Inter American University of Puerto Rico

Bayamon Campus

School of Engineering

Electrical and Computer Engineering Department

COEN 2310- Discrete Mathematics for Computer Engineering

Homework Report N°: Final Project InterCovid

Student Name: Raúl A. Torres Cobián

Name of homework: Proyecto Final

Percent of task complete: 88%

Date: Mayo 12, 2021

Introduction

In this Proyecto is about an intercovid the files must calculated apply in class which is the distance, update, initialize and accuracy. The algorithm will apply each step that professor choose the students which were going to use for example: I was pick using formula Manhattan, initial first elements and average update. This code of my theorical will use Java programming to use this project. This project has 3 data in files to apply the project. This project will include graph in java using Jframe (GUI) that apply library like arraylist and hashmap.

Program

a) The program I test is using a loop files to read all elements using array list and HashMap. The run program will use class named K_cluster the output will show 3 different consoles is the average, distance, initialize value and accuracy. In this output only show distance Manhattan in class manhattan, average update, initialize first elements and accuracy I include graph based the project that must do the 3 data. The graph class will use a frame will include library class of hashmap and list array will use in padding the coordinate score that read the file that has to do in data 1 to 3.

Data1 results

```
Iterations: 100
Count of Clusters (groups): 3
Distance: 50.0
Accuracy: 1.3333333333333335%
Press 1 if you want to continue or press 0 to exit.............................
```

El input the program will ask this question:

1. “Enter the filename with path.”

Data1.txt (and hit enter)

2. “Enter the no. of cluster (group”

3 (hit enter)’

3. “Enter the maximum iteration.”

100 (or 10,000 hit enter)

This dark letter is the input that need to write to read the file data one and compile feature 1 to 4 and final cluster.

Data2 results

```
Iterations: 100
Count of Clusters (groups): 3
Distance: 29.0
Accuracy: 52.24719101123596%
Press 1 if you want to continue or press 0 to exit.............................
```

El input the program will ask this question:

1. “Enter the filename with path.”

Data2.txt (and hit enter)

2. “Enter the no. of cluster (group”

3 (hit enter because is 3 group in last column)

3. “Enter the maximum iteration.”

100 (or 10,000 hit enter)

This dark letter is the input that need to write to read the file data one and compile feature 1 to 4 and final cluster.

Data3 results

```
Iterations: 100
Count of Clusters (groups): 2
Distance: 5136.0
Accuracy: 7.6%
Press 1 if you want to continue or press 0 to exit.............................
```

El input the program will ask this question:

1. “Enter the filename with path.”

Data3.txt (and hit enter)

2. “Enter the no. of cluster (group”

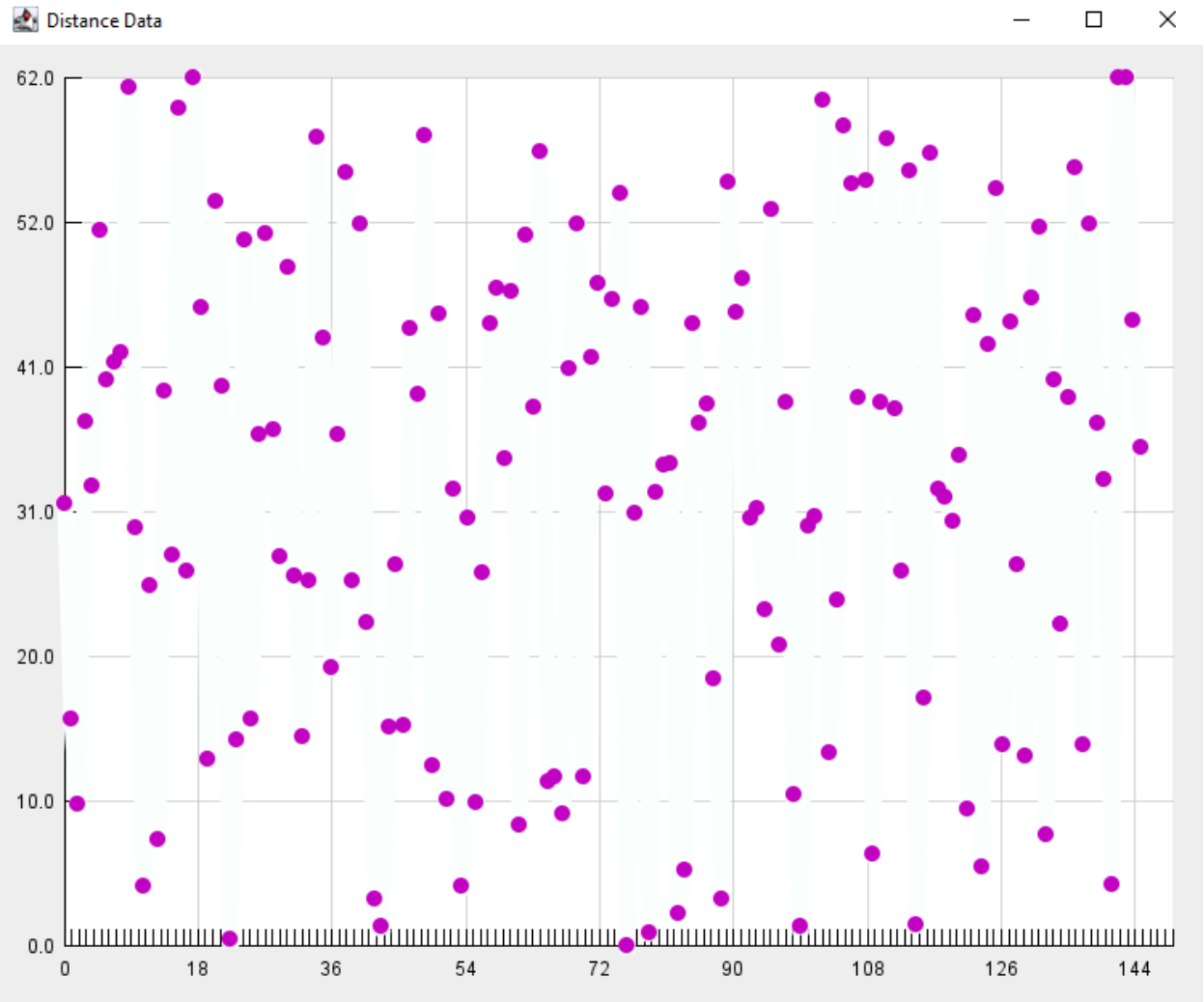
2 (this cluster contains only 2 groups only and hit enter).

3. “Enter the maximum iteration.”

100 (or 10,000 hits enter **Note**: 10,000 will take 13 seconds to read all files in data 3 if you enter 10,000 iterations).

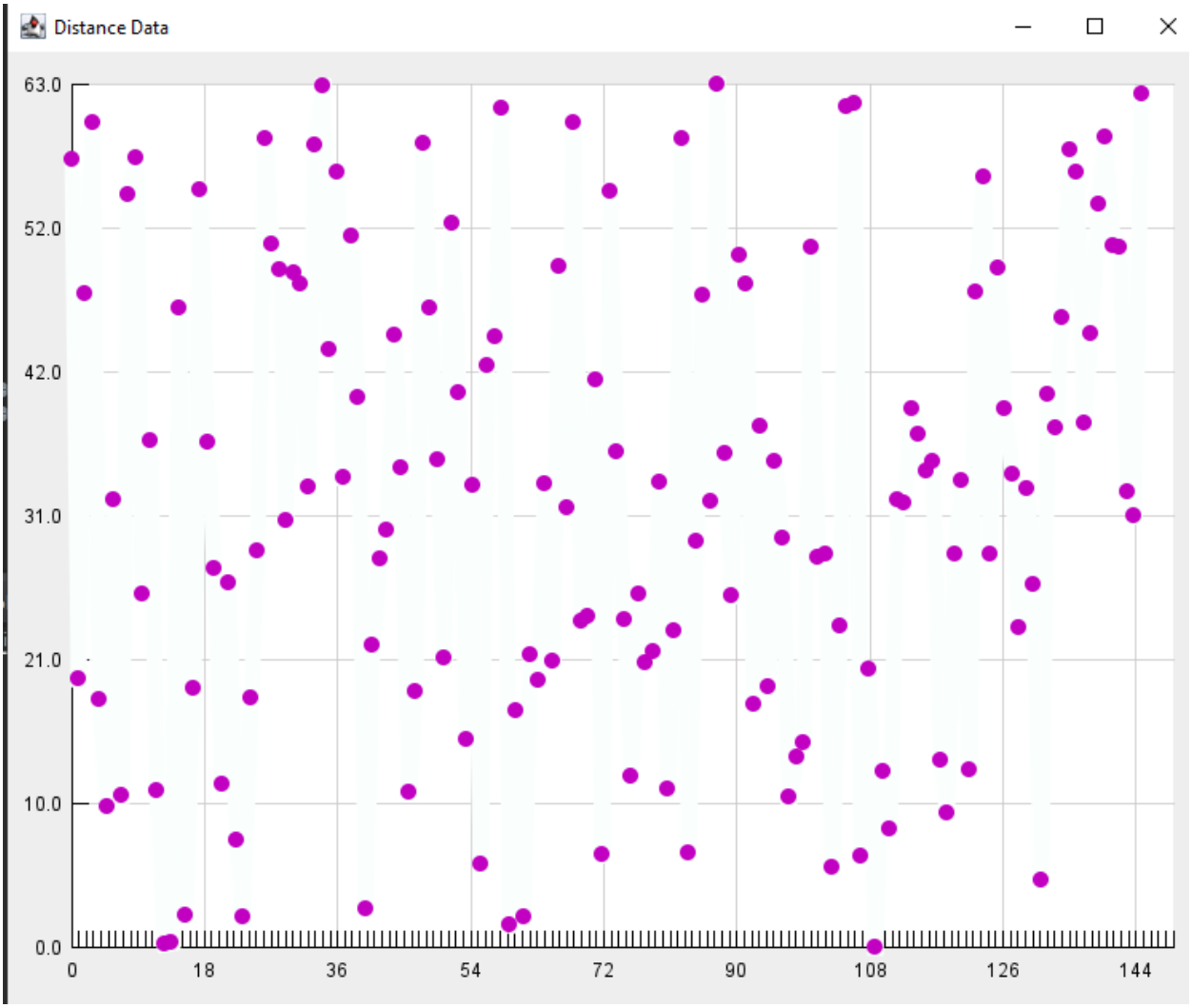
This dark letter is the input that need to write to read the file data one and compile feature 1 to 4 and final cluster.

Data1 graph



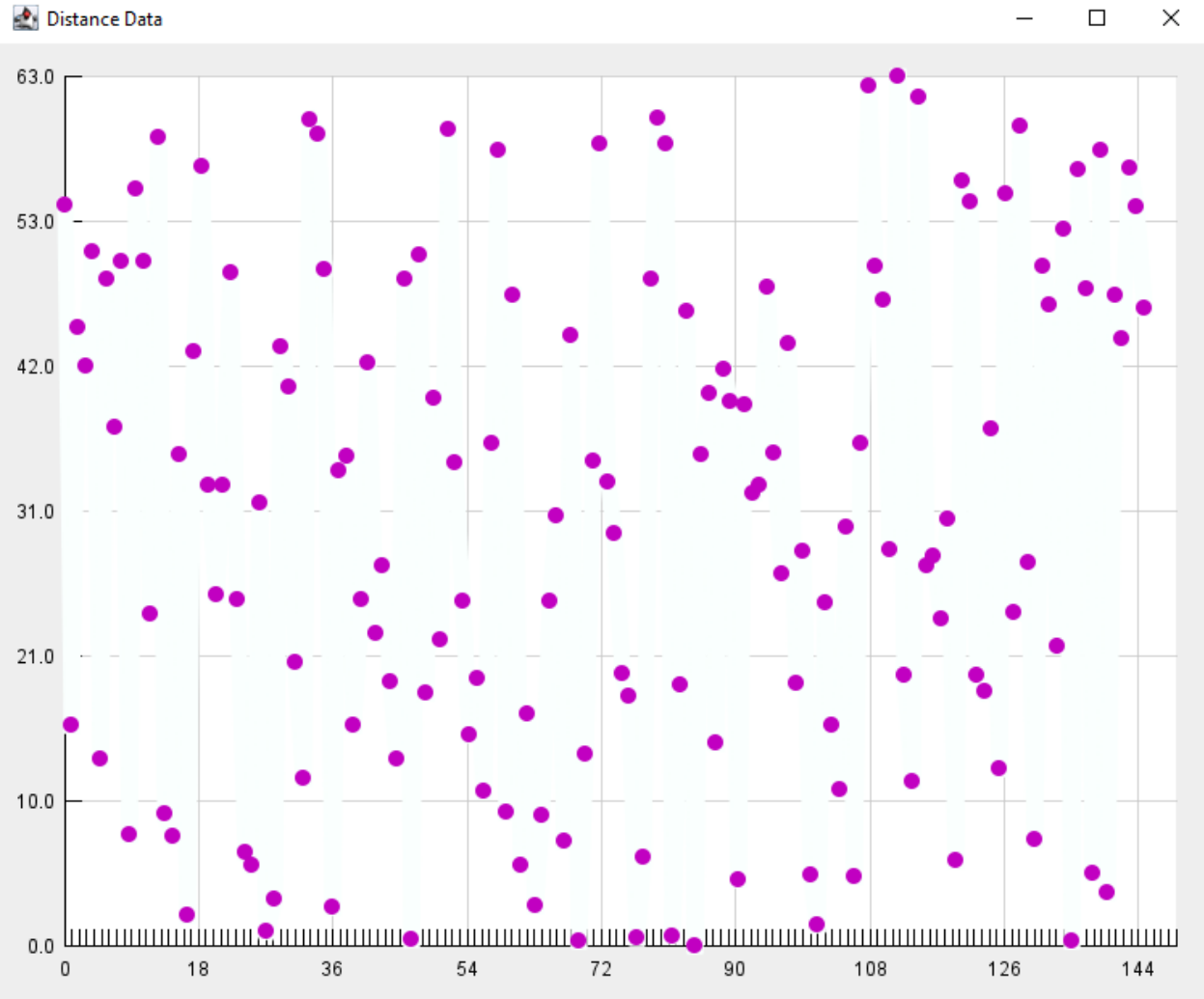
This graph works in line 191 where the main method is, I forgot to put the input the file for graph. The new file Reader are in parenthesis we put “Data1.txt” to output the graph with coordinates based the file read the panel.

