Neeraj Vashistha
190573735

Assignment 5

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

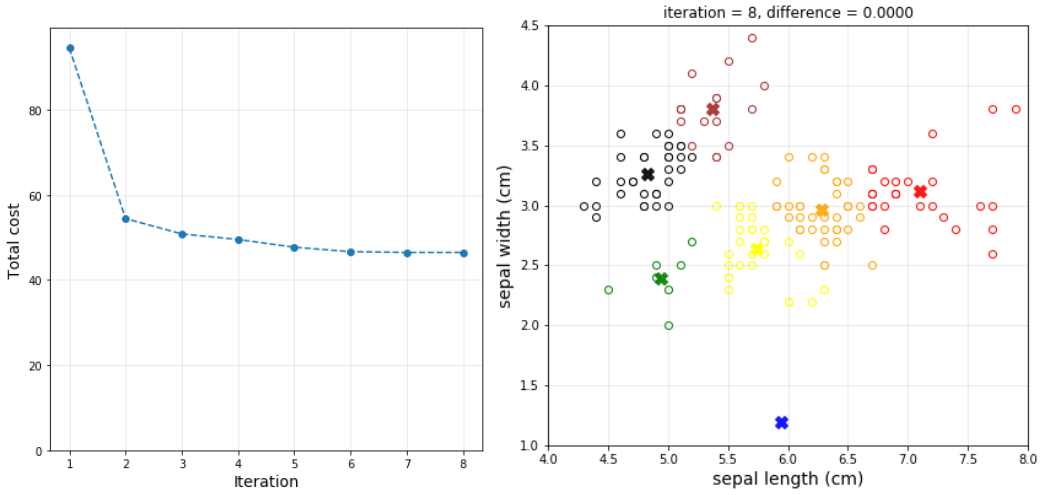
Below observation are on seed = 3

As we increase the number of iterations, for k=3 and k=4 the cost/distance decreases to lowest and is observed around 62 and 58 respectively and good clusters are obtained. With this keeping in mind we observe, if k = 7, the cost is very low, but we do not get good clusters, and as we increase the k from 2 to 7 the cost/distance decreases but clusters quality decreases too.

k	cost/distance	quality
2	83.30	good
3	62.699	good
4	58.05	good
5	56.09	Not good
6	50.24	Not good
7	46.46	Not good

This could be explained as the number of clusters increases the number of datapoint for each cluster decreases and thus the total Euclidean distance between the centroid to the data point decreases.

When k=7, there are 6 clusters and we can see that one of the clusters is not correctly formed (no data points).

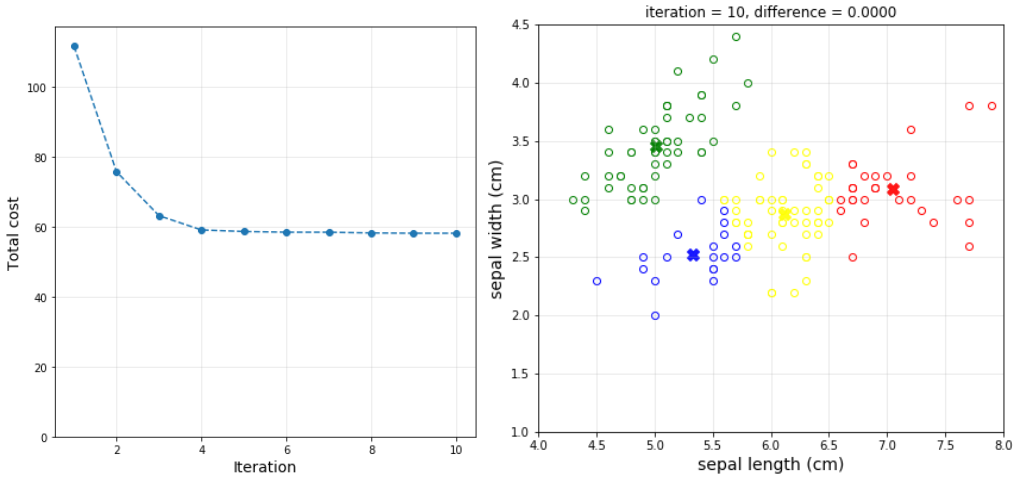


When k=7 the cost/distance drops from 95 to 55 at iteration 2 then stablises to 46 on increasing the iterations to 8.

