ASSIGNMENT :

Implementation of k-means clustering.

PROBLEM STATEMENT :

Implement k-means for clustering data of children belonging to different age groups to perform some specific activities. Formulate the Feature vector for following parameters:
i. height
ii. weight
iii. age
iv. IQ
Formulate the data for 40 children to form 3 clusters.

OBJECTIVE:

• To develop problem solving abilities for gamifications .
• To apply algorithmic strategies while solving problems
• To develop time and space efficient algorithms
• To study algorithmic examples in distributed, concurrent and parallel environments

THEORY :

Cluster analysis or simply clustering is the process of partitioning a set of data objects (or observations) into subsets. Each subset is a cluster, such that objects in a cluster are similar to one another, yet dissimilar to objects in other clusters. The set of clusters resulting from a cluster analysis can be referred to as a clustering. In this context, different clustering methods may generate different clusterings on the same data set. The partitioning is not performed by humans, but by the clustering algorithm. Hence, clustering is useful in that it can lead to the discovery of previously unknown groups within the data.

k-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed apriori. k-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster.

Suppose a data set, D, contains n objects in Euclidean space. Partitioning methods distribute the objects in D into k clusters, $C_1, \ldots, C_k$, that is, $C_i \subset D$ and $C_i \cap C_j = \emptyset$ for $(1 \le i, j \le k)$. An objective function is used to assess the partitioning quality so that objects within a cluster are similar to one another but dissimilar to objects in other clusters. This is, the objective function aims for high intracluster similarity and low intercluster similarity. A centroid-based partitioning technique uses the centroid of a cluster, C_i, to represent that cluster. Conceptually, the centroid of a cluster is its center point. The centroid can be defined in various ways such as by the mean or medoid of the objects (or points) assigned to the cluster. The difference between an object $p \in C_i$ and c_i, the representative of the cluster, is measured by dist(p, c_i), where dist(x, y) is the Euclidean distance between two points x and y. The quality of cluster C_i can be measured by the within cluster variation, which is the sum of squared error between all objects in C_i and the centroid c_i, defined as

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} dist(\boldsymbol{p}, \boldsymbol{c_i})^2,$$

The k-means algorithm defines the centroid of a cluster as the mean value of the points within the cluster. It proceeds as follows. First,it randomly selects k of the objects in D, each of which initially represents a cluster mean or center. For each of the remaining objects, an object is assigned to the cluster to which it is the most similar, based on the Euclidean distance between the object and the cluster mean. The k-means algorithm then iteratively improves the within-cluster variation.
For each cluster, it computes the new mean using the objects assigned to the cluster in the previous iteration. All the objects are then reassigned using the updated means as the new cluster centers. The iterations continue until the assignment is stable, that is, the clusters formed in the current round are the same as those formed in the previous round.

ALGORITHM:

The k-means algorithm for partitioning, where each cluster's center is represented by the mean value of the objects in the cluster.

```
(1) arbitrarily choose k objects from D as the initial cluster centers;
(2) repeat
(3)     (re)assign each object to the cluster to which the object is the most similar,
            based on the mean value of the objects in the cluster;
(4)     update the cluster means, that is, calculate the mean value of the objects for
            each cluster;
(5) until no change;
```

PSEUDO CODE:

```
def kmeans(dataSet, k):
    # Initialize centroids randomly
    numFeatures = dataSet.getNumFeatures()
    centroids = getRandomCentroids(numFeatures, k)

    # Initialize book keeping vars.
    iterations = 0
    oldCentroids = None

    # Run the main k-means algorithm
    while not shouldStop(oldCentroids, centroids, iterations):
        # Save old centroids for convergence test. Book keeping.
        oldCentroids = centroids
        iterations += 1

        # Assign labels to each datapoint based on centroids
        labels = getLabels(dataSet, centroids)

        # Assign centroids based on datapoint labels
        centroids = getCentroids(dataSet, labels, k)

   return centroids
```

INPUT : k: the number of clusters,

D: a data set containing n objects.

OUTPUT : A set of k clusters.

MATHEMATICAL MODEL :

Let S be the solution perspective .