Data2 graph



This graph works in line 191 where the main method is, I forgot to put the input the file for graph. The new file Reader are in parenthesis we put “Data2.txt” to output the graph with coordinates based the file read the panel. The graph is different than Data1.txt

Data3



This graph works in line 191 where the main method is, I forgot to put the input the file for graph. The new file Reader are in parenthesis we put “Data3.txt” to output the graph with coordinates based the file read the panel. The graph is different than Data2.txt and Data1.txt.

c) Using a table, list all the created functions. The columns of the table will be name, description, input, output of the function.

1. distance manhattan (minima)

$g_1 = |6 - 5.6| - |-0.1 - 0.8|$

g1	6, 5.6 -0.1, 0.8
g2	3, 4, 8, 7

= 17.5 distancia mínima

2. Element first initialize (center group)

Data:

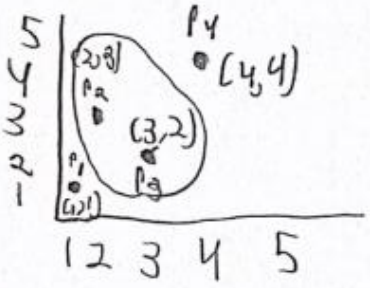
f1	f2
8	2
9	8
10	9

9, 8 center groups

3. Update center (average)

$g_1 = \frac{2+3}{2} = 2.5$

$g_2 = \frac{3+2}{2} = 2.5$



Ends iterator 100 or 10,000

4. Accuracy

$Accuracy = \frac{2}{4} \times 100\% = 50\%$

lines	data read	final cluster
T 1	1	1
F 2	1	0
F 3	2	1
T 4	2	2

Result

a) Table 2: results.

Data1

Distance	50
Updating	N/A
Initialize	5.1, 3.5, 1.4, 0.2 ,1.0

Data2

Distance	3.8
Updating	N/A
Initialize	14.23, 1.71, 2.43, 15.6, 127.0

Data3

Distance	3.8
Updating	N/A
Initialize	5.1, 3.5, 1.4, 0.2 ,1.0

Name files	K	Accuracy
Data1	3	1.33%
Data2	3	52.24%
Data3	2	7.6%

b) The result of this table I think the data 2 us better because the accuracy is higher than data 3 and 1. The accuracy of data 2 is 50% which the theoretical good for solve the groups of cases the Inter covid.

Complexity of time (Big O)

(Find worst case possible)

K_cluster (main)

```
1  package projectFinal;
2  import java.io.IOException;
4
5  //class with inheritance readdataset class
6  public class K_Clusterer extends ReadDataset {
7
8
9  //main method
10     public static void main(String args[]) throws IOException {
11
12         //file
13         ReadDataset r1 = new ReadDataset();  +1
14         r1.features.clear();
15         //input scanner
16         Scanner sc = new Scanner(System.in);  +1
17         //name file
18         System.out.println("Enter the filename with path");
19         String file = sc.next();  +1
20         r1.read(file); //load data
21         int ex = 1;  +1
22
23         //creating do while
24         do{
25             //input cluster (group)
26             System.out.println("Enter the no. of clusters (group)");
27             int group = sc.nextInt();  +1
28
29             System.out.println("Enter maximum iterations");
30             int max_iterations = sc.nextInt();  +1
31
32
33             //Hashmap to store center to calculate average
34             Map<Integer, double[]> center = new HashMap<>();
35             // calculating initial centroids
36             double[] x1 = new double[numberOfFeatures];
37             int r =0; +1
38             for (int i = 0 ; i < group; i++) {
39
40                 x1 = r1.features.get(r ++);
41                 center.put(i, x1);
42
43             }
44
45
46             //Hashmap for finding cluster indexes using hashmap
47             Map<double[], Integer> clusters = new HashMap<>();  +1
48
49             clusters = kmeans(r1.features, center, group);  +1
50             // initial cluster print
51             double db[] = new double[numberOfFeatures];  +1
52
53             //reassigning to new clusters (group) using nested loop
54             for (int i = 0 ; i < max_iterations; i++) {
55
56                 for (int j = 0 ; j < group; j ++) {
57
58                     //arraylist libary
59                     List<double[]> list = new ArrayList<>();
```

