VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

DATA ANALYZING AND SUMMARIZING SYSTEM

A PROJECT REPORT

Submitted in partial fulfillment for the award of the course

PROGRAMMING IN JAVA

(SWE1007)

By

NITHISH KUMAR S (20MIS0024)

STEVE MATHEW D A (20MIS0047)

DURGA SHREE N (20MIS0054)

Under the guidance of

Prof. ELANGO N M

ABSTRACT:

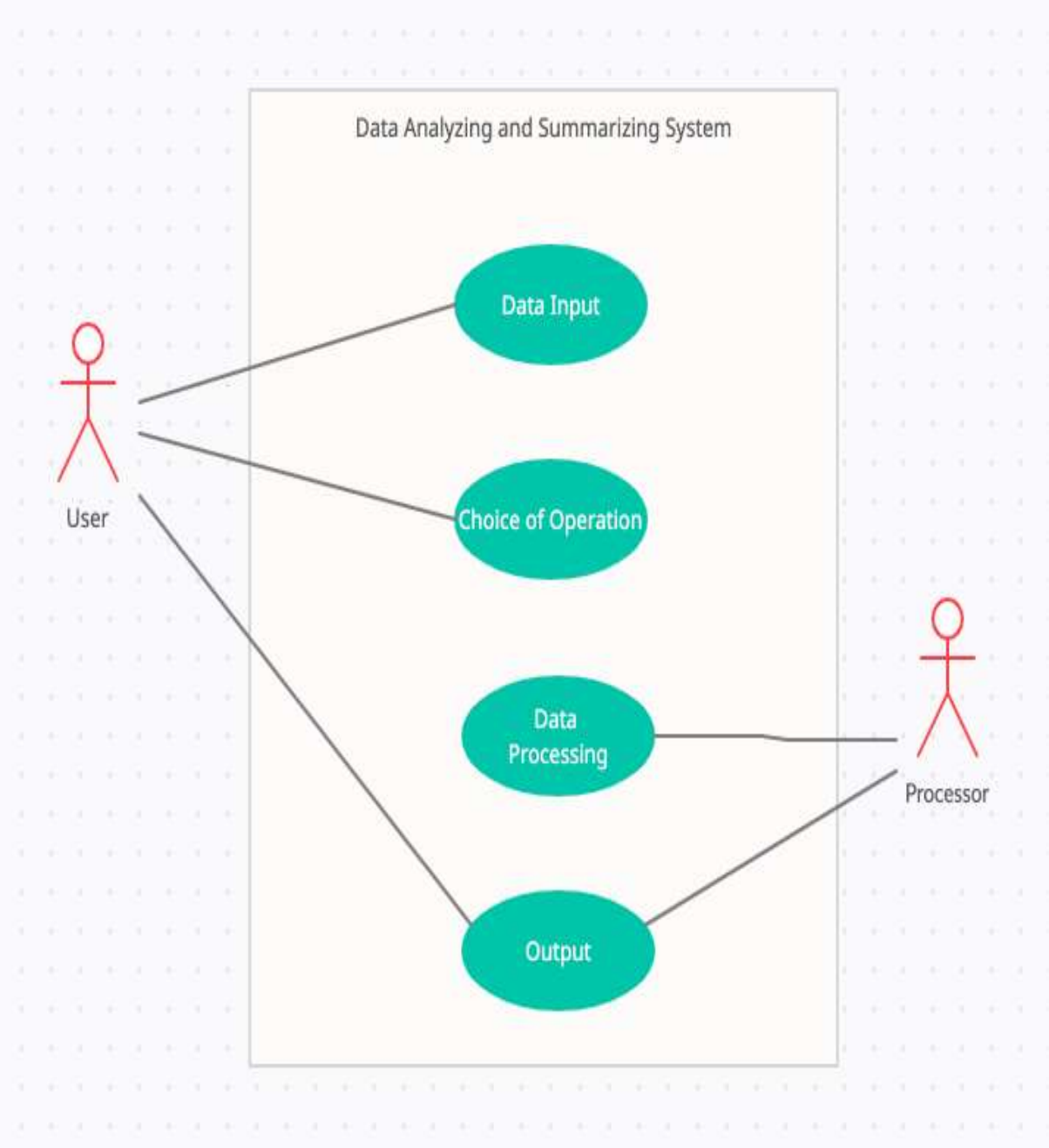
The System being proposed deals with statistical analysis. The Functionalities we include are summarization, analysis and modelling of data. Summarization deals with computing the mean, median, mode, standard deviation, variance and mean deviation of the given data using standard, conventional formulae. The system performs fundamental data analysis using statistical concepts such as covariance, correlation, point biserial correlation and formation of contingency tables. We extract results from the analysis phase and in turn, apply them to the data modelling phase to perform simple future predictions. Algorithms such as linear regression, which is the linear approach for modelling the relationship between a scalar response and one or more explanatory variables, is being applied to perform future predictions based on the analytical results obtained from the previous (analysis) phase. Furthermore, the outcomes from the correlation analysis are used to find the principal components of the dataset and the potential relationships between various combinations of data attributes.

INTRODUCTION:

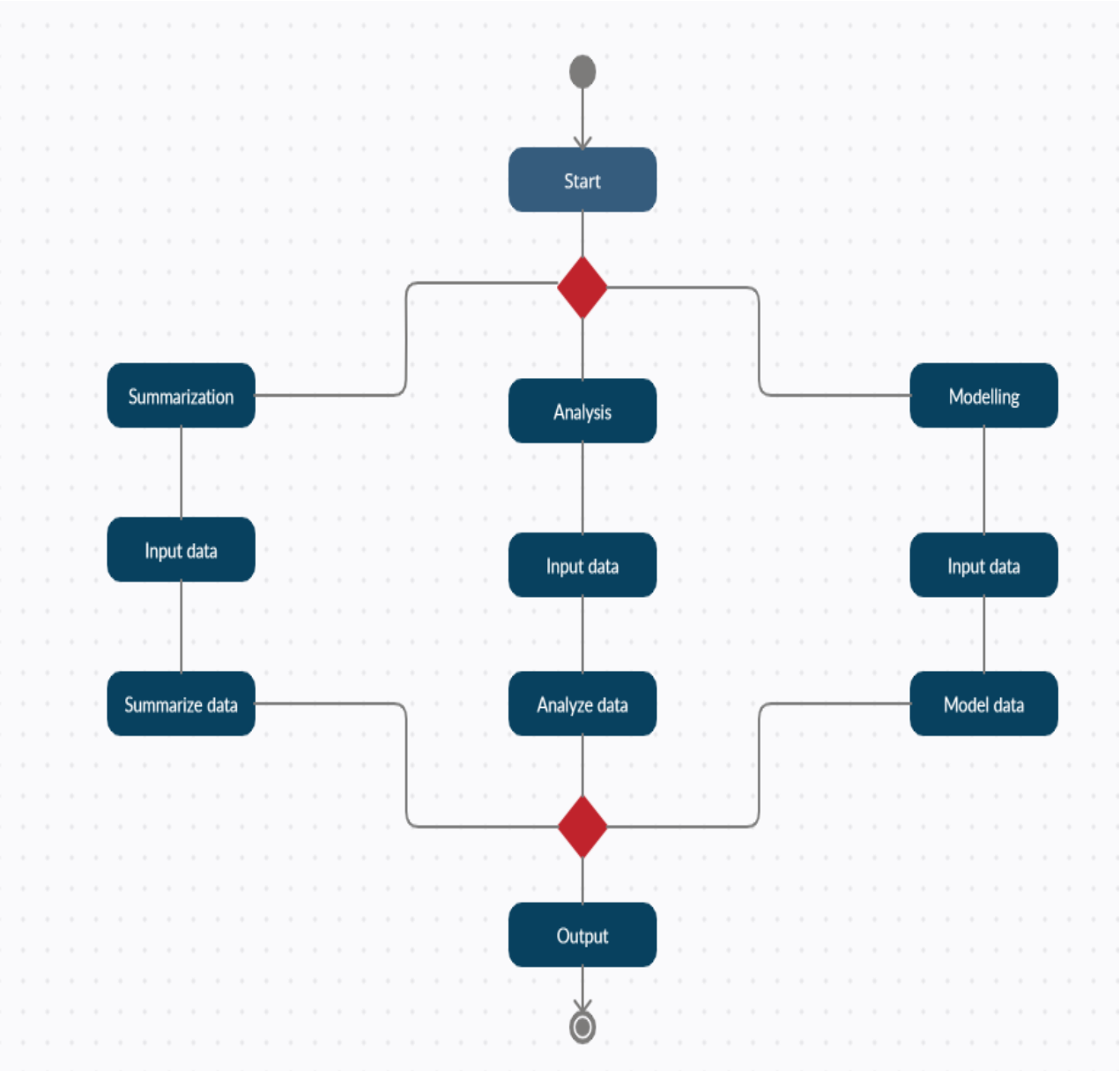
Mathematical computations on the input datasets are performed to obtain the measures of central tendency, measures of deviation and the distribution of data to understand its behaviour. The Relationship between numerical variables is obtained using the concepts of covariance and correlation coefficients while the relationship between a numerical variable and a categorical variable is estimated using the method of Point biserial correlation. The Analysis we carry out using the aforementioned concepts paves the way to model the data and make predictions.