S={s, e, i, o, f, DD, NDD, success, failure}

s = { Initial state of the 8-tile puzzel consisting of all tiles at particular places }

I = Input of the system → { I1, I2 }
where I1 = { no of clusters k}
I2 = { data set of 'n' objects }

o = Output of the system → { O1 }
where O1 = { k clusters }

f = Functions used → { f1}
where f1 = { k-means algorithm }

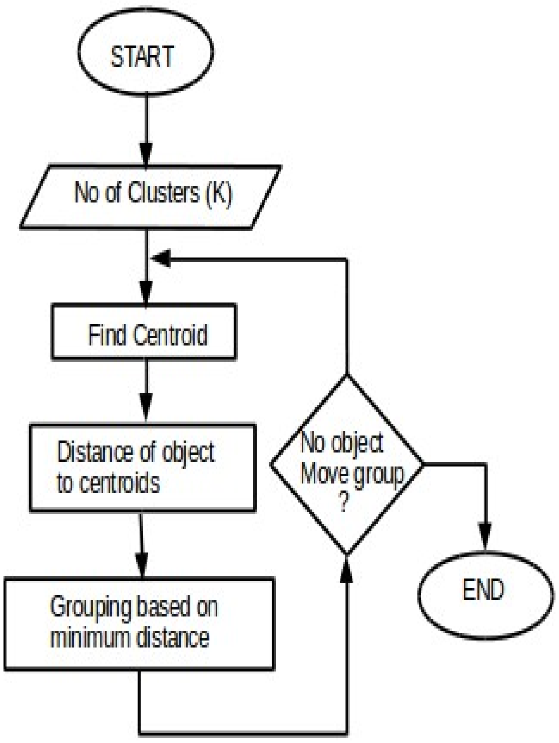
DD - Deterministic data → { }

NDD - Non deterministic data → { Cluster objects, size of the cluster}

Success - Desired outcome generated → { k clusters }

Failure - Desired outcome not generated or forced exit due to system error.

FLOWCHART :


START
No of Clusters (K)
Find Centroid
Distance of object to centroids
Grouping based on minimum distance
No object Move group ?
END