n+6

n x n = n^2

```
56              for (int j = 0 ; j < group; j ++) {
57
58          //arraylist libary
59          List<double[]> list = new ArrayList<>();
60                  //another loop with
61              for (double[] key : clusters.keySet()) {
62
63              if (clusters.get(key)==j) {
64
65              list.add(key);
66                      }
67              }//end for loop cluster
68
69                  //calculator average in center update
70                  db = centerCalculator(list);
71                  center.put(j, db);
72              }
73              //clear culster to clean output
74              clusters.clear();
75              //
76              clusters = kmeans(r1.features, center, group);
77
78          }//end loop iteration
79
80          //final cluster print line (optional for purpose check)
81          System.out.println("\nFinal Clustering of Data file");
82          System.out.println("Feature1\tFeature2\tFeature3\tFeature4\tCluster");
83
84          //using nested for loop cluster key set
85          for (double[] key : clusters.keySet()) {
86              for (int i = 0; i < key.length; i++) {
87                  System.out.print(key[i] + "\t \t");
88              }
89              System.out.print(clusters.get(key) + "\n");
90          }
91
92
93          //calulate Accuracy
94          int accrate = 0 , total = 0;
95          for (double[] key : clusters.keySet()) {
96              if (clusters.get(key) + 1 == r1.getLabel().get(key)) {
97
98                  //increment accuraccy
99                  accrate ++;
100             }
101             //increment
102             total++;
103         }
104         //accuracy operator calculator
105         double accuracy = (double)accrate / (double)total * 100;
106
107         //end accuracy
108
109
110         //Calculate manhattan distance
111         double distance = 0;
112
113         for(int i = 0 ; i < group ; i++){
```

```
            for(int i = 0 ; i < group ; i++){

                double sse = 0;
                for (double[] key : clusters.keySet()) {
                    if (clusters.get(key)==i) {
                        Distance.manhattanDistance(key, center.get(i));
                        //increment sse
                        sse++;
                    }
                    //disntace assign sse
                    distance = sse;
                }

            }
            //end manhattan distance

            //print final results iteration,grous,distabce and accuracy
            System.out.println("Iterations: " + max_iterations);
            System.out.println("Count of Clusters (groups): " + group);
            System.out.println("Distance: " + distance);

            System.out.println("Accuracy: " + accuracy + "%");
            //print of sentineel cero
            System.out.println("Press 1 if you want to continue or press 0 to exit.....................");

            //input the counter sentinel
            ex=sc.nextInt();
        } while(ex==1); //while loop ends the do while
    }

    //method to calculate average to update the cluster
    public static double[] centerCalculator(List<double[]> a) {

        //variables of two int and double
        int count = 0;
        double sum=0.0;

        double[] average = new double[ReadDataset.numberOfFeatures];
        for (int i = 0; i < ReadDataset.numberOfFeatures; i++) {
            sum=0.0;
            count = 0;
            //single for loop assig a method variables
            for(double[] x:a){
                count++;
                //math operator average
                sum = sum + x[i];
            }

            average[i] = sum / count;
            //System.out.println("average: " + average);
        }
        return average;


    }

    //method for putting features to clusters and reassignment of clusters with hashmap,list to assign d
```

```
//method for putting features to clusters and reassignment of clusters with hashmap,list to assign distance minim

public static Map<double[], Integer> kmeans(List<double[]> features, Map<Integer, double[]> centroids, int k) {

    //create hashmap for cluster
    Map<double[], Integer> clusters = new HashMap<>();
    int k1 = 0;
    double dist=0.0;

    //nested for loop
    for( double[] x : features) {
        double minimum = 999999.0;
        for (int j = 0 ; j < k ; j ++) {

            //declaring distance minimum
                dist = Distance.manhattanDistance(centroids.get(j), x);

            //distance if statements minimum distance
            if (dist < minimum) {
                //assign equal minimum to distance
                minimum = dist;
                //assign k1 to j colms
                k1 = j;
            }
        }
        clusters.put(x, k1);

    }

    //System.out.println("Distance: " + dist);
    return clusters;
}//end method
nd program
```

+1

+1

+1

+1

$n \times n + 2 = n^2 + 2$

11) $f(n) = 5n^2 + 4n + 37$

$f(n) \in O(n^2)$

Datareadset.java

```
1  package projectFinal;
2
3  import java.io.*;
5
6  public class ReadDataset {
7
8      //create protected for inheritance
9      protected List<double[]> features = new ArrayList<>();
10     protected Map<double[] , Integer> label = new HashMap<>();
11     //variable using protect with static for center number
12     protected static int numberOfFeatures;
13     private BufferedReader readFile;
14
15     //createmethod list libary feature
16     public List<double[]> getFeatures()
17     {
18         return features;
19     }
20
21     //create map of labels
22     public Map<double[] , Integer> getLabel()
23     {
24         return label;
25     }
26
27     //void method read file
28     void read(String s) throws NumberFormatException, IOException {
29
30         File file=new File(s);
31
32         //try catch file read
33     try {
34         readFile = new BufferedReader(new FileReader(file));
35         String line;
36
37         //read file using while loop
38         while((line = readFile.readLine()) != null)
39             {
40
41             String[] split = line.split(",");
42                     double[] feature = new double[split.length - 1];
43
44                 //number feature of loops will include -1
45                 numberOfFeatures = split.length-1;
46
47                 //for loop for split elements
48                 for (int i = 0; i < split.length - 1; i++)
49
50             feature[i] = Double.parseDouble(split[i]);
51            features.add(feature);
52             String labels = split[feature.length];
53            label.put(feature , Integer.parseInt(labels));
54            }
55     }
56     //catch error exception for files
57     catch (FileNotFoundException e) {
58         // TODO Auto-generated catch block
59         e.printStackTrace();
```

```
    //catch error exception for files
    catch (FileNotFoundException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

}//end void method

//disokay for iterator
void display(){
        Iterator<double[]> itr = features.iterator();

        while(itr.hasNext()) {
            //iterate variable with single array
            double db[] = itr.next();
            //whiule loop next the 4 x element axis.
            for(int i = 0 ; i < 4 ; i ++){
                //output the iterator
            System.out.print(db[i] + " ");
        }

        }

    }//end void method display
}//end program
```

