Appendix

Appendix A. References:

Matt. (2017). Logistic Regression. Retrieved February 27, 2018, from http://ml-cheatsheet.readthedocs.io/en/latest/logistic_regression.html#gradient-descent

Ng, A. (2017, December 25). Machine learning. Retrieved February 27, 2018, from https://www.coursera.org/learn/machine-learning/home/week/1

Javainterviewpoint. (2017, January 05). How to Read Excel File in Java using POI. Retrieved February 27, 2018, from http://www.javainterviewpoint.com/read-excel-file-java-using-poi/

MrJavaHelp. (2009, July 24). GridBag Layouts JAVA SWING - Arranging objects in a Panel. Retrieved February 27, 2018, from https://www.youtube.com/watch?v=FB_wJpIdA8k

Appendix B: Source code:

```
import java.awt.BorderLayout;

import java.io.*;

import javax.swing.JPanel;

import dk.ange.octave.OctaveEngine;

import dk.ange.octave.OctaveEngineFactory;


        public class tester_IA {

                public static OctaveEngineFactory factory = new OctaveEngineFactory();

                public static OctaveEngine octave;

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

                        octave = new OctaveEngineFactory().getScriptEngine();

                  octavePath octpath = new octavePath();

                  octpath.setVisible(true);


                }
```

```
	}



package IA;
import java.awt.BorderLayout;
import java.awt.FlowLayout;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.Insets;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import javax.swing.JButton;
import javax.swing.JFileChooser;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextArea;
import javax.swing.JTextField;

public class suc_result extends JFrame{
	private JTextArea textArea = new JTextArea(2, 20);
	private JButton back;
	private JButton srchB;
	private JButton outputS;
	private ActionListener listener;
	private JTextField nameSrch;
	private String name;
	private GridBagConstraints c;
	private JPanel panel;
	private boolean exist=false;
	private String sucPath;
	private JPanel panel1 = new JPanel();
	public suc_result(){
		class ChoiceListener implements ActionListener
		{
			public void actionPerformed(ActionEvent event)
			{
				if (event.getSource()==back){   //if the user presses
back button
					directory dir = new directory(); //goes back to
directory
					dir.setVisible(true);
					setVisible(false);
				}
				else if (event.getSource()==srchB){    //if the user
presses search
					name = nameSrch.getText();             //get the
name of the student
					exist=false;
					textArea.setText("");
					getStudSuc();            //call getStudSuc()
				}
				else if (event.getSource()==outputS){  //output the
students' successfulness into a textfile
```

```java
                                outputSuc();       //call  outputSuc()
                          }
                    }
             }
            listener= new ChoiceListener();
            setLayout(new FlowLayout());
            setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            setBounds(460,210,550,350);
            setTitle("Search for student results");
            textArea.setEditable(false);
            msg();
      }
      public void msg(){
            panel = new JPanel(new GridBagLayout());
            c = new GridBagConstraints();
            getContentPane().add(panel);
            JLabel label = new JLabel("The student's name: ");
            c.insets = new Insets( 8, 8, 8, 8);
            c.gridx=0; c.gridy=0;
            panel.add(label, c);
            nameSrch = new JTextField(30);
            c.gridx=1; c.gridy=0;
            panel.add(nameSrch, c);
            srchB=new JButton("Search");
            srchB.addActionListener(listener);
            c.gridx=0; c.gridy=1;
            c.gridwidth=5;
            panel.add(srchB, c);
       back = new JButton("Back");
       back.addActionListener(listener);
       c.gridx=0; c.gridy=3;
       c.gridwidth=8;
       panel.add(back, c);
       c.gridx=1; c.gridy=2;
       c.gridwidth=1;
       outputS = new JButton("Output the students' success into a designated
location");
       outputS.addActionListener(listener);
       panel.add(outputS, c);

      panel1.add(octavePath.swclogo);
       add(panel1, BorderLayout.PAGE_END);

      }
      public void getStudSuc(){ //display the students' successfulness by
searching their name
            c.gridx=0; c.gridy=2; c.gridwidth=1;
            JLabel res = new JLabel("The result: ");
             panel.add(res, c);
       c.gridx=1; c.gridy=2;
       panel.add(textArea, c);
       c.gridx=0; c.gridy=3;
       c.gridwidth=8;
       panel.add(back, c);
       c.gridx=1; c.gridy=4;
       c.gridwidth=1;
       panel.add(outputS, c);
       for (int i=0;i<succ_crit.success.length;i++){ //search for the student
```

```
			if(succ_crit.success[i][0].equalsIgnoreCase(name)){ //if the student
is found
				textArea.append(succ_crit.success[i][0] +": " +
succ_crit.success[i][1]);
				exist=true;	//exist becomes true
			}
			else if (i==succ_crit.success.length-1 && exist==false){ //if the
student is not found
					if (succ_crit.success[i][0].equalsIgnoreCase(name)==false){
						textArea.append("This person does not exist!");
						exist=false; //exist becomes false
					}
			}
		 }
		}
		public void outputSuc(){  //if the user wants to output the students'
successss result into a textfile
			JFileChooser fc = new JFileChooser();
			fc.setCurrentDirectory(new java.io.File("C:/Users/Owner/Desktop"));
//initial directory
			fc.setDialogTitle("Please select where you want to save it: ");
			try{
			if(fc.showSaveDialog(outputS) == JFileChooser.APPROVE_OPTION){
					sucPath = fc.getSelectedFile().getAbsolutePath(); //get
filepath
					FileWriter fw = new FileWriter(sucPath  + ".txt"); //save the
file as a text file
					PrintWriter write = new PrintWriter(fw);
					for (int i=0;i<succ_crit.success.length;i++){	//print
out each student's successfulness
						write.println(succ_crit.success[i][0] +": " +
succ_crit.success[i][1]);
					}
					JOptionPane.showMessageDialog(null, "The file is saved!");
//pop up that tells the user the file is saved.
