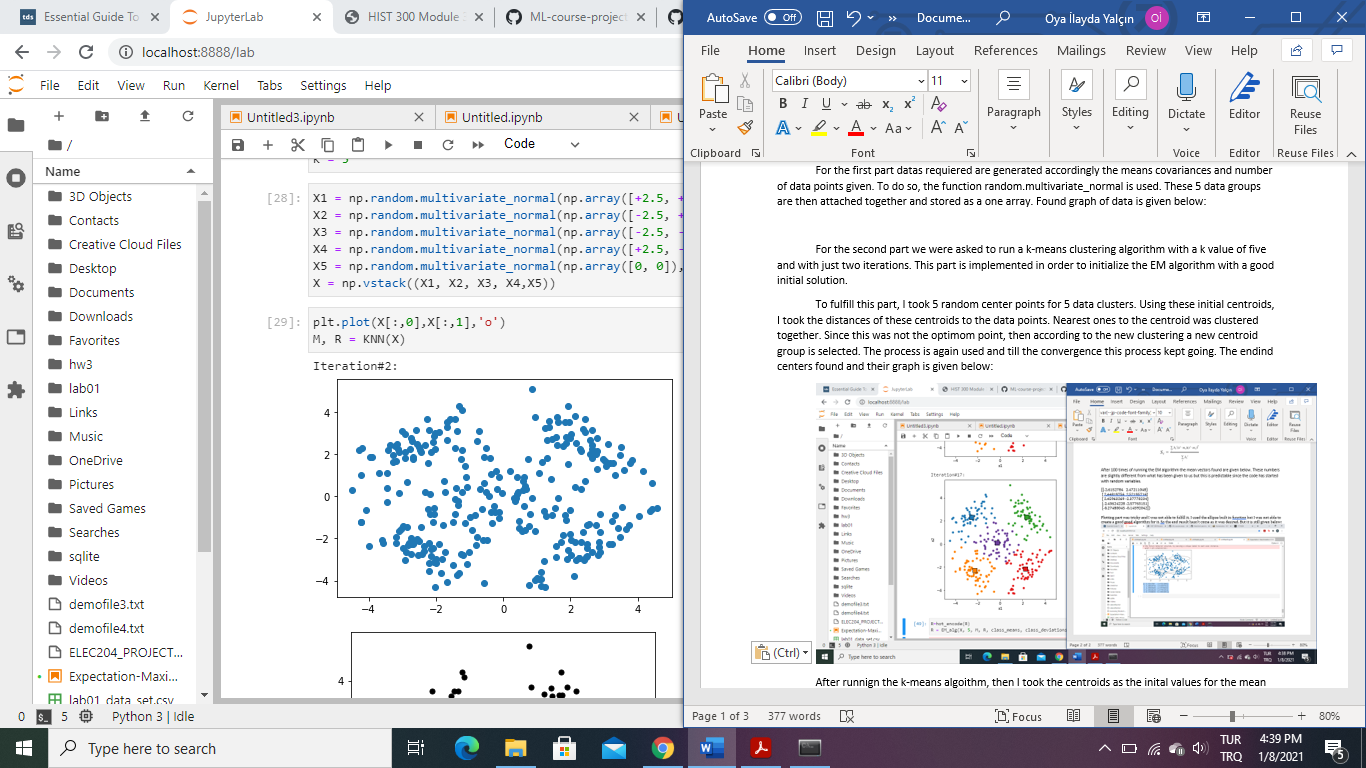
Oya İlayda Yalçın

60692

**ENGR 421- INTRODUCTION TO MACHIENE LEARNING HOMEWORK #7 REPORT**

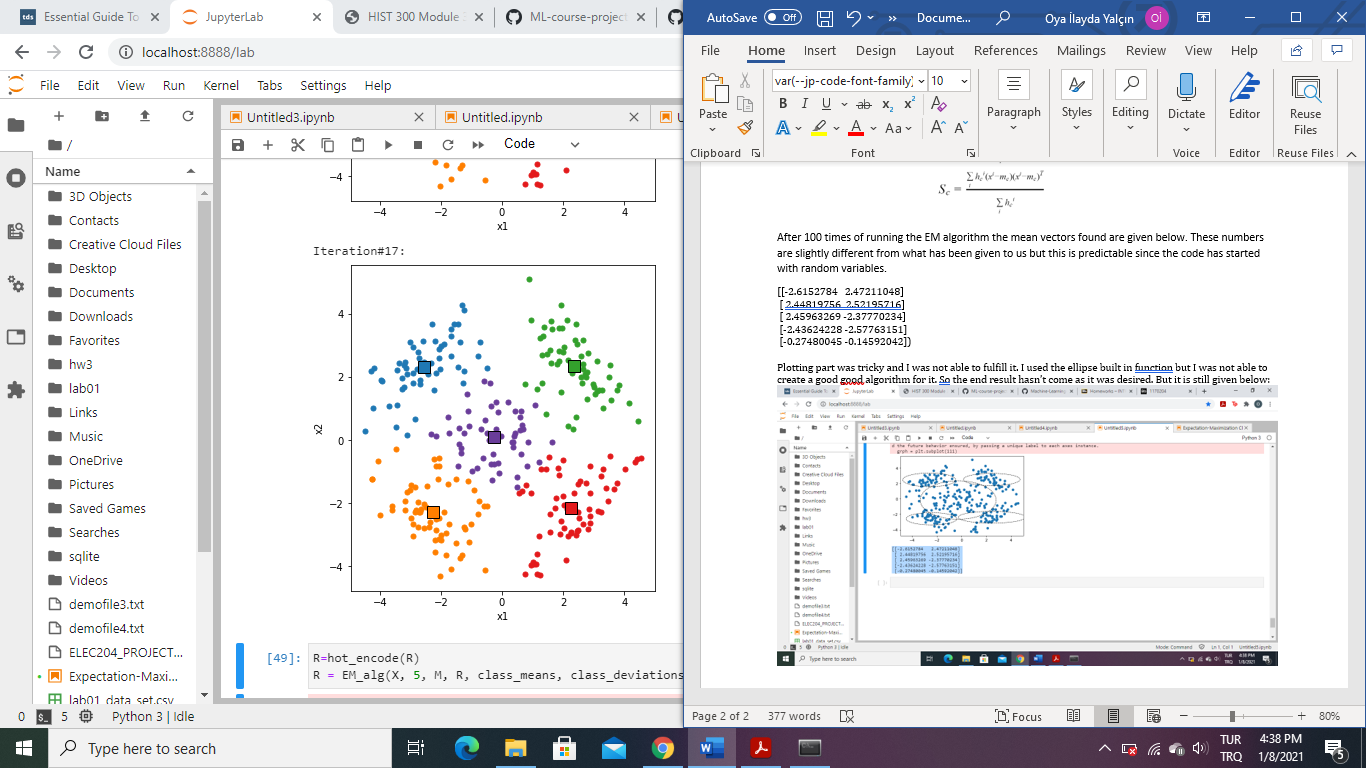
In this homework we were asked to implement an expectation maximization clustering.

For the first part datas requiered are generated accordingly the means covariances and number of data points given. To do so, the function random.multivariate\_normal is used. These 5 data groups are then attached together and stored as a one array. Found graph of data is given below:

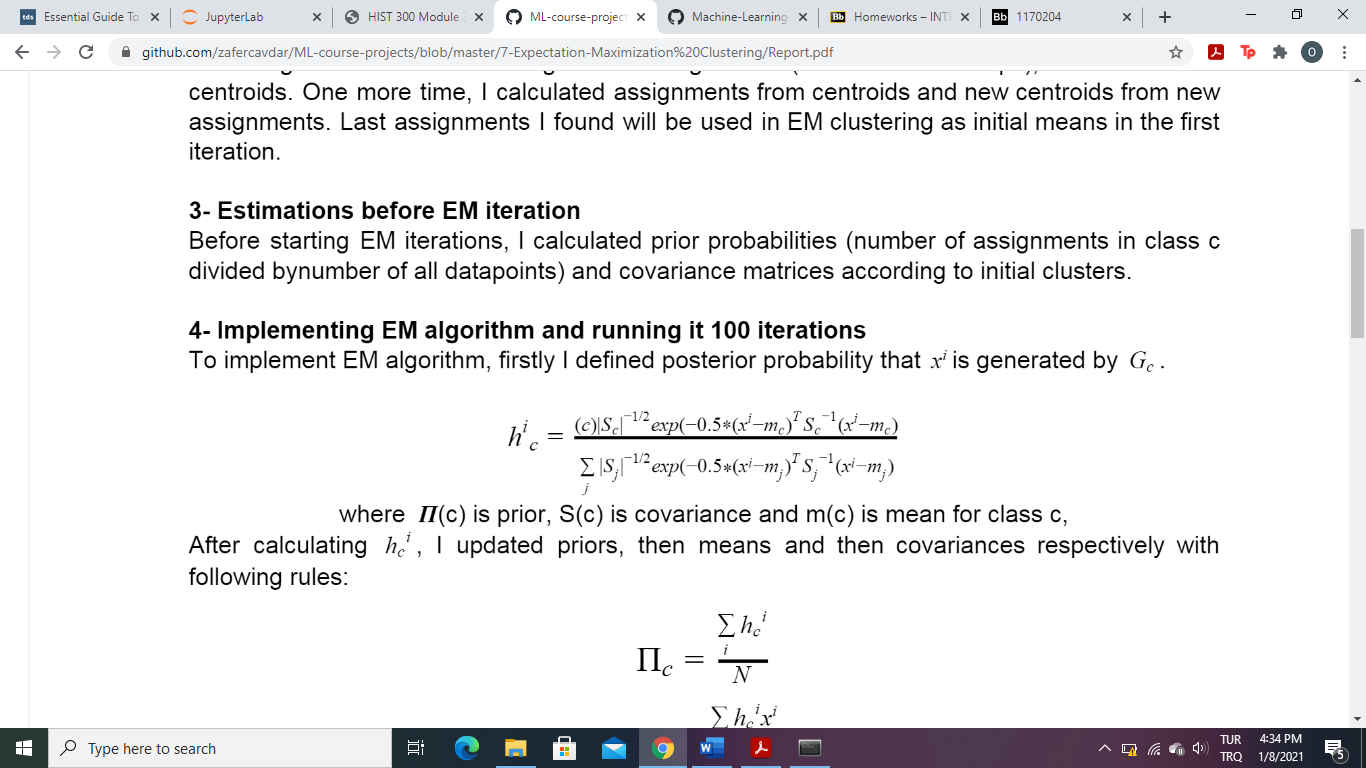


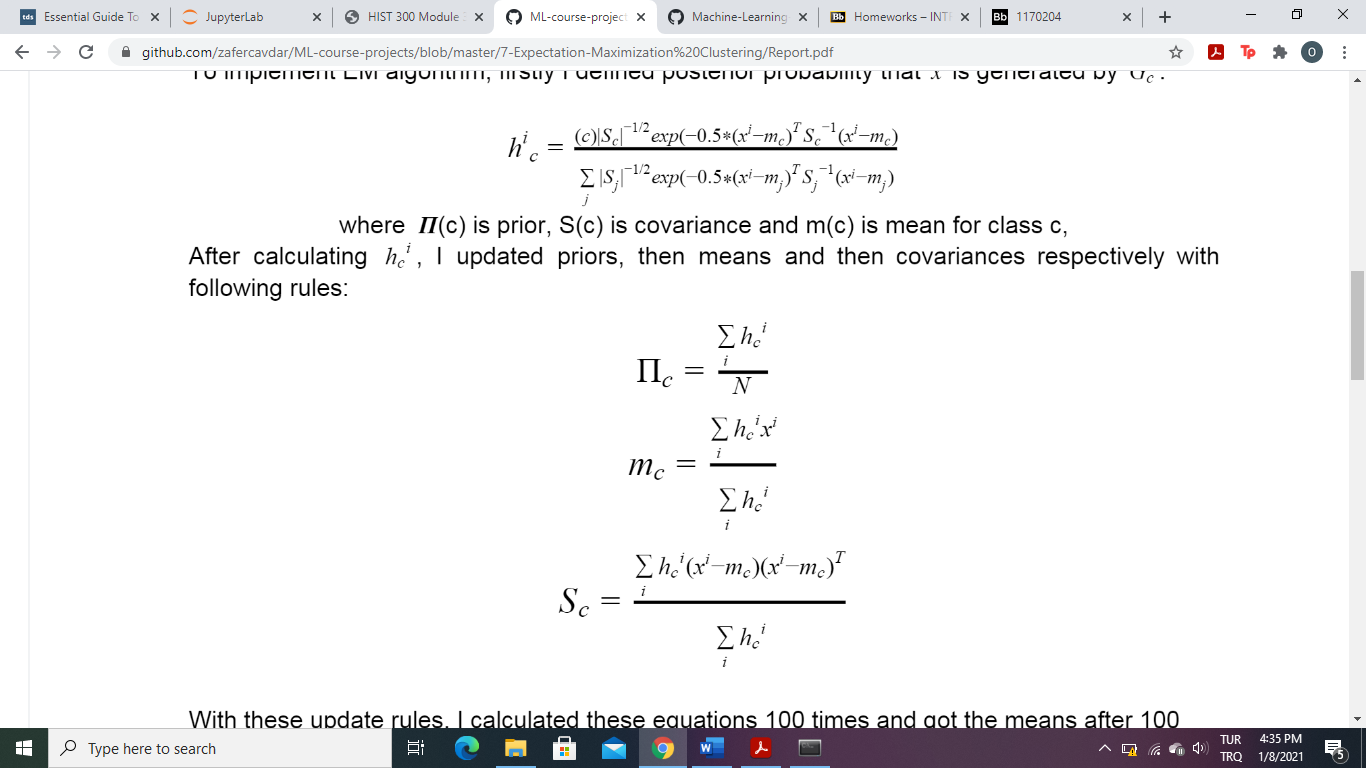
For the second part we were asked to run a k-means clustering algorithm with a k value of five and with just two iterations. This part is implemented in order to initialize the EM algorithm with a good initial solution.

To fulfill this part, I took 5 random center points for 5 data clusters. Using these initial centroids, I took the distances of these centroids to the data points. Nearest ones to the centroid was clustered together. Since this was not the optimom point, then according to the new clustering a new centroid group is selected. The process is again used and till the convergence this process kept going. The endind centers found and their graph is given below:



After runnign the k-means algoithm, then I took the centroids as the inital values for the mean vectros in my EM algorithm. Using the data points covered by each center, I estimated the initial covariance matrices and prior probabilities in my EM algorithm. The used formula to find the posterior probability and the update formulas for prior, mean and covariances are given below. These are given and found to the EM algorithm to give a start point data. The algorithm has ran for 100 times as required.





After 100 times of running the EM algorithm the mean vectors found are given below. These numbers are slightly different from what has been given to us but this is predictable since the code has started with random variables.

[[-2.6152784 2.47211048]

[ 2.44819756 2.52195716]

[ 2.45963269 -2.37770234]

[-2.43624228 -2.57763151]

[-0.27480045 -0.14592042])

Plotting part was tricky and I was not able to fulfill it. I used the ellipse built in function but I was not able to create a good algorithm for it. So the end result hasn’t come as it was desired. But it is still given below:

