ASSIGNMENT NO

AIM : Implement a simple approach for k-means clustering using Java.

OBJECTIVE :

- To understand the basic concept of K-means Clustering.
- To implement K-means clustering algorithm using Java.

SOFTWARE REQUIREMENTS :

- Linux Operating System
- Java compiler
- Weka Tool

MATHEMATICAL MODEL :

Consider a set S consisting of all the elements related to a program.The mathematical model is given as below,

S={s,e,X,Y,Fme,DD,NDD,Mem shared}

Where,

s = Initial State
e = End State
X = Input. Here it is number of elements,actual element,number of clusters
Y = Output. Here output is _nal cluster by K-means.
Fme = Algorithm/Function used in program. for eg.fcal mean(),cal di_()g
DD=Deterministic Data
NDD=Non deterministic Data
Mem shared=Memory shared by processor.

THEORY :

K-means Clustering :

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) _xed apriori. The main idea is to de_ne k centers, one for eachncluster. These centers should be placed in a cunning way because of di_erent location causes di_erent result. So, the better choice is to place them as much as possible far away from each other.

The next step is to take each point belonging to a given data set and as sociate it to the nearest center. When no point is pending, the _rst stepis completed and an early group age is done. At this point we need to re-calculate k new centroids as barycenter of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new center. A loop has been generated.

As a result of this loop we may notice that the k centers change their location step by step until no more changes are done or in other words centers do not move any more. Finally, this algorithm aims at minimizing an objective function know as squared error function given by:

J(V)= (xi-vj)2 (1)

Where,

xi-vjis the Euclidean distance between xi and vj.
`ci' is the number of data points in ith cluster.
`c' is the number of cluster centers.

Algorithmic steps for k-means clustering :

Let X = fx1,x2,x3,..,xng be the set of data points and V = fv1,v2,.,vcg
be the set of centers.
1. Randomly select `c' cluster centers.
2. Calculate the distance between each data point and cluster centers.
3. Assign the data point to the cluster center whose distance from the cluster center is minimum of all the cluster centers.
4. Recalculate the new cluster center using:

Vi = (1=Ci)Xi (2)

where, `ci' represents the number of data points in i th cluster.
5. Recalculate the distance between each data point and new obtained cluster centers.
6. If no data point was reassigned then stop,otherwise repeat from step 3.

Advantages :
1. Fast, robust and easier to understand.
2. Relatively e_cient:O(tknd), where n is #objects, k is # clusters, d is # dimension of each object, and t is # iterations.Normally, k, t, d << n.
3. Gives best result when data set are distinct or well separated from each other.

Disadvantages :
1. The learning algorithm requires apriori speci_cation of the number of cluster centers.
2. The use of Exclusive Assignment - If there are two highly overlapping data then k-means will not be able to resolve that there are two clusters.
3. The learning algorithm is not invariant to non-linear transformations i.e. with di_erent representation of data we get di_erent results (data represented in form of cartesian co-ordinates and polar co-ordinates will give di_erent results).
4. Euclidean distance measures can unequally weight underlying factors.
5. The learning algorithm provides the local optima of the squared error function.
6. Randomly choosing of the cluster center cannot lead us to the fruitful result.
7. Applicable only when mean is de_ned i.e. fails for categorical data.
8. Unable to handle noisy data and outliers.
9. Algorithm fails for non-linear data set.

K-means : Step-By-Step Example

As a simple illustration of a k-means algorithm, consider the following data set consisting of the scores of two variables on each of seven individuals:

Subject	A	B
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

This data set is to be grouped into two clusters. As a first step in finding a sensible initial partition, let the A and B values of the two individuals furthest apart (using the Euclidean distance measure), de_ne the initial cluster means, giving:

	Individual	Mean Vector(centriod)
Group 1	1	(1.0,1.0)
Group 2	4	(5.0,7.0)

The remaining individuals are now examined in sequence and allocated to the cluster to which they are closest, in terms of Euclidean distance to the cluster mean. The mean vector is recalculated each time a new member is added. This leads to the following series of steps:

Step	Individual	Mean Vector(centriod)	Individual	Mean Vector
1	1	(1.0,1.0)	4	(5.0,7.0)
2	1,2	(1.2,1.5)	4	(5.0,7.0)
3	1,2,3	(1.8,2.3)	4	(5.0,7.0)
4	1,2,3	(1.8,2.3)	4,5	(4.2,6.0)
5	1,2,3	(1.8,2.3)	4,5,6	(4.3,5.7)
6	1,2,3	(1.8,2.3)	4,5,6,7	(4.1,5.4)

Now the initial partition has changed, and the two clusters at this stage having the following characteristics:

	Individual	Mean Vector(centriod)
Cluster 1	1,2,3	(1.8,2.3)
Cluster 2	4,5,6,7	(4.1,5.4)

But we cannot yet be sure that each individual has been assigned to the right cluster. So, we compare each individuals distance to its own cluster mean and to that of the opposite cluster. And we find:

Individual	Dist. to mean of Cluster 1	Dist. to mean of Cluster 2
1	1.5	5.4
2	0.4	4.3
3	2.1	1.8
4	5.7	1.8
5	3.2	0.7
6	3.8	0.6
7	2.8	1.1

Only individual 3 is nearer to the mean of the opposite cluster (Cluster 2) than its own (Cluster 1). In other words, each individual's distance to its own cluster mean should be smaller that the distance to the other cluster's mean (which is not the case with individual 3). Thus, individual 3 is relocated to Cluster 2 resulting in the new partition:

	Individual	Mean Vector(centriod)
Cluster 1	1,2	(1.3,1.5)
Cluster 2	3,4,5,6,7	(3.9,5.1)

The iterative relocation would now continue from this new partition until no more relocations occur. However, in this example each individual is now nearer its own cluster mean than that of the other cluster and the iteration stops, choosing the latest partitioning as the final cluster solution.

Also, it is possible that the k-means algorithm won't find a final solution. In this case it would be a good idea to consider stopping the algorithm after a pre-chosen maximum of iterations.

PSEUDO CODE :

Pseudo code for cal mean() :

```
static void cal mean()
{
	int temp1=0;
	for(int i=0;i<p;++i)
	{
		if(a>m[i])
			di_[i]=a-m[i];
		else
			di_[i]=m[i]-a;
	}
	int val=0;
	double temp=di_[0];
	for(int i=0;i<p;++i)
	{
		if(di_[i]<temp)
		{
			temp=di_[i];
			val=i;
		}
	}
	return val;
}
```

CONCLUSION :

Thus, we have implemented K-means clustering algorithm using Java.