Project 2: Unsupervised Learning (K-means)

Name: Wenzhe Zheng

Date:3/27/2021

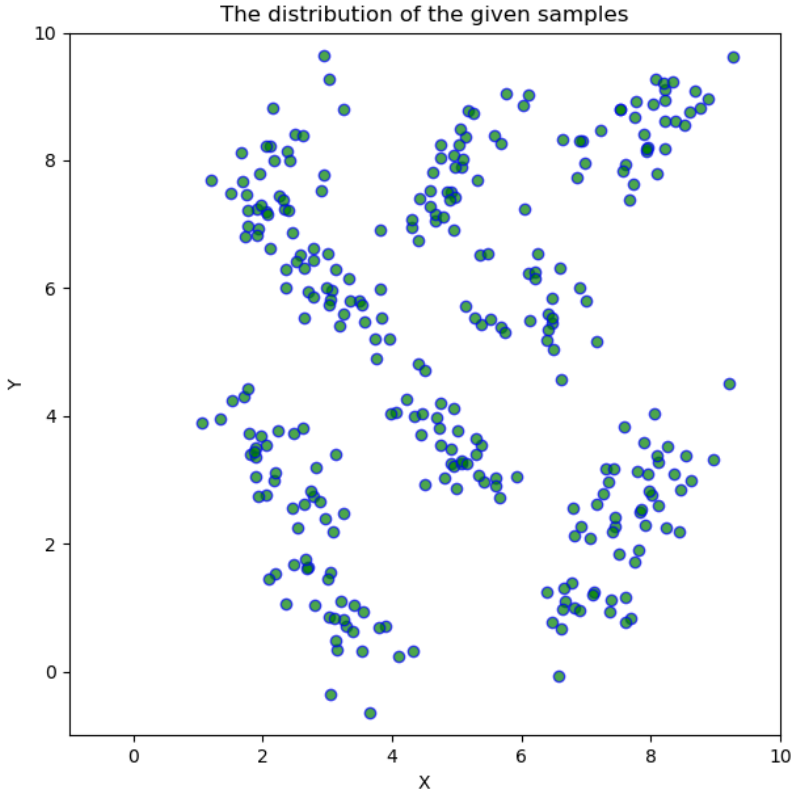
Introduction:

K-means clustering is one of the simplest unsupervised machine learning algorithms in artificial learning so far. Unsupervised algorithms use only input vectors to make inferences from the data set without reference to known or labeled results.

The main goal of this project, Part 2, is to use the K-means clustering algorithm to perform clustering on a provided set of data points containing two-dimensional points. According to the Project specifications, this project selects two different implementations of the initial cluster center using two different strategies in two ways. After learning Python, this project has greatly influenced my view on and use of artificial intelligence, especially for k-means.

Dataset:

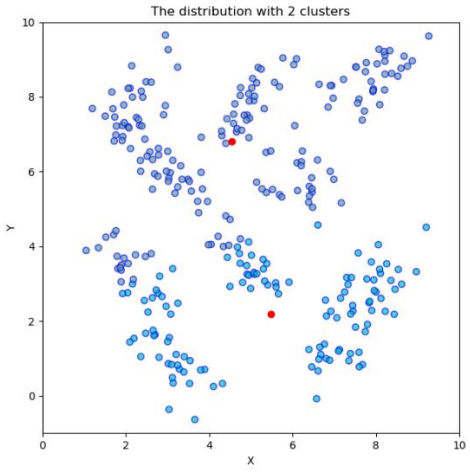
In this project, the dataset is a collection of 2-dimensional dataset.

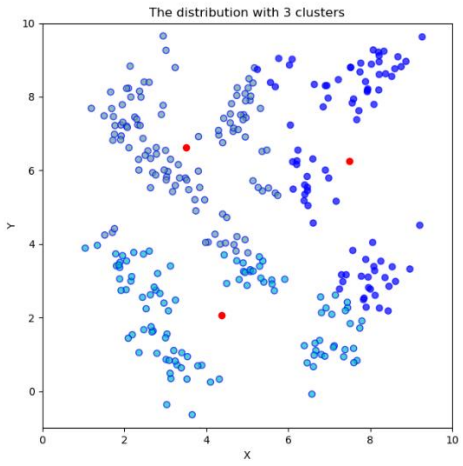


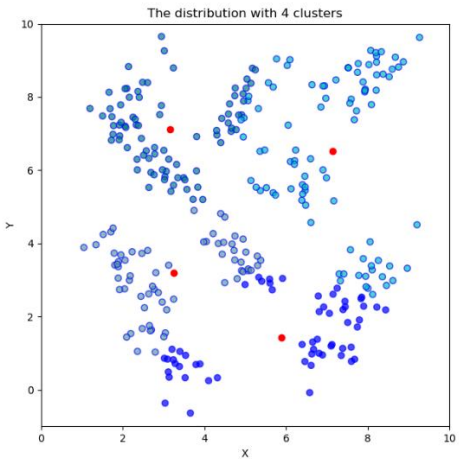
Implementation of K-means algorithm (Strategy 1: random)