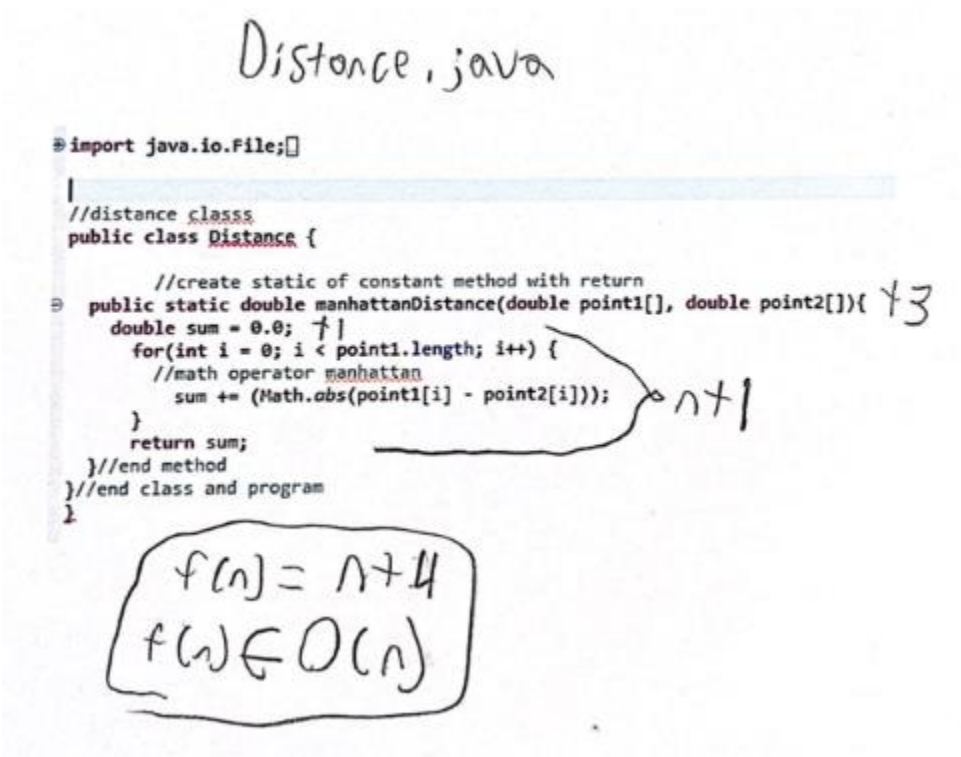
+1

n x n = n² + 2

f(n) = 2n² + 12 + C

f(n) ∈ O(n²)

Distance.java



Graph (Jframe)

```
1  package projectFinal;
2
3  import java.awt.*;
11
12
13 //libary grapg final project
14 public class GraphFinalProject extends JPanel{
15
16     //instance fields
17
18     //instance fiedls of Jframe
19     private static final long serialVersionUID = 5L;
20
21     //thew line axis x and y extend GUI aplication
22     private int labelPadding = 20;
23
24     //color line
25     private Color lineColor = new Color(250,255,254);
26
27     private Color pointColor = new Color(193,0,193 );
28
29     private Color gridColor = new Color(200, 200, 200, 200);
30
31     private static final Stroke GRAPH_STROKE = new BasicStroke(10f);
32
33     private static int pointWidth = 10;
34
35     private int numberDivisions = 6;
36     //variable coordinateds for points
37     private List<Double> coordinates;
38     //extend padd
39     private int padding = 20;
40
41     public GraphFinalProject(List<Double> scores) {
42         this.coordinates = scores;
43     }
44
45 //paint component with ovverdiding method
46     protected void paintComponent(Graphics g) {
47         super.paintComponent(g);
48         Graphics2D g2 = (Graphics2D) g;
49         g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING, RenderingHints.VALUE_ANTIALIAS_ON);
50
51         //create points using math operator with type casting
52  double xScale = ((double) getWidth() - (3 * padding) - labelPadding) / (coordinates.size() - 1);
53   double yScale = ((double) getHeight() - 2 * padding - labelPadding) / (getMaxScore() - getMincoordinates());
54
55         //include pad points between x axis and y axis
56         List<Point> graphPoints = new ArrayList<>();
57         for (int i = 0; i < coordinates.size(); i++) {
58             int x1 = (int) (i * xScale + padding + labelPadding);
59             int y1 = (int) ((getMaxScore() - coordinates.get(i)) * yScale + padding);
60             graphPoints.add(new Point(x1, y1));
61         }
62        //color background
63       g2.setColor(Color.WHITE);
64      //fill the rect
65    g2.fillRect(padding + labelPadding, padding, getWidth() - (2 * padding)
66           - labelPadding, getHeight() - 2 * padding - labelPadding);
```

```
      //loop x axis , y axis
       for (int i = 0; i < numberDivisions + 1; i++) {
         int x0 = padding + labelPadding;
         int x1 = pointWidth + padding + labelPadding;
         int y0 = getHeight() - ((i * (getHeight() - padding * 2 -
        labelPadding)) / numberDivisions + padding + labelPadding);
          //assign y1 to y0
          int y1 = y0;
          //size coordinates
            if (coordinates.size() > 0) {
              g2.setColor(gridColor);
              g2.drawLine(padding + labelPadding + 1 + pointWidth, y0, getWidth() - padding, y1);
             g2.setColor(Color.BLACK);
              String yLabel = ((int) (getMincoordinates() + (getMaxScore() -
           getMincoordinates()) * ((i * 8.0) / numberDivisions)) * 100) / 100.0 + "";
              //metric for lines
              FontMetrics metrics = g2.getFontMetrics();
              int labelWidth = metrics.stringWidth(yLabel);
           g2.drawString(yLabel, x0 - labelWidth - 6, y0 + (metrics.getHeight() / 2) - 3);
           }
              //line with 4 lines
              g2.drawLine(x0, y0, x1, y1);
          }
//coordinates size distance
          for (int i = 0; i < coordinates.size(); i++) {
              if (coordinates.size() > 1) {
           int x0 = i * (getWidth() - padding * 2 - labelPadding) /
                    (coordinates.size() - 1) + padding + labelPadding;
           int x1 = x0;
            int y0 = getHeight() - padding - labelPadding;
             int y1 = y0 - pointWidth;
             if ((i % ((int) ((coordinates.size() / 10.0)) + 3)) == 0) {
               g2.setColor(gridColor);
                g2.drawLine(x0, getHeight() - padding - labelPadding - 1 - pointWidth, x1, padding);