					fw.close();
					write.close();
			}
			}
			 catch (IOException e) {
					e.printStackTrace();
			}
		}

}



package IA;

public class succ_crit{
		private String studdata[][];
		public static String success [][] = new String [read_data.totalrows][2];
//for storing each student's successfulness
		private int x;
		public succ_crit (String[][] data, int a){
			studdata = data;
			x=a;
		}
```

```
	public int checkSuccess(){
			success[x][0]=studdata[0][0];	//store the name
			double Sc20 =
Double.parseDouble((studdata[0][3]).substring(studdata[0][3].length()-4));
			double Sc24 =
Double.parseDouble((studdata[0][6]).substring(studdata[0][6].length()-4));
			double Sc30 =
Double.parseDouble((studdata[0][4]).substring(studdata[0][4].length()-4));
			double Bio30 =
Double.parseDouble((studdata[0][8]).substring(studdata[0][8].length()-4));
			double
Chem30=Double.parseDouble((studdata[0][10]).substring(studdata[0][10].length()-
4));
			double
Phy30=Double.parseDouble((studdata[0][12]).substring(studdata[0][12].length()-4));
			//if passed 20lv course
			if (((Sc20>=50) || (Sc24>=50)) || (((Sc20==0)) &&
((Sc24)==0))){
				////if passed 30lv course
				if ((Sc30>=50) ||
(Bio30>=50)||(Chem30>=50)||(Phy30>=50)){
					//if the student got at least one 30lv course
above 75%
					if ((Sc30>=75) ||
(Bio30>=75)||(Chem30>=75)||(Phy30>=75)){
						// check if a student got >75% on one 30lv
course but not the other
						if ((Sc30<75 && Sc30>0) || (Bio30<75 &&
Bio30>0)||(Chem30<75 && Chem30>0)||(Phy30<75 && Phy30>0)){
							success[x][1]= "Moderately
Successful";
							return 3;	//these return
statements will return a success level that will be outputted in suc_markX
						}
						else
						success[x][1]= "Maximally Successful";
						return 4;
					}
					else {
						success[x][1]= "Moderately Successful";
						return 3;
					}
				}
				else{//if did not pass 30lv course
					success[x][1]= "Minimally Successful";
					return 2;
				}
			}
			else{ //if did not pass 20lv course
				success[x][1]= "Not Successful";
				return 1;
			}
	}
}
```

package IA;

import java.awt.BorderLayout;

```
import java.awt.GridBagConstraints;

import java.awt.GridBagLayout;

import java.awt.Insets;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.BufferedReader;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import java.io.PrintWriter;

import java.nio.file.Files;

import java.nio.file.Path;

import java.nio.file.Paths;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.Iterator;

import javax.swing.JButton;

import javax.swing.JFileChooser;

import javax.swing.JFrame;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.filechooser.FileNameExtensionFilter;

import org.apache.poi.hssf.OldExcelFormatException;

import org.apache.poi.hssf.usermodel.HSSFSheet;

import org.apache.poi.hssf.usermodel.HSSFWorkbook;
import org.apache.poi.poifs.filesystem.OfficeXmlFileException;

import org.apache.poi.ss.usermodel.Cell;

import org.apache.poi.ss.usermodel.Row;

import org.apache.poi.xssf.usermodel.XSSFSheet;

import org.apache.poi.xssf.usermodel.XSSFWorkbook;


public class read_data extends JFrame

{

private String filePath;

private  JPanel panel = new JPanel(new GridBagLayout());

private JPanel panel1 = new JPanel();

private GridBagConstraints c;

private ActionListener listener;

private JButton fileopen;

private JButton addFile;

private JButton cont;

private String coursNames[] = {"ELLSc",
"Sc10","Sc20","Sc30","Sc14","Sc24","Bio20","Bio30","Chm20","Chm30","Phys20","Phys3
0","Math10C","Math103","Math201","Math202","Math203","Math301","Math302","Math31
"};

private int Rows;

public static int totalrows;

public String stud_data[][];

private ArrayList <String> excelPath = new ArrayList <String>();

int rowC=0;

private FileInputStream fileInputStream; //is where to read excel files

public int i=0;

private PrintWriter output;
private FileWriter fw;
private FileWriter writer;
private PrintWriter out;
private FileWriter writer1;
private PrintWriter out1;
private FileWriter writer2;
private PrintWriter out2;

public read_data()
{
        class ChoiceListener implements ActionListener
        {
                public void actionPerformed(ActionEvent event)
                {
                        if(event.getSource()==fileopen){      //if click "open excel"
                        chooseFile();  // go choose
                        }
                        else if(event.getSource()==addFile){ //if click "add more excel"
                                chooseNewFile();      //choose again
                        }
                        else if(event.getSource()==cont){     //go to the next class
                                next();
                        }
                }
        }
        listener = new ChoiceListener();
        setBounds(553,250,300,200);
```

```
            setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

            setTitle("Program");

            setVisible(true);

            add(panel);

            fileOpenB();

}


public void fileOpenB(){

           c = new GridBagConstraints();

           fileopen = new JButton("Please choose the excel document");

           fileopen.addActionListener(listener);

           c.gridx=0; c.gridy=0;

           panel.add(fileopen, c);

           panel1.add(octavePath.swclogo);

            add(panel1, BorderLayout.PAGE_END);

            //return panel;

}


public void chooseFile(){ // read the number of files and store them like: data1, data2 if only
2 files are selected. make am integer a static thing

                    JFileChooser fc = new JFileChooser();

                    fc.setCurrentDirectory(new java.io.File("C:/Users/Owner/Desktop")); //initial
directory

                    fc.setDialogTitle("Please choose the excel document: ");

                    // allowed types of Excel format

                    FileNameExtensionFilter filter = new FileNameExtensionFilter("Excel
file","xls","xlt","xlsx","xlsm","xltx","xltm","xlam");

                    fc.setFileFilter(filter);
```

```
        //try{
        if(fc.showOpenDialog(fileopen) == JFileChooser.APPROVE_OPTION){
            filePath = fc.getSelectedFile().getAbsolutePath(); //get filepath
            //excelNum++;
            //System.out.println(filePath);
            excelPath.add(filePath);
            fileopen.setVisible(false);
            c.insets = new Insets( 4, 4,4, 4);
            addFile = new JButton ("add another excel file");
            c.gridx=0; c.gridy=0;
            panel.add(addFile, c);
            addFile.addActionListener(listener);
            cont = new JButton("continue");
            c.gridx=0; c.gridy=1;
            cont.addActionListener(listener);
            panel.add(cont, c);
        }
}

public void chooseNewFile(){ // read the number of files and store them like: data1, data2 if
only 2 files are selected. make am integer a static thing
    JFileChooser fc = new JFileChooser();
    fc.setCurrentDirectory(new java.io.File("C:/Users/Owner/Desktop")); //initial
directory
    fc.setDialogTitle("Please choose the excel document: ");
    // allowed types of Excel format
    FileNameExtensionFilter filter = new FileNameExtensionFilter("Excel
file","xls","xlt","xlsx","xlsm","xltx","xltm","xlam");
```

```
        fc.setFileFilter(filter);

        if(fc.showOpenDialog(fileopen) == JFileChooser.APPROVE_OPTION){

                filePath = fc.getSelectedFile().getAbsolutePath(); //get filepath

                excelPath.add(filePath);

                fileopen.setVisible(false);

        }

}


public void next(){

        try{

                for (int a= 1;a<=excelPath.size();a++){        //to get the totalrows

                        int rows=0;     //the number of rows in this excel document

                        String path = excelPath.get(a-1);

                        String lastLet = path.substring(path.length()-2, path.length());

                        fileInputStream = new FileInputStream(new File(excelPath.get(a-1)));

                        //checking the excel version

                        if(lastLet.equals("sx") || lastLet.equals("sm") || lastLet.equals("tx")||
lastLet.equals("tm") || lastLet.equals("am")){

                                XSSFWorkbook wb = new XSSFWorkbook(fileInputStream);

                                XSSFSheet sheet = wb.getSheetAt(0);

                           rows=sheet.getPhysicalNumberOfRows()-2;

                           totalrows += rows;  //add to the totalrows

                        }

                                else if (lastLet.equals("ls") || lastLet.equals("lt")){

                                        HSSFWorkbook workbook = new
HSSFWorkbook(fileInputStream);

                                         HSSFSheet sheet = workbook.getSheetAt(0);

                                           rows=sheet.getPhysicalNumberOfRows()-2;
                                        totalrows += rows; //add to the totalrows
                                    }
                }
        for (int aa=1;aa<=excelPath.size();aa++){     //for each excel file, choose either
readFile1() or readFile2()
                String path = excelPath.get(aa-1); //get the file path
                String lastLet = path.substring(path.length()-2, path.length()); //get the last
letter of the filepath
                fileInputStream = new FileInputStream(new File(excelPath.get(aa-1))); //read
this file
            fw = new FileWriter(octavePath.octPath + "\\data_mark" + aa +".txt"); //output the
file to as data_markX
                output = new PrintWriter(fw);
                writer = new FileWriter(octavePath.octPath + "\\suc_mark" + aa +".txt") ;
//output the file to as suc_markX
                out = new PrintWriter(writer);
                //checking the excel version
                if(lastLet.equals("sx") || lastLet.equals("sm") || lastLet.equals("tx")||
lastLet.equals("tm") || lastLet.equals("am"))
                readFile2();
                else if (lastLet.equals("ls") || lastLet.equals("lt"))
                    readFile1();
                else
                    JOptionPane.showMessageDialog(null, "No files support the excel
reader. Please restart program and select a newer version of excel file!");
        }


        combineFile(); //combine the files
        directory dir = new directory();
        dir.setVisible(true);
```

```
            setVisible(false);

            }

             catch (IOException e) {

                        e.printStackTrace();

            }

            catch (OfficeXmlFileException e){  //cant read the file type

                        e.printStackTrace();

            }

            catch (OldExcelFormatException e){ //if the Excel version too old

                        JOptionPane.showMessageDialog(null, "The version of Excel is outdated.
