Sakshi Udeshi
1003197
14 November 2017

Machine Learning
Assignment 3

Parameters to be estimated (Q1)

We need to estimate the mean (μ) and variance (σ^2) for the two clusters for each of the given data.

The labels are hidden. We do not know which cluster a point belongs to.

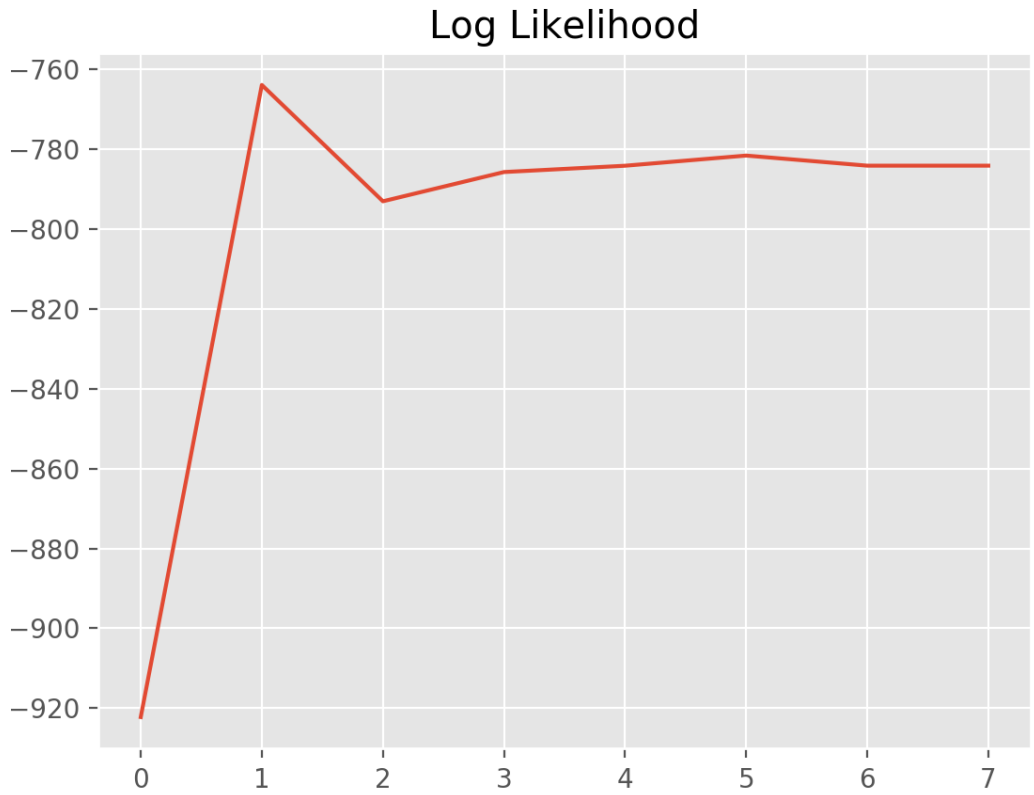
Based on the gaussian distribution, we can generate probabilities (in the soft EM case) to try and guess which cluster the points belong to.

And after we guess, we re-generate the probabilities again. This is the expectation maximisation algorithm.

Hard EM (Q2)

