THE UNIVERSITY OF THE WEST INDIES
ST. AUGUSTINE, TRINIDAD & TOBAGO, WEST INDIES
FACULTY OF ENGINEERING
Department of Electrical & Computer Engineering
BSc. in Electrical & Computer Engineering

ECNG 1009

Introduction to Programming

MINI PROJECT REPORT

Heart Disease UCI

Anthony Meade
816019469

Course Lecturer: Mr. Azim Abdool

Date Submitted: May 25th, 2020.

THE UNIVERSITY OF THE WEST INDIES
ST. AUGUSTINE, TRINIDAD & TOBAGO, WEST INDIES
FACULTY OF ENGINEERING
Department of Electrical & Computer Engineering
B. Sc. in Electrical & Computer Engineering

CHEATING, PLAGIARISM AND COLLUSION DECLARATION FORM

According to Rules 3.31 and 3.32 of 'The UWI Faculty of Engineering Undergraduate Regulations and Syllabuses 2019/2020:

3.31 "**Cheating, Plagiarism and Collusion are serious offences under University Regulations.**

(a) Cheating is any attempt to benefit one's self or another by deceit or fraud.

(b) Plagiarism is the unauthorised and/or unacknowledged use of another person's intellectual efforts and creations howsoever recorded, including whether formally published or in manuscript or in typescript or other printed or electronically presented form and includes taking passages, ideas or structures from another work or author without proper and unequivocal attribution of such source(s), using the conventions for attributions or citing used in this University. Plagiarism is a form of cheating.

(c) For the purposes of these Regulations, 'collusion' shall mean the unauthorised or unlawful collaboration or agreement between two or more students in the preparation, writing or production of a course assignment for examination and assessment, to the extent that they have produced the same or substantially the same paper, project report, as the case may be, as if it were their separate and individual efforts, in circumstances where they knew or had reason to know that the assignment or a part thereof was not intended to be a group project, but was rather to be the product of each student's individual efforts.

3.32 Cheating, plagiarism and collusion shall be reported to the Campus Committee on Examinations and the penalties would be in accordance with the University Examination Regulations."

I, ANTHONY MEADE, have read and understood Rules 3.31 and 3.32 of The UWI Faculty of Engineering Undergraduate Regulations and Syllabuses 2019/2020 on Cheating, Plagiarism and Collusion.

I understand that my submission is subject to the electronic plagiarism checker Turnitin.

I declare that this assignment is my own work and does not involve cheating, plagiarism or collusion.

Signature: Anthony Meade Date: 15/05/2020

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Chol	Cholesterol
Cp	Chest pain
CVD	Cardiovascular disease
ECG	Electrocardiogram
Exang	Exercise induced angina
Fbs	Fasting blood sugar
LDL	Low-density lipoprotein
NHLBI	National Heart Lung and Blood Institute
Thal	Thalassemia
Trestbps	Resting blood pressure
WHO	World Health Organization

1 Introduction

Heart disease is a term covering any disorder of the heart. Heart diseases are a major cause for human mortality rate and its death toll increases every year. Moreover, the toll is expected to grow more than 23.6 million by 2030[1]. Correct diagnosis and treatment at an early stage will save people from heart disease and will decrease the mortality rate due to heart problems. This project aimed to use data mining and analysis to determine which particular attributes could be used to predict heart disease. This report consists of the following:

1. analysing the problem, the dataset and reviewing literature on existing articles on heart disease prediction;
2. producing a clear and easy to understand algorithms and flowcharts on how the objectives were achieved;
3. implementing the code to match the algorithms and flowcharts;
4. applying the code to test cases to document the achievement of objectives and solution’s response to invalid input; and
5. discussing the performance and limitations of the solution as well as recommendations for improvement.

2 Problem Analysis

2.1 Rationale

According to the National Heart Lung and Blood Institute (NHLBI), heart disease is a catch-all phrase for a variety of conditions that affect the heart’s structure and function[2]. Furthermore, the World Health Organization (WHO) states that, cardiovascular diseases (CVDs) are the number one cause of death globally[3]. The term “heart disease” is often used interchangeably with the term “cardiovascular disease” in discussing this topic. Persons suffering from heart diseases are at risk to acute events such as heart attacks and strokes which are mainly caused by a blockage that prevents blood from flowing to the heart or brain. The most common reason for the blockage is a build up of plaque, which is made up of fat and cholesterol, on the inner walls of the blood arteries that supply the brain or heart.

Due to heart diseases being one of the biggest causes of mortality in the world, its prediction is regarded as one of the most important topics in the world. The WHO, also states that, “People with cardiovascular disease or who are at high cardiovascular risk (due to the presence of one or more risk factors such as hypertension, diabetes, hyperlipidaemia or already established disease) need early detection.” Therefore, this project aimed to analyse the Heart Disease UCI Cleveland dataset to determine which attribute of patients, such as angina or diabetes could be most suitable to determine whether or not a patient had a cardiovascular disease.

2.2 Background

As stated, the dataset used for this project was the Cleveland heart disease dataset taken from the UCI Machine Learning Repository. The processed dataset was utilised because it consisted of less attributes, than the raw data. Also, from literature it is found that 14 attributes were being used for predicting heart diseases. The dataset contained 303 rows representing data for 303 individuals and 14 columns for the 14 attributes.