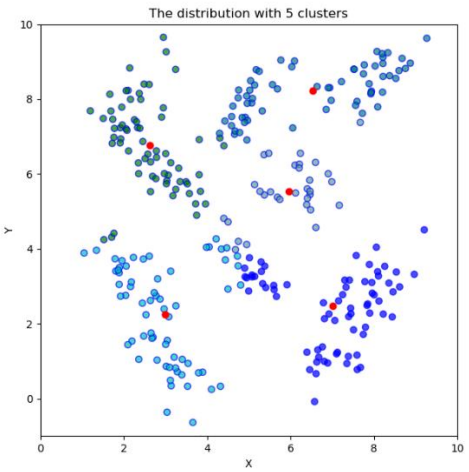
```
n = len(points[0])
# Strategy 1: randomly pick the initial centers from the given samples.
centroids = {}
for i in range(k):
    index = np.random.randint(0, n)
    centroids[i+1] = [points[0][index], points[1][index]]
```

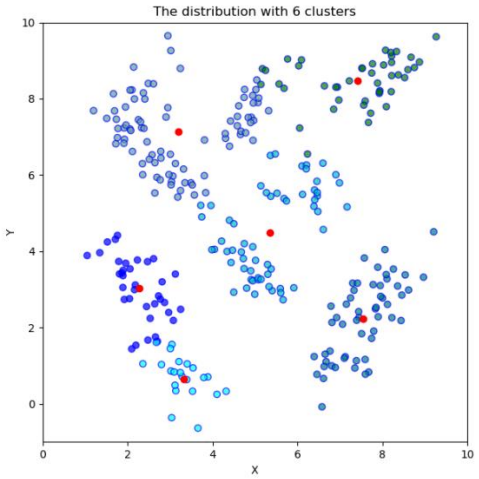
The result of the first run of Strategy 1:

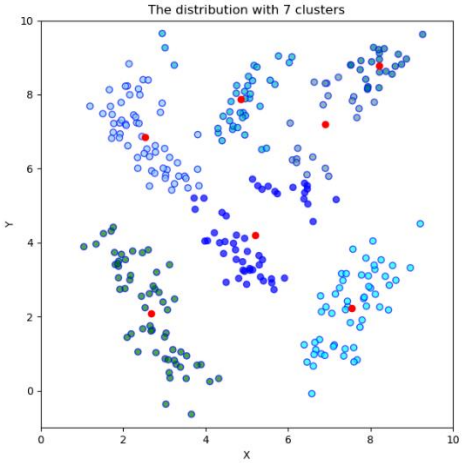


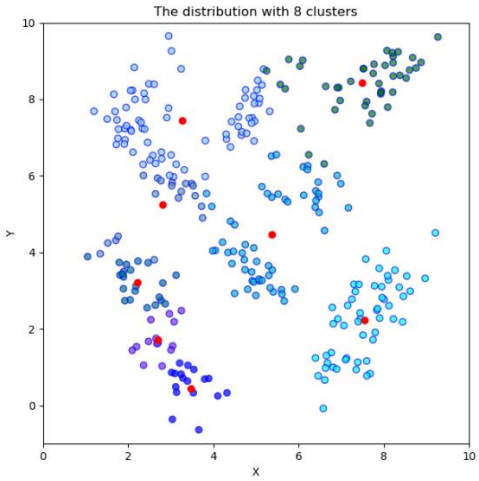


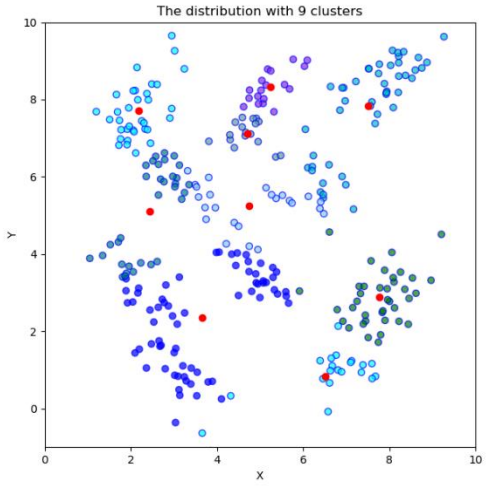


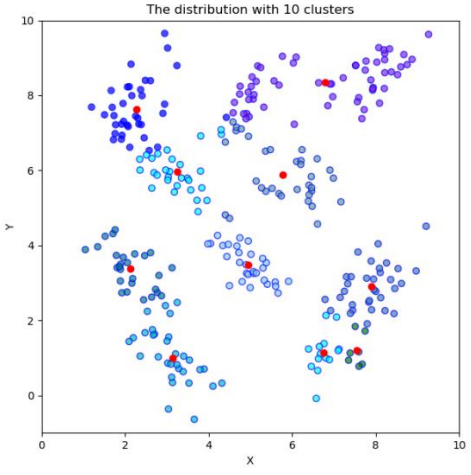


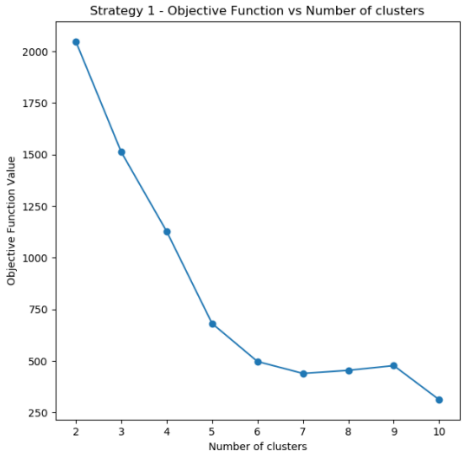

The distribution with 6 clusters
X
Y


The distribution with 7 clusters
X
Y

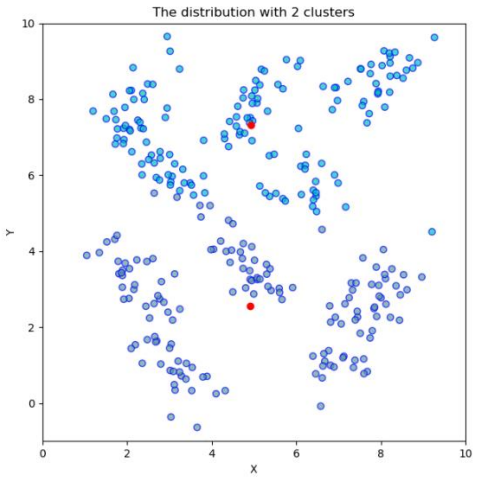

The distribution with 8 clusters
X
Y