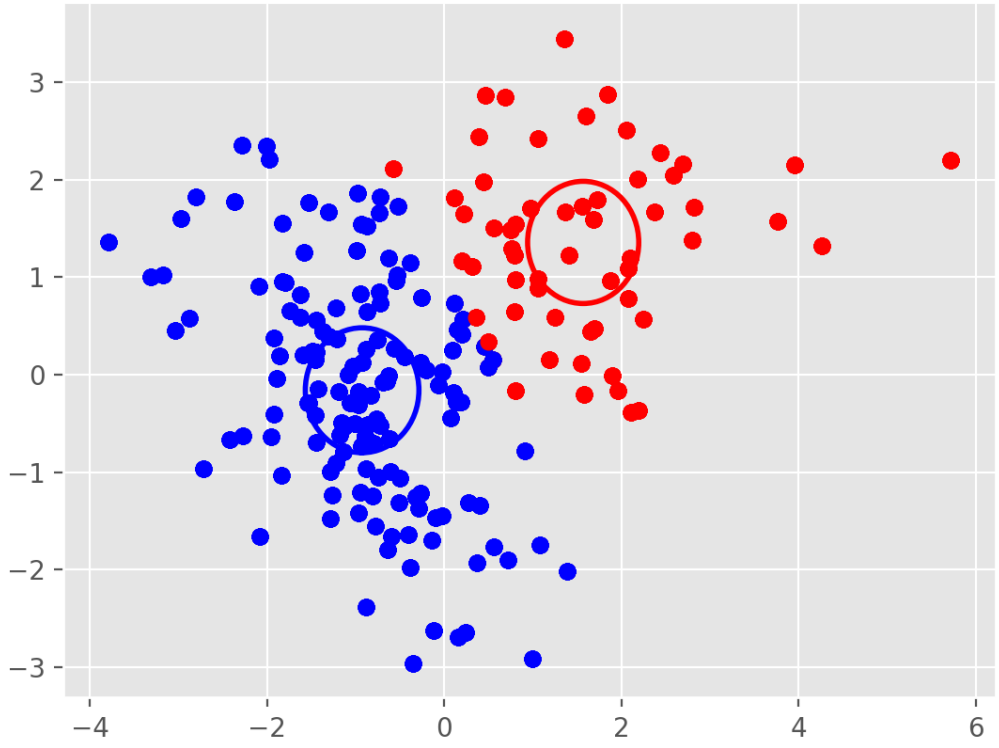
I randomly initialise two values μ_1 and μ_2 representing the mean of the the two clusters. I assign the points to the means depending on their euclidian distance from the clusters and run the hard expectation maximisation algorithm.

At each iteration the log likelihood is calculated and when the difference between the previous iteration's log likelihood and this iteration's log likelihood goes below a threshold value (0.005), I terminate the algorithm.



The graph isn't optimal because it is Hard EM.

For large dataset 1, the data looks like this. One can run the visualisation and the final result of the code here.



Soft EM (Q3)

We try to maximise the following function.

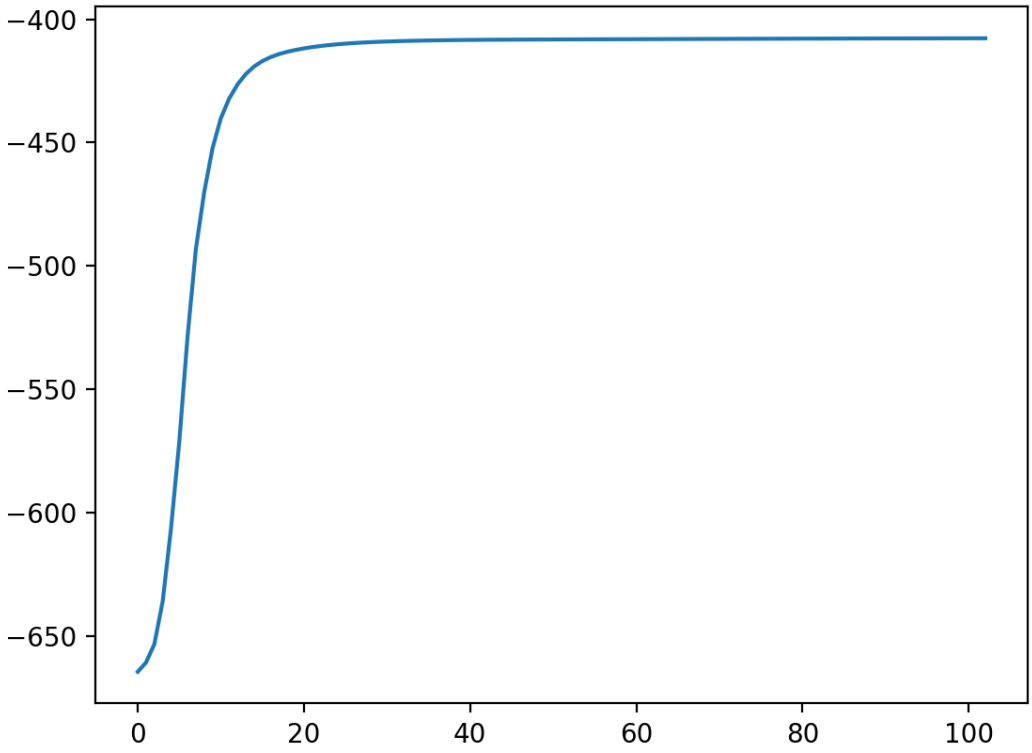
$$\sum_{t=1}^{n} logP(x^{(t)}|\theta) = \sum_{t=1}^{n} log\Big(\sum_{t=1}^{k} p_t P(x^{(t)}|\mu^{(t)}, \sigma_t^2)\Big)$$