The 14 attributes are described below[4]:

1. Age: displays the numerical value of the age of the patient in years.
2. Sex: displays the gender of the patient using binary format where 1 represents males and 0 represents females.
3. Cp: displays the chest pain type that the patient experienced in the following categories:
    - 0 = asymptomatic
    - 1 = atypical angina
    - 2 = non-anginal pain
    - 3 = typical angina
4. Trestbps: displays the numerical value of the resting blood pressure of the patient in millimetres of mercury (mm Hg), on admission to the hospital.
5. Chol: displays the numerical value of the serum cholesterol level of the patient in milligrams per decilitre (mg/dL).
6. Fbs: Compares the fasting blood sugar level of the patient to diabetes level (>120 mg/dL) and categorizes into 1 = true and 0 = false.
7. Rest ECG: displays the resting electrocardiographic results in the following categories:
    - 0 = hypertrophy
    - 1 = normal
    - 2 = having st-t wave abnormality
8. Thalach: displays the numerical value of the maximum heart rate achieved by the patient.
9. Exang: displays if the patient experiences exercised induced angina in two categroies:
    - 1 = yes
    - 2 = no
10. Oldpeak: displays the numerical value of ST depression of the patient, induced by exercise relative to rest.
11. Slope: displays the slope of the peak exercise ST segment in the following categories:
    - 0 = downsloping
    - 1 = flat

- 2 = upsloping

12. Ca: displays the number of major vessels (0-3) coloured by cardiac fluoroscopy. The value “4” was interpreted as NaN’s.
13. Thal: displays thalassemia in the following categories:
    - 0 = NaN’s
    - 1 = fixed defect (no blood flow in some part of the heart)
    - 2 = normal blood flow
    - 3 = reversible defect (a blood flow is observed but it is not normal)
14. Num: displays the diagnosis of heart disease in two categories:
    - 0 = yes
    - 1 = no

For this project, only 8 of the attributes were checked for their correlation to heart diseases. The 8 attributes used were chosen for the following reasons:

- Age: Age is an important risk factor in developing heart diseases, with an increasing risk with each decade of life. The WHO states that, “Out of 17 million premature deaths (under the age of 70), due to non-communicable diseases in 2015, 37% are caused by cardiovascular diseases.” This is as a result of changing lifestyle such as food habits and other bad habits like smoking has led people to heart disease at younger ages[4].
- Sex: The NHLBI states that, in men the risk of heart diseases is higher compared pre-menopausal women. However, once past menopause ($\sim 50\ years$), the risk of heart diseases increases for women.
- Exang: Angina is at type of chest pain that results from reduced oxygenated blood flow to the heart. Exercise induced angina can be an indicator of heart disease, however it is sometimes the result of less serious issues such as muscle strain and asthma. Persons often mistake exercised induced angina for a heart attack.
- Chest pain: Heart disease patients, may also experience angina without being under the stress of exercise. It is often described as squeezing, pressure, heaviness, tightness or pain in the chest. Angina is the most common symptom of coronary artery disease however, it can still be hard to distinguish from other types of chest pain, such as the discomfort of indigestion.
- Fasting blood sugar: The WHO and NHLBI list diabetes as one of the main risk factors for heart diseases. If an individual’s body is not producing enough insulin or is not responding to insulin properly, this can result in high blood sugar levels increasing the risk of heart disease.
- Resting blood pressure: The WHO lists raised blood pressure as the leading risk factor for cardiovascular diseases worldwide. High blood pressure can damage the arteries that feed the heart and

normally occurs with other risk factors such as obesity and high cholesterol. Mayo Clinic, observes high blood pressure as systolic greater than 120 mmHg.

- Cholesterol: According to NHLBI, higher concentrations of low-density lipoprotein (LDL) cholesterol also known as the "bad cholesterol" and lower concentrations of high-density lipoprotein HDL cholesterol (also called the "good cholesterol"), increases an individual's risk of heart attack. High levels of LDL cholesterol narrow the arteries with a build-up of plaque, thereby limiting the flow of oxygen-rich blood to the heart muscle. Sometimes, plaque can break off and form clots which block most or all of the blood flow through the coronary artery.
- Cardiac Fluoroscopy: Cardiac fluoroscopies are similar to x-rays in that it is an imaging tool used to help healthcare providers. It is often done while a contrast dye moves through the arteries and a continuous X-ray beam is passed through and sent to a video monitor so that the blood flow can be seen in detail. The fluoroscopy, shows the number of narrowed or blocked blood vessels and the higher the value of this feature, the more likely an individual has heart disease.

2.3 Existing Literature and Solutions

Numerous literary works, related to the diagnosing of heart diseases, using data mining techniques are available on the world wide web. A brief review of some these studies is presented in the following paragraphs.

In a 2019 journal written by Shubhankar Rawat, entitled "Heart Disease Prediction using Machine Learning." A number of classification models were used to determine which could predict heart disease with the highest accuracy, using the Heart Disease UCI Cleveland pre-processed dataset. Some of the models used were support vector machines, naïve bayes, logistic regressions and decision trees. From these models, heart diseases were predicted with a highest accuracy of up to 80.32%. It was concluded that predicting heart disease based on risk factors was difficult, however machine learning techniques were useful to predict heart disease from accurate patient data.[5]

Similarly, in a 2019 journal written by Sajeev Shelda entitled "Deep Learning to Improve Heart Disease Risk Prediction." The aim of the work was to investigate the plausibility of using the deep and machine learning to predict models for heart disease. The study utilised a combination of two datasets, one being the Cleveland heart disease dataset. A basic multi-layer perceptron (MLP), which is a traditional deep learning algorithm was used in this study. In addition, four popular machine learning models, logistic regression (LR), linear discriminant analysis (LDA), support vector machine (SVM), and random forest (RF) were used for comparison. The results of the study population were that there were substantially fewer women than men

(32% vs 68%). Of the participants, 14% had diabetes and 52% had high cholesterol (>240). In addition, 51% had abnormal ECG results and 31% exhibited major vessel calcification in the fluoroscopy, while 33% experienced exercised induced angina. There 257 cases of heart disease from 567 participants. The study concludes that the deep learning approach was more effective with a 94% accuracy of predicting heart disease while the best machine learning accuracy was 93.3%. [7]