Since this system does fundamental data analysis, people from all fields of study can utilize it to better understand the datasets of their domain.

USE CASE DIAGRAM:



ACTIVITY DIAGRAM:



CLASSES:

welcome.java

This class contains the main method and hence, the program is initiated from this class. This class supports the GUI design of the program. GUI is implemented using the Swing class.

PACKAGES IMPORTED:

javax.swing.*;

java.awt.*;

java.awt.events.*;

COMPONENTS USED:

Jtabbedpane

Jpanel

Jbutton

JTextField

JTextArea

JField

JCheckBox

JRadioButton

JLabel

DESCRIPTION:

The screen displayed to the user contains a section to select the type of data interpretation (Summarization, Analysis, Modelling) preferred. Input tabbed pane is used to for collecting input and output tabbed pane is used for displaying the computed output and interpretation. Each tabbed pane has four panels. Different panels are used for Welcome page, summarization, analysis and modelling interfaces.

The process of data collection from the user, identifying the requirements based on checkboxes checked and radio buttons clicked, listening to the activity of 'submit' button, ensuring that the computation is made and the required output is displayed, is implemented by this class.

The user has the option to work on various data processing options in one execution.

WORKING:

1. Display welcome pages
2. Listen to button click of the user to identify the data processing selected
3. Display the selected data processing panel
4. Accept the input from the user and checkbox information after listening to the 'submit' action
5. The information is passed to the appropriate class for computation.
6. The computed information is received
7. The output is set to JLabels
8. The appropriate panel with outputs is displayed to the user.

sum.java

This module is responsible to carry out all the operations that come under the Summarization part of the system. It calculates values such as Mean, Median, Mode, Variance, Standard deviation and Mean deviation.

WORKING:

- The data that is input in the interface is passed onto this class by the welcome.java class.
- This class gets the array of data elements as a single string and it parses each of them into individual values.
- It feeds them into the user-defined modules such as Mean, Median, Mode, Variance, Standard deviation and Mean deviation in order to calculate them using the statistical formulae.
- The formulae used are as follows:

Mean

$$Mean = \frac{\sum(x_i)}{N}$$

Median

$$\mathrm{Med}(X) = \begin{cases} X[\frac{n}{2}] & \text{if n is even} \\ \frac{(X[\frac{n-1}{2}]+X[\frac{n+1}{2}])}{2} & \text{if n is odd} \end{cases}$$

Mode

In statistics, the mode is the most commonly observed value in a set of data.

Variance

$$Sample\ Variance = \frac{\sum(x_i - \bar{x})^2}{N-1}$$

$$Population\ Variance = \frac{\sum(x_i - \bar{x})^2}{N}$$

- $\bar{x}$ = Mean value of the numerical variable x

Standard Deviation

$$Sample\ Standard\ Deviation = \sqrt{\frac{\sum(x_i - \bar{x})^2}{N-1}}$$

$$Population\ Standard\ Deviation = \sqrt{\frac{\sum(x_i - \bar{x})^2}{N}}$$

- $\bar{x}$ = Mean value of the numerical variable x

Mean Deviation

$$Mean\ Deviation = \frac{1}{N}\sum_{i=1}^{N}|x_i - m(X)|$$

- $m(X)$= Average value of the data set
- x_i= Data values in the set

The output is again passed back onto the welcome.java class so that the output is displayed on the right pane which is dedicated for displaying the outputs.

anal.java

This module is responsible to carry out all the operations that come under the Analysis part of the system. It calculates values such as covariance and correlation between numerical variables and point biserial coefficient between a numerical and a categorical variable.

WORKING:

- The data that is input in the interface is passed onto this class by the welcome.java class.
- This class gets the array of data elements as a single string and it parses each of them into individual values.
- It feeds them into the user-defined modules such as covariance, correlation, point-biserial in order to calculate them using the statistical formulae.
- The formulae used are as follows:

Covariance

$$Sample\ Covariance(x,y) = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{N-1}$$

$$Population\ Covariance(x,y) = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{N}$$

- $\bar{x}$ = Mean value of the numerical variable x
- $\bar{y}$ = Mean value of the numerical variable y

Correlation Coefficient

$$Correlation\ Coefficient, r(x,y) = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}$$

- $\bar{x}$ = Mean value of the numerical variable x
- $\bar{y}$ = Mean value of the numerical variable y

Point Biserial Correlation Coefficient

$$Point\ Biserial\ Correlation\ Coefficient\ r_{pq} = \frac{(\overline{y_1} - \overline{y_2})\sqrt{pq}}{S_y}$$

- p = Proportion for which nominal value is 1.
- q = Proportion for which nominal value is 0.
- $\overline{y_1}$ = Conditional mean of the quantitative or numerical variable y when the nominal score is 1.
- $\overline{y_1}$ = Conditional mean of the quantitative or numerical variable y when the nominal score is 0.
- S_y= Standard deviation of the numerical feature.

The output is again passed back onto the welcome.java class so that the output is displayed on the right pane which is dedicated for displaying the outputs.

model.java

This module is responsible to carry out all the operations that come under the Modeling part of the system. It creates the regression equation for 2 or 3 variables depending upon the need of the user. This class can also utilize the derived regression equation to predict the values for custom values.

WORKING:

- The data that is input in the interface is passed onto this class by the welcome.java class.
- This class gets the array of data elements as a single string and it parses each of them into individual values.
- It feeds them into the user-defined modules such as regression equation, correlation coefficient, multiple regression in order to calculate them using the statistical formulae.
- The formulae used are as follows:

Regression Equation of x on y

$$x - \bar{x} = b_{xy}(y - \bar{y})$$

$$b_{xy} = \frac{cov(x, y)}{{S_y}^2} = r\,\frac{S_x}{S_y}$$

Regression Equation of y on x

$$y - \bar{y} = b_{yx}(x - \bar{x})$$

$$b_{yx} = \frac{cov(y, x)}{{S_x}^2} = r\,\frac{S_y}{S_x}$$

- r = Correlation coefficient between numerical variables x and y
- S_y= Standard deviation of y
- S_x= Standard deviation of x

Regression Equation of x and y and z

$$x - \bar{x} = b_{xyz}(y - \bar{y}) + b_{xzy}(z - \bar{z})$$

$$b_{xyz} = \frac{S_x}{S_y}\left[\frac{r_{xy} - r_{xz}r_{yz}}{1 - {r_{yz}}^2}\right]$$

$$b_{xzy} = \frac{S_x}{S_z}\left[\frac{r_{xz} - r_{xy}r_{yz}}{1 - {r_{yz}}^2}\right]$$

