**CSE572 Assignment - 2**

**Team members:**

**Manoj Kumar Yellojirao: 1215201117**

**Vishal Kumar: 1215200480**

**Ronak Maheshwari: 1215180408**

**Saswat Bastia: 1215053814**

**Abstract:**

Analyze and experiment with opioid-related drug overdose fatalities of different states of the United States of America.Given data of population and deaths for different states of the United States of America, apply K means clustering on the given 50 data points and plot the graph of objective function vs the number of clusters.Then apply the k-means algorithm on this similarity matrix over 50 rows and plot the graph of objective function vs the number of clusters.

**Keywords:** K means clustering, cosine similarity,Euclidean distance, centroid.

**Task 1:**

**Problem Statement:** Given data of population and deaths for different states of the United States of America, apply K means clustering on the given 50 data points and plot the graph of objective function vs the number of clusters.

**Implementation:**

**Step 1**: select the random k points among the given 50 data points.

**Step 2:** find the distances of all the remaining data points from the k initial data points. Assign the cluster for each data point based on the minimum distance of the k clusters.

**Step 3:** once all the data points are assigned the initial cluster, find the new cluster center based on the mean of all the data points inside the cluster, repeat the Step 2 until there is no change in the cluster or the number of iteration=500 whichever is achieved first.

**Output:**

1. Display the 50\*2 table, each row represents the states, and each column represents the features population count and the death count.
2. For K=5 display the table of 50 rows indicating the cluster each data point belongs to.
3. Code for K means clustering (.py)
4. Plot the graph of objective function vs the number of clusters.
5. **50 \* 2 table for 50 states:**