In another article published in 2014 by Aline Milane, titled "Association of hypertension with coronary artery disease onset in the Lebanese population." The objective, was to examine the relationship between hypertension and coronary artery disease (CAD) age of onset in the Lebanese population. The analysis was performed on data extracted from Lebanese patients, undergoing cardiac catherization. In this study, the age of CAD onset was defined as the age upon first diagnosis of CAD by catherization. Furthermore, this study accounted for gender influence on CAD by changing the standard for both genders. Male patients in the study were categorized as having early onset CAD if diagnosed at an age younger than or equal to 45 (≤45), while female patients were categorized as having early onset CAD when diagnosed at an age younger than or equal to 55 (≤55). This study also recorded different patient attributes from the Cleveland dataset, such as body mass index (BMI), cigarette use, physical activity level and hyperlipidaemia. The study population consisted of 72% males and 28% females and the mean age of CAD onset was 61 years. The majority of the CAD cases were associated with at least one of main risk factor with 63.9% reported having high blood pressure. The non-risk factor CAD patients represented only 1.7% of the total population. In addition, the study used a logistic regression model to asses the impact of other risk factors on CAD onset. The model showed that smoking and obesity were positively associated with early onset coronary artery disease while diabetes and hyperlipidaemia did not show significant association with early disease onset. [8]

A study titled, "Recommendation of Attributes for Heart Disease Prediction using Correlation Measure" by S.Chellammal, R. Sharmila, was published in 2019 by the International Journal for Technology and Engineering. The study utilised the UCI Cleveland heart disease pre-processed dataset. Moreover, in this work, the patient attributes were analysed for their relevancy in the prediction of heart disease using correlation techniques. The prediction results were obtained using three different classifiers, Naive Bayes (NB), Multi-Layer Perceptron (MLP) and Sequential Minimal Optimization (SMO). The proposed approach for this study was to firstly rank the attributes according their correlation measures, then perform the classification using the three outlined classifiers and finally recommend the relevant attributes based on their accuracy. The results showed that 10 attributes, age, sex, cp, restecg, thalach, exang, oldpeak, slope, ca and thal were the most relevant attributes in predicting heart diseases. [9]

2.4 Objectives

For this project, the objectives were set based on the findings of past studies as well as personal preferences. The goals set out to be achieved were:

- To determine the correlation between gender and heart disease. The function outputs to the console screen the total number of males and females, the number and percentage of persons with and without heart disease and the quantity of males and females with and without disease.
- To determine the most common age group and average age of heart disease patients. Two functions are executed for this objective. One function shows the age information for all the patients and the other the age data for only heart disease patients. The age information is the lowest, highest, and average age of the individuals, the average male and female age, as well as a histogram with 5 age group bins, starting from the lowest age. Each bin has a range of 10 years.
- To determine the relationship between exercised induced angina and heart disease. The output of this objective displays the number of individuals who do and do not experience exercised induced angina. In addition, the number of heart disease and non-heart disease patients who do and do not experience exercise induced angina is also output.
- To ascertain which chest pain types are more prevalent amongst heart disease patients. The output of this objective shows the number of heart disease and non-heart disease patients and the type of chest pain experienced.
- To calculate the probability that a heart disease patient has either diabetes, high blood pressure and cholesterol, which are the main risk factors in developing heart diseases. When this objective is selected, the user has to choose specifically which risk factor to display data for. For each factor, a probability tree displays the probability of an individual in this dataset, having heart disease as well as the probability of the specific risk factor.
- To determine the correlation between narrowed blood vessels and heart disease. This output the number of blood vessels highlighted by the fluoroscopy for all patients and heart disease patients. The percentage of heart patients is also output.

For all of the objectives stated, the outputs to the console were also written to text files for ease of access for the user.

3 Algorithms and Flow Charts

To ensure that the program was user friendly, a menu driven system was implemented. Each of the objectives stated were implemented with one function, with the exception of objective two. The menu system operated with the use of switch statements, which executed a particular function or group of functions based on the users input.

To start, the program works by finding and opening the input file. If the file is not found, an error is displayed on the console and the program quits. However, once the file is found, each line is added to a vector of strings. Following that, each line is goes into the split function where each string separated by a comma is added into a different vector. Each of the strings are then passed into the "isInteger" function which checks if each character in the string is an integer and returns an error if it is not. However, this error does not stop the program. Then, the respective strings are converted to integers or doubles depending on which attribute is being checked. The attribute values are then assigned to their respective struct members and stored in a vector of structs. The steps stated are then repeated again, however only lines where the target value is zero are added to the vector or structs.

At this time, the program enters a while loop, prints the menu and prompts the user to make a choice. Each of the choices on the menu are assigned to a case and when a match is found the function in that block executes. It should be noted that if the user enters a letter or a choice not available, the program outputs an error to the console screen and prompts for a new input. The choices in the menu, labelled 1-7, achieve the six objectives and has an exit option.

The gender function, loops through the entire input file, checks if all the gender values are binary and throws an error, to the main function if they are not. A second loop runs through the file and increments different variables, based on the criteria being checked. For example, if the sex on a particular line is equivalent to 1, the number of males will be increased, or if the sex is equivalent to 1 and target is equivalent to 0, the number of heart disease males will be increased. Outside of the loop, the percentages of male, female, heart disease and non-heart disease patients are calculated and a series of output statements, will output the results to the console and a text file, so a correlation between gender and heart disease can be determined.