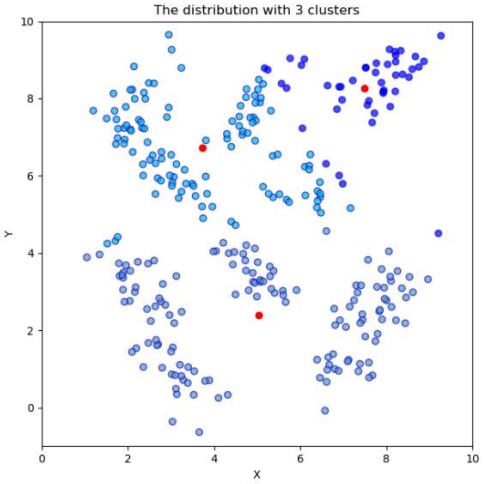

The distribution with 9 clusters
X
Y

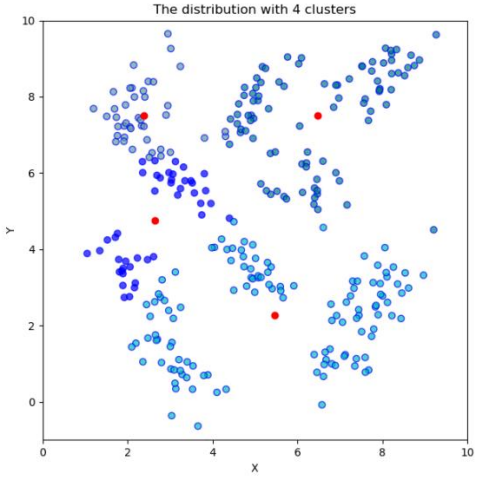

The distribution with 10 clusters
X
Y

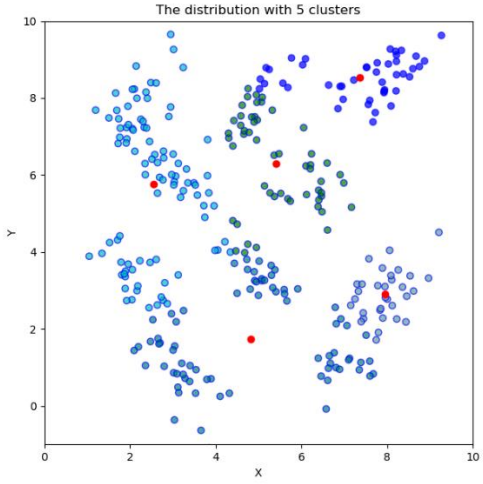

Strategy 1 - Objective Function vs Number of clusters
Number of clusters
Objective Function Value

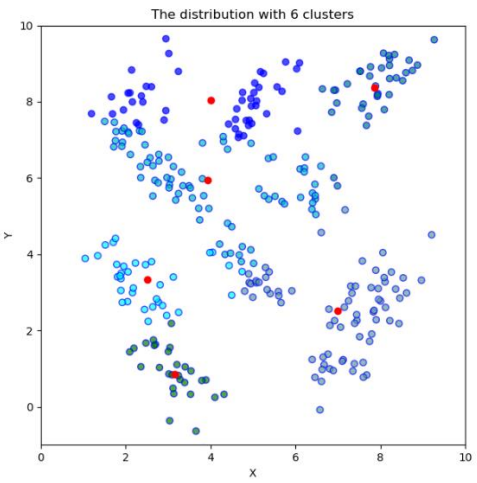
The result of the second run of Strategy 1:

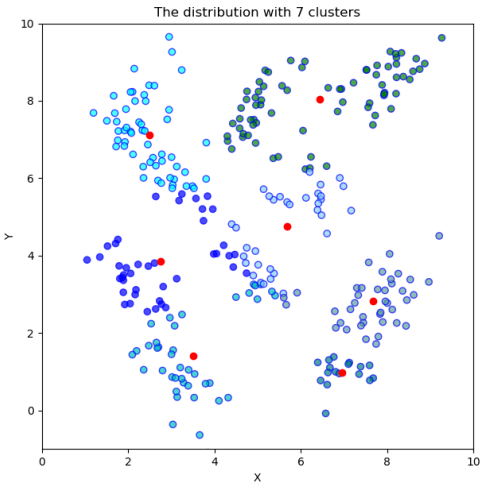


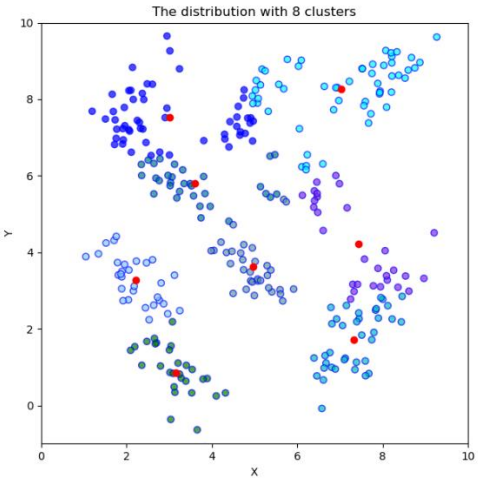


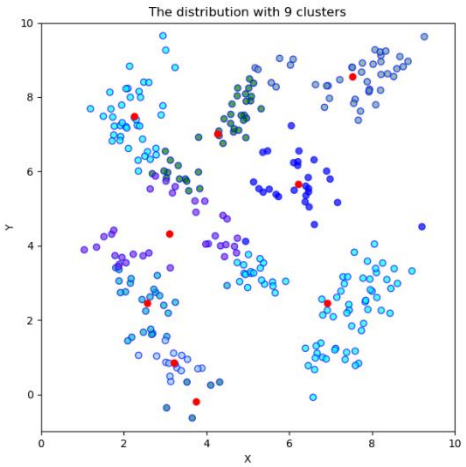


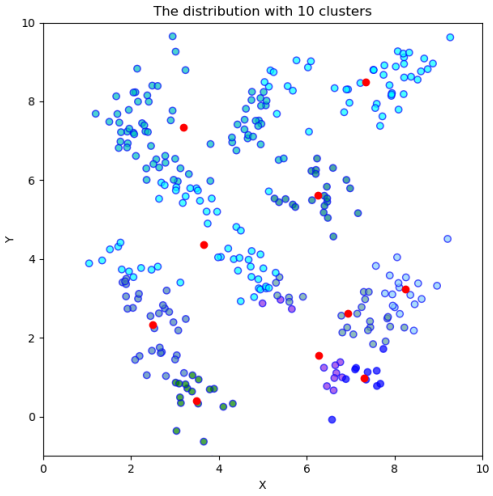


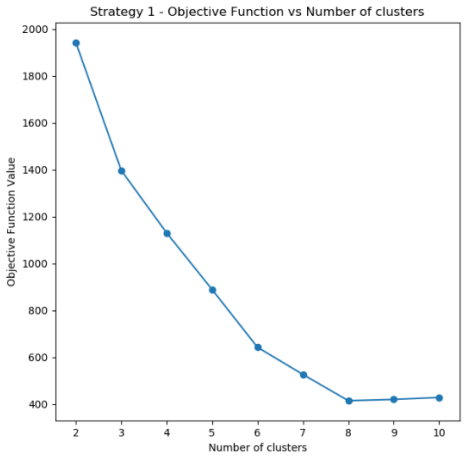

The distribution with 6 clusters
X
Y


The distribution with 7 clusters
X
Y


The distribution with 8 clusters
X
Y


The distribution with 9 clusters
X
Y

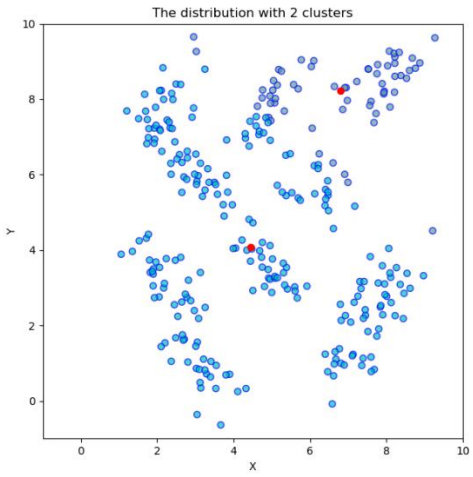

The distribution with 10 clusters
X
Y

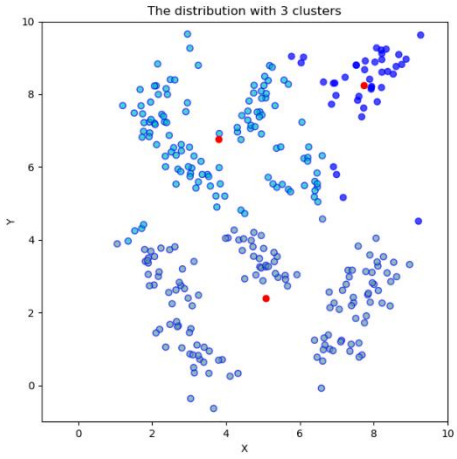

Strategy 1 - Objective Function vs Number of clusters
Objective Function Value
Number of clusters

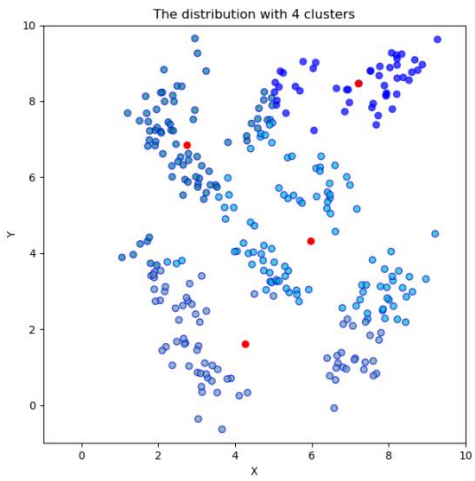
Implementation of K-means algorithm (Strategy 2)

