

Week 3: Lab 3 Machine Learning Part II

2 Part B: Implementing K-Means Clustering Algorithm

Write a code for k-means clustering algorithm in the provided file. Initialize random points, and then write three major functions for KMeans clustering including *initializing random centroids*, *assign each data sample to initialized centroids*, and then *update centroids*.

Step 1: Initialize Random Points

```
import numpy as np

points = np.vstack(((np.random.randn(150, 2) * 0.75 + np.array([1, 0])),
                    (np.random.randn(50, 2) * 0.25 + np.array([-0.5, 0.5])),
                    (np.random.randn(50, 2) * 0.5 + np.array([-0.5, -0.5]))))

c = initialize_centroids(points, 3)
closest_centroid(points, c)
move_centroids(points, closest_centroid(points, c), c)
```

Step 2: Initializing random centroids

```
def initialize_centroids(points, k):
    """returns k centroids from the initial points"""
    centroids = points.copy()
    np.random.shuffle(centroids)
    return centroids[:k]
```

Step 3: Assign each data sample to initialized centroids

```
def closest_centroid(points, centroids):
    """returns an array containing the index to the nearest centroid for each
    point"""
    distances = np.sqrt(((points - centroids[:, np.newaxis]) ** 2).sum(axis=2))
    return np.argmin(distances, axis=0)
```

Step 4: Update centroids

```
def move_centroids(points, closest, centroids):
    """returns the new centroids assigned from the points closest to them"""
    return np.array([points[closest == k].mean(axis=0) for k in
                     range(centroids.shape[0])])
```