Option 2 in the menu executes two functions consecutively. Both functions that execute in this case are very similar, with the main difference being the input arguments for each. At the start of each function, each of the variables are initialized to 0 with the exception of one, the lowest age which is initialized to 100. A loop runs through each line of the file and if the age is lower than the lowest age variable, that age becomes the lowest age. Similarly, if the age is higher than the highest age variable, that age becomes the highest age. The total age is also calculated and divided by the number of patients to determine the average age. Furthermore, a second loop sums all the male and all the female ages and divides by the number of males and females, respectively to determine the average male and female age. Next, five ranges of 10 years are created starting from the lowest age and each range is populated from an array. A scaled histogram is then created using the data stored in the array and the “ * ” character. All of the values that are calculated are output to the console and a text file, in order to see the most common age groups as well as average ages.

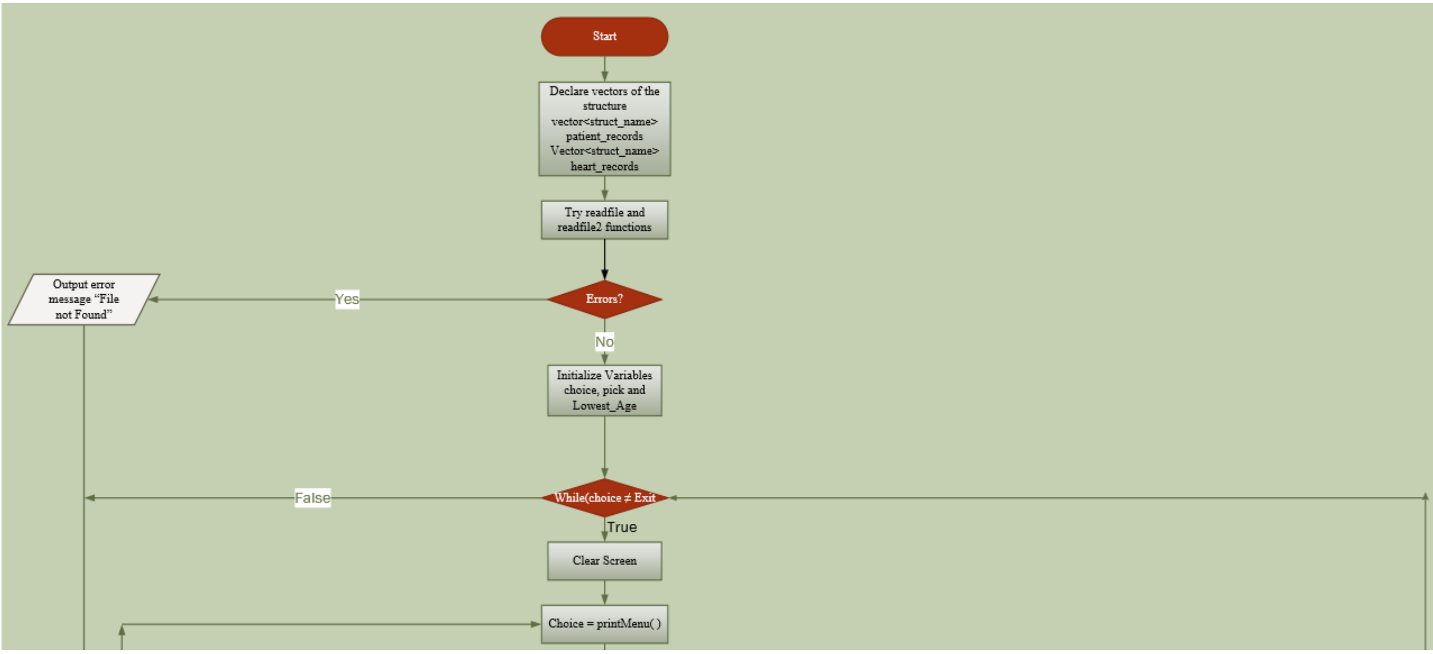
Next, is the exang function which starts of by looping through the file and checking if any of the exercised induced angina entries are non-binary. If a non-binary entry is found, the line on which it occurs as well as an error message is output to the console. The subsequent loop in the function increments respective variables to determine the total number of patients who experience and do not experience exercised induced angina. This loop also determines the number heart disease and non-heart disease patients who experience and do not experience exercised induced angina. Furthermore, the percentage of heart disease patients who do and do not experience exang is also determined and output to the console window, so that a correlation between heart disease and exang could be observed.

