Report

Project Part 2: Unsupervised Learning (K-means)

Submitted by

Shashank Davalgi

1219510734

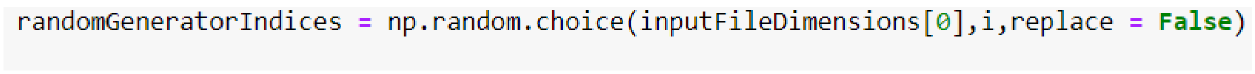
sdavalgi@asu.edu

Unsupervised Learning: K-means clustering.

Strategy 1: Randomly picking the initial centers from the sample.

Implementation:

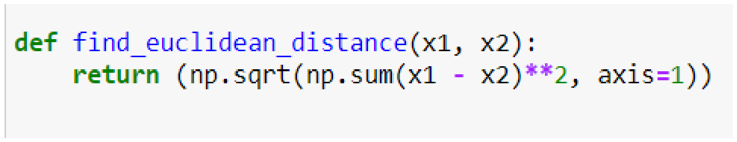
1. Iterating the logic 2 times to get the generalized result.
2. K values ranges from 2 to 10, where K means the number of clusters.
3. Below code is used to get the random data point from the given sample:



```
randomGeneratorIndices = np.random.choice(inputFileDimensions[0],i,replace = False)
```

   randomGeneratorIndices will pick k number of centers from the given data sample.

4. For each data sample, calculate the Euclidean distance with the centers generated from the previous step.
5. Euclidean distance is calculated using:



```
def find_euclidean_distance(x1, x2):
    return (np.sqrt(np.sum(x1 - x2)**2, axis=1))
```

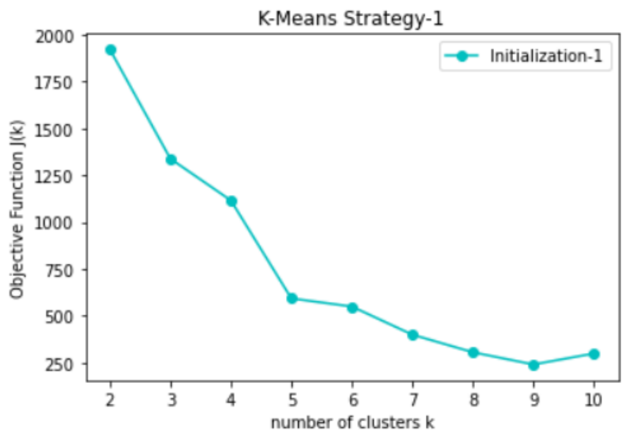
6. Classify the data points to the clusters which has the minimum Euclidean distance.
7. Calculate the average of the 2 data points to find the new centers in each cluster found from the previous step.
8. Repeat step-4 to step-7, until the new center remains the same with the previous center.

Output:

Objective function for Iteration 1:

```
Objective function is [1921.033485856205, 1338.087854201209, 1115.5344812362398, 592.5283842592463, 549.4502743616728, 399.3736
1987424936, 305.45409851538983, 240.3794892166908, 298.92660831452383]
Objective function values Initialization:1
K:[2, 3, 4, 5, 6, 7, 8, 9, 10]
J(K):[1921.033485856205, 1338.087854201209, 1115.5344812362398, 592.5283842592463, 549.4502743616728, 399.37361987424936, 305.4
5409851538983, 240.3794892166908, 298.92660831452383]
```

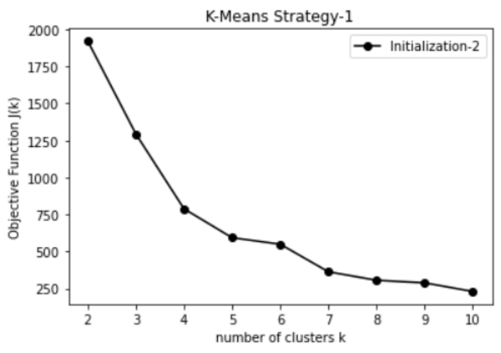
Plot for Iteration 1:



Objective function for Iteration 2:

```
Objective function is [1921.033485856206, 1293.777452391135, 788.2734352397214, 592.877929265473, 549.0022323491788, 362.933114
0450499, 305.45409851538966, 288.1028628297239, 229.72789036274077]
Objective function values Initialization:2
K:[2, 3, 4, 5, 6, 7, 8, 9, 10]
J(K):[1921.033485856206, 1293.777452391135, 788.2734352397214, 592.877929265473, 549.0022323491788, 362.9331140450499, 305.4540
9851538966, 288.1028628297239, 229.72789036274077]
```

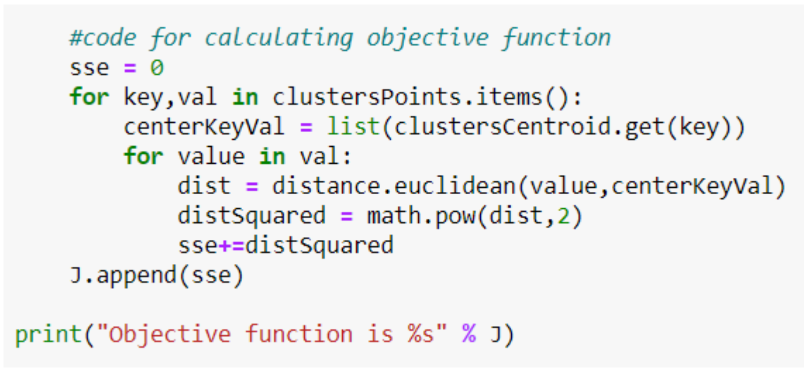
Plot for Iteration 2:



Strategy 2:

Implementation:

1. Randomly picking the first center.
2. For the i-th center, choose a sample (among all possible samples) such that the average distance of this chosen one to all previous (i-1) centers is maximal.
3. Steps 3 to 8 are similar to strategy 1.
4. Code for calculating objective function:

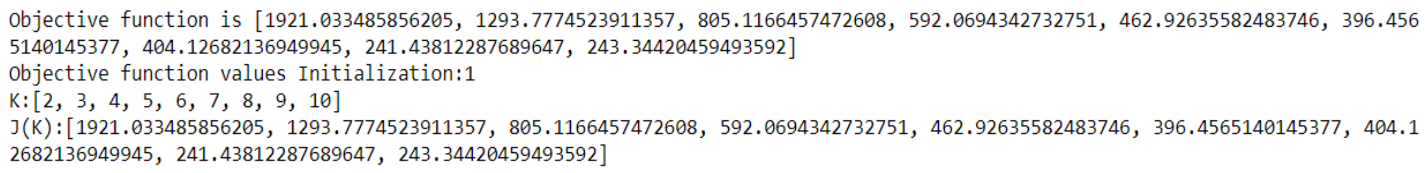


```python
    #code for calculating objective function
    sse = 0
    for key,val in clustersPoints.items():
        centerKeyVal = list(clustersCentroid.get(key))
        for value in val:
            dist = distance.euclidean(value,centerKeyVal)
            distSquared = math.pow(dist,2)
            sse+=distSquared
    J.append(sse)

print("Objective function is %s" % J)
```

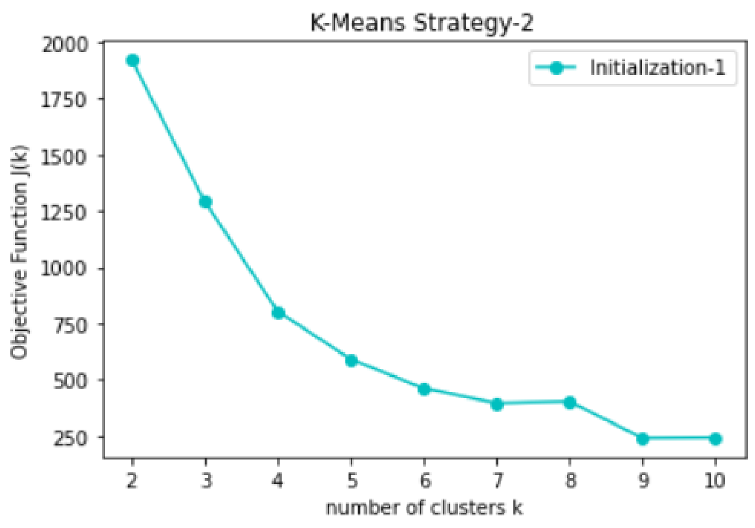
Output:

Objective function for Iteration 1:



```
Objective function is [1921.033485856205, 1293.7774523911357, 805.1166457472608, 592.0694342732751, 462.92635582483746, 396.456
5140145377, 404.12682136949945, 241.43812287689647, 243.34420459493592]
Objective function values Initialization:1
K:[2, 3, 4, 5, 6, 7, 8, 9, 10]
J(K):[1921.033485856205, 1293.7774523911357, 805.1166457472608, 592.0694342732751, 462.92635582483746, 396.4565140145377, 404.1
2682136949945, 241.43812287689647, 243.34420459493592]
```

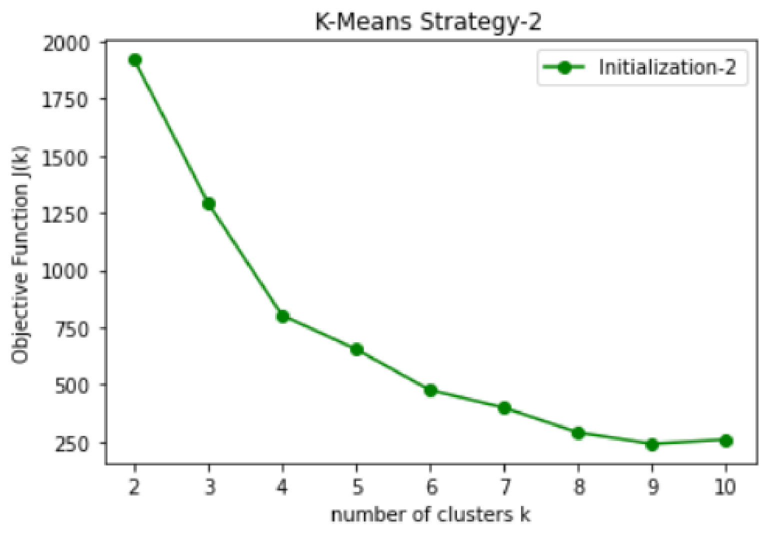
Plot for Iteration 1:



Objective function for Iteration 2:

```
Objective function is [1921.033485856205, 1294.2984174853163, 803.514506197507, 654.8779090667977, 476.29657052696626, 399.6800
1855863355, 290.92433447443744, 241.43812287689647, 260.04019829095625]
Objective function values Initialization:2
K:[2, 3, 4, 5, 6, 7, 8, 9, 10]
J(K):[1921.033485856205, 1294.2984174853163, 803.514506197507, 654.8779090667977, 476.29657052696626, 399.68001855863355, 290.9
2433447443744, 241.43812287689647, 260.04019829095625]
```

Plot for Iteration 2:


K-Means Strategy-2
Initialization-2
Objective Function J(k)
number of clusters k