**OOP CA**

**Video:**

Please follow the below link to see the demo of my program

<https://youtu.be/eWqwwebKk8A>

**Read me:**

Classes:

* Control:

This is used as the main class for my program, in this class I call my ReadData constructor and pass the name of the document that I want to read the data for (in this case MLdata.csv)

I then open the file and read its contents. After that process is executed it creates a Hashtable (closes variable to a dictionary that I could understand) and add the data read from the ReadData class.

Lastly, I open the GUI and pass the name of the GUI, dataDictionary (Hashtable created above) and dataByRow (from ReadData)

* ReadData:

This is used to read the csv file and store the totals of each feature (male, female, etc)

It firsts gets the name of the csv file from the control, opens the file and read it,

When it reads the file, it stores the data in a 2D array (as rows) and sums the occurrences of each feature.

After that is just a lot of getters and setters.

* GUI:

For the GUI I followed the same ideas we used before in past lectures, but I also added the “JComboBox” which I looked up in internet and found it better looking than making the user type an answer.

Inside the actions performed I do a bit of error checking, if user doesn’t select a feature it will not let the button event work and will display a message of what is wrong or missing, I do this by using multiple Booleans and setting them to true or false if user is missing a value.

Lastly if user is not missing any values and clicks the button to calculate if the student will become an entrepreneur or not it will then open the MachineLearning class and pass the values the user entered.

After MachineLearning class process is executed it then displays the answer in the GUI saying if student will become an entrepreneur or not.

* MachineLearning:

In this class I calculate the probabilities of becoming an entrepreneur or not,

First thing I do is create two more Hashtable to store the probabilities of becoming an entrepreneur or not.

First It calculates the probability of prior probabilities (I cast the dataDictionary to a double in order to get the type of values I want), then the probability of evidence (denominator)

I Create a few more variables (1 in the name means they are entrepreneurs - 0 in the name means they are not entrepreneurs) to store the totals for features when they become entrepreneurs or not (EG: Male AND Entrepreneur / Urban AND No Entrepreneur)

Then I sum the occurrences of features that became entrepreneurs or not

Therefore, I add to the Hashtables the calculated probabilities of each feature AND entrepreneur or not.

Lastly, I calculate the probabilities of the student details passed by the GUI to become an entrepreneur or not, and if the probability of becoming an entrepreneur is higher than the probability of not becoming an entrepreneur, I return the answer to the GUI and display it in the GUI.

Core Functionality:

* Read in the training set and do training dynamically
* Model trains on data set

Optional Functionality:

* Model trains on 100% of data
* Better looking GUI (JComboBox)

If I had more time:

* I would add function to add more rows and allow the program to improve with more data added by user