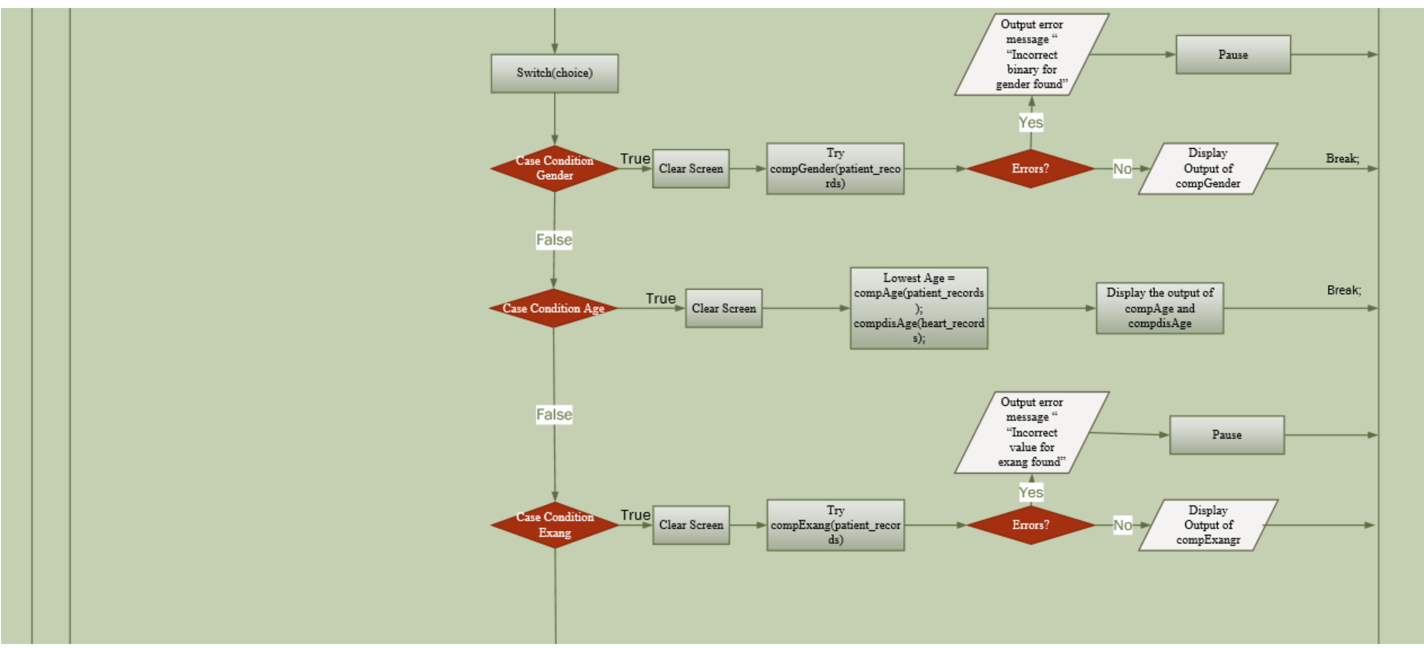
The function, in the fourth switch case statement, aims to determine, the types of chest pain experienced by all the patients in the dataset, but the those with heart disease are of greater interest. The functions operate similar to the others, by first running a loop to determine if there are any invalid data entries and throwing and error is there is. The second loop simply increments the value of different assigned variables based on if certain criteria are met. Then a series of output statements are used to format the data into a presentable form onto the console screen and a text file. The output from this function allows the user to see if a certain chest pain type is more prevalent amongst heart disease patients.

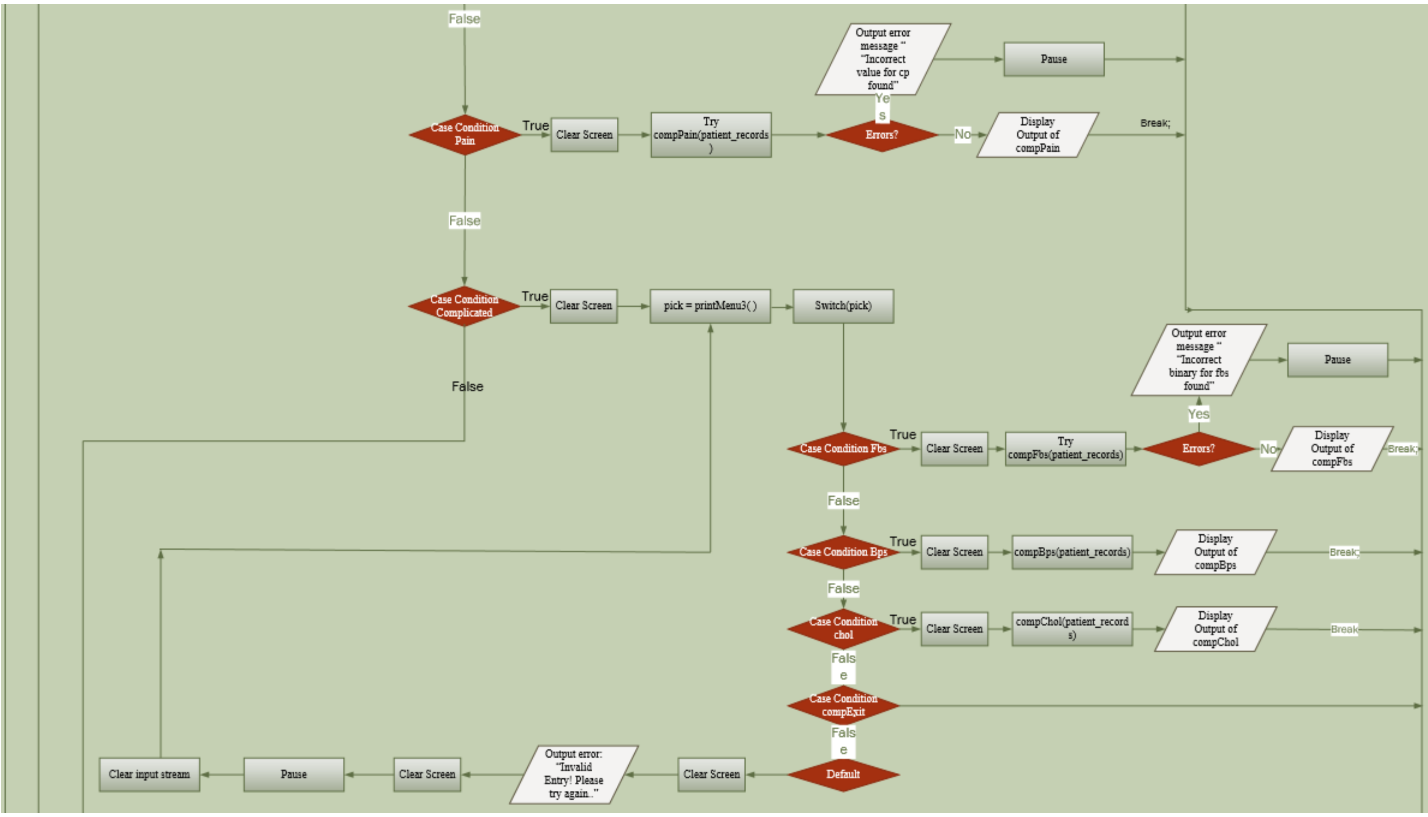
The fifth case in the menu, opens a secondary menu that prompts the user for an input based on three risk factors, diabetes, hypertension and cholesterol. Each of the functions in the three case blocks have very similar outputs. The diabetes function, initially runs a loop which ensures that the data entries are binary and throws an error if non-binary entries are found. The function then runs a second loop which like the other functions, simply increment certain variables on each passing of the loop depending on which criteria are met. Since the output of each function in this case block is a probability tree, the probabilities that a patient has diabetes, hypertension and high cholesterol are calculated. The trees are formatted using “cout” and output to the console, with the respective percentages.

