**Project Part 1: Feature Extraction, Density Estimation and Bayesian Classification**

Data:

This dataset consists of the handwritten images of numeric 3 and 7. The statistics of the data which we will using are 5713 are the number of samples in the training set for handwritten image of 3 and 1428 are the number of samples in the testing set for handwritten image of 3. Then, 5835 are the number of samples in the training set for handwritten image of 7 and 1458 are the number of samples in the testing set for handwritten image of 7. And in this image is stored as 28x28 array of pixels.

Task 1: Feature Extraction and Normalization

First of all, in this we will import the training data then after that we will check the type of data is it and then convert it into flatten form as the data is in form of 2d array. After that we will compute the mean and standard deviation for each image of 784 pixels. Then, we mean Mj and standard deviation Sj, j=1,2, for the first and the second feature of 2-d vector Xi= [mi, si]t. After that we computed the Yi from it

Yi=[y1i,y2i]t= [(mi–M1)/S1,(si–M2)/S2]t

In this,

M1 is the mean of the feature1(image 3)

M2 is the mean of the feature 2(image 7)

S1 is the standard deviation of feature1(image 3)

S2 is the standard deviation of feature2(image7)

mi is the mean of all the features

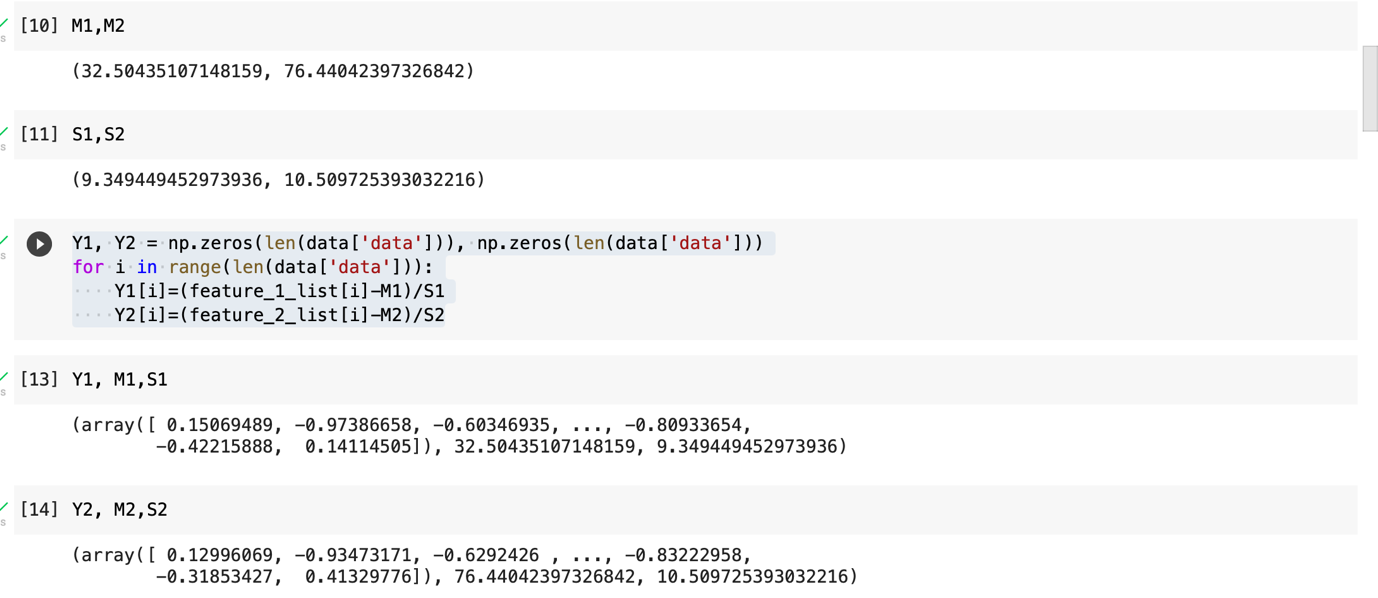
si is the standard deviation of all the features

from it we will compute the Y1i, Y2i for each image in the training set and after that from Y1i and Y2i we will compute the Yi which is going to be used as feature vectors defining our training sample.

ScreenShot of Code

Graphical user interface, text, application, email

Description automatically generated



M1 = 32.50435107148159

M2 = 76.44042397326842

S1= 9.349449452973936

S2 = 10.509725393032216

Y1= [ 0.15069489, -0.97386658, -0.60346935, ..., -0.80933654,

-0.42215888, 0.14114505]

Y2 = [ 0.12996069, -0.93473171, -0.6292426 , ..., -0.83222958,

-0.31853427, 0.41329776]

Task 2. Density estimation

The Yi is being computed from the above task then we will use the MLE to estimate the parameters for the 2-d normal distribution. In this first we will split the Y1 and Y2 is terms of data labels and then we will find the mean and standard deviation from the Y1 and Y2. Once the mean is calculated of the Y1 and Y2 we will compute the mu for both the images and after that from the above values we computed the discriminant for both the images and the discriminant serves as the parameter for MLE.

The formulas used to find the mean and covariance matrix for each image in the training set is given by:

Screenshot of the code

Graphical user interface, text, application, email

Description automatically generatedGraphical user interface, text, application

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Mean of image 3 is [0.37687996, 0.31851855]

Mean of image 7 is [-0.36900004, -0.31185886]

Covariance matrix for image 3 is [[1.04956469, 0.98760563],

[0.98760563, 0.96080009]]

Covariance matrix for image 7 is [[0.67669136, 0.74435619],

[0.74435619, 0.842203 ]]

Task 3. Bayesian Decision Theory for optimal classification

In this we will compute the minimum error rate classification for the two given cases.

In this first of all we computed the minimum error rate for the training data set for both the class labels (images of 3 and 7). Now using the mean and covariance matrix derived from MLE, we find the conditional probability of ‘x’ belonging to class label 3 or 7.

<<<<Give the formula p(x|w1) >>>>

p(x|w1) = \* ]

Now we say: if p(w1|x) > p(w2|x) we select w1 else select w2

To find the posterior probability, we use the equation:

P(wi|x) = p(wi) \* p(x|wi) / P(x)

Here p(x|wi) is found from the multivariate normal density equation.

P(wi) is the class prior and P(x) is derived from the total probability rule:

P(x) = P(w1)\*p(x|w1) + P(w2)\*p(x|w2). ------ total probability rule

Hence our decision rule here comes to:

If p(w1) \* p(x|w1) / P(x) > p(w2) \* p(x|w2) / P(x). ---- choose w1 and vice versa

Error occurs in our system if P(error|x) = P(w2|x) when p(w1|x)>p(w2|x) and vice versa

Screenshot of the code

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated with medium confidenceGraphical user interface, text, application

Description automatically generated

Graphical user interface, application, Teams

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

P(error) for P(3)=P(7)=0.5

P(error) for training it is 0.25098557306484687

P(error) for testing is 0.23738030097678892

P(error) for P(3)=0.3 and P(7)=0.7

P(error) for training it is 0.2550935170023689

P(error) for testing is 0.17567475918338987