Neeraj Vashistha
190573735

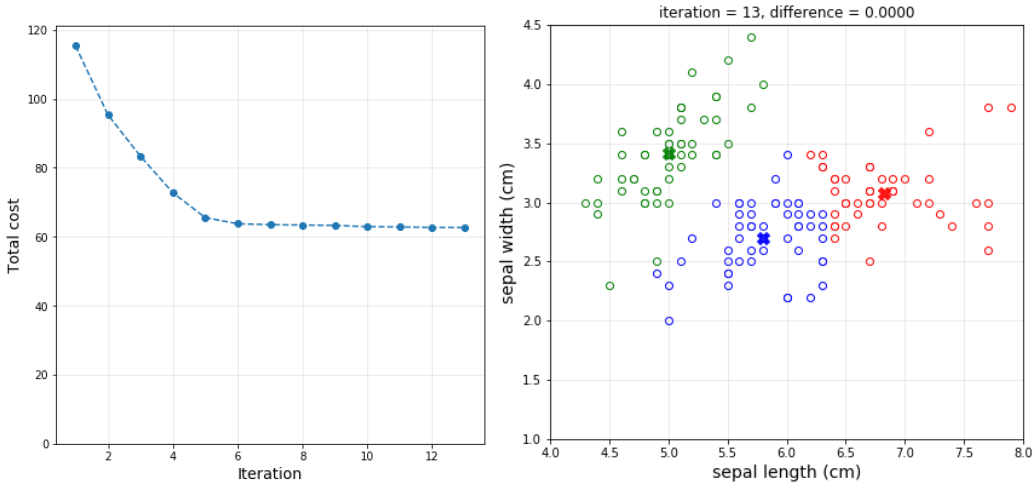
When k=6, there are 5 clusters and we can see that one of the clusters is not correctly formed (no data points). When k=6 the cost/distance drops from 95 to 55 at iteration 2 then stablises to 50 on increasing the iterations to 8.

When k=5, there are 4 clusters and we can see that one of the clusters is not correctly formed (no data points). When k=5 the cost/distance drops from 105 to 65 at iteration 2 then stablises to 56 on increasing the iterations to 5.
When k=4, there are 4 clusters and we can see all the clusters are formed correctly.



When k=4 the cost/distance drops from 105 to 75 at iteration 2 then further decreases and smoothens out at iteration 4 and finally stablises to 58 on increasing the iterations to 10.

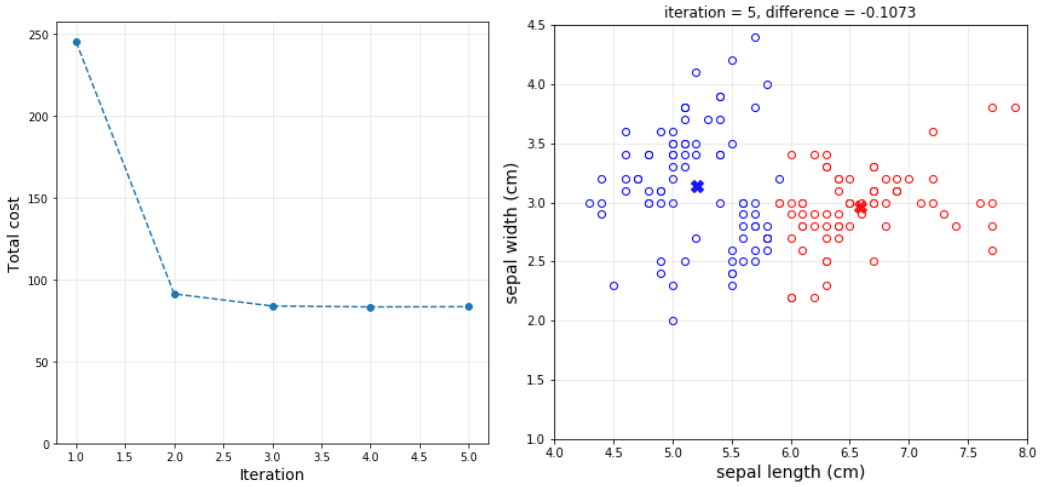
When k=3, there are 3 clusters and we can see all the clusters are formed correctly.



When k=3 the cost/distance gradually decreases from 115 to 63 over first five iteration and then finally stablises to 62 on increasing the iterations to 13.

When k=2, there are 2 clusters and we can see all the clusters are formed correctly.