The final function that can execute from the menu shows, if heart disease patients show higher amounts of narrowed blood vessels. The function initially runs a loop to determine if the data is good and throws an error if it is not. The second loop increments variables vessel (0,1,2,3) and HDvessel (0,1,2,3) based on the number of highlighted blood vessels in the fluoroscopy. The result is then output onto the console screen to be analysed by the user.

3.1 Flow Chart for Main Function







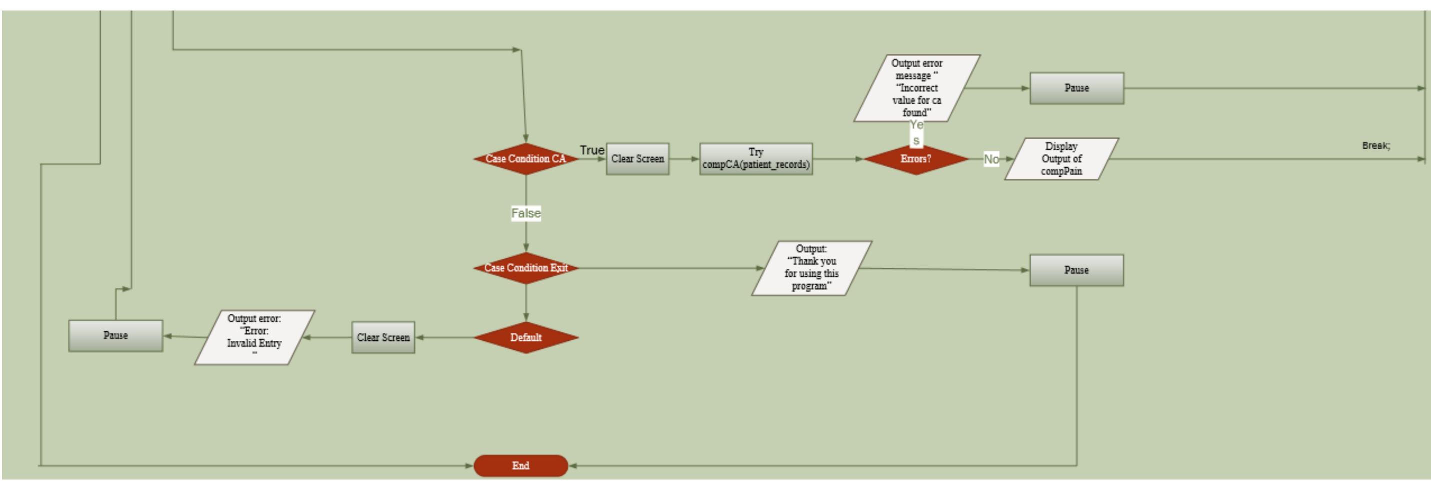


Figure 1: Showing a flow chart of the main function

Pseudocode for Functions

For this project, all non-error outputs to the console were also output to a text. Hence this is shown only in the gender function's pseudocode. Similarly, the convert and convert double function are the essentially the same with the only difference being that convert double changes the string to a double and convert changes the string to an integer. Therefore, the pseudocode is shown for the covert function only. The probability tree functions for diabetes, hypertension and cholesterol are very similar as well, with the main difference being, the attributes being analysed. In addition, it was possible to calculate the average blood pressure and cholesterol levels but the blood sugar was represented in binary, making it not possible to the same. Hence, the pseudocode was done for only the cholesterol function.

3.2 Read File Function

```
void readfile(vector<Records>& v)

Start

    Input stream infile;

    Open infile ("heart.csv");

    if (infile is not found)

        throw domain_error("File Not Found");

    end if

    else

        output to console << "Let's Begin: \n";

        string line;

        vector <string> usefuldata;

        while (getline(infile, line))

            add line to the useful data vector;

        end while loop


        Records temp;

        for (vector<string>::size_type i = 1; i ≠ usefuldata vector size(); increment i)

            temp = split(usefuldata[i]);

            add temp to vector<Records>;

        end for loop

    end else
```

```
end
```

3.3 Split Function

```
Records split(string line)

start

    Records p;

    vector<string> temp_records_items;

    string temp;

    stringstream ss;

    ss << line;

    while (getline(ss, temp, ','))

        temp_records_items.push_back(temp);

    end while loop

    for (vector<string>::size_type i = 0; i ≠ 9; increment i)


        if (isInteger(temp_records_items[i]))

            continue;

        end if

        else

            cerr << "Non interger type found" << endl;

            pause console

        end else

    end for loop

    for (vector<string>::size_type i = 10; i ≠  temp_records_items vector size(); increment i)

        if (isInteger(temp_records_items[i]))
```

```
                continue;
            end if
            else
                cerr << "Non interger type found" << endl;
                pause console
            end else
        end for loop

        p.Age = convert(temp_records_items[0]);
        p.Sex = convert(temp_records_items[1]);
        p.Chest_Pain = convert(temp_records_items[2]);
        p.Threstbps = convert(temp_records_items[3]);
        p.chol = convert(temp_records_items[4]);
        p.fbs = convert(temp_records_items[5]);
        p.ECG = convert(temp_records_items[6]);
        p.thalach = convert(temp_records_items[7]);
        p.exang = convert(temp_records_items[8]);
        p.slope = convert(temp_records_items[10]);
        p.fluorscopy = convert(temp_records_items[11]);
        p.thalas = convert(temp_records_items[12]);
        p.target = convert(temp_records_items[13]);
        return p;
    end
```