I randomly initialise two values μ_1 and μ_2 representing the mean of the the two clusters.

I assign random probabilities to each point, representing their probability that they belong to that cluster. After that, I run the expectation maximisation algorithm, and store the log likelihood.

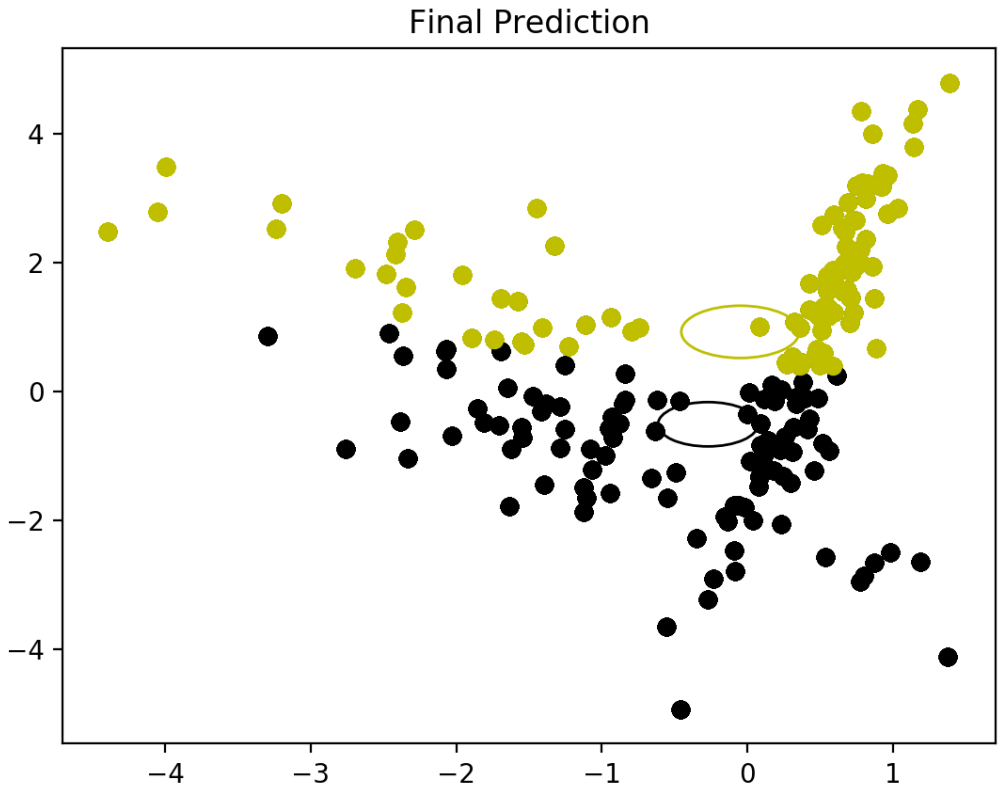
At each iteration the log likelihood is calculated and when the difference between the previous iteration's log likelihood and this iteration's log likelihood goes below a threshold value (0.005), I terminate the algorithm.

The graph shows the log likelihood as a function of no of iterations.