Please choose a newer version of Excel!");

            }

}


public void combineFile() throws IOException{

    fw = new FileWriter(octavePath.octPath + "\\data_mark.txt"); //output the file as
"data_mark" so octave can read it

            output = new PrintWriter(fw);

            //the data in each line outputted into data_mark (final) txt is the same as the lines in
data_markX. This loop will transfer all the lines

            //in data_markX into data_mark.

            for (int q=1;q<=excelPath.size();q++){

                        BufferedReader buf = new BufferedReader(new
FileReader(octavePath.octPath + "\\data_mark" + q +".txt")); //read data_markX file

                        String line  ;

        while ((line = buf.readLine()) != null) {   //copy the entire line into data_mark

            output.println(line);

          }
        buf.close();

        Path delPath = Paths.get(octavePath.octPath + "\\data_mark" + q +".txt");

        Files.deleteIfExists(delPath);      //delete the temporary file data_markX

           }

           output.close();

           fw.close();

     writer = new FileWriter(octavePath.octPath + "\\suc_mark.txt"); //output the file as
"data_mark" so octave can read it

           out = new PrintWriter(writer);

     writer1 = new FileWriter(octavePath.octPath + "\\suc_mark_round1.txt"); //output the file
as "data_mark_round1" so octave can read it

           out1 = new PrintWriter(writer1);

     writer2 = new FileWriter(octavePath.octPath + "\\suc_mark_round2.txt"); //output the file
as "data_mark_round2" so octave can read it

           out2 = new PrintWriter(writer2);

           for (int q=1;q<=excelPath.size();q++){

                      BufferedReader buf = new BufferedReader(new
FileReader(octavePath.octPath + "\\suc_mark" + q +".txt")); //read suc_markX file

                      String line  ;

        while ((line = buf.readLine()) != null) {

           if((line.charAt(line.length()-1))>='3'){ //if the student's success level is bigger than 3,
they go to round1

           //create a stringbuilder that will make a new string that is same as the line in
suc_markX, but will change the last character to be 1 or 0

                      StringBuilder newl = new StringBuilder(line);

                      newl.setCharAt(line.length()-1, '1');   //Parse 1 being true (success level >=3).

                      out.println(newl);                                          //print to
suc_mark

           //create another stringbuilder that will change the last character to be 1 or 0 for round1

                      StringBuilder newl1 = new StringBuilder(line);
        if((line.charAt(line.length()-1))=='3'){ //for round1, if success level is 3, put 0
(false)
            newl1.setCharAt(line.length()-1, '0');
            out1.println(newl1);
        }
        else if((line.charAt(line.length()-1))=='4'){ //for round1, if success level is 4,
put 1 (true)
            newl1.setCharAt(line.length()-1, '1');
            out1.println(newl1);
    }
   }
    else if((line.charAt(line.length()-1))<='2'){ //if the student's success level is less than
3, they go to round2
    //create a stringbuilder that will make a new string that is same as the line in
suc_markX, but will change the last character to be 1 or 0
        StringBuilder newl = new StringBuilder(line);
        newl.setCharAt(line.length()-1, '0');   //0 being false, and print to suc_mark
        out.println(newl);
    //create another stringbuilder that will change the last character to be 1 or 0 for round1
        StringBuilder newl1 = new StringBuilder(line);
        if((line.charAt(line.length()-1))=='2'){ //for round2, if success level is 2 put 1
            newl1.setCharAt(line.length()-1, '1');
            out2.println(newl1);
        }
        else if((line.charAt(line.length()-1))=='1'){ //for round2, if success level is 1
put 0
            newl1.setCharAt(line.length()-1, '0');
            out2.println(newl1);
    }
```

```
          }
            }
        buf.close();
        Path delPath = Paths.get(octavePath.octPath + "\\suc_mark" + q +".txt");
        Files.deleteIfExists(delPath);     //delete the temporary file data_markX
            }
            out.close();
            writer.close();
            out1.close();
            writer1.close();
            out2.close();
            writer2.close();
}
public void readFile1() throws IOException{
            HSSFWorkbook workbook = new HSSFWorkbook(fileInputStream);
             HSSFSheet sheet = workbook.getSheetAt(0);
             Iterator<Row> rowIterator = sheet.iterator();
               Rows=sheet.getPhysicalNumberOfRows()-2;
               stud_data= new String [1][21]; //make a totalrows x 21 size matrix
               rowIterator.next();
                      rowIterator.next();
               while (rowIterator.hasNext()) //assume the file is correctly chosen
               {
                  Row row = rowIterator.next();
                  // Iterating through Each column of Each Row
                  for (int i=0;i<21;i++) // Checking the cell format
                  {
                      Cell cell = row.getCell(i);
                      String cellValue;
                        // Checking the cell format
                      if(cell==null || cell.getCellType()==Cell.CELL_TYPE_BLANK){
                                  cellValue="0.00";
                       stud_data[0][i]= coursNames[i-1] + "-" + cellValue;


