Name: Chandramohan Natarajan

Student number: 190617866

Assignment number: 5

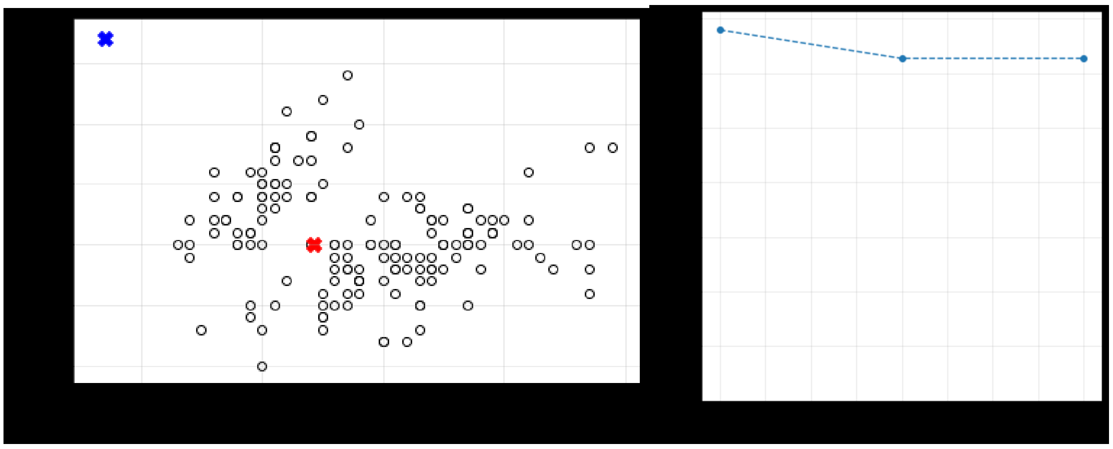
Module: ECS766

Exercise 0: What do you observe about the dependence of the final cluster quality on the number of clusters K used? Why? [1 mark]

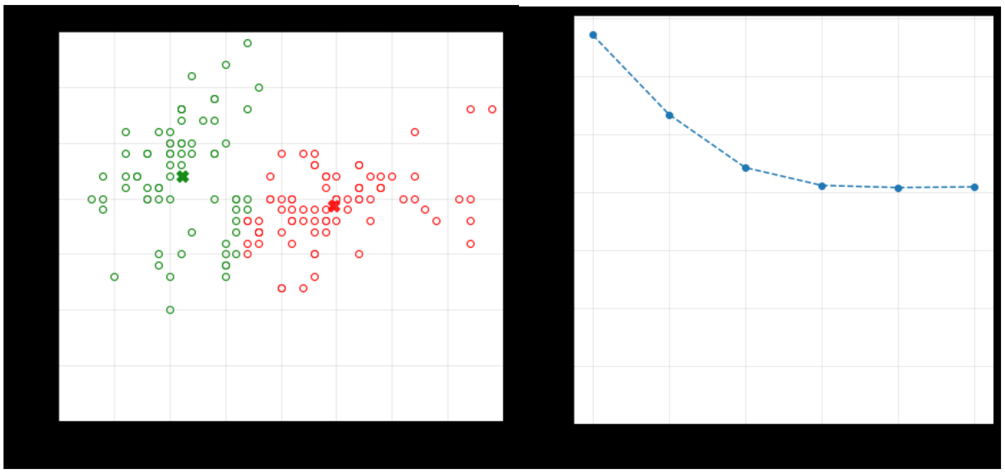
The exercise is performed   to segregate iris parameters with similar traits and assign them into clusters. The goal of the k-means algorithm is to find colours of chosen dimension in iris data, with the number of clusters represented by the variable K. Cluster means from the k-means algorithm are nonparametric estimators of principal points. A parametric k-means approach is introduced for estimating principal points by running the k-means algorithm on a very large simulated data set from a distribution whose parameters are estimated using ML. Expectation–maximization (E–M) is a powerful algorithm that comes up in a variety of contexts and in k-means is a particularly simple and easy-to-understand application of the algorithm. The expectation–maximization approach here consists of the following procedure. Cluster points are chosen in random and converged repeatedly. We have coding to execute following tasks. E-Step would assign points to the nearest cluster centre and M-Step set the cluster centres to the mean.In the above mentioned the "E-step" involves updating the expectation of which cluster each point belongs to. The "M-step" is involves maximizing some fitness function that defines the location of the cluster centres. In the exercise that maximization is accomplished by taking a simple mean of the data in each cluster.