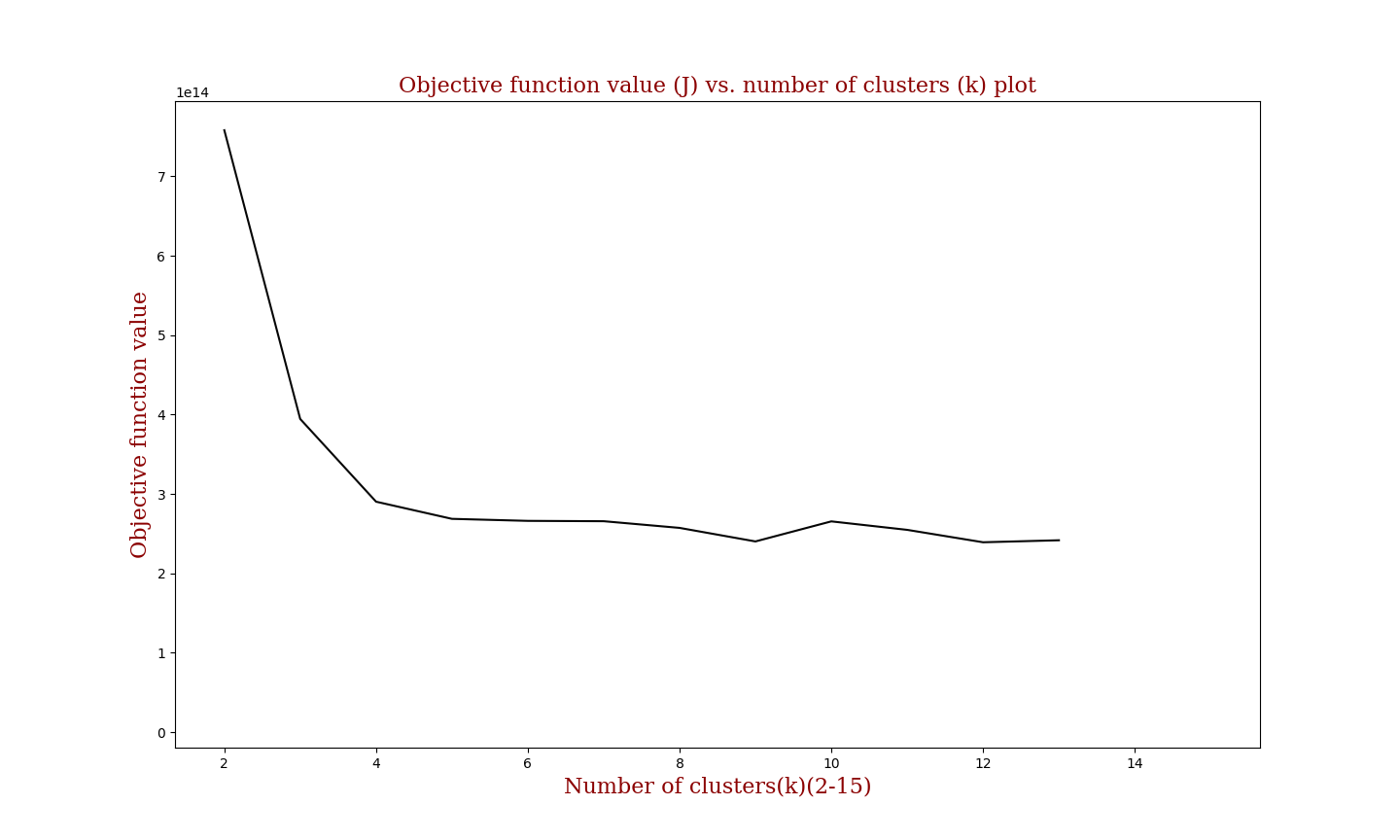
|  |  |  |
| --- | --- | --- |
|  | **Population** | **Deaths** |
| **0** | 4833722 | 723 |
| **1** | 735132 | 124 |
| **2** | 6626624 | 1211 |
| **3** | 2959373 | 356 |
| **4** | 38332521 | 4521 |
| **5** | 5268367 | 899 |
| **6** | 3596080 | 623 |
| **7** | 925749 | 189 |
| **8** | 19552860 | 2634 |
| **9** | 9992167 | 1206 |
| **10** | 1404054 | 157 |
| **11** | 1612136 | 212 |
| **12** | 12882135 | 1705 |
| **13** | 6570902 | 1172 |
| **14** | 3090416 | 264 |
| **15** | 2893957 | 332 |
| **16** | 4395295 | 1077 |
| **17** | 4625470 | 777 |
| **18** | 1328302 | 216 |
| **19** | 5928814 | 1070 |
| **20** | 6692824 | 1289 |
| **21** | 9895622 | 1762 |
| **22** | 5420380 | 517 |
| **23** | 2991207 | 336 |
| **24** | 6044171 | 1067 |
| **25** | 1015165 | 125 |
| **26** | 1868516 | 125 |
| **27** | 2790136 | 545 |
| **28** | 1323459 | 334 |
| **29** | 8899339 | 1253 |
| **30** | 2085287 | 547 |
| **31** | 19651127 | 2300 |
| **32** | 9848060 | 1358 |
| **33** | 723393 | 43 |
| **34** | 11570808 | 2744 |
| **35** | 3850568 | 777 |
| **36** | 3930065 | 522 |
| **37** | 12773801 | 2732 |
| **38** | 1051511 | 247 |
| **39** | 4774839 | 701 |
| **40** | 844877 | 63 |
| **41** | 6495978 | 1269 |
| **42** | 26448193 | 2601 |
| **43** | 2900872 | 603 |
| **44** | 626630 | 83 |
| **45** | 8260405 | 980 |
| **46** | 6971406 | 979 |
| **47** | 1854304 | 627 |
| **48** | 5742713 | 853 |
| **49** | 582658 | 109 |