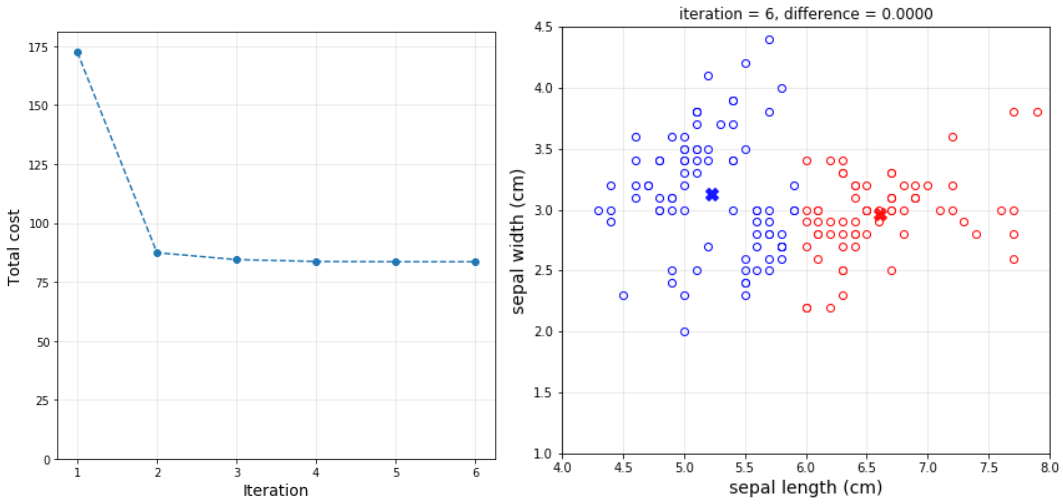
Neeraj Vashistha
190573735



When k=2 the cost/distance drops from 245 to 95 at iteration 2 then stablises to 83 on increasing the iterations to 5.

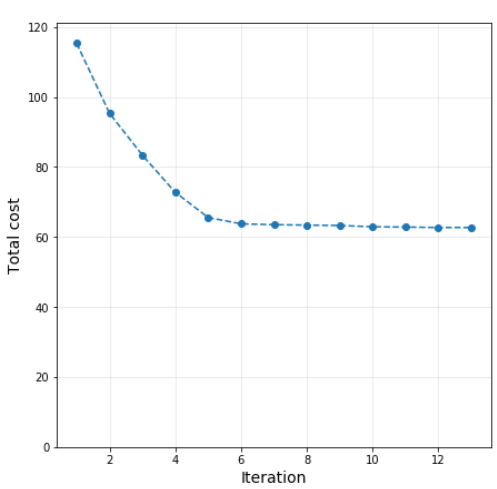
Exercise 1: *Find a seed that gives a different final quality of clusters (in terms of total distance). Include both the values of the seed, the final distance and the picture of the cluster with your answer.* *[1 mark]*

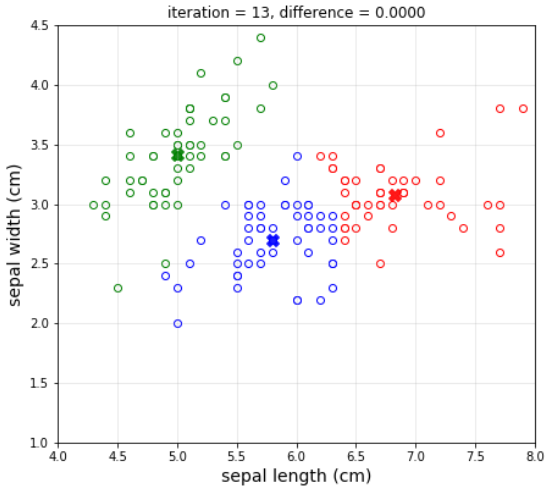
k = 3, seed = 1 at iteration = 6, distance/cost = 83.52922001837975



k = 3, seed = 3 at iteration = 13, distance/cost = 62.69987288875754

Neeraj Vashistha
190573735





Exercise 2: Has the clustering accuracy improved from before? Why? [1 mark]

Yes, the clustering accuracy has increased, from 81% to 89% and cost/distance has increased from 62 to 97 although the iterations have decreased from 14 to 8. This is because this time, the centers has been initialized with complete iris.data

```
newX = iris.data
d = len(newX[0]) # here d is 4
centers3 =  np.random.normal(size = [k, d]) + np.ones((k,1)) * np.mean(newX, axis=0)
```

As seen below, previously, these were randomly selected, and k-means clustering took place.