                      }
                      else {
                       if(cell.getCellType()==Cell.CELL_TYPE_STRING){
                                   cellValue=cell.getStringCellValue();
                                   stud_data[0][i]= cellValue;
                                  // System.out.println(cellValue);


                       }
                       else if (cell.getCellType()==Cell.CELL_TYPE_NUMERIC){
                                   cellValue=Double.toString(cell.getNumericCellValue());
                                   stud_data[0][i]= coursNames[i-1] + "-" + cellValue;


                       }
                      }
                  }
                                  double
EllSc=Double.parseDouble((stud_data[0][1]).substring(stud_data[0][1].length()-4));
                                  double
Math10=Double.parseDouble((stud_data[0][13]).substring(stud_data[0][13].length()-4));
                                  double
Math103=Double.parseDouble((stud_data[0][14]).substring(stud_data[0][14].length()-4));
```

```
                    double
Math201=Double.parseDouble((stud_data[0][15]).substring(stud_data[0][15].length()-4));

                    double
Math202=Double.parseDouble((stud_data[0][16]).substring(stud_data[0][16].length()-4));

                    double
Math203=Double.parseDouble((stud_data[0][17]).substring(stud_data[0][17].length()-4));

                    double
Sc14=Double.parseDouble((stud_data[0][5]).substring(stud_data[0][5].length()-4));

                    double
Sc10=Double.parseDouble((stud_data[0][2]).substring(stud_data[0][2].length()-4));

                    if (EllSc>0 &&
(Math10>0||Math103>0||Math201>0||Math202>0||Math203>0)){ //if the student took a math
course and ELLScience

                            output.print(EllSc +",");

                            out.print(EllSc+","); //printing it to the suc_mark

                            if(Math10>0){            //if the student took math10

                                    output.print(Math10+",");

                                    out.print(Math10+",");

                            }

                            else if (Math10==0){  //if the student did not take math10 but
took another math course

                                    if(Math103>0){

                                            output.print(Math103+",");

                                            out.print(Math103+",");

                                            }

                                    else if (Math201>0){

                                            output.print(Math201+",");

                                            out.print(Math201+",");

                                            }

                                    else if (Math202>0){
```

```
                        output.print(Math202+",");
                        out.print(Math202+",");
                    }
                    else if (Math203>0){
                        output.print(Math203+",");
                        out.print(Math203+",");
                    }
                }

                if (Sc10>0){    //if the student took Sc10, and 1 being true
                    if(Sc14>0) //if the student took Sc14 during their
second semester, 0 being false
                        output.println("0");
                    else
                        output.println("1");
                }
                else
                    output.println("0"); //if the student did not take Sc10,
and 0 being false

              succ_crit succrit = new succ_crit(stud_data, i);
              out.println(succrit.checkSuccess());
              i++;
            }
            else {   //still put the student in successful crit, but not in data set
                 succ_crit succrit = new succ_crit(stud_data, i);
             succrit.checkSuccess();
             i++;
```

```
                    }



            }
               fileInputStream.close();
               fileInputStream.close();
               fw.close();
               output.close();
               writer.close();
               out.close();
}
public void readFile2() throws IOException{
            XSSFWorkbook wb = new XSSFWorkbook(fileInputStream);
    XSSFSheet sheet = wb.getSheetAt(0);
    Iterator<Row> rowIterator = sheet.iterator();
    Rows=sheet.getPhysicalNumberOfRows()-2;
    stud_data= new String [1][21]; //make a totalrows x 21 size matrix
    rowIterator.next();
            rowIterator.next();
    while (rowIterator.hasNext()) //assume the file is correctly chosen
    {
        Row row = rowIterator.next();
        // Iterating through Each column of Each Row
        for (int i=0;i<21;i++) // Checking the cell format
        {
            Cell cell = row.getCell(i);
            String cellValue;
              // Checking the cell format
            if(cell==null || cell.getCellType()==Cell.CELL_TYPE_BLANK){
                        cellValue="0.00";
             stud_data[0][i]= coursNames[i-1] + "-" + cellValue;


            }
            else {
             if(cell.getCellType()==Cell.CELL_TYPE_STRING){
                         cellValue=cell.getStringCellValue();
                         stud_data[0][i]= cellValue;
                        // System.out.println(cellValue);


             }
             else if (cell.getCellType()==Cell.CELL_TYPE_NUMERIC){