3.4 Integer Check Function

```
bool isInteger(string s)

start

        for (string::size_type i = 0; i ≠ string size(); increment i)

                if (isdigit(s[i])== false)


                        return false;

                end if

        end for loop

        return true;

end
```

3.5 Convert Function

```
int convert(string s)

start

        stringstream ss;

        ss << s;

        int num;

        ss >> num;

        return num;

end
```

3.6 Print Menu Function

```
int printMenu()

start


        int choice → 0;
```

```
output to console "Menu: newline"
          "(1) Compare Heart Disease by Gender newline"
          "(2) Compare Heart Disease by Age Group newline"
          "(3) Exercise Induced Angina newline"
          "(4) Most Common Chest Pain Types newline"
          "(5) Volume with Diabetes, High Blood Pressure and Cholesterol newline"
          "(6) Number of Vessels Highlighted by Fluoroscopy newline"
          "(7) Exit newline";
output to console "Choose an item (1-7) newline";
input set width(1)  choice;


while (cin is not good)
        clear console
        output error  "Faulty Input! Please try again..." newline newline;
        pause console
        clear console;


        clear input stream;
        ignore all inputs until newline;


        output to console "Menu: newline"
                  "(1) Compare Heart Disease by Gender newline"
                  "(2) Compare Heart Disease by Age Group newline"
                  "(3) Exercise Induced Angina newline"
```

```
                    "(4) Most Common Chest Pain Types newline"

                    "(5) Volume with Diabeties, High Blood Pressure and Cholesterol newline"

                    "(6) Number of Vessels Highlighted by Fluoroscopy newline"

                    "(7) Exit newline";

          output to console "Choose an item (1-7) newline";

          input set width(1)  choice;


          clear input stream;

          ignore all inputs until newline;

     end while loop

     return choice;

end
```

3.7 Gender Function

```
Void compGender(const vector<Records> v)

Start

     Clear console window

     Output stream outfile("Gender_Results.txt");

     numMales →  0;           malePerc → 0;

     numFemales →   0;        femalePerc → 0;

     max → 0;                 diseasePerc → 0;

     disease → 0;             nodisease → 0;

     nodiseasePerc → 0;

     disMales → 0;            nodisMale → 0;

     disFemales → 0;          nodisFem → 0;
```

```
Output to console newline "Gender Summary" newline;

Output to console newline "Number of Patients: " vector size() newline;

Output to file newline "Gender Summary" newline;

Output to file newline "Number of Patients: "  vector size() newline;


For (vector<Records>::size_type i = 0; i ≠ vector size(); ++i)

        If (v[i].Sex = 0 or v[i].Sex = 1)

                Continue;

        End if

        Else

                Output to console "line: " << i + 1 << newline;

                Throw domain_error("Incorrect Value for Gender Found");

        End else

End for loop

for (vector<Records>::size_type i = 0; i ≠ vector size(); increment i)


        if (v[i].target = 0)

                increment disease;

        end if


        if (v[i].target = 1)

                increment nodisease;

        end if

        if (v[i].Sex = 1)
```

```
                increment numMales;
        end if

        if (v[i].Sex = 0)
                increment numFemales;
        end if
        if (v[i].Sex = 1 & v[i].target = 0)
                increment disMales;
        end if

        if (v[i].Sex = 0 & v[i].target = 0)
                increment disFemales;
        end if
        if (v[i].Sex = 1 & v[i].target = 1)
                increment nodisMale;
        end if
        if (v[i].Sex = 0 & v[i].target = 1)
                incremet nnodisFem;
        end if
end for loop

malePerc = round ((disMales) ÷ numMales) x 100);
femalePerc = round((disFemales) ÷ numFemales) x 100);
diseasePerc = round((disease) ÷ vector size) x 100);
```

nodiseasePerc = round((nodisease) ÷ vector size) x 100);

Output to console newline

Output to file newline

Output to console "Number of Males: " set width(6) numMales newline;

Output to file "Number of Males: " set width(6) numMales newline;

Output to console "Number of Females: " set width (4) numFemales newline;

Output to file "Number of Females: " set width(4) numFemales newline;

Output to console string(30, '-') newline;

Output to file string(30, '-') endl;

Output to console newline "Number without Heart Disease: " nodisease " (" nodiseasePerc "%)" newline newline;

Output to file "Number without Heart Disease: " nodisease " (" nodiseasePerc "%)" newline newline;

Output to console "Males without Heart Disease: " set width (4) nodisMale newline;

Output to file"Males without Heart Disease: " nodisMale newline;

Output to console "Females without Heart Disease: " nodisFem newline;

Output to file "Females without Heart Disease: " nodisFem newline;

Output to console string(30, '-') newline;

Output to file string(30, '-') newline;

Output to console newline "Number with Heart Disease: " disease " (" diseasePerc "%)" newline newline;

Output to file "Number with Heart Disease: " disease " (" diseasePerc "%)"newline newline;

Output to console "Males with Heart Disease: " setw(4) disMales " (" malePerc "%)" newline;

Output to file "Males with Heart Disease: " disMales " (" malePerc "%)" newline;

Output to console "Females with Heart Disease: " disFemales " (" femalePerc "%)" newline;

Output to file "Females with Heart Disease: " disFemales " (" femalePerc << "%)" newline;

Output to console string(30, '-') newline;

Output to filestring(30, '-') newline;

end

3.8 Age Function

int compAge(const vector<Records>& v)

start

Output stream outfile("Age_Results.txt");