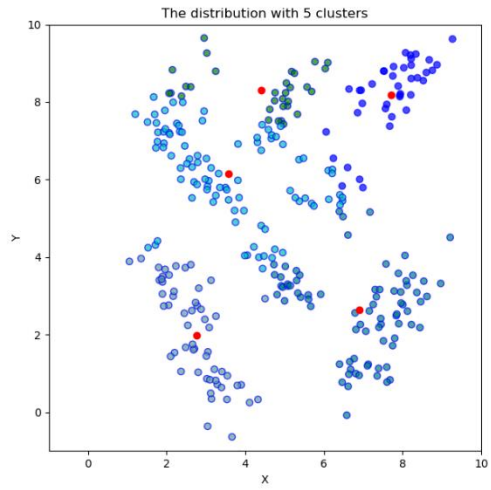
```
for i in range(0, k - 1):
    averageDistanceFromAllCentroids = {}
    for value in samples:
        # remove the selected centroids
        if list(value) in list(centroids.values()):
            continue
        distanceFromCentroids = [np.linalg.norm(value - [centroids.get(b)]) for b in centroids.keys()]
        averageDistanceFromAllCentroids[tuple(value)] = np.mean(distanceFromCentroids)
    centroids.update({i+2: list(max(averageDistanceFromAllCentroids, key=averageDistanceFromAllCentroids.get))})
```

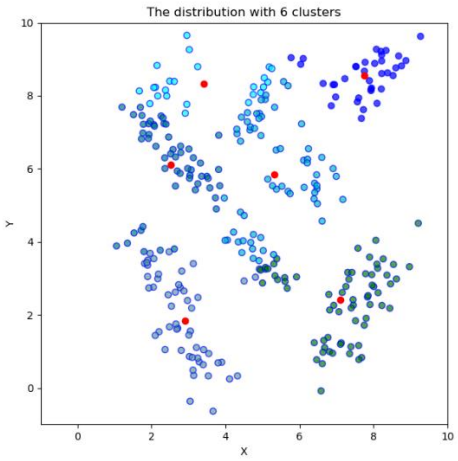
The result of the first run of Strategy 2:

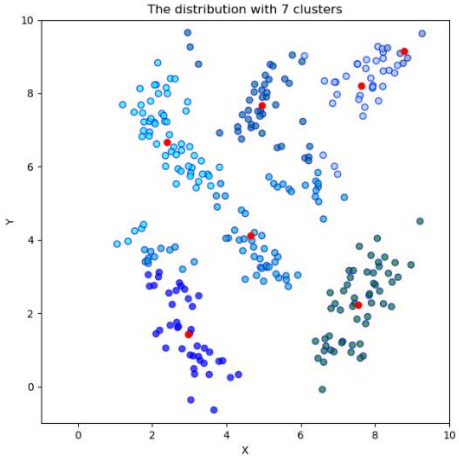


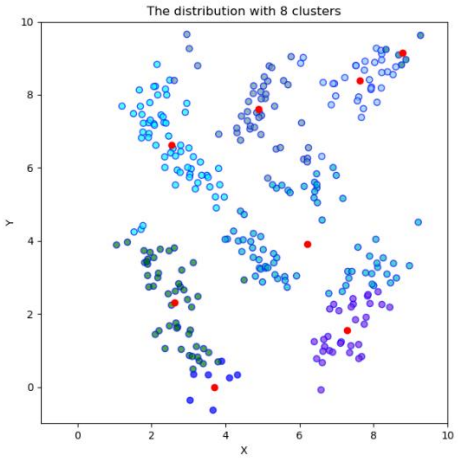


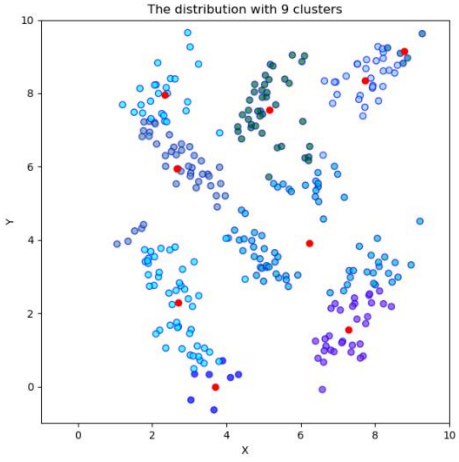


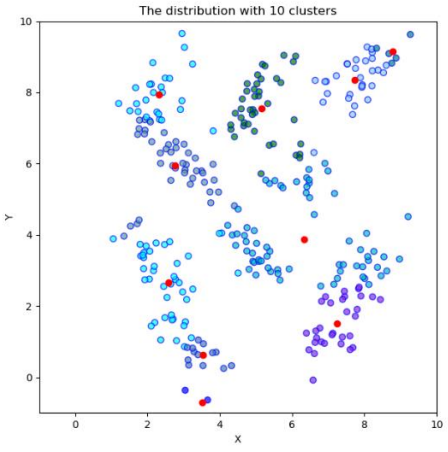


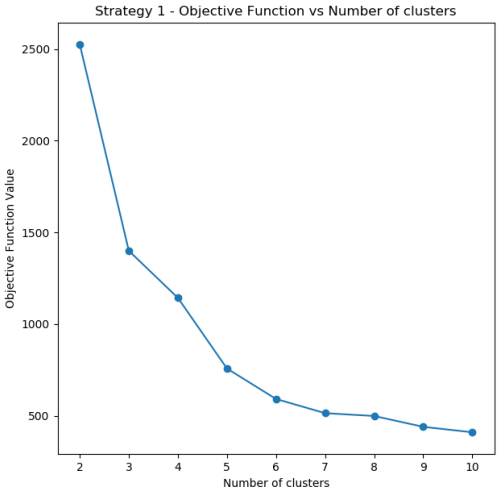

The distribution with 6 clusters
X
Y


The distribution with 7 clusters
X
Y

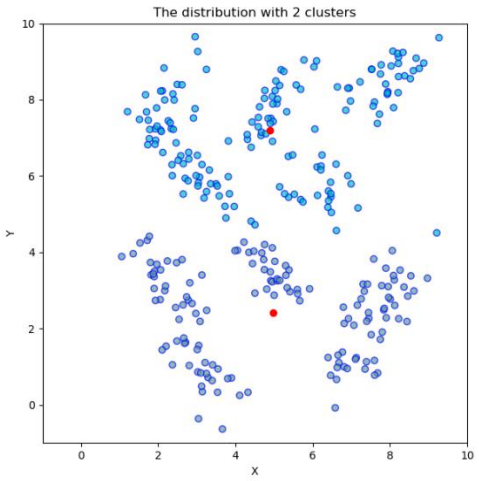

The distribution with 8 clusters
X
Y

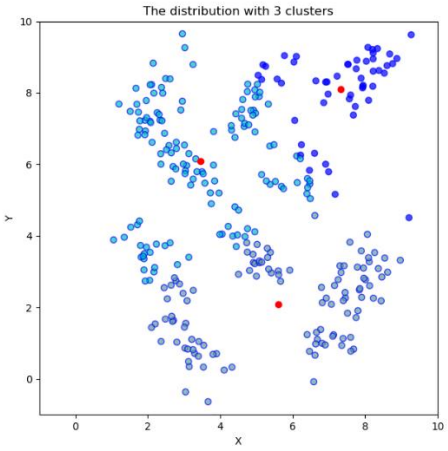

The distribution with 9 clusters
X
Y

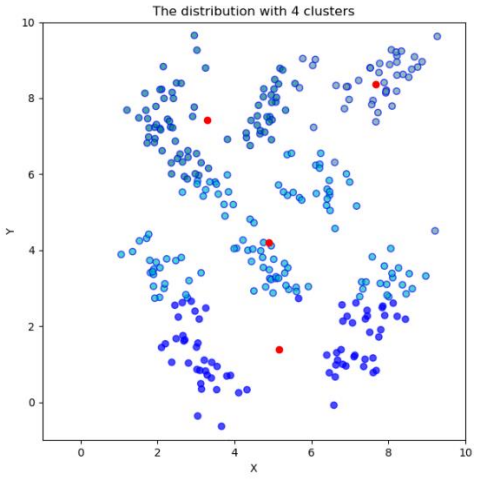

The distribution with 10 clusters
X
Y

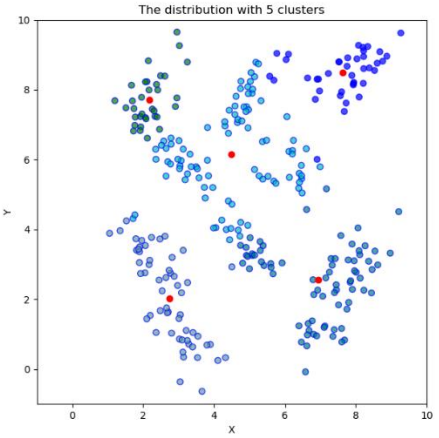

Strategy 1 - Objective Function vs Number of clusters
Objective Function Value
Number of clusters

