

EXP.NO : 05

DATE :

IMPLEMENTATION OF K-MEANS CLUSTERING USING MAPREDUCE

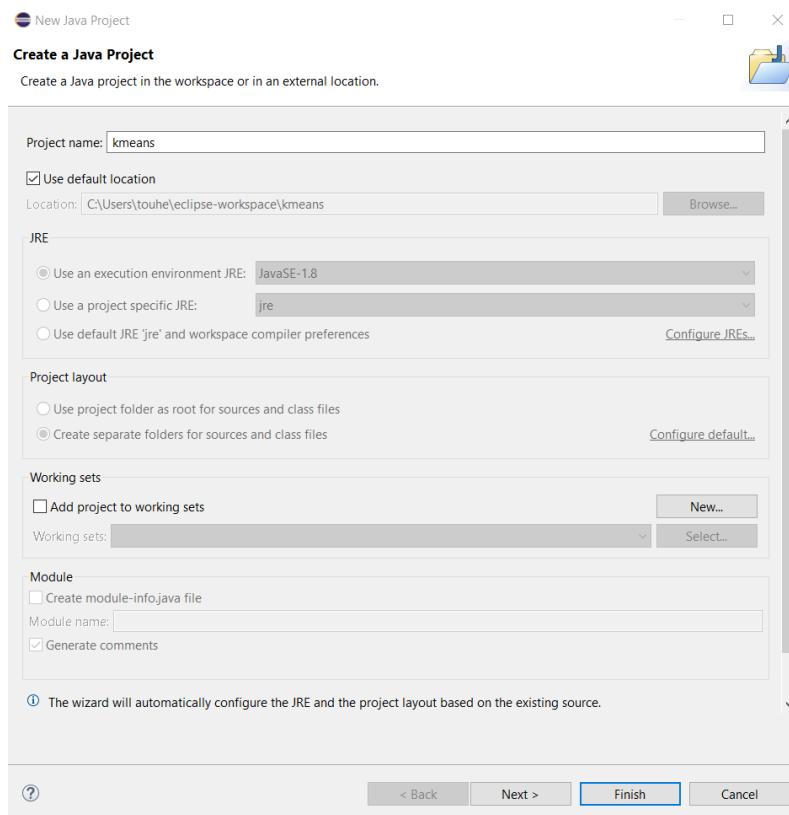
AIM : To implement K-means clustering using mapreduce.

STEPS :

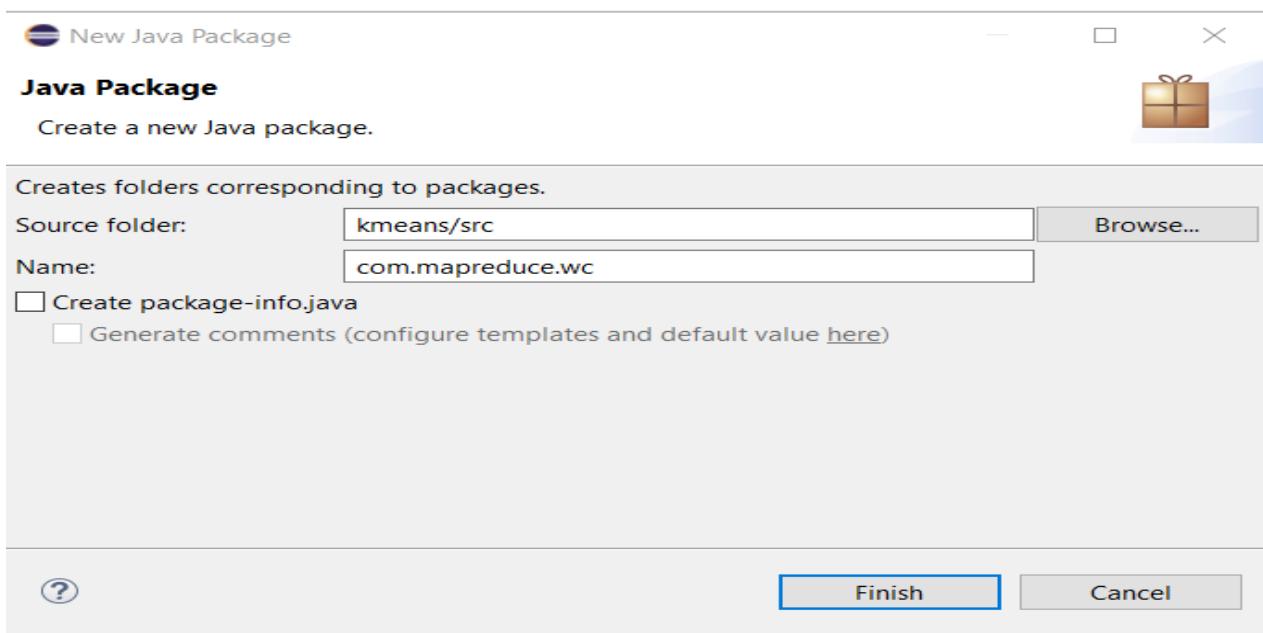
STEP 1: Run Eclipse for Java Developers

STEP 2: Create a new Java Project with name “Kmeans”.

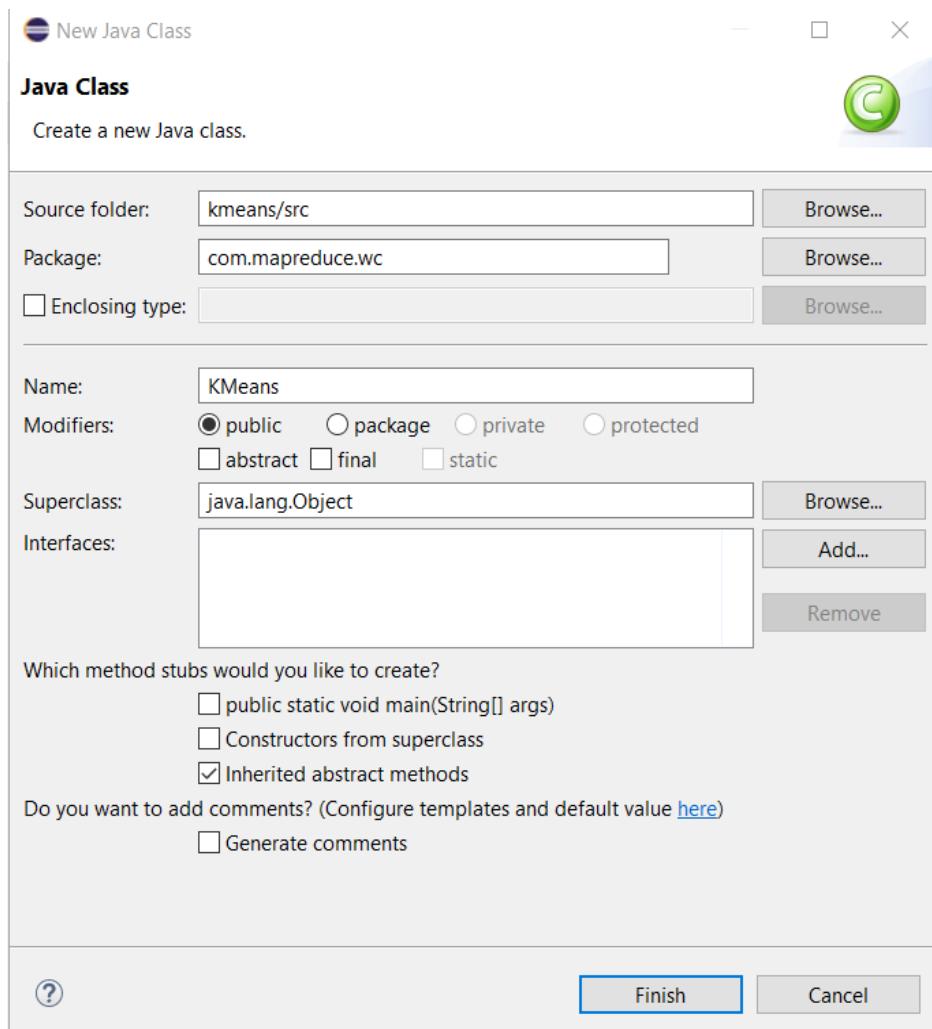
STEP 3: Set the Java Environment Version to your current version of Java (JRE - 1.8)



STEP 4: Add a Package with name “com.mapreduce.java” and Create three Classes in it.



STEP 5 :Create a New Class With name “KMeans.java”



STEP 6: Now write the below program in the “KMeans.java” Class

PROGRAM :

```
package com.mapreduce.wc;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Iterator;
import java.util.List;
import java.util.Map;

public class KMeans {

    List<Record> data = new ArrayList<Record>();
    List<Cluster> clusters = new ArrayList<Cluster>();

    Map<Cluster, List<Record>> clusterRecords = new HashMap<Cluster, List<Record>>();

    public static void main(String[] args) {
        int clusterNumber = 2;
        KMeans demo = new KMeans();
        demo.genereateRecord();
        demo.initiateClusterAndCentroid(clusterNumber);
        demo.printRecordInformation();
        demo.printClusterInformation();
    }

    private void genereateRecord() {
        Record record = new Record(1, 19, 15, 39);
        data.add(record);
        record = new Record(2, 21, 15, 81);
        data.add(record);
        record = new Record(3, 20, 16, 6);
        data.add(record);
        record = new Record(4, 23, 16, 77);
        data.add(record);
        record = new Record(5, 31, 17, 40);
        data.add(record);
        record = new Record(6, 22, 17, 76);
        data.add(record);
    }

    private void initiateClusterAndCentroid(int clusterNumber) {
```

```

int counter = 1;
Iterator<Record> iterator = data.iterator();
Record record = null;
while(iterator.hasNext()) {
    record = iterator.next();
    if(counter <= clusterNumber) {
        record.setClusterNumber(counter);
        initializeCluster(counter, record);
        counter++;
    }else {
        System.out.println(record);
        System.out.println("** Cluster Information **");
        for(Cluster cluster : clusters) {
            System.out.println(cluster);
        }
        System.out.println("*****");
    }
    double minDistance = Integer.MAX_VALUE;
    Cluster whichCluster = null;

    for(Cluster cluster : clusters) {
        double distance = cluster.calculateDistance(record);
        System.out.println(distance);
        if(minDistance > distance) {
            minDistance = distance;
            whichCluster = cluster;
        }
    }

    record.setClusterNumber(whichCluster.getClusterNumber());
    whichCluster.updateCentroid(record);
    clusterRecords.get(whichCluster).add(record);
}

System.out.println("** Cluster Information **");

for(Cluster cluster : clusters) {
    System.out.println(cluster);
}

```

```

        }
        System.out.println("*****");
    }

private void initializeCluster(int clusterNumber, Record record) {Cluster
    System.out.println("***** Each Record INFORMATIN *****");
    for(Record record : data) {
        System.out.println(record);
    }clusterCluster(clusterNumber,record.getAge(),record.getIncome(),record.getScore());
    clusters.add(cluster);
    List<Record> clusterRecord = new ArrayList<Record>();
    clusterRecord.add(record);
    clusterRecords.put(cluster, clusterRecord);
}

private void printRecordInformation() {
}

private void printClusterInformation() {
    System.out.println("***** FINAL CLUSTER INFORMATIN *****");
    for (Map.Entry<Cluster, List<Record>> entry : clusterRecords.entrySet()) {
        System.out.println("Key = " + entry.getKey() +
                           ", Value = " + entry.getValue());
    }
}
}
}

```

STEP 7: Now Create another class with name “Cluster.java” and write the below program in it.

PROGRAM :

```
package com.mapreduce.wc;

public class Cluster {
    private int ageCentroid;
    private int incomeCentroid;
    private int scoreCentroid;
    private int clusterNumber;

    public Cluster(int clusterNumber, int ageCentroid, int incomeCentroid, int scoreCentroid) {
        super();
        this.clusterNumber = clusterNumber;
        this.ageCentroid = ageCentroid;
        this.incomeCentroid = incomeCentroid;
        this.scoreCentroid = scoreCentroid;
    }

    public int getAgeCentroid() {
        return ageCentroid;
    }

    public void setAgeCentroid(int ageCentroid) {
        this.ageCentroid = ageCentroid;
    }

    public int getIncomeCentroid() {
        return incomeCentroid;
    }

    public void setIncomeCentroid(int incomeCentroid) {
        this.incomeCentroid = incomeCentroid;
    }

    public int getScoreCentroid() {
        return scoreCentroid;
    }

    public void setScoreCentroid(int scoreCentroid) {
        this.scoreCentroid = scoreCentroid;
    }

    public int getClusterNumber() {
        return clusterNumber;
    }
}
```

```

}

public void setClusterNumber(int clusterNumber) {
    this.clusterNumber = clusterNumber;
}

@Override
public String toString() {
    return "Cluster [ageCentroid=" + ageCentroid + ", incomeCentroid=" + incomeCentroid + ", scoreCentroid="
           + scoreCentroid + ", clusterNumber=" + clusterNumber + "]";
}

// Euclidean distance calculation
public double calculateDistance(Record record) {
    return Math.sqrt(Math.pow((getAgeCentroid() - record.getAge()), 2) +
Math.pow((getIncomeCentroid() - record.getIncome()), 2) + Math.pow((getScoreCentroid() -
record.getScore()), 2));
}

// Binod Suman Academy YouTube Video on K-Mean Algorithm
public void updateCentroid(Record record) {
    setAgeCentroid((getAgeCentroid() + record.getAge()) / 2);
    setIncomeCentroid((getIncomeCentroid() + record.getIncome()) / 2);
    setScoreCentroid((getScoreCentroid() + record.getScore()) / 2);
}
}

```

STEP 8: Now Create another class with name “Record.java” and write the below program in it.

PROGRAM :

```

package com.mapreduce.wc;

public class Record {

    private int id;
    private int age;
    private int income;
    private int score;
    private int clusterNumber

    public Record(int id, int age, int income, int score) {
        super();
    }
}

```

```
        this.id = id;
        this.age = age;
        this.income = income;
        this.score = score;
    }
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        this.age = age;
    }
    public int getIncome() {
        return income;
    }
    public void setIncome(int income) {
        this.income = income;
    }
    public int getScore() {
        return score;
    }
    public void setScore(int score) {
        this.score = score;
    }
    public int getClusterNumber() {
        return clusterNumber;
    }
    public void setClusterNumber(int clusterNumber) {
        this.clusterNumber = clusterNumber;
    }
}
```

```

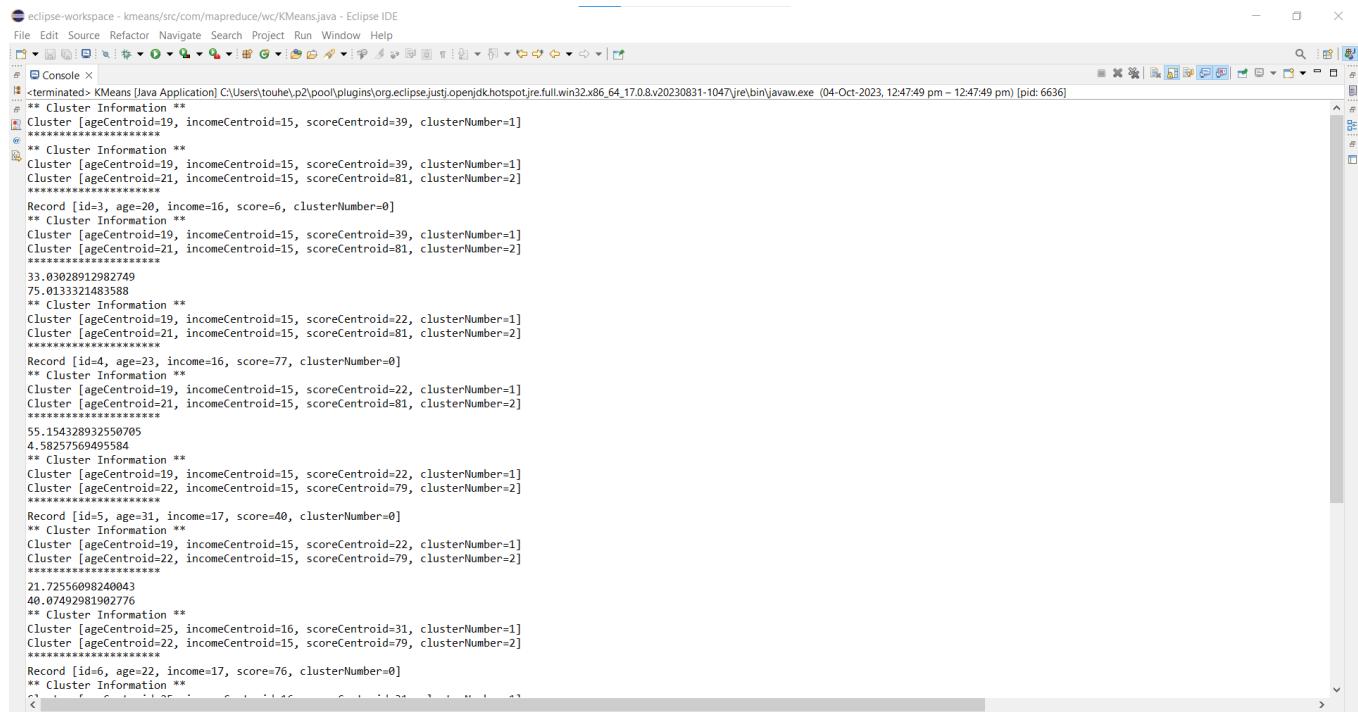
}

@Override
public String toString() {
    return "Record [id=" + id + ", age=" + age + ", income=" + income + ", score=" + score + ",
clusterNumber="+ clusterNumber + "]";
}
}

```

STEP 9: Run the “KMeans.java” class from the project to get the output.

OUTPUT:



The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. The output window displays the execution of the KMeans Java application. It shows the initial centroids and the iterative process of assigning data points to clusters and updating centroids until convergence. The output includes numerical values for centroids and data points, along with cluster assignment information.

```

eclipse-workspace - kmeans/src/com/mapreduce/wc/KMeans.java - Eclipse IDE
File Edit Source Refactor Navigate Project Run Window Help
Console
<terminated> KMeans [Java Application] C:\Users\touhe\p2\pool\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86_64_17.0.8.v20230831-1047\jre\bin\javaw.exe (04-Oct-2023, 12:47:49 pm - 12:47:49 pm) [pid: 6636]
** Cluster Information **
Cluster [ageCentroid=19, incomeCentroid=15, scoreCentroid=39, clusterNumber=1]
*****
** Cluster Information **
Cluster [ageCentroid=19, incomeCentroid=15, scoreCentroid=39, clusterNumber=1]
Cluster [ageCentroid=21, incomeCentroid=15, scoreCentroid=81, clusterNumber=2]
*****
Record [id=3, age=20, income=16, score=6, clusterNumber=0]
** Cluster Information **
Cluster [ageCentroid=19, incomeCentroid=15, scoreCentroid=39, clusterNumber=1]
Cluster [ageCentroid=21, incomeCentroid=15, scoreCentroid=81, clusterNumber=2]
*****
Record [id=4, age=23, income=16, score=77, clusterNumber=0]
** Cluster Information **
Cluster [ageCentroid=19, incomeCentroid=15, scoreCentroid=22, clusterNumber=1]
Cluster [ageCentroid=21, incomeCentroid=15, scoreCentroid=81, clusterNumber=2]
*****
55.154328932550705
4.58257569495584
** Cluster Information **
Cluster [ageCentroid=19, incomeCentroid=15, scoreCentroid=22, clusterNumber=1]
Cluster [ageCentroid=22, incomeCentroid=15, scoreCentroid=79, clusterNumber=2]
*****
Record [id=5, age=31, income=17, score=40, clusterNumber=0]
** Cluster Information **
Cluster [ageCentroid=19, incomeCentroid=15, scoreCentroid=22, clusterNumber=1]
Cluster [ageCentroid=22, incomeCentroid=15, scoreCentroid=79, clusterNumber=2]
*****
21.72556098240043
40.07492981902776
** Cluster Information **
Cluster [ageCentroid=25, incomeCentroid=16, scoreCentroid=31, clusterNumber=1]
Cluster [ageCentroid=22, incomeCentroid=15, scoreCentroid=79, clusterNumber=2]
*****
Record [id=6, age=22, income=17, score=76, clusterNumber=0]
** Cluster Information **

```

```
eclipse-workspace - kmeans/src/com/mapreduce/wc/KMeans.java - Eclipse IDE
File Edit Source Refactor Navigate Project Run Window Help
Console X
terminated: KMeans [Java Application] C:\Users\touhe\p2\pool\plugins\org.eclipse.jst.openjdk.hotspot.jre.full.win32.x86_64_17.0.8.v20230831-1047\jre\bin\javaw.exe (04-Oct-2023, 12:47:49 pm - 12:47:49 pm) [pid: 6636]
Cluster [ageCentroid=19, incomeCentroid=15, scoreCentroid=22, clusterNumber=1]
Cluster [ageCentroid=21, incomeCentroid=15, scoreCentroid=81, clusterNumber=2]
*****
55.154328932550705
4.58257569495584
** Cluster Information **
Cluster [ageCentroid=19, incomeCentroid=15, scoreCentroid=22, clusterNumber=1]
Cluster [ageCentroid=22, incomeCentroid=15, scoreCentroid=79, clusterNumber=2]
*****
Record [id=5, age=31, income=17, score=40, clusterNumber=0]
** Cluster Information **
Cluster [ageCentroid=19, incomeCentroid=15, scoreCentroid=22, clusterNumber=1]
Cluster [ageCentroid=22, incomeCentroid=15, scoreCentroid=79, clusterNumber=2]
*****
21.72556098240043
40.07492981902776
** Cluster Information **
Cluster [ageCentroid=25, incomeCentroid=16, scoreCentroid=31, clusterNumber=1]
Cluster [ageCentroid=22, incomeCentroid=15, scoreCentroid=79, clusterNumber=2]
*****
Record [id=6, age=22, income=17, score=76, clusterNumber=0]
** Cluster Information **
Cluster [ageCentroid=25, incomeCentroid=16, scoreCentroid=31, clusterNumber=1]
Cluster [ageCentroid=22, incomeCentroid=15, scoreCentroid=79, clusterNumber=2]
*****
45.110974274559844
3.605551275463989
** Cluster Information **
Cluster [ageCentroid=25, incomeCentroid=16, scoreCentroid=31, clusterNumber=1]
Cluster [ageCentroid=22, incomeCentroid=16, scoreCentroid=77, clusterNumber=2]
*****
***** Each Record INFORMATIN *****
Record [id=1, age=19, income=15, score=39, clusterNumber=1]
Record [id=2, age=21, income=15, score=81, clusterNumber=2]
Record [id=3, age=20, income=16, score=6, clusterNumber=1]
Record [id=4, age=23, income=16, score=77, clusterNumber=2]
Record [id=5, age=31, income=17, score=40, clusterNumber=1]
Record [id=6, age=22, income=17, score=76, clusterNumber=2]
*****
***** FINAL CLUSTER INFORMATIN *****
Key = Cluster [ageCentroid=25, incomeCentroid=16, scoreCentroid=31, clusterNumber=1], Value = [Record [id=1, age=19, income=15, score=39, clusterNumber=1], Record [id=3, age=20, income=16, score=6, clusterNumber=1], Record [id=5, age=31, income=17, score=40, clusterNumber=1], Record [id=6, age=22, income=17, score=76, clusterNumber=2]]
Value = [Record [id=2, age=21, income=15, score=81, clusterNumber=2], Record [id=4, age=23, income=16, score=77, clusterNumber=2]]
```

RESULT: Thus the program to implement K-means clustering using mapreduce is executed the output is verified successfully.