Q2

March 31, 2024

[ ]:

```
# Name: Carlos Sanchez
# Student ID: 21111910
```

1 Question 2

1.1 Part 2

[ ]:

```
# Import useful libraries. Feel free to use sklearn.
from sklearn.datasets import make_blobs


# Construct a 2D toy dataset for clustering.
X, _ = make_blobs(n_samples=1000,
                  centers=[[0, 0], [1, 1], [-1, 1], [-1, -1], [1, -1]],
                  cluster_std=[0.2, 0.3, 0.3, 0.3, 0.3],
                  random_state=26)

# Conduct clustering on X using k-Means, and determine the best k with the␣
 ↪elbow method.
```

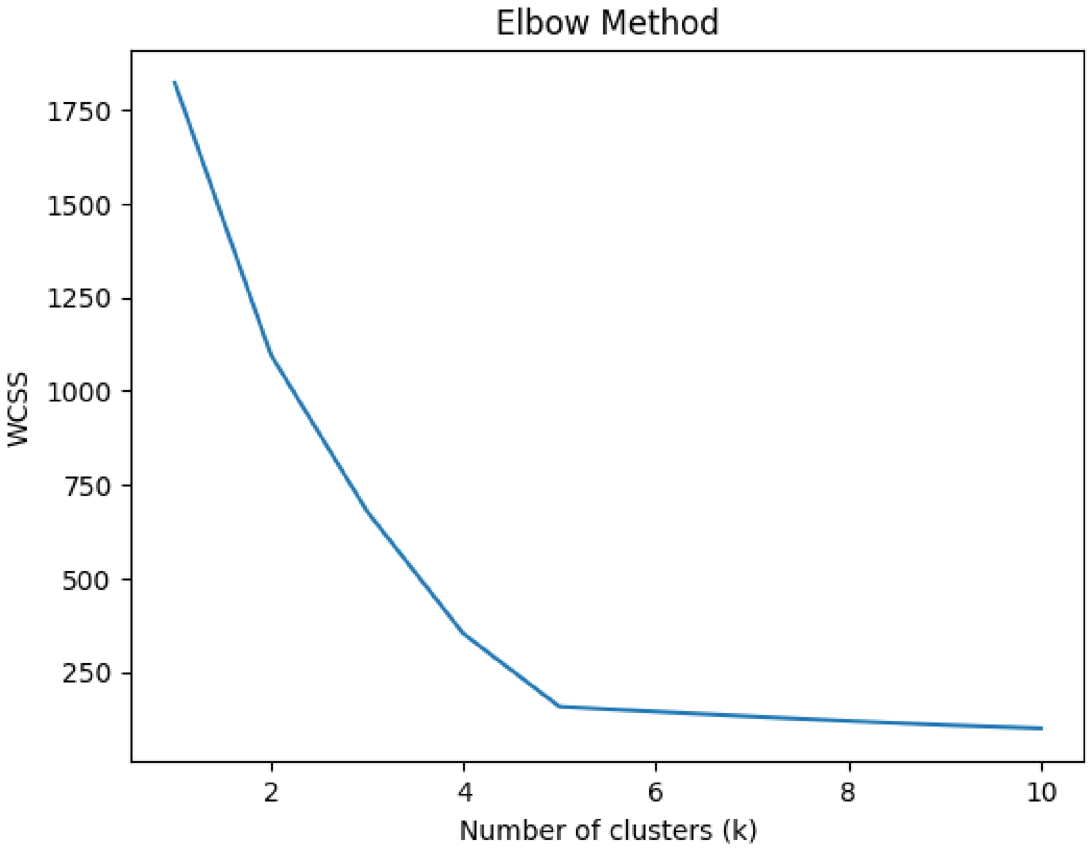
[ ]:

```
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
import numpy as np

distances = []

for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=300, n_init=10,␣
 ↪random_state=0)
    kmeans.fit(X)
    distances.append(kmeans.inertia_)

plt.plot(range(1, 11), distances)
plt.title('Elbow Method')
plt.xlabel('Number of clusters (k)')
plt.ylabel('WCSS')
plt.show()
```



As can be seen in the graph, for the first values of k, the improvement of the WCSS value calculated by the elbow method is considerable. For values of k greater than 5, the resulting value continues to decrease, but at a much slower rate than in the previous cases.