Regression Equation of y on x and z.

$$y - \bar{y} = b_{yxz}(x - \bar{x}) + b_{yzx}(z - \bar{z})$$

$$b_{yxz} = \frac{S_y}{S_x}\left[\frac{r_{yx} - r_{yz}r_{xz}}{1 - {r_{xz}}^2}\right]$$

$$b_{yzx} = \frac{S_y}{S_z}\left[\frac{r_{yz} - r_{yx}r_{xz}}{1 - {r_{xz}}^2}\right]$$

Regression Equation of z on x and y

$$z - \bar{z} = b_{zxy}(x - \bar{x}) + b_{zyx}(y - \bar{y})$$

$$b_{zxy} = \frac{S_z}{S_x}\left[\frac{r_{zx} - r_{yz}r_{xy}}{1 - {r_{xy}}^2}\right]$$

$$b_{zyx} = \frac{S_z}{S_y}\left[\frac{r_{yz} - r_{xz}r_{xy}}{1 - {r_{xy}}^2}\right]$$

- r_{ab}= Coefficient Correlation of numerical variables a and b
- S_a = Standard deviation of numerical variable a
- $\bar{a}$ = Mean value of numerical variable a.

SOURCE CODE:

Connection with sum.java

```
private void sum_submitActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    String summary_input=sum_inp.getText();
    int arr[]={0,0,0,0,0,0};
    if(mean.isSelected()){
       arr[0]=1;
    }
    if(median.isSelected()){
       arr[1]=1;
    }
    if(mode.isSelected()){
       arr[2]=1;
    }
    if(variance.isSelected()){
       arr[3]=1;
    }
    if(std_dev.isSelected()){
       arr[4]=1;
    }
    if(mean_dev.isSelected()){
       arr[5]=1;
    }
    sum s=new sum();
    String val1[]=s.data(summary_input,arr);
    mean_ans.setText(val1[0]);
    median_ans.setText(val1[1]);
    mode_ans.setText(val1[2]);
```

```
        var_pop_ans.setText(val1[3]);
        var_sam_ans.setText(val1[4]);
        std_pop_ans.setText(val1[5]);
        std_sam_ans.setText(val1[6]);
        meandev_ans.setText(val1[7]);
        //sumo.setVisible(true);
        output.setSelectedIndex(1);
    }
```

Connection with anal.java

```
private void anal_submitActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        String analysis_input1=anal_inp1.getText();
        String analysis_input2=anal_inp2.getText();
        int arr[]={0,0,0};
        if(covar.isSelected()){
            arr[0]=1;
        }
        if(corr.isSelected()){
            arr[1]=1;
        }
        if(point_bi.isSelected()){
            arr[2]=1;
        }
        anal a=new anal();
        String val2[]=a.data2(analysis_input1,analysis_input2,arr);
        covar_pop_ans.setText(val2[0]);
        covar_sam_ans.setText(val2[1]);
        corr_ans.setText(val2[2]);
        point_bi_ans.setText(val2[3]);
```

```
        //analo.setVisible(true);
        output.setSelectedIndex(2);
    }
```

Connection with model.java

```
private void model_submitActionPerformed(java.awt.event.ActionEvent evt) {
        int arr[]={0,0,0,0,0};
        if(var2.isSelected()){
            arr[0]=1;
        }
        else if(var3.isSelected()){
            arr[1]=1;
        }
        if(x.isSelected()){
            arr[2]=1;
        }
        else if(y.isSelected()){
            arr[3]=1;
        }
        else if (z.isSelected()){
            arr[4]=1;
        }
        model m=new model();
        String modelling_input1=model_inp1.getText();
        String modelling_input2=model_inp2.getText();
        String val1[];
        if(arr[1]==1){
            String modelling_input3=model_inp3.getText();
            String ind1=indep1.getText();
            String ind2=indep2.getText();
```

```
            String ind=ind1+","+ind2;
            val1=m.data_three(modelling_input1, modelling_input2, modelling_input3, arr,ind);
        }
        else{
            String ind1=indep1.getText();
            val1=m.data_two(modelling_input1, modelling_input2, arr,ind1);
        }
        reg_eq.setText(val1[0]);
        future.setText(val1[1]);
        output.setSelectedIndex(3);
    }
```

Class - sum.java

```
package javaapplication2.home;
import java.util.*;
public class sum {
    //mean
    static float mean(float arr[]){
        float sum=0;
        for(float i:arr){
            sum+=i;
        }
        return sum/arr.length;
    }
    //median
    static float median(float arr[]){
        Arrays.sort(arr);
        int l=arr.length;
        if(l%2!=0){
```

```
        return arr[(l+1)/2];
    }
    else{
        float x=arr[l/2-1]+arr[l/2];
        return x/2;
    }
}
//mode
static float mode(float arr[]){
    float maxValue = 0;
    int maxCount = 0,i, j;
    for (i = 0; i < arr.length; i++) {
        int count = 0;
        for (j = 0; j < arr.length; j++) {
            if (arr[j] == arr[i])
            count++;
        }
        if (count > maxCount) {
            maxCount = count;
            maxValue = arr[i];
        }
    }
    return maxValue;
}
//variance
static float[] variance(float arr[]){
    float var=0;
    float m=mean(arr);
```

```
    for(float i:arr){
        var+=Math.pow(i-m,2);
    }
    float a[]={var/arr.length,var/(arr.length-1)};
    return a;
}
//standard deviation
static float[] standardDeviation(float arr[]){
    float x,y;
    float v[];
    v=variance(arr);
    x=(float)(Math.sqrt(v[0]));
    y=(float)(Math.sqrt(v[1]));
    float a[]={x,y};
    return a;
}
//mean deviation
static float meanDeviation(float arr[]){
    float sum=0;
    float m=mean(arr);
    for(float i:arr){
        sum+=Math.abs(i-m);
    }
    return sum/arr.length;
}
//data collection
//methods collection
//data conversion
```

```
String[] data(String val, int arr_selected[]){
    String val1 = val.replaceAll(" ", "");
    String[] csval = val1.split(",",val1.length());
    int leng = csval.length;
    float[] arr = new float[leng];
    int i = 0;
    for (String str:csval){
        arr[i] = Float.parseFloat(str);
        i++;
    }
    String mean_value="-";
    String median_value="-";
    String mode_value="-";
    String variance_pop_value="-";
    String variance_sam_value="-";
    String standardDeviation_pop_value="-";
    String standardDeviation_sam_value="-";
    String meanDeviation_value="-";
    for(i=0;i<6;i++){
        if(arr_selected[i]!=0 && i==0){
            mean_value=Float.toString(mean(arr));
        }
        if(arr_selected[i]!=0 && i==1){
            median_value=Float.toString(median(arr));
        }
        if(arr_selected[i]!=0 && i==2){
            mode_value=Float.toString(mode(arr));
        }
```

```
            if(arr_selected[i]!=0 && i==3){
                float variance[]=variance(arr);
                variance_pop_value=Float.toString(variance[0]);
                variance_sam_value=Float.toString(variance[1]);
            }
            if(arr_selected[i]!=0 && i==4){
                float standardDeviation_value[]=standardDeviation(arr);
                standardDeviation_pop_value=Float.toString(standardDeviation_value[0]);
                standardDeviation_sam_value=Float.toString(standardDeviation_value[1]);
            }
            if(arr_selected[i]!=0 && i==5){
                meanDeviation_value=Float.toString(meanDeviation(arr));
            }
        }
        String
values[]={mean_value,median_value,mode_value,variance_pop_value,variance_sam_value,standard
Deviation_pop_value,standardDeviation_sam_value,meanDeviation_value};
        return values;
    }
}
```