TEST CASE :

```
rohan@rohan-HP-ProBook-4410s:~/Dropbox/7_SEMESTER$ python k-means-new.py
Initial Centroid 1 :	[172, 45, 20, 105]
Initial Centroid 2 :	[180, 65, 23, 160]
Initial Centroid 3 :	[189, 69, 23, 143]

CLUSTER 1 : [[172, 160, 162, 128, 102, 103, 105, 120, 125, 134, 127, 129, 123, 124, 134, 200, 168, 165, 167, 160, 157, 159, 155, 139, 147, 178], [45, 73, 52, 98, 76, 74, 72, 79, 80, 82, 67, 65, 40, 45, 100, 89, 95, 73, 75, 79, 55, 56, 57, 58, 54, 53], [20, 12, 19, 19, 12, 11, 10, 10, 9, 8, 23, 34, 35, 56, 28, 29, 20, 27, 31, 30, 33, 42, 43, 47, 49, 65], [105, 123, 112, 108, 120, 110, 111, 112, 115, 114, 117, 119, 141, 132, 90, 89, 88, 100, 102, 109, 108, 123, 124, 126, 122, 121]]

CLUSTER 2 : [[175, 180, 145], [48, 65, 95], [21, 23, 17], [156, 160, 145]]

CLUSTER 3 : [[189, 169, 197, 172, 174, 172, 176, 190, 173, 165, 179], [69, 89, 67, 63, 71, 87, 88, 49, 50, 102, 60], [23, 21, 18, 35, 40, 45, 32, 78, 80, 25, 62], [143, 119, 147, 123, 124, 127, 140, 133, 134, 137, 130]]
```

```
--

New Centroid 1 :        [143, 68, 27, 113]
New Centroid 2 :        [166, 69, 20, 153]
New Centroid 3 :        [177, 72, 41, 132]

CLUSTER 1 : [[172, 160, 162, 128, 102, 103, 105, 120, 125, 134, 127, 129, 123, 124, 134, 168,
165, 167, 160, 157, 155, 139, 147], [45, 73, 52, 98, 76, 74, 72, 79, 80, 82, 67, 65, 40, 45, 100, 95,
73, 75, 79, 55, 57, 58, 54], [20, 12, 19, 19, 12, 11, 10, 10, 9, 8, 23, 34, 35, 56, 28, 20, 27, 31, 30, 33,
43, 47, 49], [105, 123, 112, 108, 120, 110, 111, 112, 115, 114, 117, 119, 141, 132, 90, 88, 100, 102,
109, 108, 124, 126, 122]]

CLUSTER 2 : [[175, 180, 145, 197], [48, 65, 95, 67], [21, 23, 17, 18], [156, 160, 145, 147]]

CLUSTER 3 : [[189, 169, 172, 174, 172, 176, 190, 173, 165, 200, 159, 178, 179], [69, 89, 63, 71,
87, 88, 49, 50, 102, 89, 56, 53, 60], [23, 21, 35, 40, 45, 32, 78, 80, 25, 29, 42, 65, 62], [143, 119,
123, 124, 127, 140, 133, 134, 137, 89, 123, 121, 130]]
------------------------------------------------------------------------------------------------------------------
--

New Centroid 1 :        [139, 69, 25, 113]
New Centroid 2 :        [174, 68, 19, 152]
New Centroid 3 :        [176, 71, 44, 126]

CLUSTER 1 : [[160, 162, 128, 102, 103, 105, 120, 125, 134, 127, 129, 123, 124, 134, 168, 165,
160, 157, 139, 147], [73, 52, 98, 76, 74, 72, 79, 80, 82, 67, 65, 40, 45, 100, 95, 73, 79, 55, 58, 54],
[12, 19, 19, 12, 11, 10, 10, 9, 8, 23, 34, 35, 56, 28, 20, 27, 30, 33, 47, 49], [123, 112, 108, 120, 110,
111, 112, 115, 114, 117, 119, 141, 132, 90, 88, 100, 109, 108, 126, 122]]

CLUSTER 2 : [[175, 180, 189, 145, 197, 165], [48, 65, 69, 95, 67, 102], [21, 23, 23, 17, 18, 25],
[156, 160, 143, 145, 147, 137]]

CLUSTER 3 : [[172, 169, 172, 174, 172, 176, 190, 173, 200, 167, 159, 155, 178, 179], [45, 89, 63,
71, 87, 88, 49, 50, 89, 75, 56, 57, 53, 60], [20, 21, 35, 40, 45, 32, 78, 80, 29, 31, 42, 43, 65, 62],
[105, 119, 123, 124, 127, 140, 133, 134, 89, 102, 123, 124, 121, 130]]
```

```
--------------------------------------------------------------------------------------------------------------
-

New Centroid 1 :        [135, 70, 24, 113]
New Centroid 2 :        [175, 74, 21, 148]
New Centroid 3 :        [174, 66, 44, 121]

CLUSTER 1 : [[160, 128, 102, 103, 105, 120, 125, 134, 127, 129, 123, 124, 134, 168, 139], [73, 98,
76, 74, 72, 79, 80, 82, 67, 65, 40, 45, 100, 95, 58], [12, 19, 12, 11, 10, 10, 9, 8, 23, 34, 35, 56, 28,
20, 47], [123, 108, 120, 110, 111, 112, 115, 114, 117, 119, 141, 132, 90, 88, 126]]

CLUSTER 2 : [[175, 180, 189, 145, 197, 176, 165], [48, 65, 69, 95, 67, 88, 102], [21, 23, 23, 17, 18,
32, 25], [156, 160, 143, 145, 147, 140, 137]]

CLUSTER 3 : [[172, 162, 169, 172, 174, 172, 190, 173, 200, 165, 167, 160, 157, 159, 155, 147,
178, 179], [45, 52, 89, 63, 71, 87, 49, 50, 89, 73, 75, 79, 55, 56, 57, 54, 53, 60], [20, 19, 21, 35, 40,
45, 78, 80, 29, 27, 31, 30, 33, 42, 43, 49, 65, 62], [105, 112, 119, 123, 124, 127, 133, 134, 89, 100,
102, 109, 108, 123, 124, 122, 121, 130]]
--------------------------------------------------------------------------------------------------------------


New Centroid 1 :        [128, 73, 22, 115]
New Centroid 2 :        [175, 76, 22, 146]
New Centroid 3 :        [169, 64, 41, 116]

CLUSTER 1 : [[128, 102, 103, 105, 120, 125, 134, 127, 129, 123, 124, 134], [98, 76, 74, 72, 79, 80,
82, 67, 65, 40, 45, 100], [19, 12, 11, 10, 10, 9, 8, 23, 34, 35, 56, 28], [108, 120, 110, 111, 112, 115,
114, 117, 119, 141, 132, 90]]

CLUSTER 2 : [[175, 180, 189, 160, 169, 145, 197, 176, 165], [48, 65, 69, 73, 89, 95, 67, 88, 102],
[21, 23, 23, 12, 21, 17, 18, 32, 25], [156, 160, 143, 123, 119, 145, 147, 140, 137]]

CLUSTER 3 : [[172, 162, 172, 174, 172, 190, 173, 200, 168, 165, 167, 160, 157, 159, 155, 139,
147, 178, 179], [45, 52, 63, 71, 87, 49, 50, 89, 95, 73, 75, 79, 55, 56, 57, 58, 54, 53, 60], [20, 19, 35,
40, 45, 78, 80, 29, 20, 27, 31, 30, 33, 42, 43, 47, 49, 65, 62], [105, 112, 123, 124, 127, 133, 134, 89,
```

88, 100, 102, 109, 108, 123, 124, 126, 122, 121, 130]]

New Centroid 1 : [121, 73, 21, 115]

New Centroid 2 : [172, 77, 21, 141]

New Centroid 3 : [167, 64, 41, 115]

CLUSTER 1 : [[128, 102, 103, 105, 120, 125, 134, 127, 129, 123, 124, 134], [98, 76, 74, 72, 79, 80, 82, 67, 65, 40, 45, 100], [19, 12, 11, 10, 10, 9, 8, 23, 34, 35, 56, 28], [108, 120, 110, 111, 112, 115, 114, 117, 119, 141, 132, 90]]

CLUSTER 2 : [[175, 180, 189, 160, 169, 145, 197, 176, 165], [48, 65, 69, 73, 89, 95, 67, 88, 102], [21, 23, 23, 12, 21, 17, 18, 32, 25], [156, 160, 143, 123, 119, 145, 147, 140, 137]]

CLUSTER 3 : [[172, 162, 172, 174, 172, 190, 173, 200, 168, 165, 167, 160, 157, 159, 155, 139, 147, 178, 179], [45, 52, 63, 71, 87, 49, 50, 89, 95, 73, 75, 79, 55, 56, 57, 58, 54, 53, 60], [20, 19, 35, 40, 45, 78, 80, 29, 20, 27, 31, 30, 33, 42, 43, 47, 49, 65, 62], [105, 112, 123, 124, 127, 133, 134, 89, 88, 100, 102, 109, 108, 123, 124, 126, 122, 121, 130]]

TIME AND SPACE COMPLEXITY :

Time Complexity of the algorithm:

To analyze the complexity of the algorithm in terms of time required, we need to identify the operations required in each step and in each iteration of the algorithm. Here, for k-means algorithm, the operations for each iteration are:

- Calculation of distances: To calculate the distance from a point to the centroid, we can use the squared Euclidean proximity function. For this, we need two subtractions, one summation, two multiplications and one square - root operations, i.e., 6-operations.
- Comparisons between distances
- Calculation of centroids

So, the total number of operations for an iteration

= distance calculation operations + comparison operations + centroids calculation operations

= 6*[k*m*n] operations + [(k-1)*m*n] operations + [k*((m-1) + 1)*n] operations

Assume that the algorithm converges after I iterations, then total number of operations for the algorithm
= 6*[I*k*m*n] operations + [I*(k-1)*m*n] operations + [I*k*((m-1) + 1)*n] operations
= [6Ikmn + I(k-1)mn + Ikmn] operations
≈ O (I*k*m*n)

Here, we’ve applied the basic assumption of each operation requiring the same amount of time. Since, normally, k << m & n << m, the analysis shows that the k-means is linear in m, the number of points, and is scalable and efficient in processing large data sets.

Space Complexity of the Algorithm:

The space requirement for k-means are modest because only the data points and centroids are needed to be stored. Specifically, the storage required is O((m+k)n), where m is the number of objects (points) & n is the number of attributes considering n-dimensional objects.

CONCLUSION

Hence we have successfully implemented the k-means algorithm to perform clustering of children based on the data given.

COURSE OUTCOMES :

COURSE OUTCOME	**ACHIEVED**(√)
Problem solving abilities for smart devices.	
Problem solving abilities for gamifications.	
Problem solving abilities of pervasiveness,embedded security and NLP.	
To solve problems for multicore or distributed,concurrent/Parallel environments	√