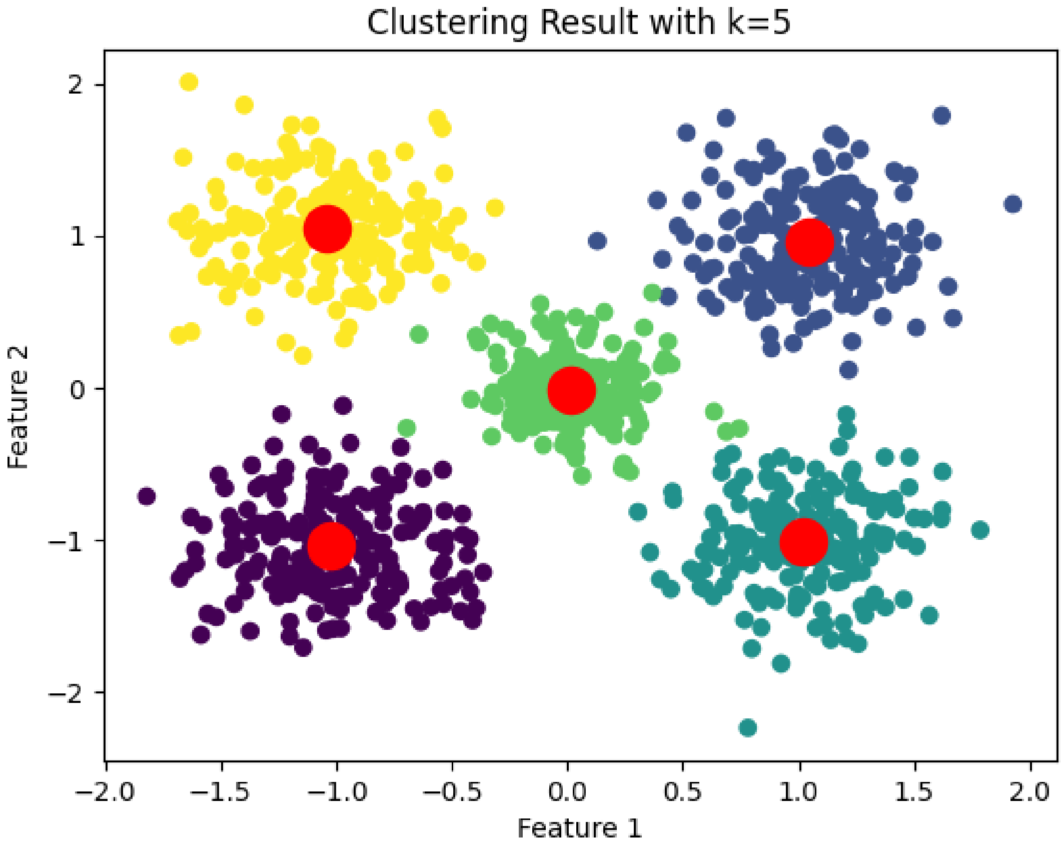
In this method, we are looking for the right point to go from a big improvement to a not so big improvement. This is because although the result continues to improve, for each larger value of k, the computational cost increases, and a balance is sought between this computational cost and a k-means result that is as accurate as possible.

Therefore, as can be seen in the graph, the best value is k=5. The values of the centroids of each cluster are shown in the following graph, with all points related with each cluster(coloured for each one with a different color):

[ ]:

```
k = 5
kmeans = KMeans(n_clusters=k, init='k-means++', max_iter=300, n_init=10,␣
 ↪random_state=0)
pred_y = kmeans.fit_predict(X)

# Plotting the clusters
plt.scatter(X[:,0], X[:,1], c=pred_y, cmap='viridis')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1],␣
 ↪s=300, c='red')
plt.title('Clustering Result with k={}'.format(k))
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.show()
```


Clustering Result with k=5
Feature 2
Feature 1