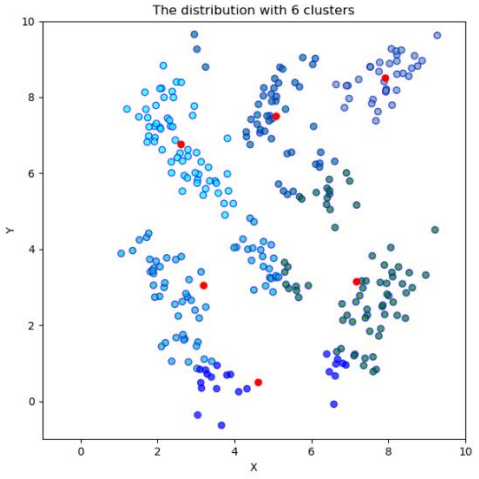
The results of the second run of Strategy 2:

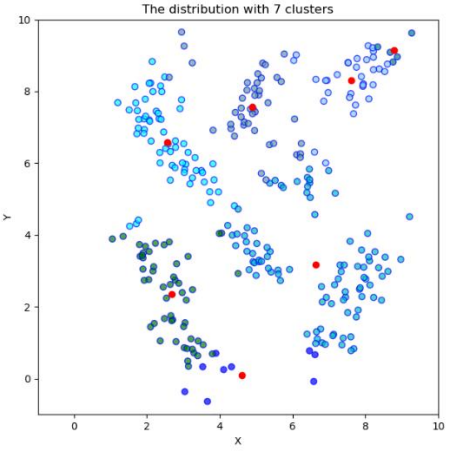


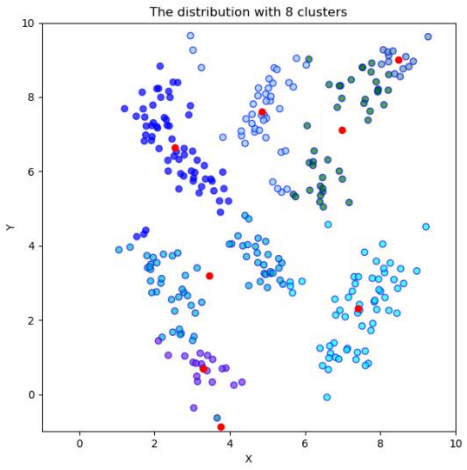


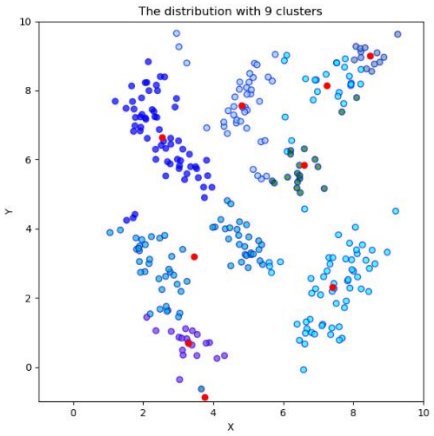


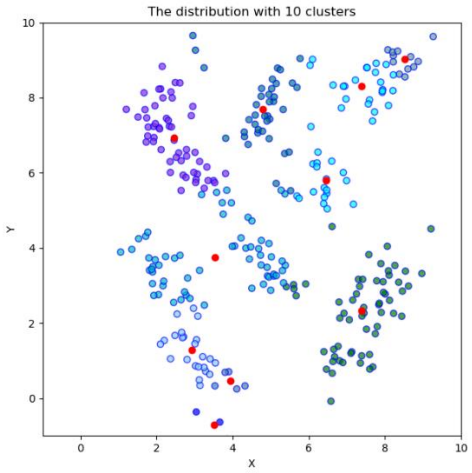


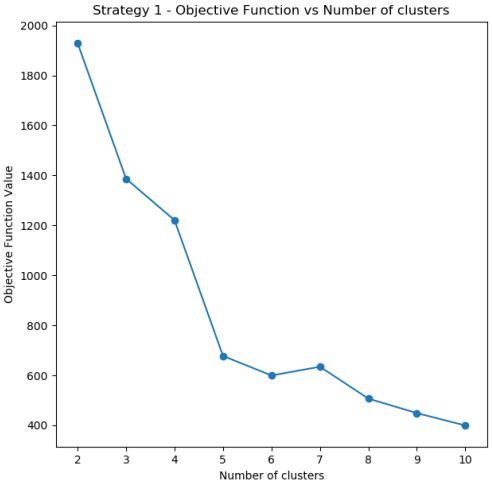












Conclusion:

So far, the result of this Project is very good.The random strategy is very simple, but the result is quite good! For learning procedure, a random initialization is a good choice.