correspond to this image, likely being that t-shirt, bag, etc.

To calculate a y_predicted value, I iterated through the score_values value and got the argmax value from it. Since we have (30000,5) shape, every image has a tendency to be similar to one of the 5 objects. Getting the maximum argument thus takes this maximum likelihood of that object being similar to one of the 5 objects.

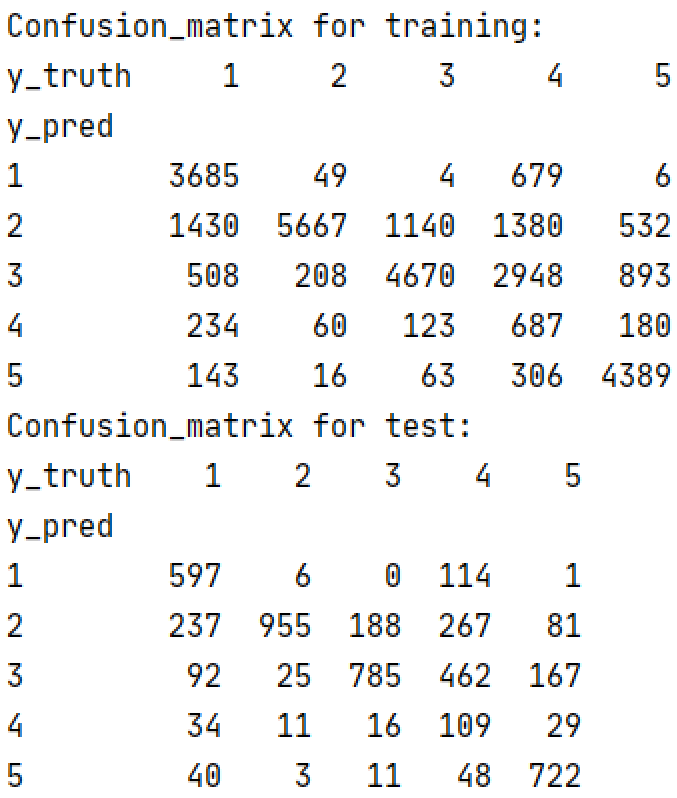
I calculated the confusion matrix two times, one for train and one for test.
For simplicity, I created a function get_score_values(x) that is used for getting score values of that given x. As x is either x_train or x_test.

Screenshots of my outputs:
Sample means, deviations and class priors:



```
sample_means:
[[254.99866667 254.98416667 254.85616667 ... 254.679      254.87816667
  254.95933333]
 [254.99733333 254.99733333 254.9965     ... 254.96883333 254.99216667
  254.98866667]
 [254.99933333 254.99933333 254.99233333 ... 251.52483333 254.4725
  254.97483333]
 [254.99666667 254.98983333 254.91416667 ... 252.39516667 254.44166667
  254.93666667]
 [254.999      254.98433333 254.93783333 ... 250.673      253.23333333
  254.79083333]]
sample_deviations:
[[ 0.09127736  0.25609108  1.31090756 ...  5.29826629  3.9117332
   1.93959091]
 [ 0.2065419   0.2065419   0.2163818  ...  1.04076669  0.47057267
   0.70062226]
 [ 0.05163547  0.04081939  0.16002465 ... 18.43665868  6.7881694
   1.1061344 ]
 [ 0.18436076  0.21617116  1.81046936 ... 15.67799977  6.34549162
   1.79971911]
 [ 0.04471018  0.64582342  3.03248555 ... 23.62576428 13.9167006
   4.4727787 ]]
class_priors:
[0.2, 0.2, 0.2, 0.2, 0.2]
```

Confusion matrix for training and testing:



```
Confusion_matrix for training:
y_truth     1     2     3     4     5
y_pred
1        3685    49     4   679     6
2        1430  5667  1140  1380   532
3         508   208  4670  2948   893
4         234    60   123   687   180
5         143    16    63   306  4389
Confusion_matrix for test:
y_truth    1    2    3    4    5
y_pred
1        597    6    0  114    1
2        237  955  188  267   81
3         92   25  785  462  167
4         34   11   16  109   29
5         40    3   11   48  722
```

Acknowledgements:
I understand the university rules for plagiarism and I have never shared or used any code or slice of code while doing this project. Thus, the effort belongs only to me.
Additionally, I push all my progress to a private github repository, in case need of any proof to display my effort.