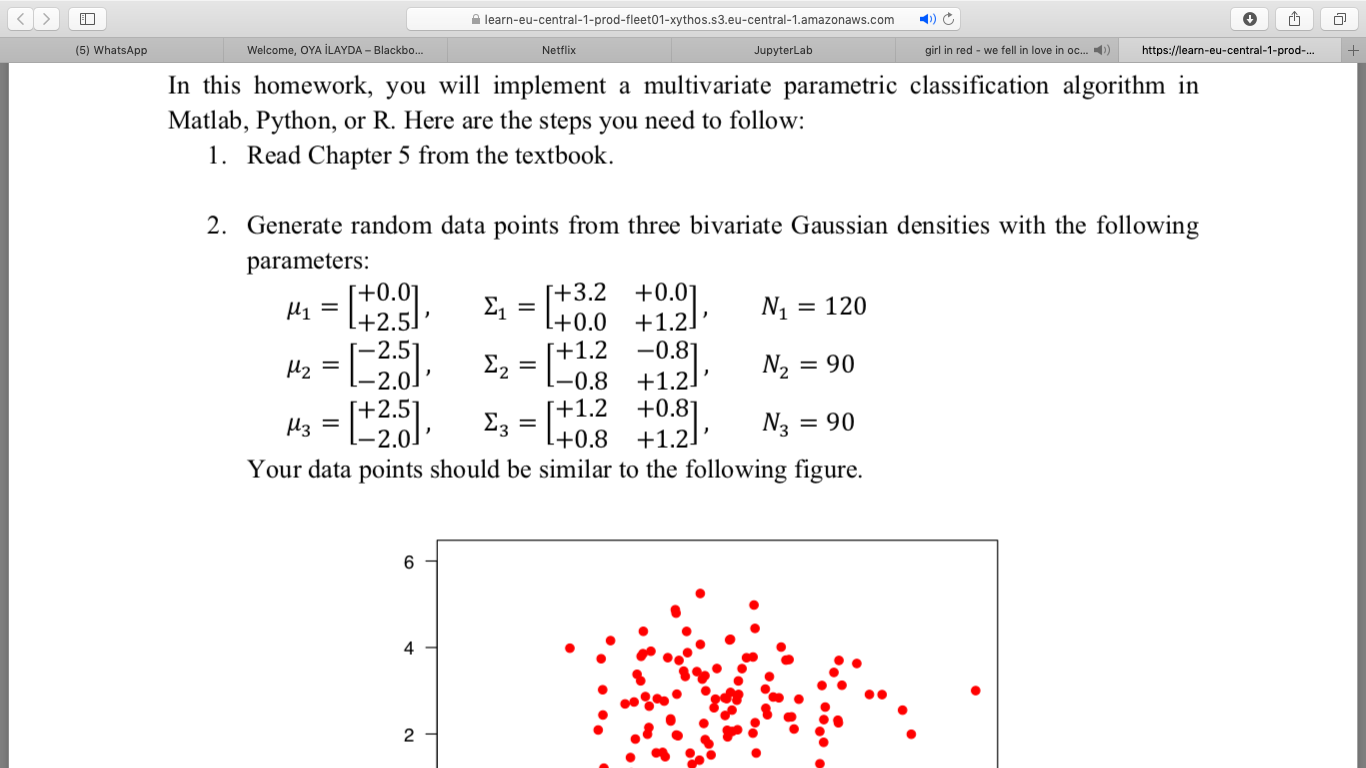
Oya İlayda Yalçın

60692

09/11/2020

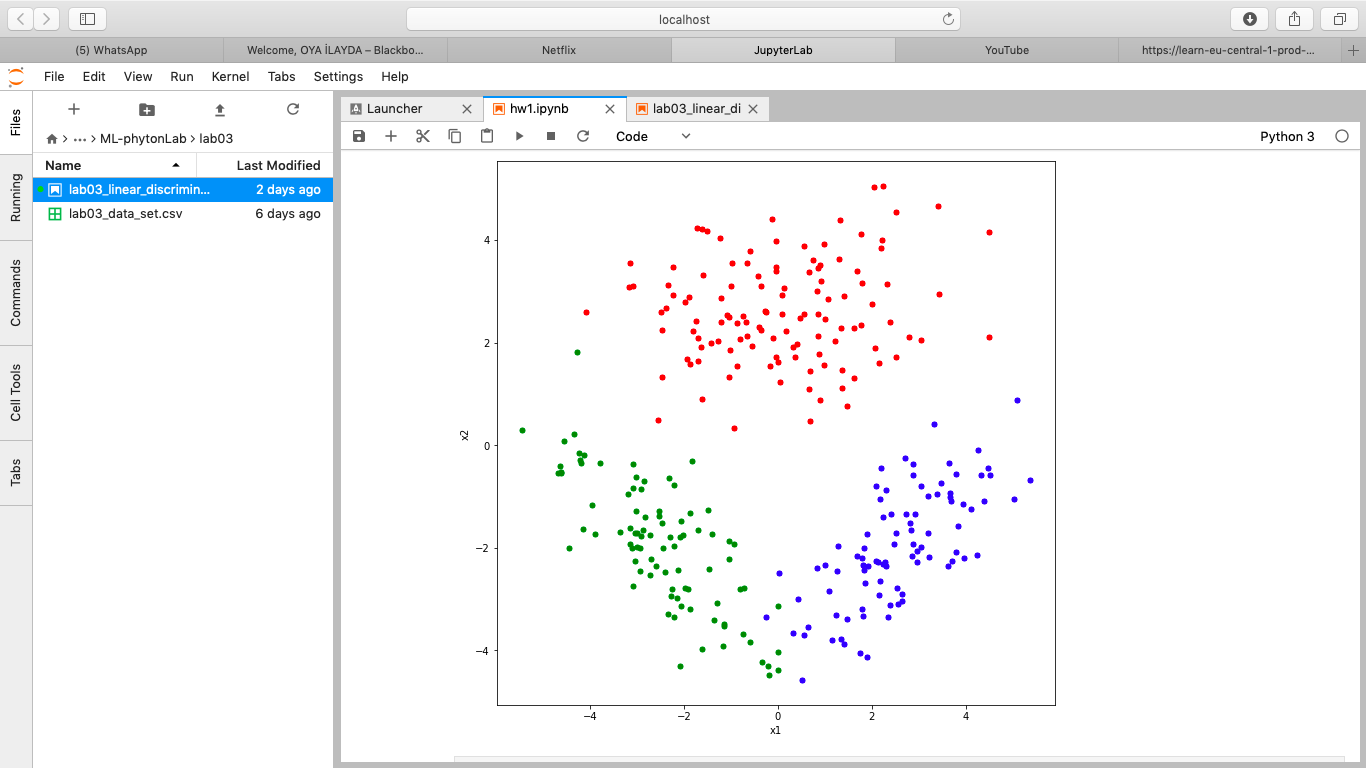
ENGR 421 HOMERWORK#1 REPORT

In this homework it was wanted from us to do an implementation of multivariate parametres classification. The given paramteres were as shown below:



In the first part of the code I have defined these given parameters and created multiple arrays in order to store them. Later on, using the built in function: np.random.multivariate\_normal

Putting the default parameters above to this function I have obtained three data groups found by Gaussian Distribution. Plotting these data, I have found the graph given below. In this graph 3 different colors represent the three different groups of data.



Looking at this graph, It can already be visualized that red points have a mean of x1 near 0, mean of x2 near 2.5; blue points have a mean of x1 near -2.5, mean of x2 near -2 and green points have a mean of x1 near 2.5, mean of x2 near -2.

It is also possible to talk about the covariances by this graph since the allocation of green and blue have a mirror like reflection among themselves. This point is also clear at the covariances of the two groups. In addition, red points are generally allocated horizontallly so it is logical that some of the elements of this group’s covariance matrixes are equal to zero.

Moving in the code, it was required for us to find the sample means, sample covariances and class priors of these random data.

Starting from the sample mean, I have used the built in mean function of phyton and then used the formula given below:

Caltulating the mean of each x1 and x2 of the groups, I came up with given means below:

1 2

1 (0.04453807022577969, 2.612251275421772)

2 (-2.4849106679801958, -1.9481798439846398)

3 (2.5495473485309432, -1.9888099037097666)

It is seen that there are some differences in these found means and it is expected to find so. In this exercise we generated random arrays of points. So, it is normal to find some variances

From the actual means given. It is important to underline that these variances are neglectable which can be interpreted from the real covariance and mean values.

               After this step, the covariance matrices of the random values are calculated. In order to calculate this value two loops are implemented. We had to calculate the Cov00, Cov01,

 Cov10 and COV11. Cov01 and Cov10 must be equal to each other. The equation  implemented, to calculate these values are given below:

Matrix multiplying a 2x1 and a 1x2 matrixes we end up with a 2x2 covariance matrix. Found results are given below:

**1:** 2.81619317, 0.22436503

0.22436503, 1.00404699

**2:** 1.53076989, -1.18878261

-1.18878261, 1.572812

**3:** 1.39930019, 0.92292252

0.92292252, 1.24783096

Again same argument fort he sample means pass also here. Sample covariances are a little