Class - anal.java

```
package javaapplication2.home;
public class anal {
    static float[] covariance(float num1[], float num2[]){
        float x, y, cv=0;
        int l, i;
        l=num1.length;
        float xd[]=new float[l];
        float yd[]=new float[l];
```

```
    float sum1=0, sum2=0;
    for(i=0; i<l; i++){
        sum1+=num1[i];
        sum2+=num2[i];
    }
    x=sum1/l;
    y=sum2/l;
    for(i=0; i<l; i++){
        xd[i]+=num1[i]-x;
        yd[i]+=num2[i]-y;
    }
    for(i=0; i<l;i++){
        cv+=xd[i]*yd[i];
    }
    float a[]={cv/l,cv/(l-1)};
    return a;
}
static float correlation(float num1[], float num2[]){
    float x, y, cv=0, a;
    int l, i;
    l=num1.length;
    float xd[]=new float[l];
    float yd[]=new float[l];
    float sum1=0, sum2=0;
    for(i=0; i<l; i++){
        sum1+=num1[i];
        sum2+=num2[i];
    }
    x=sum1/l;
    y=sum2/l;
```

```
    for(i=0; i<l; i++){
        xd[i]+=num1[i]-x;
        yd[i]+=num2[i]-y;
    }
    for(i=0; i<l;i++){
        cv+=xd[i]*yd[i];
    }

    float varx=0, vary=0;
    for(i=0; i<l; i++){
        varx+=Math.pow(num1[i]-x,2);
        vary+=Math.pow(num2[i]-y,2);
    }
    x=(float)(Math.sqrt(varx));
    y=(float)(Math.sqrt(vary));
    a=cv/(x*y);
    return a;
}
static float PointBiSerial(float cat[], float num[]){
    int sum0=0, count0=0, sum1=0, count1=0, mean=0;
    for(int i=0;i<cat.length;i++){
        if((int)cat[i]==0){
            sum0+=num[i];
            count0++;
        }
        else{
            sum1+=num[i];
            count1++;
        }
    }
```

```
    float var=0,pbs=0;
    mean=(sum0+sum1)/2;
    for(float i:num){
        var+=Math.pow(i-mean,2);
    }
    float std=(float)Math.sqrt(var/num.length);
    pbs=((sum0/count0)-(sum1/count1))/std;
    return Math.abs(pbs*(float)Math.sqrt(count0*count1/num.length*num.length));
}
String[] data2(String arr1, String arr2,int arr[]){
    String cval1= arr1.replaceAll(" ", "");
    String cval2= arr2.replaceAll(" ", "");
    String[] csval1 = cval1.split(",",cval1.length());
    String[] csval2 = cval2.split(",",cval2.length());
    int leng1 = csval1.length;
    int leng2 = csval2.length;
    float[] num1 = new float[leng1];
    float[] num2 = new float[leng2];
    int i = 0;
    for (String str:csval1){
        num1[i] = Float.parseFloat(str);
        i++;
    }
    i=0;
    for (String str:csval2){
        num2[i] = Float.parseFloat(str);
        i++;
    }
    String covar_pop_value="-";
    String covar_sam_value="-";
```

```
        String corr_value="-";
        String point_bi_value="-";
        for(i=0;i<3;i++){
           if(arr[i]!=0 && i==0){
              float covariance[]=covariance(num1,num2);
              covar_pop_value=Float.toString(covariance[0]);
              covar_sam_value=Float.toString(covariance[1]);
           }
           if(arr[i]!=0 && i==1){
              corr_value=Float.toString(correlation(num1,num2));
           }
           if(arr[i]!=0 && i==2){
              point_bi_value=Float.toString(PointBiSerial(num1,num2));
           }
        }

        String values[]={covar_pop_value,covar_sam_value,corr_value,point_bi_value};
        return values;
    }
}
```

Class – model.java

```
package javaapplication2.home;
import java.util.*;
import java.lang.Math;
public class model {
    public static float[] varcalc(float x[], float y[])
    {
        float sumx = 0, sumy = 0, sumdevx = 0, sumdevy = 0, avgx = 0, avgy = 0;
        int n = x.length;
```

```
for (int i = 0; i<n; i++)
{
   sumx += x[i];
   sumy += y[i];
}

avgx = sumx/(n);
avgy = sumy/(n);

float devx[];
float devy[];
devx = new float[n];
devy = new float[n];

for (int i = 0; i<n; i++)
{
   devx[i] = x[i]-avgx;
   devy[i] = y[i]-avgy;

}

for (int j = 0; j<n; j++)
{
   sumdevx += devx[j];
   sumdevy += devy[j];
}

float devxsquare[];
float devysquare[];
float devxy[];
```

```
    devxsquare = new float[n];
    devysquare = new float[n];
    devxy = new float[n];

    for (int i = 0 ;i<n; i++)
    {
        devxsquare[i] = devx[i]*devx[i];
        devysquare[i] = devy[i]*devy[i];
        devxy[i] = devx[i]*devy[i];
    }

    float sumdevxsquare = 0, sumdevysquare = 0, sumdevxy = 0;

    for (int i = 0; i<n; i++)
    {
        sumdevxsquare += devxsquare[i];
        sumdevysquare += devysquare[i];
        sumdevxy += devxy[i];
    }
    float covxy = (sumdevxy/n) - ((sumdevx/n)*(sumdevy/n));
    float varx = (sumdevxsquare/n)-((sumdevx/n)*(sumdevx/n));
    float vary = (sumdevysquare/n)-((sumdevy/n)*(sumdevy/n));

    float[] ans = new float[]{ covxy,varx,vary,avgx,avgy };
    return (ans);
}
public static String[] regressionequation (float x[], float y[], int a,float var1)
{
    float ansset[] = varcalc(x,y);
```

```
float covxy = ansset[0];
float varx = ansset[1];
float vary = ansset[2];
float avgx = ansset[3];
float avgy = ansset[4];

float byx = covxy/varx;
float bxy = covxy/vary;

String output[]={"","",""};
String p="-";
if (a == 2)
{
   if (byx>1 || byx<0)
   {

      output[0] = "Regression of y on x does not exist";
   }
   else
   {
      output[0] = "y - "+avgy+" = "+byx+" (x - "+avgx+")";
      float q=byx*(var1-avgx)+avgy;
      p=Float.toString(q);
   }
   output[1]=p;
}

if (a==1)
{
   if (bxy>1 || bxy<0)
```

```
        {
            output[0] = "Regression of x on y does not exist";
        }
        else
        {
            output[0] = "x - "+avgx+" = "+bxy+" (y - "+avgy+")";
            float q=bxy*(var1-avgy)+avgx;
            p=Float.toString(q);
        }
        output[1]=p;
    }
    return output;
}


static float correlationcoefficient(float x[], float y[])
{
    float[] ansset = varcalc(x,y);
    float stdx = (float)Math.sqrt(ansset[1]);
    float stdy = (float)Math.sqrt(ansset[2]);
    float covxy = ansset[0];
    float corrcoeff = covxy/(stdx*stdy);
    return (corrcoeff);
}

public static String[] multipleregression(float x[], float y[], float z[], int a, float var1, float var2)
{
    float sumx = 0, sumy = 0, sumdevx = 0, sumdevy = 0, avgx = 0, avgy = 0, sumz = 0, avgz = 0, sumdevz = 0;
    int n = x.length;
    for (int i = 0; i<n; i++)
```

```
{
   sumx += x[i];
   sumy += y[i];
   sumz += z[i];
}

avgx = sumx/(n);
avgy = sumy/(n);
avgz = sumz/n;

float devx[];
float devy[];
float devz[];
devx = new float[n];
devy = new float[n];
devz = new float[n];

for (int i = 0; i<n; i++)
{
   devx[i] = x[i]-avgx;
   devy[i] = y[i]-avgy;
   devz[i] = z[i]-avgz;
}

for (int j = 0; j<n; j++)
{
   sumdevx += devx[j];
   sumdevy += devy[j];
   sumdevz += devz[j];
}
```

```
float devxsquare[];
float devysquare[];
float devzsquare[];

devxsquare = new float[n];
devysquare = new float[n];
devzsquare = new float[n];

for (int i = 0 ;i<n; i++)
{
   devxsquare[i] = devx[i]*devx[i];
   devysquare[i] = devy[i]*devy[i];
   devzsquare[i] = devz[i]*devz[i];
}

float sumdevxsquare = 0, sumdevysquare = 0, sumdevzsquare = 0;

for (int i = 0; i<n; i++)
{
   sumdevxsquare += devxsquare[i];
   sumdevysquare += devysquare[i];
   sumdevzsquare += devzsquare[i];
}

float stdx = (float)Math.sqrt((sumdevxsquare/n)-((sumdevx/n)*(sumdevx/n)));
float stdy = (float)Math.sqrt((sumdevysquare/n)-((sumdevy/n)*(sumdevy/n)));
float stdz = (float)Math.sqrt((sumdevzsquare/n)-((sumdevz/n)*(sumdevz/n)));

float r12 = correlationcoefficient(x,y);
```

```
float r13 = correlationcoefficient(x,z);
float r23 = correlationcoefficient(y,z);

float b123 = ((stdx/stdy)*((r12 - (r13*r23))/(1-(r23*r23))));
float b132 = ((stdx/stdz)*((r13 - (r12*r23))/(1-(r23*r23))));

float b231 = ((stdy/stdz)*((r23 - (r12*r13))/(1-(r13*r13))));
float b213 = ((stdy/stdx)*((r12 - (r23*r13))/(1-(r13*r13))));

float b321 = ((stdz/stdy)*((r23 - (r13*r12))/(1-(r12*r12))));
float b312 = ((stdz/stdx)*((r13 - (r23*r12))/(1-(r12*r12))));

String ans[]= {"","",""};
String p="-";
if (a == 1)
{
   if (b123>1 || b132>1 || b123<0 || b132<0)
   {
      ans[0] = "Regression equation of x on y and z does not exist";
   }
   else
   {
      String x_on_yz = ("(x - "+avgx+") = "+b123+" (y - "+avgy+") + "+b132+"(z - "+avgz+")");
      ans[0] =  x_on_yz;
      float q=b123*(var1-avgy)+b132*(var2-avgz)+avgx;
      p=Float.toString(q);
   }
   ans[1]=p;
}
```