                         cellValue=Double.toString(cell.getNumericCellValue());
                         stud_data[0][i]= coursNames[i-1] + "-" + cellValue;


             }
            }
        }
                        double
EllSc=Double.parseDouble((stud_data[0][1]).substring(stud_data[0][1].length()-4));
                        double
Math10=Double.parseDouble((stud_data[0][13]).substring(stud_data[0][13].length()-4));
                        double
Math103=Double.parseDouble((stud_data[0][14]).substring(stud_data[0][14].length()-4));
                        double
Math201=Double.parseDouble((stud_data[0][15]).substring(stud_data[0][15].length()-4));
                        double
Math202=Double.parseDouble((stud_data[0][16]).substring(stud_data[0][16].length()-4));
```

```
		double
Math203=Double.parseDouble((stud_data[0][17]).substring(stud_data[0][17].length()-4));

		double
Sc14=Double.parseDouble((stud_data[0][5]).substring(stud_data[0][5].length()-4));

		double
Sc10=Double.parseDouble((stud_data[0][2]).substring(stud_data[0][2].length()-4));

		if (EllSc>0 &&
(Math10>0||Math103>0||Math201>0||Math202>0||Math203>0)){ //if the student took a math
course and ELLScience

			output.print(EllSc +","); //printing to data_mark

			out.print(EllSc+","); //printing it to the suc_mark

			if(Math10>0){		//if the student took math10

				output.print(Math10+",");

				out.print(Math10+",");

			}

			else if (Math10==0){  //if the student did not take math10 but took
another math course

				if(Math103>0){

					output.print(Math103+",");

					out.print(Math103+",");

					}

				else if (Math201>0){

					output.print(Math201+",");

					out.print(Math201+",");

					}

				else if (Math202>0){

					output.print(Math202+",");

					out.print(Math202+",");

				}
```

```
                                        else if (Math203>0){

                                                  output.print(Math203+",");

                                                  out.print(Math203+",");

                                        }

                              }

                              if (Sc10>0){    //if the student took Sc10, and 1 being true

                                        if(Sc14>0) //if the student took Sc14 during their second
semester, 0 being false

                                                  output.println("0");

                                        else

                                                  output.println("1");

                              }

                              else

                                        output.println("0"); //if the student did not take Sc10, and 0
being false

                          succ_crit succrit = new succ_crit(stud_data, i);

                          out.println(succrit.checkSuccess());                    //determine  the
successulfness of the student, ranges from 1 to 4

                          i++;

                    }

                    else {   //still put the student in successful crit, but not in data set

                               succ_crit succrit = new succ_crit(stud_data, i);

                         succrit.checkSuccess();

                         i++;

                    }

}

   fileInputStream.close();

   fw.close();
```

```
    output.close();

    writer.close();

    out.close();

}

}


package IA;


import java.awt.BorderLayout;

import java.awt.Graphics;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.IOException;

import javax.swing.ImageIcon;

import javax.swing.JButton;

import javax.swing.JFileChooser;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;


public class octavePath extends JFrame{

        private JPanel panel = new JPanel();

        private JPanel panel1 = new JPanel();

        private ActionListener listenerO;

        private JButton octpathB;

        public static JLabel swclogo = new JLabel();

        public static String octPath;
public octavePath(){
        class ChoiceListener implements ActionListener
        {
                public void actionPerformed(ActionEvent event)
                {
                        if(event.getSource()==octpathB){
                        try {
                                getOctPath();
                        } catch (IOException e) {
                                e.printStackTrace();
                        }
                        }
                }
        }
        listenerO= new ChoiceListener();
        setVisible(true);
        setBounds(460,234,600,230);
        setTitle("Octave path");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        octpathGUI();
}

public void octpathGUI(){


        octpathB = new JButton("Please choose the location of the octave files: ");
        octpathB.addActionListener(listenerO);
        panel.add(octpathB);
```

```
                    //panel.add(swclogo);

                    swclogo.setHorizontalAlignment(JLabel.RIGHT);

                    add(panel, BorderLayout.NORTH);

                    swclogo.setIcon(new
ImageIcon(octavePath.class.getResource("/logo/swclogo.jpg")));

                   panel1.add(swclogo);

                    add(panel1, BorderLayout.PAGE_END);



          }



          public void getOctPath() throws IOException{          //pops up filechooser to let the
user to select the path of octave files

                   JFileChooser fileChooser = new JFileChooser();

                   fileChooser.setDialogTitle("the location of the octave files: ");