                g2.setColor(Color.BLACK);
                 String xLabel = i + "";
               FontMetrics metrics = g2.getFontMetrics();
           int labelWidth = metrics.stringWidth(xLabel);
            g2.drawString(xLabel, x0 - labelWidth / 2, y0 + metrics.getHeight() + 3);
             }
               g2.drawLine(x0, y0, x1, y1);
             }
          }//end loop corrdinates

  //creating lines
 g2.drawLine(padding + labelPadding, getHeight() - padding -
 labelPadding, padding + labelPadding, padding);
```

```
    //creating lines
   g2.drawLine(padding + labelPadding, getHeight() - padding -
   labelPadding, padding + labelPadding, padding);
   g2.drawLine(padding + labelPadding, getHeight() - padding -
labelPadding, getWidth() - padding, getHeight() - padding - labelPadding);

       //the line color with graph stroke (2D)
      Stroke oldStroke = g2.getStroke();
     g2.setColor(lineColor);
     g2.setStroke(GRAPH_STROKE);

       for (int i = 0; i < graphPoints.size() - 1; i++) {
         int x1 = graphPoints.get(i).x;
         int y1 = graphPoints.get(i).y;
         int x2 = graphPoints.get(i + 1).x;
        int y2 = graphPoints.get(i + 1).y;
        g2.drawLine(x1, y1, x2, y2);
         }

        g2.setStroke(oldStroke);
        g2.setColor(pointColor);
         for (int i = 0; i < graphPoints.size(); i++) {
             int x = graphPoints.get(i).x - pointWidth / 2;
             int y = graphPoints.get(i).y - pointWidth / 2;
             int X1 = pointWidth;
             int Y2 = pointWidth;
             g2.fillOval(x, y, X1, Y2);
         }
     }

     //min coordinates.
     private double getMincoordinates() {
         double minScore = Double.MAX_VALUE;
         for (Double score : coordinates) {
             minScore = Math.min(minScore, score);
         }
         return minScore;
     }

     //apply max socre for prevent out of graph
     private double getMaxScore() {
         double maxScore = Double.MIN_VALUE;
         for (Double score : coordinates) {
             maxScore = Math.max(maxScore, score);
         }
         return maxScore;
     }
 //set score of coordinates
     public void setScores(List<Double> scores) {
         this.coordinates = scores;
         invalidate();
         this.repaint();
     }
 //return using method like get and set
     public List<Double> getScores() {
         return coordinates;
     }
 //creating the method createAndShowGui in the main method, where we create t
     private static void createAndShowGui() {
```

```
private static void createAndShowGui() {
    List<Double> scores = new ArrayList<>(); +1
    Random random = new Random(); +1
    int maxDataPoints = 150;      +1
    int maxScore = 8;             +1
    for (int i = 0; i < maxDataPoints; i++) {
        scores.add((double) random.nextDouble() * maxScore);
    }                                                          n+1

         //panel main
    GraphFinalProject mainPanel = new GraphFinalProject(scores);
    mainPanel.setPreferredSize(new Dimension(750, 600));
 //frame dsiplay with exit and close
    JFrame frame = new JFrame("Distance Data");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.getContentPane().add(mainPanel);
    frame.pack();
    frame.setLocationRelativeTo(null);
    frame.setVisible(true);
}
  //the main method runs createAndShowGui and
public static void main(String[] args) throws IOException {

    //file exception try catch for files and write manual
            try (BufferedReader in = new BufferedReader(new FileReader("Data3.txt"))) { +1

     //variable string
    String str;          +1

    List<String> list = new ArrayList<String>();  +1
    while ((str = in.readLine()) != null) {
    list.add(str);                                  n+1
            }
}catch (FileNotFoundException e) {
    // TODO Auto-generated catch block
        e.printStackTrace();
    }

    //final output graph
         SwingUtilities.invokeLater(new Runnable() {
     public void run() {
        createAndShowGui();
     }
  }
); //end graph
}
/end program
```