Plots and graph

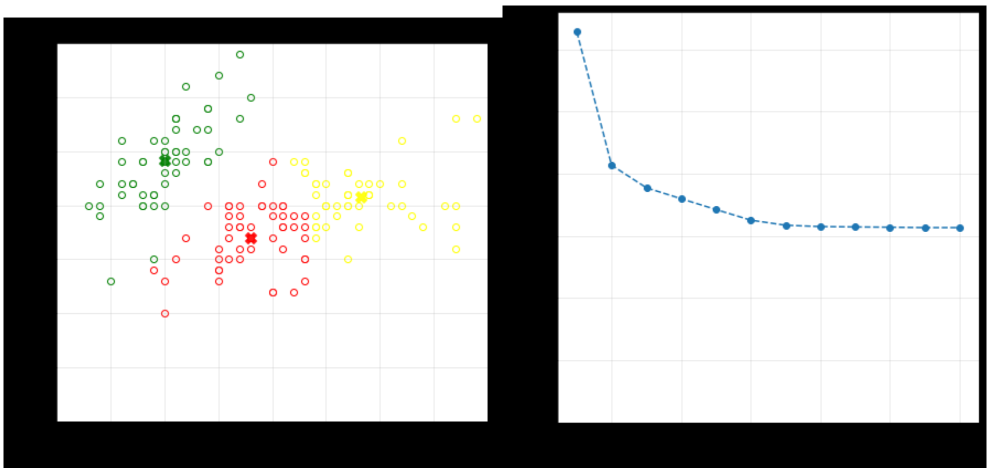
Iteration 2/k=3



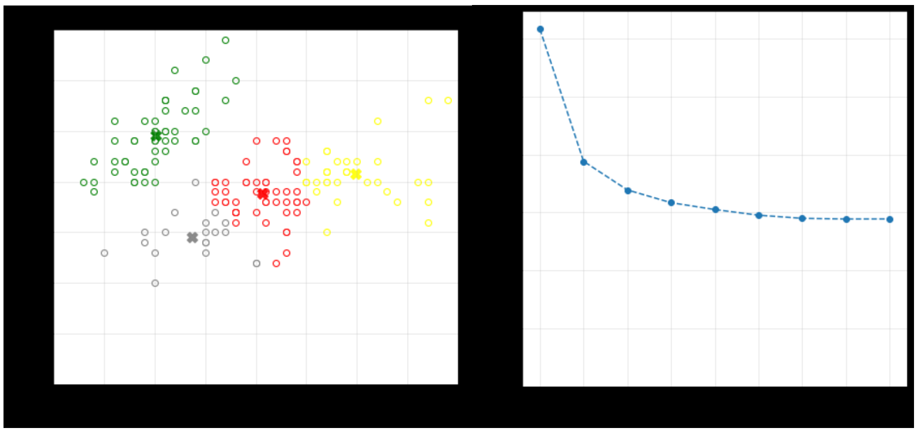
Iteration 6/K=3



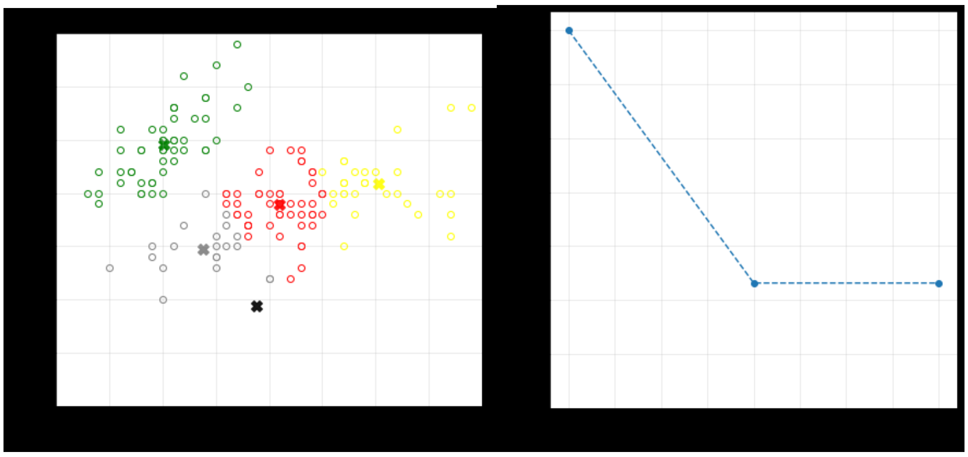
Iteration 11/K=4



Iteration 8 / k=-5



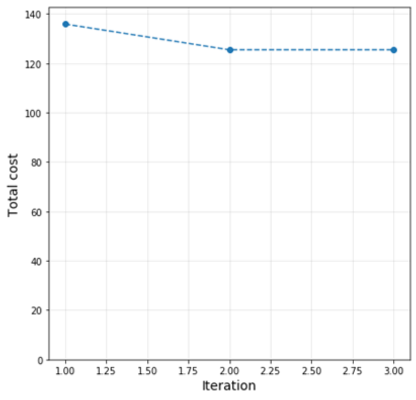
Iteration 3 /K=7

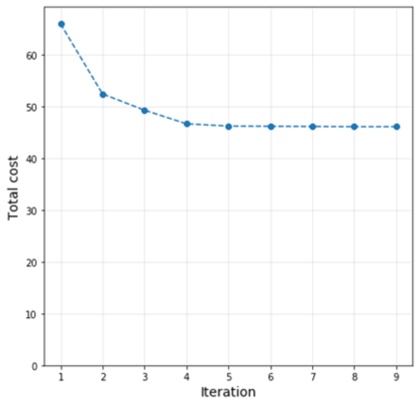


The cost function illustrated below shows the behaviour of increase in K in reflection to Cost graph.

K=2

K=7





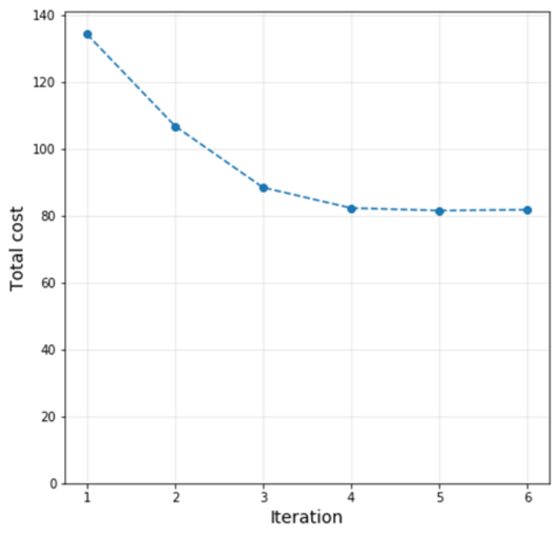
Under typical scenarios, each repetition of the E-step and M-step will always result in a better estimate of the cluster characteristics.

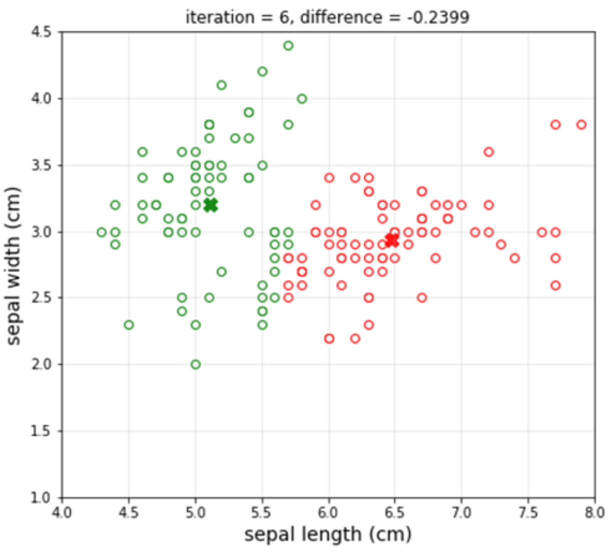
Exercise 1: Find a seed that gives a different final quality of clusters (in terms of total distance). Include both the values of the seed, the final distance and the picture of the cluster with your answer. [1 mark]

TO get more clarity on how to decide the"value of k" in k-means (k= amount of clusters), because any additional cluster improves the quality of the clustering but at a decreasing rate, and having too many clusters may be useless to explain the data.

The number of initial seeds (initial centres of clusters) is the same as number of clusters (at least in the original k-means). The problem of the values of the seeds is different than problem of number of clusters. In this exercise, we used random cluster centres, but some implementation provides us with better ways to choose them. With better seeds, k-means converges faster and the quality of the clusters is good.