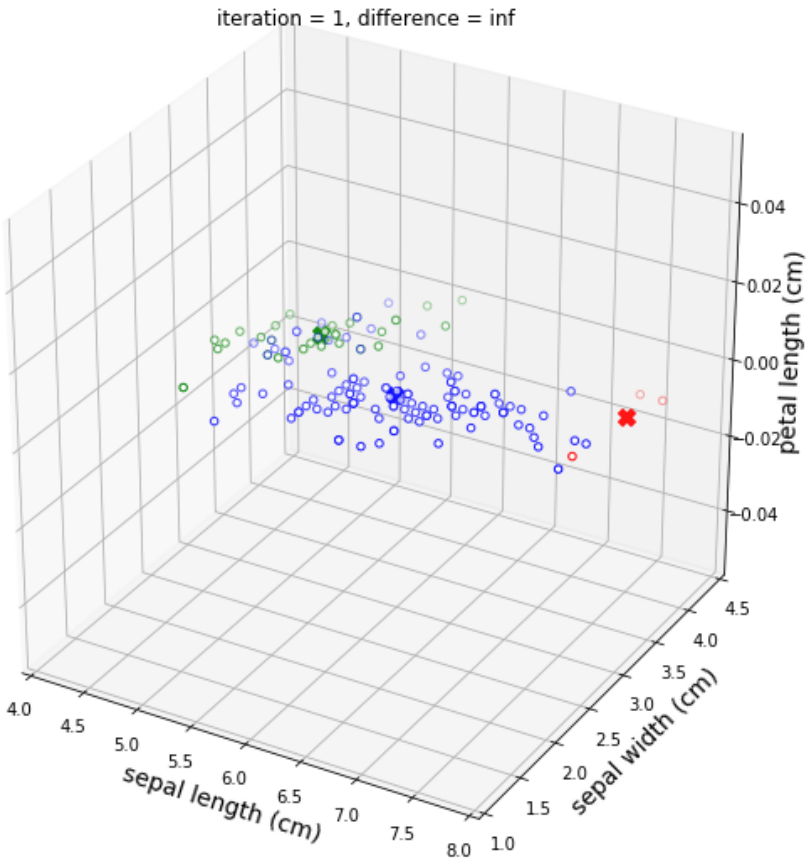
```
X = iris.data[:, :2]  # we only take the first two features.
centers3 = np.random.normal(size = [k, 2])  +  np.ones((k,1))  *  np.mean(X, axis=0)
```

Bonus: Edit the visualization to visualize three of the K-means clustering dimensions instead of just 2 [1 mark extra]

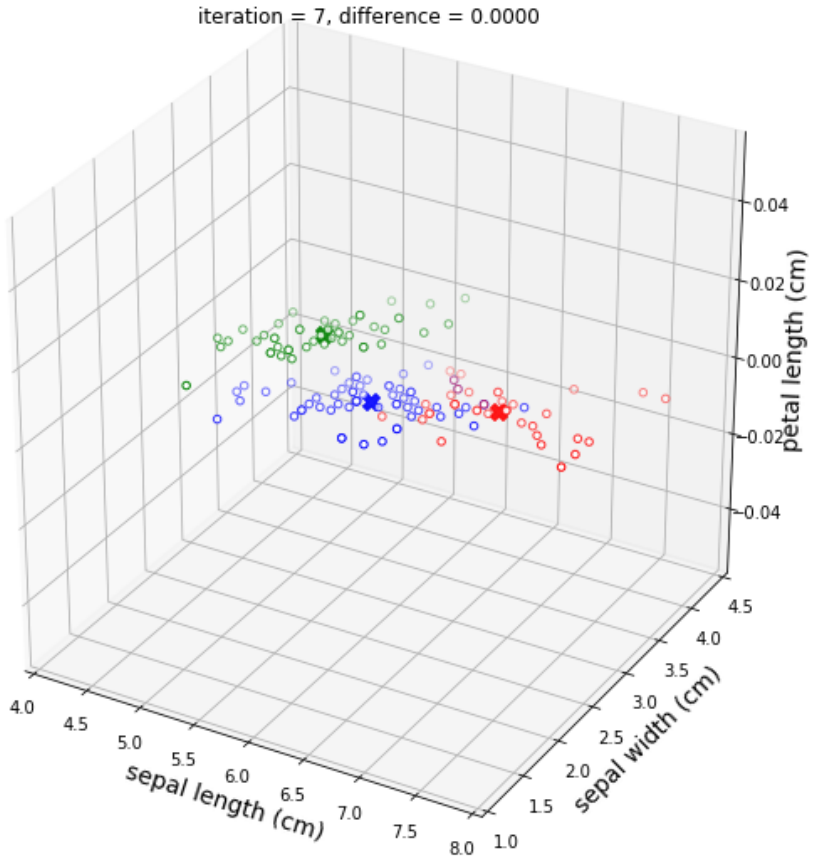
```
from mpl_toolkits.mplot3d import Axes3D
from mpl_toolkits import mplot3d
import matplotlib.pyplot as plt
newX = iris.data
d = len(newX[0])
np.random.seed(seed = 3)
k = 3 # set the k value of k-means
centers3 =  np.random.normal(size=[k, d]) + np.ones((k,1)) * np.mean(newX, axis=0)
# Find the euclidean distance between every point and every cluster.
```

```
tol = 0.000001
max_iteration = 100
difference = np.inf
iteration = 0
overalDistToClusters3 = [np.inf]
edgecolorlist = ['red', 'blue', 'green']
while difference>tol and iteration<max_iteration:
    fig = plt.figure(figsize=(10, 10))
    ax = plt.axes(projection='3d')
    # ax = fig.add_subplot(111)
    # ax.set_aspect('equal')
    ax.set_xlabel(iris.feature_names[0], fontsize=14)
    ax.set_ylabel(iris.feature_names[1], fontsize=14)
    ax.grid(alpha=0.3)
    ax.set_xlim(4, 8)
    ax.set_ylim(1, 4.5)
    overalDistToClusters3_new = 0.0
    distanceMatrix3 = distance.cdist(newX, centers3, 'euclidean')
    whichCenterNearest = np.argsort(distanceMatrix3, axis=1)[:, 0]
    for index in range(len(X)):
        overalDistToClusters3_new += distanceMatrix3[index][whichCenterNearest[index]]
    difference = overalDistToClusters3[-1] - overalDistToClusters3_new
    overalDistToClusters3.append(overalDistToClusters3_new)
    for i in range(k):
        indx = whichCenterNearest==i
        if indx.any():
            centers3[i,:] = np.mean(newX[indx,:], axis = 0)
        ax.scatter3D(newX[indx, 0], newX[indx, 1], edgecolor=edgecolorlist[i], c = 'w')
        ax.scatter3D(centers3[i, 0], centers3[i, 1], color=edgecolorlist[i], marker = 'X', s= 100,
alpha=0.9)
    iteration +=1
    ax.set_title('iteration = {:d}, difference = {:.4f}'.format(iteration, difference))
plt.show()
```