                    fileChooser.setCurrentDirectory(new
java.io.File("C:/Users/Owner/Desktop"));

                    fileChooser.setFileSelectionMode(JFileChooser.DIRECTORIES_ONLY);

                   if(fileChooser.showOpenDialog(octpathB) ==
JFileChooser.APPROVE_OPTION){ //if the location is confirmed

                             octPath=fileChooser.getSelectedFile().getAbsolutePath();

                             moveToRead();

                   }

          }

          public void moveToRead() throws IOException{

                   read_data read = new read_data();

                   setVisible(false);

                   read.setVisible(true);
```

```
        }


    }


package IA;

import java.awt.BorderLayout;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.Insets;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;

public class directory extends JFrame{
        private JPanel panel = new JPanel(new GridBagLayout());
        private ActionListener listener;
        private JButton suCrit;
        private JPanel panel1 = new JPanel();
        private JButton rec;
        public directory(){
                class ChoiceListener implements ActionListener
                {
                        public void actionPerformed(ActionEvent event)
                        {
                                if(event.getSource()==suCrit){ //if "To check
students' successfulness" is clicked
                                        suc_result res = new suc_result();
        //go to success result class
                                        setVisible(false);
                                        res.setVisible(true);
                                }
                                else if (event.getSource()==rec){       //if "To receive
a recommendation made by this prorgam" is clicked
                                    call_octave callO = new call_octave();//go to
call_octave class
                                    callO.setVisible(true);
                                    setVisible(false);
                                }
                        }
                }
                listener= new ChoiceListener();
                setVisible(true);
                setBounds(460,234,600,230);
                setTitle("Directory");
                setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                choices();
        }

        public void choices(){// let user choose 2 choices
                add(panel, BorderLayout.NORTH);
```

```
		GridBagConstraints c = new GridBagConstraints();
		c.insets = new Insets( 15, 8, 15, 8);
		JLabel label = new JLabel("Please select one of the following: ");
		c.gridx=0; c.gridy=0;
		c.gridwidth=10;
		panel.add(label, c);
		suCrit = new JButton("To check students' successfulness");
		rec = new JButton("To receive a recommendation made by this 
prorgam");
		c.gridx=0; c.gridy=1;
		c.gridwidth=2;
		panel.add(suCrit, c);
		c.gridx=2; c.gridy=1;
		c.gridwidth=2;
		panel.add(rec, c);
		suCrit.addActionListener(listener);
		rec.addActionListener(listener);
		panel1.add(octavePath.swclogo);
		 add(panel1, BorderLayout.PAGE_END);
	}

}


package IA;

import java.awt.BorderLayout;
import java.awt.FlowLayout;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.Insets;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import java.text.DecimalFormat;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import dk.ange.octave.exception.OctaveEvalException;
import dk.ange.octave.type.OctaveDouble;
public class call_octave extends JFrame{
	private ActionListener listenerC;
	private JButton back;
	private JButton getP;
	private JTextField mathG;
	private JTextField sciG;
	private GridBagConstraints c;
	private JPanel panel;
	private JPanel panel1 = new JPanel();
	private JTextArea textArea = new JTextArea(2, 10);
	private JTextArea acuText = new JTextArea(2, 10);
	private JTextArea sucText = new JTextArea(2, 10);
	private JTextArea suc1Text = new JTextArea(2, 10);
	private double MathMark, ScMark;
	private  DecimalFormat df = new DecimalFormat(".##");
	private boolean first = true;
	private JLabel label3 = new JLabel();
	private JLabel label = new JLabel();
	private JLabel label1 = new JLabel();
	private JLabel labe = new JLabel();
	private JLabel labe1 = new JLabel();
	private JLabel label2 = new JLabel();
	public call_octave() {
		class ChoiceListener implements ActionListener
		{
			public void actionPerformed(ActionEvent event)
			{
				try {
				if (event.getSource()==getP){   //if click "get
probability"
					ScMark=Double.parseDouble(sciG.getText());
	//get the science mark
					MathMark=Double.parseDouble(mathG.getText());
//get the math mark
					textArea.setText("");    //sets the textareas as
blank
					acuText.setText("");
					sucText.setText("");
					suc1Text.setText("");
					if(first==true){         //if first time to
click "get probability" (not to run unnecessary files again to save RAM)
						callMain();
					}
						getProb();          //call getProb()
				}
				else if (event.getSource()==back){    //if click back
button
					directory dir = new directory();      //go back
to directory
					dir.setVisible(true);
					setVisible(false);
				}
			}
				catch (IOException e) {  //catch file not found
exception (very little possibility of happening, but can happen)
					JOptionPane.showMessageDialog(null, "File not
found!");
				}