An example of the prediction is

The code to run all the datasets can be found here



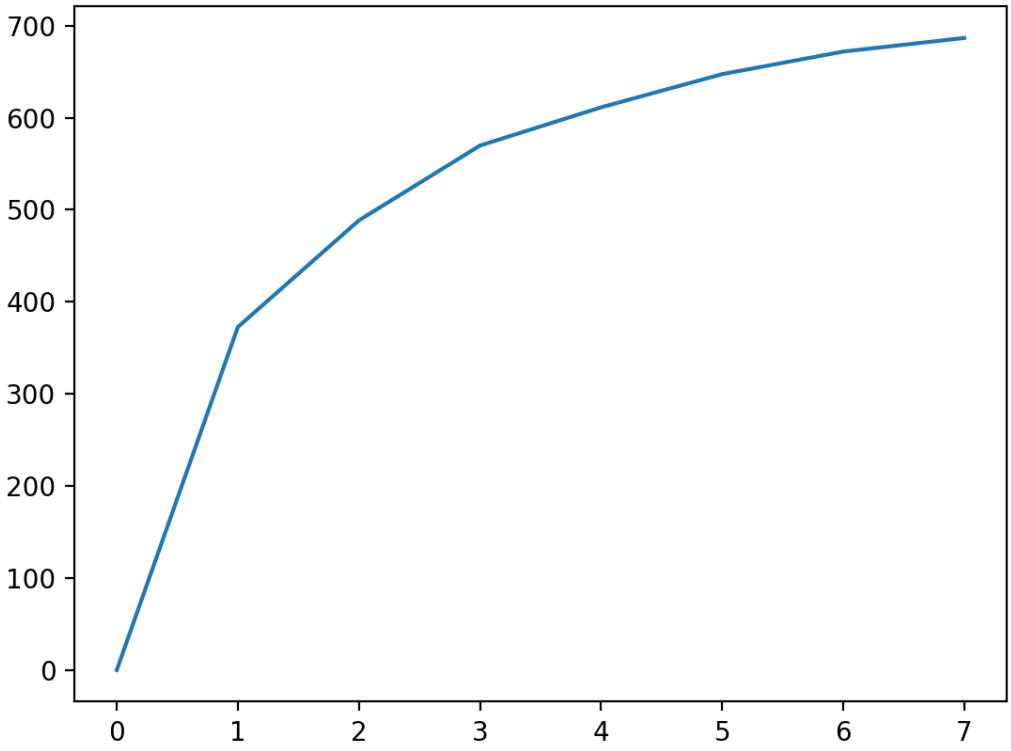
Determining cluster (Q4)

The Elbow method is a method of interpretation and validation of consistency within cluster analysis designed to help finding the appropriate number of clusters in a dataset.

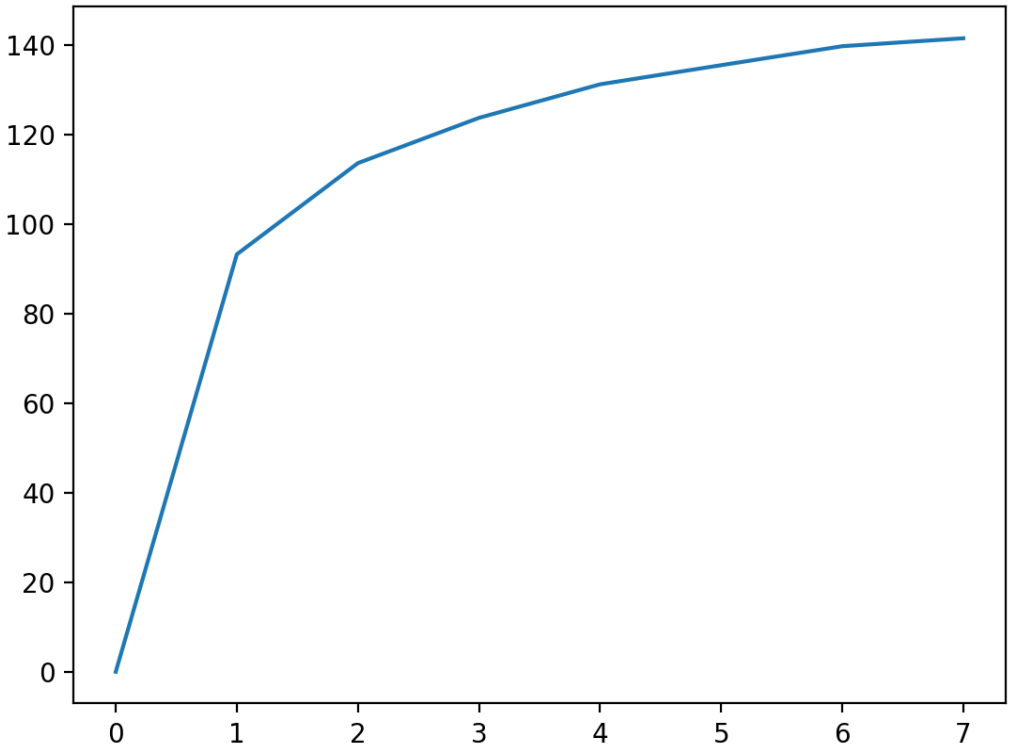
This method looks at the percentage of variance explained as a function of the number of clusters: One should choose a number of clusters so that adding another cluster doesn't give much better modelling of the data. More precisely, if one plots the percentage of variance explained by the clusters against the number of clusters, the first clusters will add much information (explain a lot of variance), but at some point the marginal gain will drop, giving an angle in the graph. The number of clusters is chosen at this point, hence the "elbow criterion". This "elbow" cannot always be unambiguously identified.

