Background and Preparation

The purpose of this project is to cluster points based on proximity to each other. This was established by creating centroids, or centers of each cluster, which represent the average coordinate of datapoints within the cluster. The centroids were iteratively shifted throughout each cluster as the mean value of the clusters shifted. A cost function as used to as an indicator of distortion amongst each cluster, which are then summed for a final value gauging performance of the algorithm.

This first step was rather straightforward, as the data was in pre-generated into a numpy array. Each strategy called for something slightly different. In the first strategy, unique centroids were pre-generated, but second strategy called for me to generate my own centroids aside from the first centroid. These other clusters were expected to be at maximum distance from the first cluster and other clusters.

Strategy 1

For this first part, I used a dataframe approach instead of using numpy, but for organizational and familiarization purposes. I identified the centroids that were given and then calculated the distances between each centroid and all points, which gave me three distances values, one per centroid, per datapoint. I took the minimum of each of these distances, effectively assigning the datapoints to the centroid, therefore creating each cluster.

Next was iteratively finding better centroids. This was done by taking the average of each clusters’ datapoints. Each of these averages became the new centroid of each cluster. The process was repeated until the centroid values stopped changing. The plots below show the initial datapoints along with the initial centroids that were given, prior to any clustering. The plots with more color, in Figure 2, show the new clusters.

Chart, scatter chart

Description automatically generated Chart, scatter chart

Description automatically generated

Figure 1. Initial centroids and datapoints prior to clustering with k = 3 (left) and k = 5 (right). These datapoints are used through the project

Chart, scatter chart

Description automatically generatedChart, scatter chart

Description automatically generated

Figure 2. Clustering after new centroids were computed for k = 3 (left) and k = 5 (right)

Strategy 2

For this portion of the project, I used the data in numpy array format instead of pandas dataframe format. For each value of k given (k = 4 and k = 6), one centroid value was also given. From this, I produced all other centroids by findings datapoints with the largest distance from the first centroid and then all other centroids afterward. By establishing clusters at maximum distance from each other, a lot of work was cut out and seemed quicker overall to produce a clustering model. The clustering is shown in the figure below for both k values.

Chart, scatter chart

Description automatically generated

Chart, scatter chart

Description automatically generated

Figure 3. Clustering for k when k = 4 (left) and k = 6 (right) using K-means++ strategy

Cost Function and Centroids Values:

|  |  |  |  |
| --- | --- | --- | --- |
| K-Means | Cost Function Value | Initial Centroids | Final Centroids |
| K = 3 | 1547.53594 | [6.79251, 2.56208], [7.56399, 7.83135], [8.52789, 8.55183] | [5.47740, 2.25498], [2.56146, 6.08861],  [6.49725, 7.52297] |
| K = 5 | 616.04404 | [6.39627, 1.24125], [5.07250, 7.89834], [6.79252, 2.56208],  [4.95729, 6.90898],  [7.85356, 2.53105] | [2.68199, 2.09462], [6.77864, 8.07968], [5.22321, 4.22503], [2.87491, 7.01082], [7.55617, 2.23517] |

|  |  |  |  |
| --- | --- | --- | --- |
| K-Means++ | Cost Function Value | Initial Centroid | Final Centroids |
| K = 4 | 789.23797 | [7.59763,7.93924] | [6.62592,7.57614], [3.22853,2.52404], [2.90547,6.90512], [7.25262,2.40015] |
| K = 6 | 476.11875 | [7.33424,2.97894] | [7.41419, 2.32169], [5.46427, 6.83771],  [3.14506, 0.90770], [7.75648, 8.55668],  [3.49556, 3.56611], [2.56333, 6.97822] |

From the above tables, it can be seen that the cost function is smaller as the number of clusters increases. Below, the two graphs shown in Figure 4, represent the cost function based on the number of clusters. The cost decreases as the number of clusters increase, which is expected; however, it should be noted that the benefit of more clusters decreases after k = 6 in both graphs, as seen visually. The true value could be found by taking the derivative of the line at each k value, but because the change in slope is very visible, this wasn’t required.

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

Figure 4. Elbow plots to represent Cost Function based on Cluster Count for both strategies.