				//catch parsing string into double error (if user enter
string other than numbers it will warn the user)
				catch (NumberFormatException e){
					JOptionPane.showMessageDialog(null, "Please
enter numbers only!");
				}
					catch (OctaveEvalException e){ //if the octave
path is wrong
						JOptionPane.showMessageDialog(null, "The
octave file location is not correct! Please restart program!");
					}
			}
		}
		listenerC= new ChoiceListener();
		setLayout(new FlowLayout());
		setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		setBounds(403,134,750,490);
		setTitle("Get recommendations for students");
		textArea.setEditable(false);
		acuText.setEditable(false);
		gui();
	}
	public void gui(){
		panel = new JPanel(new GridBagLayout());
		c = new GridBagConstraints();
		label.setText("Science grade: ");
		c.insets = new Insets( 8, 8, 8, 8);
		c.gridx=0; c.gridy=0;
		panel.add(label, c);
		c.gridx=1; c.gridy=0;
		sciG = new JTextField(8);
		panel.add(sciG, c);
		label1.setText("Math grade: ");
		c.gridx=2; c.gridy=0;
		panel.add(label1, c);
		c.gridx=3; c.gridy=0;
		mathG = new JTextField(8);
		panel.add(mathG, c);
		getP = new JButton("get the probability");
		c.gridx=1; c.gridy=1;
		getP.addActionListener(listenerC);
		panel.add(getP, c);
		c.gridx=1; c.gridy=2;
		back = new JButton("back");
		back.addActionListener(listenerC);
		panel.add(back, c);
		panel1.add(octavePath.swclogo);
		add(panel);
		add(panel1, BorderLayout.PAGE_END);
	}
	//calling the main method to minimize the theta values (these theta values
are constant after determined
	//thus there is no point of calling this method again if the user clicks
"get probability" button again
	public void callMain(){
		System.out.println(octavePath.octPath);
		tester_IA.octave.eval("cd " + octavePath.octPath);
		tester_IA.octave.eval("main");   //get the probability and theta
values of the student going into science 10
		tester_IA.octave.eval("suc0"); //get the probability and theta
values of the student getting success level 3 or above
		tester_IA.octave.eval("suc1"); //get the probability and theta
values of the student having success level 4 (round1)
		tester_IA.octave.eval("suc2"); //get the probability and theta
values of the student having success level 2 (round2)
		first=false;		//will not call this method again
	}

	public void getProb() throws IOException{
		FileWriter faw = new FileWriter(octavePath.octPath +
"\\probability.txt");
		PrintWriter out = new PrintWriter(faw);
```

```
             out.print(ScMark +","+ MathMark);      //outputting the user entered
probability into a text so the octave files can read
             out.close();
             faw.close();
             tester_IA.octave.eval("getProb");      //use the theta values to get
the probability of going into science 10
             tester_IA.octave.eval("getSuc0");      //use the theta values to get
the probability of getting success level of 3 or above
             OctaveDouble prob = (OctaveDouble)tester_IA.octave.get("prob");
//get the probability of going into science 10
             OctaveDouble acu_prob =
(OctaveDouble)tester_IA.octave.get("accuracy"); //get the accuracy of the science
10 prediction
             OctaveDouble acu0 = (OctaveDouble)tester_IA.octave.get("suc_prob0");
//get the probability of getting success level of 3 or above
             labe.setText("Probability of getting into Sc10: ");
             c.gridx=0; c.gridy=2;
             panel.add(labe, c);
             labe1.setText("The accuracy of this prediction:");
             c.gridx=0; c.gridy=3;
             panel.add(labe1, c);
             c.gridx=1; c.gridy=2;
             panel.add(textArea, c);
             c.gridx=1; c.gridy=3;
             panel.add(acuText, c);
             textArea.append(df.format(prob.get(1)*100) + "%");   //putting the
probability onto textAreas
             acuText.append(df.format(acu_prob.get(1)) + "%");    //putting the
probability onto textAreas
             label2.setText("Probability of this student being moderately
successful or above:");
             c.gridx=0; c.gridy=4;
             panel.add(label2, c);
             sucText.append(df.format(acu0.get(1)*100) + "%");    //putting the
probability onto textAreas
             c.gridx=1; c.gridy=4;
             panel.add(sucText, c);
             c.gridx=1; c.gridy=5;
             panel.add(suc1Text, c);
             if(acu0.get(1)>=0.5){ //if the student's probability in acu0 is
bigger or equal to 0.5
                    OctaveDouble acu1 =
(OctaveDouble)tester_IA.octave.get("suc_prob1");     //get the probability of
being success level 4 (round1)
                    label3.setText("Probability of this student being maximally
successful:");
                    suc1Text.append(df.format(acu1.get(1)*100) + "%"); //putting
the probability onto textAreas
             }
             else if(acu0.get(1)<0.5){ //if the student's probability in acu0 is
smaller than 0.5
                    OctaveDouble acu2 =
(OctaveDouble)tester_IA.octave.get("suc_prob2"); //get the probability of being
success level 2 (round2)
                    label3.setText("Probability of this student being moderatly
successful:");
                    suc1Text.append(df.format(acu2.get(1)*100) + "%"); //putting
the probability onto textAreas
             }
```

```
            c.gridx=0; c.gridy=5;
            panel.add(label3, c);
            c.gridx=1; c.gridy=6;
            panel.add(back, c);
        }

    }
```