**2) Cluster details for K=5:**

|  |  |
| --- | --- |
|  | **Cluster number** |
| **0** | 2 |
| **1** | 3 |
| **2** | 2 |
| **3** | 1 |
| **4** | 4 |
| **5** | 2 |
| **6** | 1 |
| **7** | 3 |
| **8** | 4 |
| **9** | 0 |
| **10** | 3 |
| **11** | 3 |
| **12** | 0 |
| **13** | 2 |
| **14** | 1 |
| **15** | 1 |
| **16** | 1 |
| **17** | 2 |
| **18** | 3 |
| **19** | 2 |
| **20** | 2 |
| **21** | 0 |
| **22** | 2 |
| **23** | 1 |
| **24** | 2 |
| **25** | 3 |
| **26** | 3 |
| **27** | 1 |
| **28** | 3 |
| **29** | 0 |
| **30** | 3 |
| **31** | 4 |
| **32** | 0 |
| **33** | 3 |
| **34** | 0 |
| **35** | 1 |
| **36** | 1 |
| **37** | 0 |
| **38** | 3 |
| **39** | 2 |
| **40** | 3 |
| **41** | 2 |
| **42** | 4 |
| **43** | 1 |
| **44** | 3 |
| **45** | 0 |
| **46** | 2 |
| **47** | 3 |
| **48** | 2 |
| **49** | 3 |

**3) code is attached in the Zip:**

**4)Objective function(J) vs Number of Clusters.**

****

**Task 2:**

**Problem Statement:** Given data of population and deaths for different states of the United States of America,construct a similarity matrix representing the closeness of state pairs with respect to their Population and Death values using cosine similarity metric. Then apply the k-means algorithm on this similarity matrix over 50 rows and plot the graph of objective function vs the number of clusters.

**Implementation:**

**Step 1**: Create vectors represented by coordinates for every state that represents population and death values.

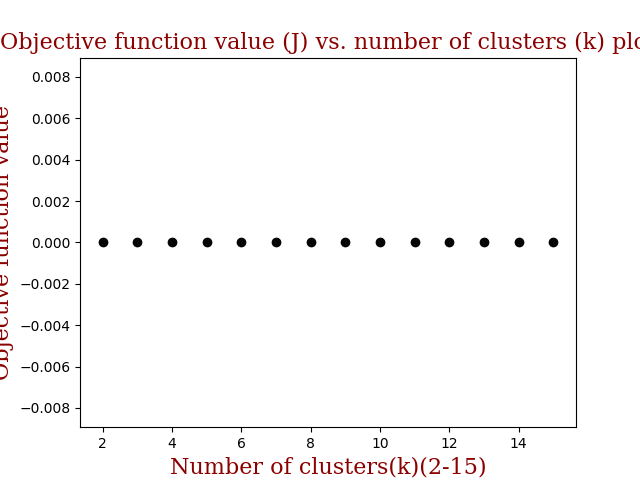
**Step 2:** Find the cosine similarity for all the states against every other state and then create a similarity matrix (50x50).

**Step 3:** Run k-means algorithm similar to task 1 on the similarity matrix. Thenassign the cluster for each data point based on the minimum distance of the k clusters.

**Step 4:** once all the data points are assigned the initial cluster, find the new cluster center based on the mean of all the data points inside the cluster, repeat the Step 2 until there is no change in the cluster or the number of iteration=500 whichever is achieved first.

**Output:**

1. Display the 50x50 similarity matrix table, each row represents the states and its 50 features.
2. For K=5 display the table of 50 rows indicating the cluster each data point belongs to.
3. Code for K means clustering (.py)
4. Plot the graph of objective function vs the number of clusters.
5. **Attached the 50\*50 csv file in the zip**
6. **Objective function(J) vs Number of Clusters.**

****

**Task 3:**

For this given data **Cosine similarity metric will work better**, the explanation is as follows:

For this dataset, the range of feature values is not similar for different samples. I.e. for one same a feature value is in terms of thousands and for some other sample that same feature is in tens place only, in simple words features are not normalized and in such case when we calculate the Euclidean distance between two samples the distance value will be dominated by the sample which has large feature values and outcomes is not a better representation of similarity considering all different samples. Thus, looking for the directions in which these different samples points in the given feature space is a better option and thus looking for the difference between these directions i.e. the **Cosine distance** is better than the Euclidean distance.