$f(n) = n8 + 44 + 6$

$f(n) \in O(n)$

Conclusion

a) The design I complete is using a function prototype for each iteration of data that include Manhattan will merge iteration with update average that is why the output it shows the accuracy which is important number of the output for each file. The calculation distance, average, iteration, and accuracy it works.

b) The problem I was facing is accuracy because it gave me error about the arrays out of bounds error which is frustrating the problem until I use library the array list and HashMap that will complete the accuracy and show the console. The second issue I face is the graph library java because is the second time how to use GUI the graph to solve the points that shows in panel and is not showing. The solution I did only is find calculation of width and length with calculation label to output the graph it has 11 operator math for graph to show the screen.

c) I test the graph and it gave me logic error because the graph is not easy for use java. I was trying to solve graph points 3 color for each file of data until the time is running out the deadline that I had no choice but summit the result.

d) Things I learned this project list:

1.hashmap

2. Jframe (for graph with operator for width and length and get without set method)

3. Array list (with List Library)

4. Accuracy formula

5. Iteration

Appendix: adjust here source code.

K_cluster class

```java
package projectFinal;
import java.io.IOException;

//class with inheritance readdataset class
public class K_Clusterer extends ReadDataset {


//main method
    public static void main(String args[]) throws IOException {

        //file
        ReadDataset r1 = new ReadDataset();
        r1.features.clear();
        //input scanner
        Scanner sc = new Scanner(System.in);
        //name file
        System.out.println("Enter the filename with path");
        String file = sc.next();
        r1.read(file); //load data
        int ex = 1;

        //creating do while
        do{
            //input cluster (group)
            System.out.println("Enter the no. of clusters (group)");
            int group = sc.nextInt();

            System.out.println("Enter maximum iterations");
            int max_iterations = sc.nextInt();


            //Hashmap to store center to calculate average
            Map<Integer, double[]> center = new HashMap<>();
            // calculating initial centroids
            double[] x1 = new double[numberOfFeatures];
            int r =0;
            for (int i = 0 ; i < group; i++) {

                x1 = r1.features.get(r ++);
                center.put(i, x1);

            }


            //Hashmap for finding cluster indexes using hashmap
            Map<double[], Integer> clusters = new HashMap<>();

            clusters = kmeans(r1.features, center, group);
            // initial cluster print
            double db[] = new double[numberOfFeatures];

            //reassigning to new clusters (group) using nested loop
            for (int i = 0 ; i < max_iterations; i++) {

                for (int j = 0 ; j < group; j ++) {

                    //arraylist libary
                    List<double[]> list = new ArrayList<>();
                    //another loop with
                    for (double[] key : clusters.keySet()) {
```

```
                for (double[] key : clusters.keySet()) {

                    if (clusters.get(key)==j) {

                        list.add(key);


                    }
            }//end for loop cluster

                //calculator average in center update
                db = centerCalculator(list);
                center.put(j, db);
            }
            //clear culster to clean output
            clusters.clear();
            //
            clusters = kmeans(r1.features, center, group);

        }//end loop iteration

        //final cluster print line (optional for purpose check)
        System.out.println("\nFinal Clustering of Data file");
        System.out.println("Feature1\tFeature2\tFeature3\tFeature4\tCluster");

        //using nested for loop cluster key set
        for (double[] key : clusters.keySet()) {
            for (int i = 0; i < key.length; i++) {
                System.out.print(key[i] + "\t \t");
            }
            System.out.print(clusters.get(key) + "\n");
        }


        //calulate Accuracy
        int accrate = 0 , total = 0;
        for (double[] key : clusters.keySet()) {
            if (clusters.get(key) + 1 == r1.getLabel().get(key)) {

                //increment accuraccy
                accrate ++;
            }
            //increment
            total++;
        }
        //accuracy operator calculator
        double accuracy = (double)accrate / (double)total * 100;

        //end accuracy


        //Calculate manhattan distance
        double distance = 0;

        for(int i = 0 ; i < group ; i++){

            double sse = 0;
            for (double[] key : clusters.keySet()) {
                if (clusters.get(key)==i) {
```

```
                double sse = 0;
                for (double[] key : clusters.keySet()) {
                    if (clusters.get(key)==i) {
                        Distance.manhattanDistance(key, center.get(i));
                        //increment sse
                        sse++;
                    }
                    //disntace assign sse
                    distance = sse;
                }

            }
            //end manhattan distance

            //print final results iteration,grous,distabce and accuracy
            System.out.println("Iterations: " + max_iterations);
            System.out.println("Count of Clusters (groups): " + group);
            System.out.println("Distance: " + distance);

            System.out.println("Accuracy: " + accuracy + "%");
            //print of sentineel cero
            System.out.println("Press 1 if you want to continue or press 0 to exit.............................");

            //input the counter sentinel
            ex=sc.nextInt();
        }  while(ex==1); //while loop ends the do while
    }

    //method to calculate average to update the cluster
    public static double[] centerCalculator(List<double[]> a) {

        //variables of two int and double
        int count = 0;
        double sum=0.0;

        double[] average = new double[ReadDataset.numberOfFeatures];
        for (int i = 0; i < ReadDataset.numberOfFeatures; i++) {
            sum=0.0;
            count = 0;
            //single for loop assig a method variables
            for(double[] x:a){
                count++;
                //math operator average
                sum = sum + x[i];
            }

            average[i] = sum / count;
            //System.out.println("average: " + average);
        }
        return average;


    }

    //method for putting features to clusters and reassignment of clusters with hashmap,list to assign distance minim

    public static Map<double[], Integer> kmeans(List<double[]> features, Map<Integer, double[]> centroids, int k) {

        //create hashmap for cluster
        Map<double[], Integer> clusters = new HashMap<>();
```

```java
    //method for putting features to clusters and reassignment of clusters with hashmap,list to assign distance minimum

    public static Map<double[], Integer> kmeans(List<double[]> features, Map<Integer, double[]> centroids, int k) {

        //create hashmap for cluster
        Map<double[], Integer> clusters = new HashMap<>();
        int k1 = 0;
        double dist=0.0;

        //nested for loop
        for( double[] x : features) {
            double minimum = 999999.0;
            for (int j = 0 ; j < k ; j ++) {

                //declaring distance minimum
                    dist = Distance.manhattanDistance(centroids.get(j), x);

                //distance if statements minimum distance
                if (dist < minimum) {
                    //assign equal minimum to distance
                    minimum = dist;
                    //assign k1 to j colms
                    k1 = j;
                }
            }
            clusters.put(x, k1);

        }

        //System.out.println("Distance: " + dist);
        return clusters;
    }//end method
}//end program

```

Read Data set class

```java
package projectFinal;

import java.io.*;

public class ReadDataset {

    //create protected for inheritance
    protected List<double[]> features = new ArrayList<>();
    protected Map<double[] , Integer> label = new HashMap<>();
    //variable using protect with static for center number
    protected static int numberOfFeatures;
    private BufferedReader readFile;

    //createmethod list libary feature
    public List<double[]> getFeatures()
    {
        return features;
    }

    //create map of labels
    public Map<double[] , Integer> getLabel()
    {
        return label;
    }

    //void method read file
    void read(String s) throws NumberFormatException, IOException {

        File file=new File(s);

        //try catch file read
    try {
        readFile = new BufferedReader(new FileReader(file));
        String line;

        //read file using while loop
        while((line = readFile.readLine()) != null)
            {

             String[] split = line.split(",");
                    double[] feature = new double[split.length - 1];

                 //number feature of loops will include -1
                numberOfFeatures = split.length-1;

                //for loop for split elements
                for (int i = 0; i < split.length - 1; i++)

             feature[i] = Double.parseDouble(split[i]);
            features.add(feature);
             String labels = split[feature.length];
            label.put(feature , Integer.parseInt(labels));
            }
    }
    //catch error exception for files
    catch (FileNotFoundException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

}//end void method
```

```
62  }//end void method
63
64  //disokay for iterator
65  void display(){
66          Iterator<double[]> itr = features.iterator();
67
68
69          while(itr.hasNext()) {
70              //iterate variable with single array
71              double db[] = itr.next();
72              //whiule loop next the 4 x element axis.
73              for(int i = 0 ; i < 4 ; i ++){
74                  //output the iterator
75              System.out.print(db[i] + " ");
76          }
77
78          }
79
80      }//end void method display
81  }//end program
82
```

Distance class

```
1   package projectFinal;
2
3
4   //distance classs
5   public class Distance {
6
7           //create static of constant method with return
8       public static double manhattanDistance(double point1[], double point2[]){
9           double sum = 0.0;
10          for(int i = 0; i < point1.length; i++) {
11              //math operator manhattan
12              sum += (Math.abs(point1[i] - point2[i]));
13          }
14          return sum;
15      }//end method
16  }//end class and program
17
```