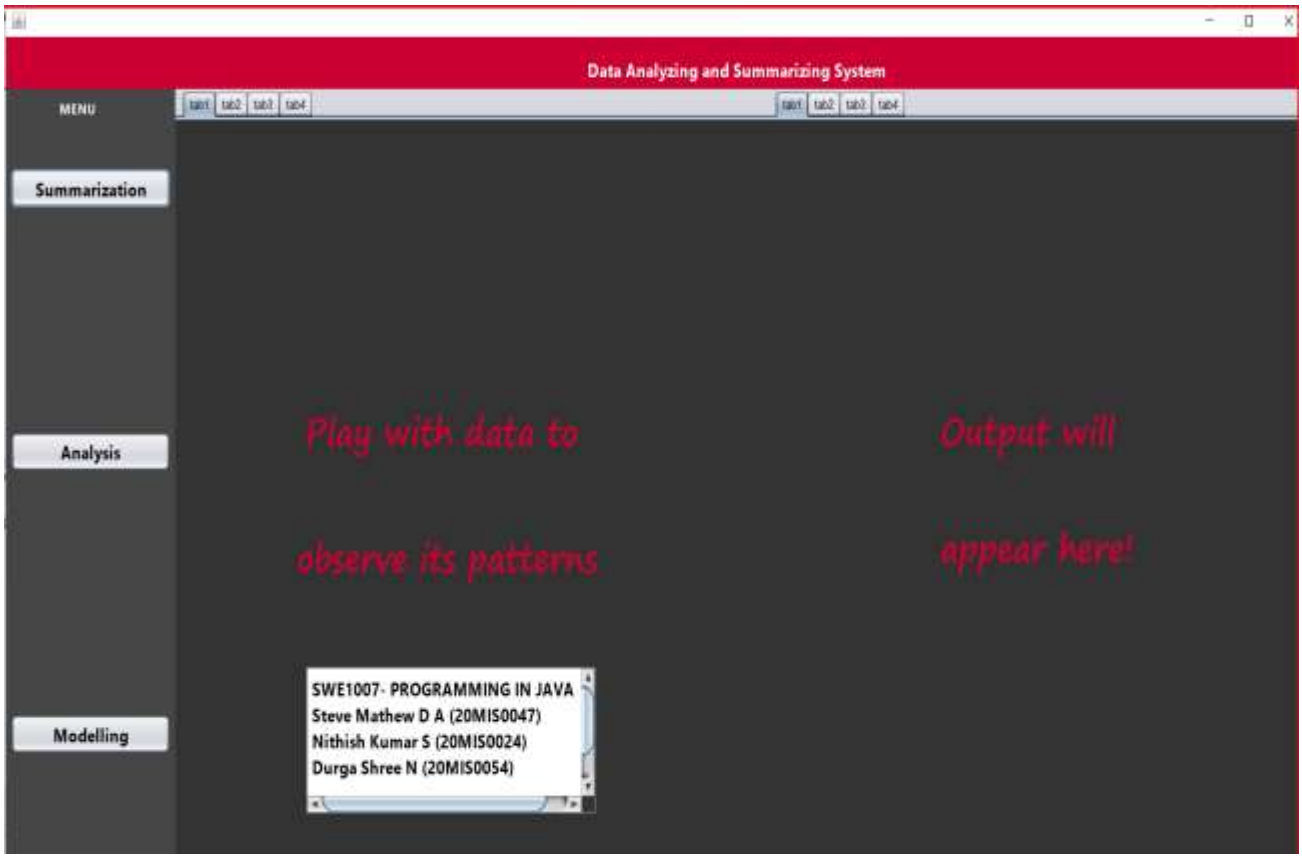
```
if (a == 2)
{
   if (b213>1 || b231>1 || b213<0 || b231<0)
   {
      ans[0] = "Regression equation of y on x and z does not exist";
   }
   else
   {
      String y_on_xz = ("(y - "+avgy+") = "+b231+" (z - "+avgz+") + "+b213+"(x - "+avgx+")");
      ans[0] =  y_on_xz;
      float q=b231*(var2-avgz)+b213*(var1-avgx)+avgy;
      p=Float.toString(q);
   }
   ans[1]=p;
}

if (a==3)
{
   if (b321>1 || b312>1 || b321<0 || b312<0)
   {
      ans[0]= "Regression equation of z on x and y does not exist";
   }
   else
   {
      String z_on_xy = ("(z - "+avgz+") = "+b312+" (x - "+avgx+") + "+b321+"(y - "+avgy+")");
      ans[0] =  z_on_xy;
      float q=b312*(var1-avgx)+b321*(var2-avgy)+avgz;
      p=Float.toString(q);
   }
   ans[1]=p;
```

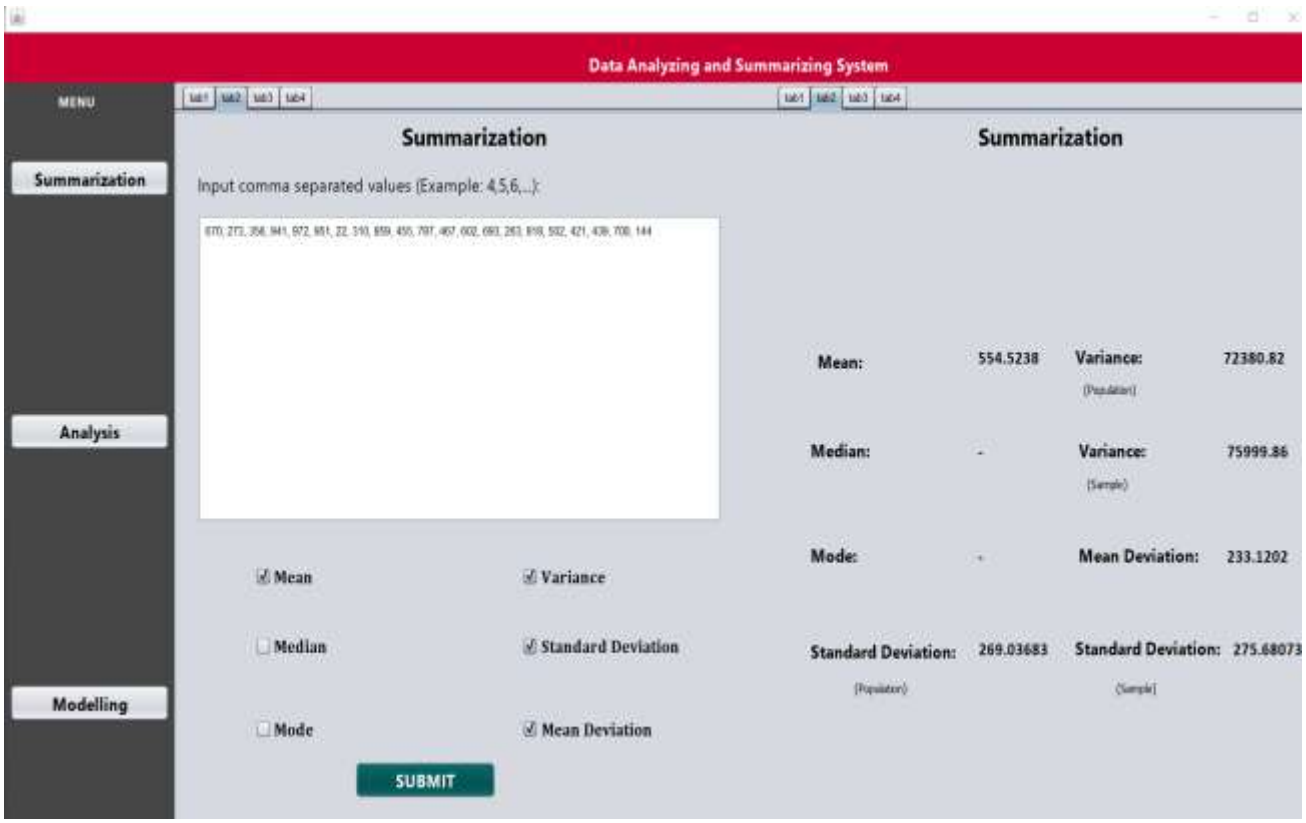
```
    }
    return ans;
}
String[] data_two(String arr1, String arr2,int arr[],String var){
    String cval1= arr1.replaceAll(" ", "");
    String cval2= arr2.replaceAll(" ", "");
    String[] csval1 = cval1.split(",",cval1.length());
    String[] csval2 = cval2.split(",",cval2.length());
    int leng1 = csval1.length;
    int leng2 = csval2.length;
    float[] num1 = new float[leng1];
    float[] num2 = new float[leng2];
    int i = 0;
    for (String str:csval1){
        num1[i] = Float.parseFloat(str);
        i++;
    }
    i=0;
    for (String str:csval2){
        num2[i] = Float.parseFloat(str);
        i++;
    }
    int x=0;
    if(arr[2]==1){
        x=1;
    }
    else{
        x=2;
    }
    float var1=Float.parseFloat(var);
```

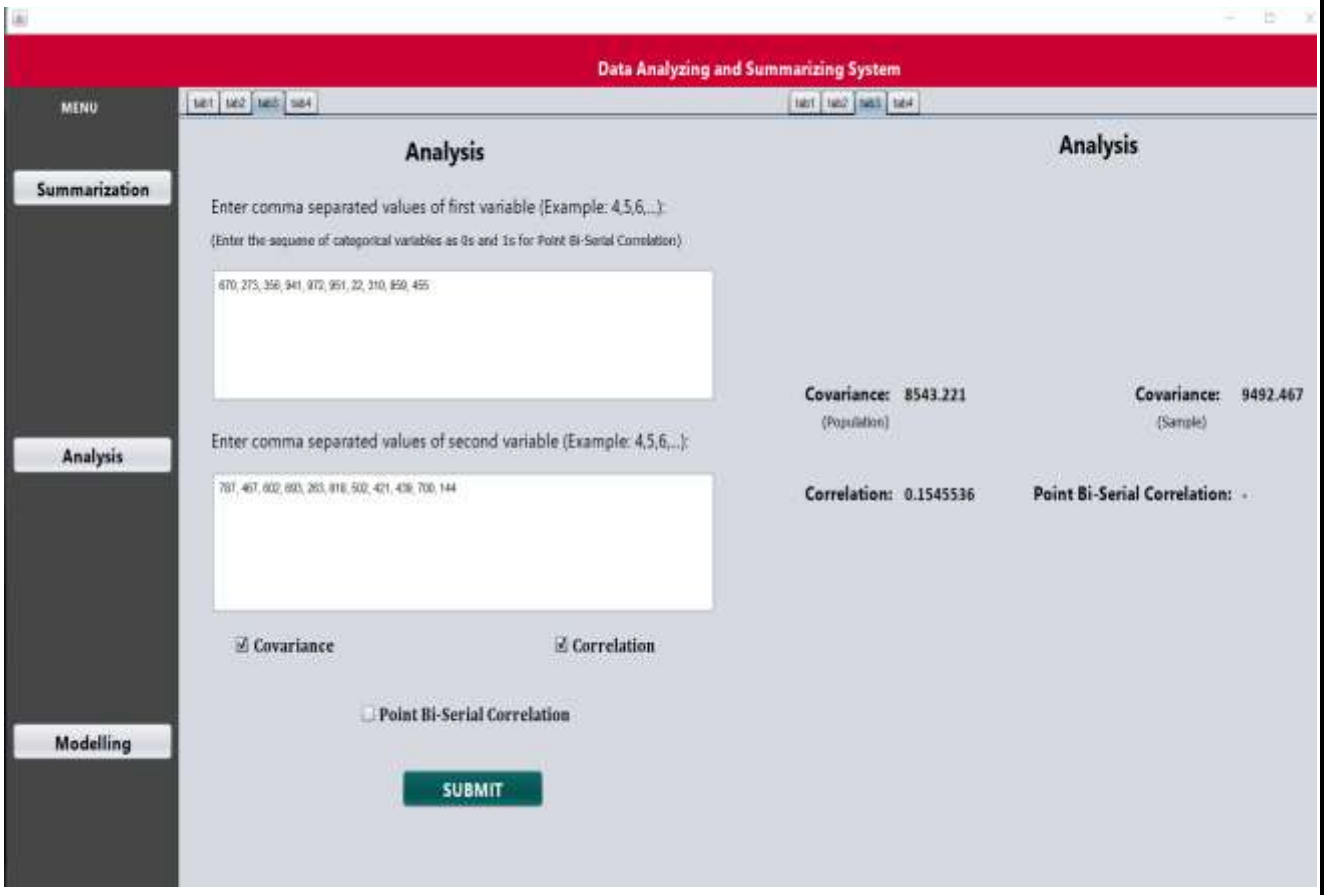
```
    String[] regression_equation_ans=regressionequation(num1,num2,x,var1);
    return regression_equation_ans;
}
String[] data_three(String arr1, String arr2, String arr3,int arr[],String var){
    String cval1= arr1.replaceAll(" ", "");
    String cval2= arr2.replaceAll(" ", "");
    String cval3= arr3.replaceAll(" ", "");
    String cval4= var.replaceAll(" ", "");
    String[] csval1 = cval1.split(",",cval1.length());
    String[] csval2 = cval2.split(",",cval2.length());
    String[] csval3 = cval3.split(",",cval3.length());
    String[] csval4 = cval4.split(",",cval4.length());
    int leng1 = csval1.length;
    int leng2 = csval2.length;
    int leng3 = csval3.length;
    int leng4 = csval4.length;
    float[] num1 = new float[leng1];
    float[] num2 = new float[leng2];
    float[] num3 = new float[leng3];
    float[] num4 = new float[leng4];
    int i = 0;
    for (String str:csval1){
        num1[i] = Float.parseFloat(str);
        i++;
    }
    i=0;
    for (String str:csval2){
        num2[i] = Float.parseFloat(str);
        i++;
    }
```