Running the elbow methods on the mystery datasets, I get

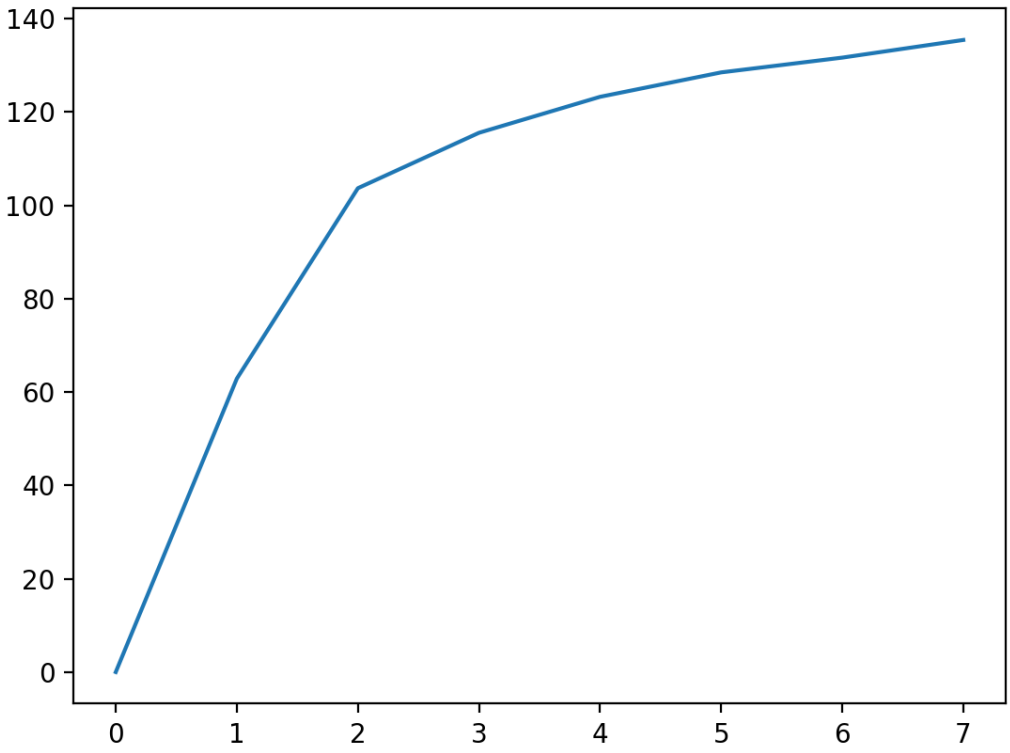
Mystery 1



Mystery 2



Mystery 3



The code can be found here