ASSIGNMENT No: 12

Title: Perform the data clustering algorithm using any Clustering algorithm

Problem Statement: Implement Page Rank Algorithm. (Use python or beautiful soup for implementation).

Prerequisite:

Basics of Python

Software Requirements: Jupyter

Hardware Requirements:

PIV, 2GB RAM, 500 GB HDD

Learning Objectives:

Learn to Perform the data clustering algorithm using any Clustering algorithm

Outcomes:

After completion of this assignment students are able to understand how to Perform the data clustering algorithm using any Clustering algorithm

Theory:

Clustering your data can provide a new way to slice that is based on the properties of the data instead of other labels. For instance, customer data is often sliced by demographic parameters like gender, age, location, etc. This data can be useful in many cases, but what if you could slice your customers by their behaviour? What they buy, how often, how much they spend, etc. This information can help with advertising because you are now looking at past behaviour that can correlate better with future actions than demographics.

k-Mean Clustering

From the results of my testing, I believe the algorithm responsible for clustering in Power BI is the k-means algorithm. I did not find any confirmation on this, but it seems reasonable given the results found below. Knowing this can help you understand how Power BI finds clusters and how it will work in the situation you are using.

The goal of k-means is to minimize the distance between the points of each cluster. Each cluster has a centre. Data points are labeled as part of a cluster depending on which centre they are closest to.

As a result, certain types of clusters are easy to find, and in others, the algorithm will fail. Below, you will see examples of both cases.

```
R Console

> # Apply K mean to iris and store result
> newiris <- iris
> newiris$Species <- NULL
> (kc <- kmeans(newiris,3))
K-means clustering with 3 clusters of sizes 62, 38, 50

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     5.901613    2.748387     4.393548    1.433871
2     6.850000    3.073684     5.742105    2.071053
3     5.006000    3.428000     1.462000    0.246000

Clustering vector:
  [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [47] 3 3 3 3 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [93] 1 1 1 1 1 1 1 1 2 1 2 2 2 2 1 2 2 2 2 2 2 1 1 2 2 2 2 1 2 1 2 1 2 2 1 1 2 2 2 2 2 1 2 2 2 2
[139] 1 2 2 2 1 2 2 2 1 2 2 1

Within cluster sum of squares by cluster:
[1] 39.82097 23.87947 15.15100
 (between_SS / total_SS =  88.4 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"
[7] "size"         "iter"         "ifault"
> |
```

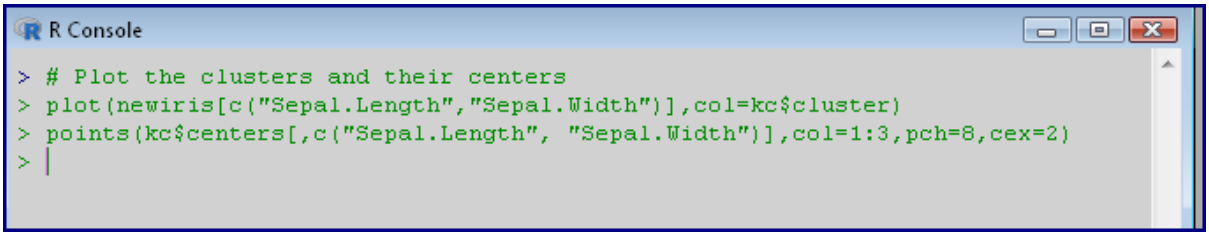
Compare the Species label with the clustering result

```
R Console

> #Compare the Species label with the clustering result
> table (iris$Species,kc$cluster)

               1  2  3
  setosa       0  0 50
  versicolor  48  2  0
  virginica   14 36  0
> |
```

Plot the clusters and their centre



Conclusion:- Thus, this way Performed data clustering algorithm using any Clustering algorithm