GraphFinalProject class

```java
package projectFinal;

import java.awt.*;


//libary grapg final project
public class GraphFinalProject extends JPanel{

    //instance fields

    //instance fiedls of Jframe
    private static final long serialVersionUID = 5L;

    //thew line axis x and y extend GUI aplication
    private int labelPadding = 20;

    //color line
    private Color lineColor = new Color(250,255,254);

    private Color pointColor = new Color(193,0,193 );

    private Color gridColor = new Color(200, 200, 200, 200);

    private static final Stroke GRAPH_STROKE = new BasicStroke(10f);

    private static int pointWidth = 10;

    private int numberDivisions = 6;
    //variable coordinateds for points
    private List<Double> coordinates;
    //extend padd
    private int padding = 20;

    public GraphFinalProject(List<Double> scores) {
        this.coordinates = scores;
    }

//paint component with ovverdiding method
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        Graphics2D g2 = (Graphics2D) g;
        g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING, RenderingHints.VALUE_ANTIALIAS_ON);

        //create points using math operator with type casting
 double xScale = ((double) getWidth() - (3 * padding) - labelPadding) / (coordinates.size() - 1);
  double yScale = ((double) getHeight() - 2 * padding - labelPadding) / (getMaxScore() - getMincoordinates());

        //include pad points between x axis and y axis
        List<Point> graphPoints = new ArrayList<>();
        for (int i = 0; i < coordinates.size(); i++) {
            int x1 = (int) (i * xScale + padding + labelPadding);
            int y1 = (int) ((getMaxScore() - coordinates.get(i)) * yScale + padding);
            graphPoints.add(new Point(x1, y1));
        }
       //color background
      g2.setColor(Color.WHITE);
     //fill the rect
    g2.fillRect(padding + labelPadding, padding, getWidth() - (2 * padding) - labelPadding, getHeight() - 2 * padding - labelPadding);


           //loop x axis , y axis
```

```
//loop x axis , y axis
  for (int i = 0; i < numberDivisions + 1; i++) {
    int x0 = padding + labelPadding;
    int x1 = pointWidth + padding + labelPadding;
    int y0 = getHeight() - ((i * (getHeight() - padding * 2 -
   labelPadding)) / numberDivisions + padding + labelPadding);
     //assign y1 to y0
     int y1 = y0;
     //size coordinates
       if (coordinates.size() > 0) {
         g2.setColor(gridColor);
         g2.drawLine(padding + labelPadding + 1 + pointWidth, y0, getWidth() - padding, y1);
        g2.setColor(Color.BLACK);
         String yLabel = ((int) (getMincoordinates() + (getMaxScore() - getMincoordinates()) * ((i * 8.0) / numberDivisions)) * 100) / 100.0 + "";
        //metric for lines
        FontMetrics metrics = g2.getFontMetrics();
        int labelWidth = metrics.stringWidth(yLabel);
     g2.drawString(yLabel, x0 - labelWidth - 6, y0 + (metrics.getHeight() / 2) - 3);
    }
       //line with 4 lines
       g2.drawLine(x0, y0, x1, y1);
   }
ordinates size distance
    for (int i = 0; i < coordinates.size(); i++) {
        if (coordinates.size() > 1) {
     int x0 = i * (getWidth() - padding * 2 - labelPadding) / (coordinates.size() - 1) + padding + labelPadding;
     int x1 = x0;
      int y0 = getHeight() - padding - labelPadding;
       int y1 = y0 - pointWidth;
       if ((i % ((int) ((coordinates.size() / 10.0)) + 3)) == 0) {
         g2.setColor(gridColor);
          g2.drawLine(x0, getHeight() - padding - labelPadding - 1 - pointWidth, x1, padding);
          g2.setColor(Color.BLACK);
           String xLabel = i + "";
        FontMetrics metrics = g2.getFontMetrics();
     int labelWidth = metrics.stringWidth(xLabel);
      g2.drawString(xLabel, x0 - labelWidth / 2, y0 + metrics.getHeight() + 3);
       }
         g2.drawLine(x0, y0, x1, y1);
      }
    }//end loop corrdinates

/creating lines
 g2.drawLine(padding + labelPadding, getHeight() - padding - labelPadding, padding + labelPadding, padding);
 g2.drawLine(padding + labelPadding, getHeight() - padding - labelPadding, getWidth() - padding, getHeight() - padding - labelPadding);

  //the line color with graph stroke (2D)
 Stroke oldStroke = g2.getStroke();
g2.setColor(lineColor);
g2.setStroke(GRAPH_STROKE);

  for (int i = 0; i < graphPoints.size() - 1; i++) {
    int x1 = graphPoints.get(i).x;
    int y1 = graphPoints.get(i).y;
    int x2 = graphPoints.get(i + 1).x;
   int y2 = graphPoints.get(i + 1).y;
   g2.drawLine(x1, y1, x2, y2);
    }
```

```
        g2.setColor(pointColor);
        for (int i = 0; i < graphPoints.size(); i++) {
            int x = graphPoints.get(i).x - pointWidth / 2;
            int y = graphPoints.get(i).y - pointWidth / 2;
            int X1 = pointWidth;
            int Y2 = pointWidth;
            g2.fillOval(x, y, X1, Y2);
        }
    }

    //min coordinates.
    private double getMincoordinates() {
        double minScore = Double.MAX_VALUE;
        for (Double score : coordinates) {
            minScore = Math.min(minScore, score);
        }
        return minScore;
    }

    //apply max socre for prevent out of graph
    private double getMaxScore() {
        double maxScore = Double.MIN_VALUE;
        for (Double score : coordinates) {
            maxScore = Math.max(maxScore, score);
        }
        return maxScore;
    }
//set score of coordinates
    public void setScores(List<Double> scores) {
        this.coordinates = scores;
        invalidate();
        this.repaint();
    }
//return using method like get and set
    public List<Double> getScores() {
        return coordinates;
    }
//creating the method createAndShowGui in the main method, where we create the frame too and pack it in the panel
    private static void createAndShowGui() {
        List<Double> scores = new ArrayList<>();
        Random random = new Random();
        int maxDataPoints = 150;
        int maxScore = 8;
        for (int i = 0; i < maxDataPoints; i++) {
            scores.add((double) random.nextDouble() * maxScore);

        }

             //panel main
        GraphFinalProject mainPanel = new GraphFinalProject(scores);
        mainPanel.setPreferredSize(new Dimension(750, 600));
     //frame dsiplay with exit and close
        JFrame frame = new JFrame("Distance Data");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.getContentPane().add(mainPanel);
        frame.pack();
        frame.setLocationRelativeTo(null);
        frame.setVisible(true);
    }
      //the main method runs createAndShowGui and
    public static void main(String[] args) throws IOException {
```

```
    }
      //the main method runs createAndShowGui and
    public static void main(String[] args) throws IOException {

        //file exception try catch for files
                try (BufferedReader in = new BufferedReader(new FileReader("Data3.txt"))) {

         //variable string
        String str;

        List<String> list = new ArrayList<String>();
        while ((str = in.readLine()) != null) {
        list.add(str);
                }
    }catch (FileNotFoundException e) {
        // TODO Auto-generated catch block
            e.printStackTrace();
        }

        //final output graph
              SwingUtilities.invokeLater(new Runnable() {
         public void run() {
            createAndShowGui();
         }
      }
  ); //end graph
   }
}//end program
```