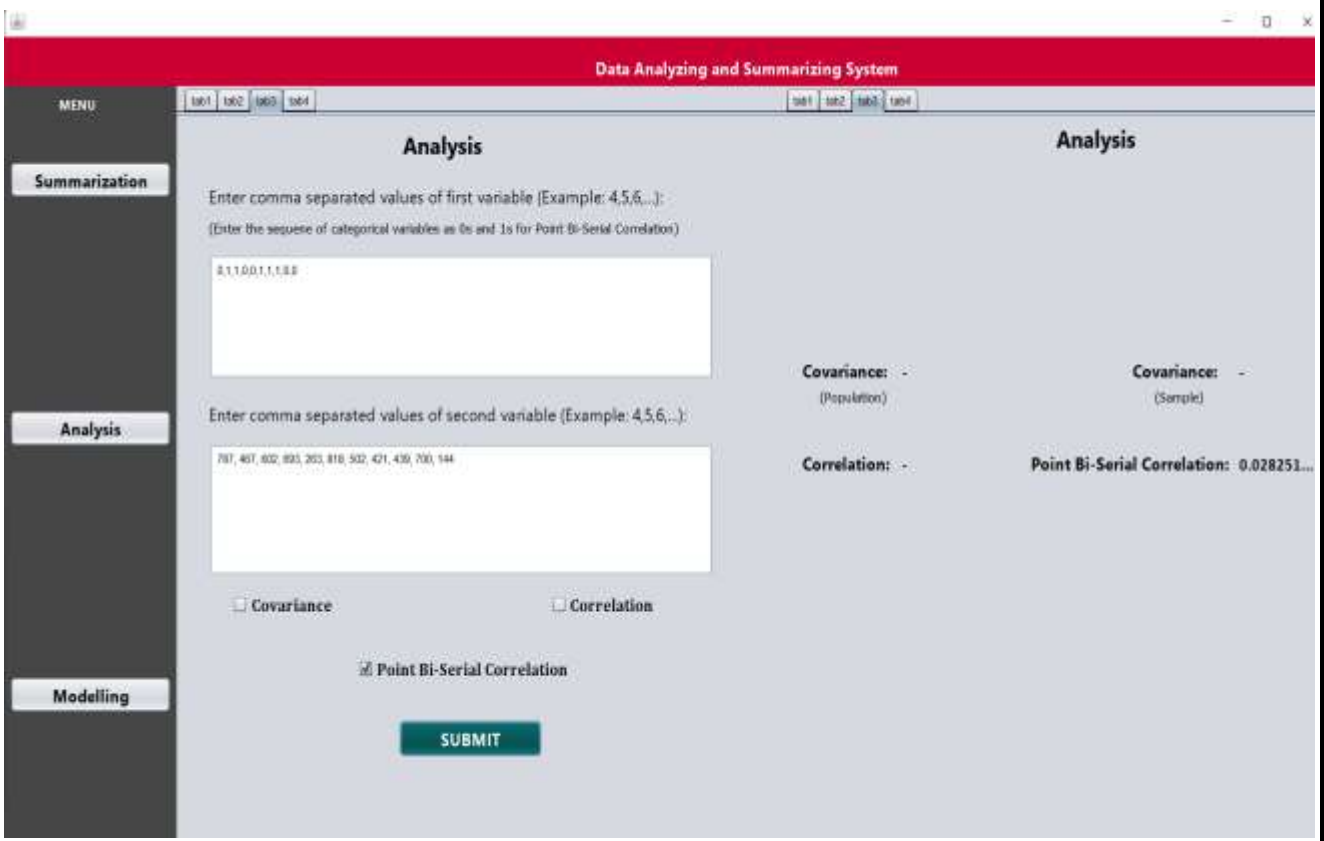
```
        i=0;
        for (String str:csval3){
            num3[i] = Float.parseFloat(str);
            i++;
        }
        i=0;
        for (String str:csval4){
            num4[i] = Float.parseFloat(str);
            i++;
        }
        float var1=num4[0];
        float var2=num4[1];
        int x=0;
        if(arr[2]==1){
            x=1;
        }
        else if(arr[3]==1){
            x=2;
        }
        else{
            x=3;
        }
        String[] regression_equation_ans=multipleregression(num1,num2,num3,x, var1,var2);
        return regression_equation_ans;
    }
}
```

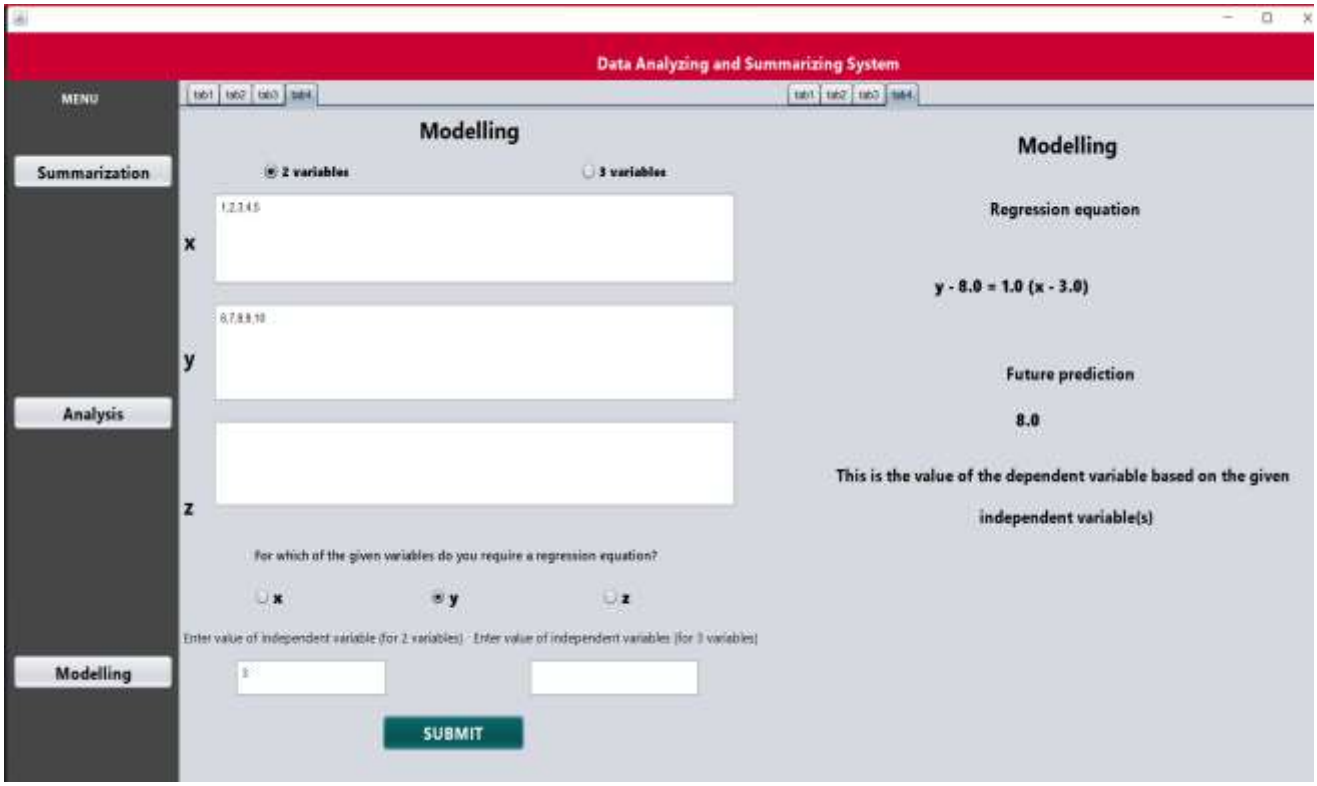
OUTPUT:

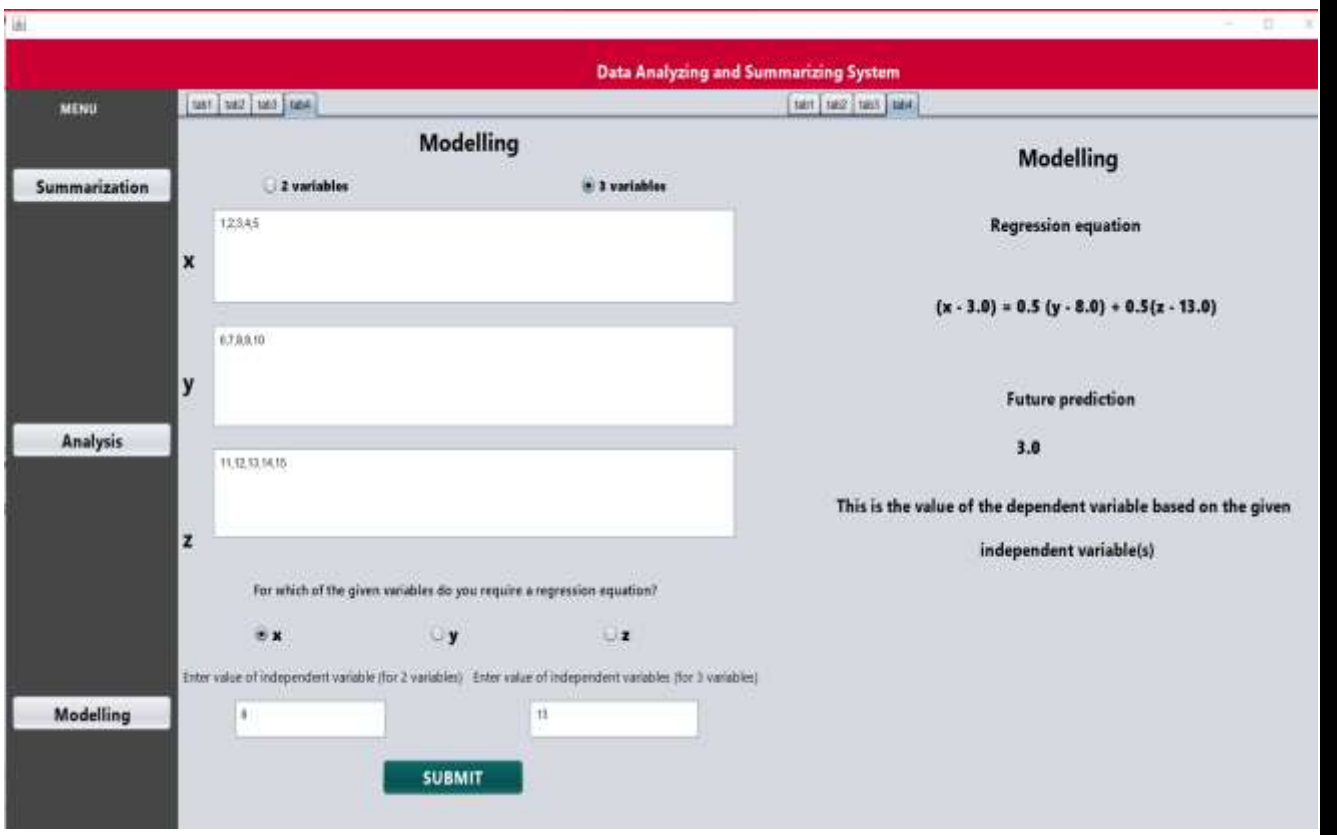

Data Analyzing and Summarizing System
MENU
Summarization
Analysis
Modelling
Play with data to
observe its patterns
Output will
appear here!
SWE1007- PROGRAMMING IN JAVA
Steve Mathew D A (20MIS0047)
Nithish Kumar S (20MIS0024)
Durga Shree N (20MIS0054)


Data Analyzing and Summarizing System
MENU
Summarization
Analysis
Modelling
Summarization
Input comma separated values (Example: 4,5,6,...):
Mean
Median
Mode
Variance
Standard Deviation
Mean Deviation
SUBMIT
Summarization
Mean: 554.5238
Median: -
Mode: -
Variance: 72380.82
(Population)
Variance: 75999.86
(Sample)
Mean Deviation: 233.1202
Standard Deviation: 269.03683
(Population)
Standard Deviation: 275.68073
(Sample)


Data Analyzing and Summarizing System
MENU
Summarization
Analysis
Modelling
Analysis
Enter comma separated values of first variable (Example: 4,5,6,...):
(Enter the sequene of categorical variables as 0s and 1s for Point Bi-Serial Correlation)
670, 275, 356, 941, 972, 951, 22, 310, 859, 455
Enter comma separated values of second variable (Example: 4,5,6,...):
787, 467, 602, 893, 263, 818, 502, 421, 439, 700, 144
Covariance
Correlation
Point Bi-Serial Correlation
SUBMIT
Analysis
Covariance: 8543.221
(Population)
Covariance: 9492.467
(Sample)
Correlation: 0.1545536
Point Bi-Serial Correlation: -


Data Analyzing and Summarizing System
MENU
Summarization
Analysis
Modelling
Analysis
Enter comma separated values of first variable (Example: 4,5,6,...):
(Enter the sequene of categorical variables as 0s and 1s for Point Bi-Serial Correlation)
0,1,1,0,0,1,1,1,0,0
Enter comma separated values of second variable (Example: 4,5,6,...):
787, 467, 602, 893, 263, 818, 502, 421, 439, 700, 144
Covariance
Correlation
Point Bi-Serial Correlation
SUBMIT
Analysis
Covariance: -
(Population)
Covariance: -
(Sample)
Correlation: -
Point Bi-Serial Correlation: 0.028251...


Data Analyzing and Summarizing System
MENU
Summarization
Analysis
Modelling
Modelling
2 variables
3 variables
x
1,2,3,4,5
y
6,7,8,8,10
z
For which of the given variables do you require a regression equation?
x
y
z
Enter value of independent variable (for 2 variables)
Enter value of independent variables (for 3 variables)
3
SUBMIT
Modelling
Regression equation
y - 8.0 = 1.0 (x - 3.0)
Future prediction
8.0
This is the value of the dependent variable based on the given independent variable(s)


Data Analyzing and Summarizing System
MENU
Summarization
Analysis
Modelling
Modelling
2 variables
3 variables
x
1,2,3,4,5
y
6,7,8,8,10
z
11,12,13,14,15
For which of the given variables do you require a regression equation?
x
y
z
Enter value of independent variable (for 2 variables)
Enter value of independent variables (for 3 variables)
11
SUBMIT
Modelling
Regression equation
(x - 3.0) = 0.5 (y - 8.0) + 0.5(z - 13.0)
Future prediction
3.0
This is the value of the dependent variable based on the given independent variable(s)

CONCLUSION:

Tools required to analyse data is inevitable in the current scenario with massive amount of data being generated every second. Various methods of analysing data give insights into various forms and patterns in data. On interpretation, we will be able to solve real world problems and make predictions. Hence, this basic application is developed as an initiative to address the requirement.