Iteration 6/K=3   Difference = -.2399





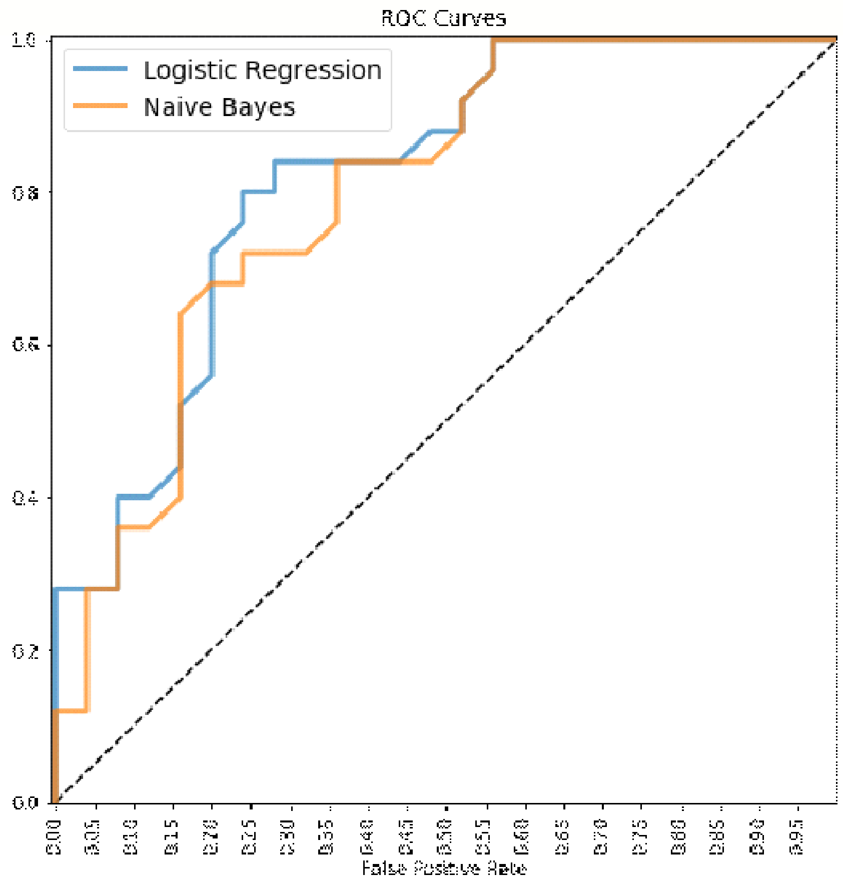
Exercise 2: Has the clustering accuracy improved from before to after of the uncommenting above? Why? [1 mark]

We are hereby increase the dimensional features of the set and obviously the accuracy has progressed. Accuracy has increased over 8 % and reached 89.3%. Whereas with less dimension accuracy just hover around 80%.

Exercise 3: The following cell contains a function that given a classifier and a threshold and some (test) samples, returns the TPR and FPR. Use this functoin to try a bunch of thresholds and fill in the TPR and FPR vectors. Plot TPR (y-axis) against FPR (x-axis) to visualise the resulting ROC curve. Provide both your code and the figure. [1 mark]

Code:

```
# Now, write your code here (to compute a 1-d arrays of TPR and FRP for different values of the threshold)

import numpy as np

TPR_CL = []*8000

FPR_CL= []*8000

for i in np.linspace(0,1,8000):
  TP,FP = compute_tpr_fpr(LR_classifier,i,Xte,Yte)
  TPR_CL.append(TP)
  FPR_CL.append(FP)

TPR_DL =[]*8000

FPR_DL =[]*8000

for i in np.linspace(0,1,8000):
  TP1,FP1=compute_tpr_fpr(NB_classifier,i,Xte,Yte)
  TPR_DL.append(TP1)
  FPR_DL.append(FP1)
```



Exercise 4: Compare the AUC of the ROCs of the two classifiers. Which one is preferable by the AUC metric? [1 mark]

AUC gives the rate of successful classification by the logistic model. The AUC makes it easy to compare the ROC curve of one model to another. The Naive Bayes is linear classifier using Bayes Theorem and strong independence condition among features. Logistic regression is mainly used in cases where the output is boolean. Multi-class logistic regression can be used for outcomes with more than two values. Logistic regression measures the relationship between a output variable Y (categorical) and one or more independent variables, which are usually (but not necessarily) continuous, by using probability scores as the predicted values of the dependent variable. Naive Bayes also assumes that the features are conditionally independent. Real data sets are never perfectly independent but they can be close. In short Naive Bayes has a higher bias but lower variance compared to logistic regression. If the data set follows the bias then Naive Bayes will be a better classifier. Both Naive Bayes and Logistic regression are linear classifiers, Logistic Regression makes a prediction for the probability using a direct functional form where as Naive Bayes figures out how the data was generated given the results.

In the above exercise , AUC of Logistic regression is 82 % whereas for Naïve, it is cloase to 80% only.

Exercise 5: Suppose for a particular application, the maximum allowed FPR is 0.16. Which classifier is preferable? Obtains the maximum TPR given this FPR constraint? [1 mark]

To draw an ROC curve, only the true positive rate (TPR) and false positive rate (FPR) are needed (as functions of some classifier parameter). Since TPR is equivalent to sensitivity and FPR is equal to 1 – specificity.

FPR = .16 which is (1-specificity)

TPR =sensitivity = TP/(TP+FN)

=TP/(TP+.16)

The maximum TPR obtained for Naïve Bayes is .66 whereas for Logistics regression is .54. So its concluded that Naïve Bayes is much preferred classifier.