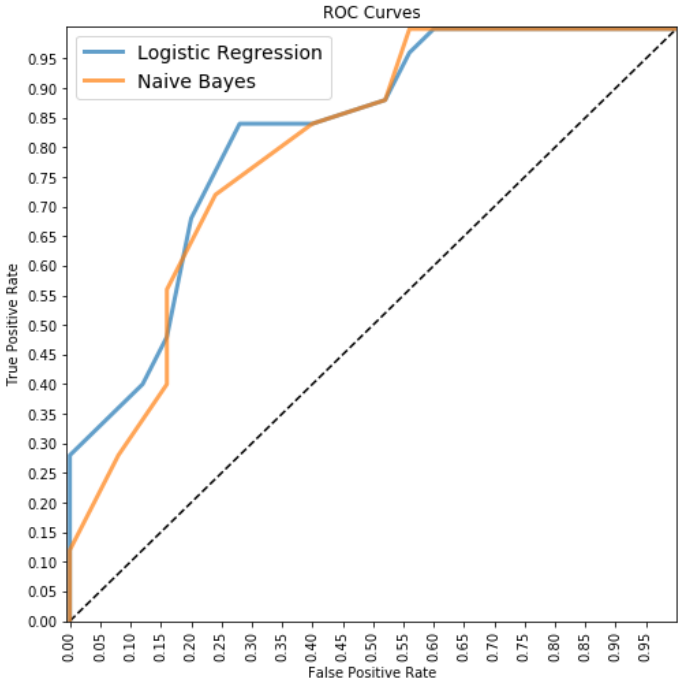
Neeraj Vashistha
190573735



Neeraj Vashistha
190573735



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



Neeraj Vashistha
190573735

Logistic regression is preferred as AUC is higher 0.82 than Naïve Baye 0.80

```
from sklearn import metrics
print(metrics.auc(FPR_LR, TPR_LR))
print(metrics.auc(FPR_NB,TPR_NB))
```

```
0.8224
0.8
```

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]*

From the above graph, we can confirm that for FPR = 0.16, we can choose either Logistic Regression or Naive Bayes as they both intersect at the same point, but we would consider a classifier whose area under the curve is maximum. Thus, we would select Logistic Regression. Given this constraint on FPR = 0.16 and taking into consideration AUC, TPR is 0.47.