different from the given ones due to randomness.

Lastly, I have found the class priors. Simply the formula is:” “ found results are:

**1 2 3**

0.4 0.3 0.3

After finding these parameters I calculated the estimated y’s. For this part first I found the    score functions of these random data points one by one. To do this I used the equations given below:

[P(x|y =c) P(y=c)]

Class conditional Class prior

The class conditional equatio for a multivariate normal distrubition is given below:

Putting this function to the one given above we end up with the score function use in the code:

Putting this function as the score function to the code, the next was to calculate each score function of the each data point. Every data point is put in to the position of x. Later, the parameters of the equation are changed 3 times. Every group’s parameters are implemented for a single data point and then maximum value among these values are chosen. The maximum value found is accepted as the score value.

Later on the found data is matched with the corresponding label. This label was the one which gave the maximum score function. Appending the found labels to a list the y predicted is then used to find the confusion matrix. Using the built in function of phyton the found confusion matrix is given below:

**y\_truth 0 1 2**

**y\_pred**

**0 120 1 0**

**1 0 89 1**

**2 0 0 89**

Looking at these tables it can be said that, out of 120 data points, the 1st group is allocated all correctly. Out of 90 data points, the 2nd group is allocated with 1 mistake. A data point is labeled as  1. Out of 90 data points, the 3rd group is allocated with 1 mistake. A data point is labeled as  2.

From this point on I plotted the end estimations. First of all the logic behind my code was to divide the whole plane into little points. Dividing the plane into grids, I created two 200 point linear spaces for both x1 and x2 and the end result was a 2D 200x200 grid. Then these points are scanned through the grid. These data points are then put in the score function explained above and calculations for each group parameter is made. In the end, these score functions found are put in a discrimination array with. All the classes are predicted for each data points by this way. Whenever a point has differing predictions from any of its immediate surroundings, it is accepted as a boundary point.

This logic that I found could not be plotted due to my errors in the code which I could not find the source of. The codes that do these steps are included in the source code that I provided but they are inactivated in the code. Please look at them to see the logic behind it and that I tried to implement it. Without the boundaries, the graph which has the data points that are wrongly estimated in circles are given below:

