**Practical 2: Write a Program to Perform Data Transformation Using Various Transformation Techniques**

import java.util.Arrays;

class NormalizationTransformation {

public static void main(String[] args) {

double[] arr3 = {200, 300, 400, 600, 1000};

Arrays.sort(arr3);

System.out.println(Arrays.toString(minMaxNormalization(arr3)));

System.out.println(Arrays.toString(zScoreNormalization(arr3)));

System.out.println(Arrays.toString(decimalScalingNormalization(arr3)));

}

static double[] minMaxNormalization(double[] arr)

{

double[] minMaxNormalizedArr = new double[arr.length];

double new\_min = 0.0;

double new\_max = 1.0;

for (int i = 0; i < arr.length; i++) {

minMaxNormalizedArr[i] = ((arr[i]-arr[0])/(arr[arr.length-1] - arr[0])\*(new\_max-new\_min))+new\_min;

}

return minMaxNormalizedArr;

}

static double[] zScoreNormalization(double[] arr)

{

double[] zScoreNormalizedArr = new double[arr.length];

double sum = 0.0f ; double sumDiffSquare = 0.0f;

for (double d : arr) {

sum += d;

}

double mean = sum / arr.length;

for (double d : arr) {

sumDiffSquare += (d - mean)\*(d-mean);

}

double sd = Math.sqrt(sumDiffSquare/arr.length);

for (int i = 0; i < arr.length; i++) {

zScoreNormalizedArr[i] = (arr[i]-mean)/sd;

}

return zScoreNormalizedArr;

}

static double[] decimalScalingNormalization(double[] arr)

{

double[] deciamlScalingNormalizedArr = new double[arr.length];

double largestValue = Math.abs(arr[0]) > Math.abs(arr[arr.length-1])?Math.abs(arr[0]):Math.abs(arr[arr.length-1]);

long factor = Math.round(Math.log10(largestValue));

for (int i = 0; i < arr.length; i++) {

deciamlScalingNormalizedArr[i] = arr[i]/(Math.pow(10,factor));

}

return deciamlScalingNormalizedArr;

}

}

**Output**

