ENGR-421 HOMEWORK-2 REPORT

Name-Surname: Barış KAPLAN

Initially, I have imported the necessary libraries. I have imported the necessary libraries as follows:

```
import pandas as pd
import numpy as np
import math
import matplotlib.pyplot as plt
```

For reading the given csv files, I have utilized read_csv function of the pandas library. To get rid of the headers in the csv files, I have assigned the header to None. I have read the csv files as follows:

```
img_file= 'hw02_images.csv'
lbl_file= 'hw02_labels.csv'
imgD= pd.read_csv(img_file, header=None, delimiter=",")
lblD= pd.read_csv(lbl_file, header=None, delimiter=",")
```

For dividing the label data and clothing image data into training set and test set, I have utilized iloc[ ] function. I have divided the clothing image data and label data into training set and test set as follows:

```
clothingImgNum=35000

train_data_of_clothing_images = imgD.iloc[0:30000]
test_data_of_clothing_images = imgD.iloc[30000:clothingImgNum]

train_data_of_clothing_image_labels = lblD.iloc[0:30000]
test_data_of_clothing_image_labels = lblD.iloc[30000:clothingImgNum]
```

For estimating the mean parameters, I have used apply() functions. For each of the 784 pixels in a clothing image, I have estimated the mean parameters. For estimating the standard deviation parameters, I have also used apply() functions. For each of the 784 pixels in a clothing image, I have estimated the standard deviation parameters.

I have repeated this process for all of the clothing image labels (labels= 1 for T-shirt, 2 for Dress, 3 for Coat, 4 for Shirt, and 5 for Bag. Total number of repetitions= 5). I have utilized np.std() function of numpy library to obtain the estimates of the standard deviation parameters for each of the 784 pixels in a clothing image. I have defined and utilized a function called my_custom_avr_func to obtain the estimates of the mean parameters for each of the 784 pixels

in a clothing image. You can find the estimations of the mean parameters in Figure 1, and the standard deviation parameters in Figure 2.

The purpose of the function called my_custom_avr_func is to find the average of a list.

```
The Estimation of the Mean Parameters :

[[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]
 [254.999      254.98433333 254.93783333 ... 250.673      253.23333333
  254.79083333]
 [254.9982     254.991      254.9394     ... 252.84816667 254.40356667
  254.93006667]]

------------------------------------------------------------------------
```

Figure 1: The estimations of the mean parameters for all of the clothing image labels

```
------------------------------------------------------------------------

The Estimation of the Standard Deviation Parameters :

[[ 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 ]]

------------------------------------------------------------------------
```

Figure 2: The estimations of the standard deviation parameters for all of the clothing image labels

For estimating the prior probability parameters, initially, I have calculated the length of the part of the training data set (training data set of the clothing image labels) where the labels are equal to 1. I have repeated this process for the other labels (other labels= 2, 3, 4, and 5). Then, to obtain the prior probability estimations, I have divided these subset lengths with the lengths of the training data set of the clothing image labels. You can find the found estimations of the prior probability parameters in the Figure 3.

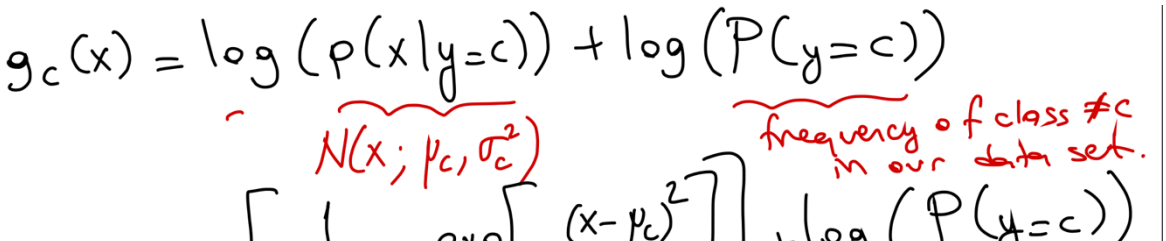
```
------------------------------------------------------------------------

The Estimation of the Prior Probabilities :

[0.2 0.2 0.2 0.2 0.2]
```

Figure 3: The estimations of prior probabilities for all of the clothing image labels

For finding the score values of the classes, I have utilized the formula in Figure 4.



Then, depending on the score values of the classes, I have selected a class and classified the data. By using the apply() function, to obtain the predictions, I have choosed a class for each image in the training set called “train_data_of_clothing_images” and in the test set called “test_data_of_clothing_images”.

By using the pd.crosstab function of the pandas library, I have displayed the confusion matrix for the training set and the confusion matrix for the test set. You can see the confusion matrix I found for the training set in Figure 5 and the confusion matrix I found for the test set in Figure 6.

I have written the following to display the confusion matrices:

```
conf_mat1 = pd.crosstab(trnPrd,lbls,rownames = ['y_pred'],colnames = ['y_truth'])
print(conf_mat1)
conf_mat2 = pd.crosstab(tstPrd,tstLbls,rownames = ['y_pred'],colnames = ['y_truth'])
print(conf_mat2)
```

```
------------------------------------------

The Confusion Matrix for the training set:

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
```

Figure 5: The Confusion Matrix for the training set

```
---------------------------------------
The Confusion Matrix for the test set:

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
```

Figure 6: The Confusion Matrix for the test set

We can observe from the confusion matrices that y_pred and y_truth are close for the class2 (There are a few misclassified points in the class2, approximately %5 of the data points are misclassified). For the class4, there are a lot of misclassified data points (for class4, approximately 90 percent of the data points are misclassified). Hence, the accuracy is not good for the class4. For the class1, approximately 40 percent of the data points are misclassified. This is better than the results of class4. For class3, approximately %22 of the data points are misclassified. This is better than the results of the class1. For class5, approximately %28 of the data points are misclassified. This is better than the results of the class1 but worse than the results of the class5.

Misclassification Percentages

Class1: %40
Class2: %5
Class3: %22
Class4: %90
Class5: %28

The accuracy order of the results for the classes (from low accurate to high accurate):

class4<class1<class5<class3<class2