Clear console window

int numMales → 0;

int numFemales → 0;

double totalAgeM → 0.0;

double totalAgeF → 0.0;

double totalAge → 0.0;

int Lowest_Age →100;

int Highest_Age → 0;

for (vector<Records>::size_type i = 0; i ≠ vector size(); increment i)

if (v[i].Age < Lowest_Age)

Lowest_Age = v[i].Age;

end if

if (v[i].Age > Highest_Age)

Highest_Age = v[i].Age;

end if

totalAge += v[i].Age;

```
end for loop
double AverageAge = round(totalAge) / vector size);

Output to console "Age Data for all Patients" newline;
Output to console "Lowest Age: " Lowest_Age newline;
Output to console "Higest Age: " Highest_Age newline;
Output to console "Average Age: " AverageAge newline;

for (vector<Records>::size_type i = 0; i ≠ vector size; increment i)
    if (v[i].Sex = 1)
        increment numMales;
        totalAgeM += v[i].Age;
    end if
    else
        increment numFemales;
        totalAgeF += v[i].Age;
    end else
end for loop
double avgMaleAge = round((totalAgeM) / numMales);
double avgFemaleAge = round((totalAgeF) / numFemales);

Output to console "Average Male Age: " avgMaleAge newline;

Output to console "Average Female Age: " avgFemaleAge newline newline;
```

```
for (int i = 0; i < 5; increment i)
        Output to console  i  ": "  (Lowest_Age + (i * 10))  "-"  (Lowest_Age + (i + 1) * 10)     ", ";
end for loop
int n[5] = { 0,0,0,0,0 };


for (vector<Records>::size_type i = 0; i ≠ vector size(); increment i)
        if (v[i].Age >= Lowest_Age & v[i].Age <= (Lowest_Age + 10))
                n[0]++;
        end if
        if (v[i].Age > (Lowest_Age + 10) & v[i].Age <= (Lowest_Age + 20))
                n[1]++;
        end if
        if (v[i].Age > (Lowest_Age + 20) & v[i].Age <= (Lowest_Age + 30))
                n[2]++;
        end if
        if (v[i].Age > (Lowest_Age + 30) & v[i].Age <= (Lowest_Age + 40))
                n[3]++;
        end if
        if (v[i].Age > (Lowest_Age + 40) & v[i].Age <= (Lowest_Age + 50))
                n[4]++;
        end if
end for loop
Output to console newline newline;
```

```
    Output to console "0 " << n[0] << "  " << string((n[0] / 5), '*') newline;

    Output to console "1 " << n[1] << "  " << string((n[1] / 5), '*') newline;

    Output to console "2 " << n[2] << " " << string((n[2] / 5), '*') newline;

    Output to console "3 " << n[3] << "  " << string((n[3] / 5), '*') newline;

    Output to console "4 " << n[4] << "  " << string((n[4] / 5), '*') newline;

    Output to console newline "Each Star Represents 5 Persons." newline;

    Output to console string(41, '-') newline;

    return Lowest_Age;

end
```

3.9 Second Age Function

```
void compdisAge(const vector<Records>& v, int L)

start

    Output stream outfile("Age_Results.txt");

    Clear console window

    int disMales → 0;

    int disFemales → 0;

    double distotalAgeM → 0.0;

    double distotalAgeF → 0.0;

    double distotalAge → 0.0;

    int disLowest_Age → 100;

    int disHighest_Age → 0;


    for (vector<Records>::size_type i = 0; i ≠ vector size(); increment i)

        if (v[i].Age < disLowest_Age)
```

```
        disLowest_Age = v[i].Age;
    end if

    if (v[i].Age > disHighest_Age)
        disHighest_Age = v[i].Age;
    end if

    distotalAge += v[i].Age;
end for loop
double disAverageAge = round(distotalAge) / vector size);

Output to console "Age Data for Patients with Heart Disease" newline;
Output to console "Lowest Age: " disLowest_Age newline;
Output to console "Higest Age: " disHighest_Age newline;
Output to console "Average Age: " disAverageAge newline;

for (vector<Records>::size_type i = 0; i ≠ vector size; increment i)
    if (v[i].Sex = 1)
        increment disMales;
        distotalAgeM += v[i].Age;
    end if
    else
        increment disFemales;
        distotalAgeF += v[i].Age;
```

```
        end else
end for loop
double avgMaleAge = round((distotalAgeM) / disMales);
double avgFemaleAge = round((distotalAgeF) / disFemales);

Output to console "Average Male Age: " avgMaleAge newline;
Output to console "Average Female Age: " avgFemaleAge newline newline;

for (int i = 0; i < 5; increment i)
         Output to console  i  ": "  (L + (i * 10))  "-"  (L + (i + 1) * 10)  ", ";
end for loop
int n[5] = { 0,0,0,0,0 };

for (vector<Records>::size_type i = 0; i ≠ vector size(); increment i)
        if (v[i].Age >= L & v[i].Age <= (L + 10))
                n[0]++;
        end if
        if (v[i].Age > (L + 10) & v[i].Age <= (L + 20))
                n[1]++;
        end if
        if (v[i].Age > (L + 20) & v[i].Age <= (L + 30))
                n[2]++;
        end if
        if (v[i].Age > (Lowest_Age + 30) & v[i].Age <= (L + 40))
```

```
                n[3]++;
            end if
            if (v[i].Age > (L + 40) & v[i].Age <= (L + 50))
                n[4]++;
            end if
        end for loop
        Output to console newline newline;


        Output to console "0 " << n[0] << "  " << string((n[0] / 5), '*') newline;
        Output to console "1 " << n[1] << "  " << string((n[1] / 5), '*') newline;
        Output to console "2 " << n[2] << " " << string((n[2] / 5), '*') newline;
        Output to console "3 " << n[3] << "  " << string((n[3] / 5), '*') newline;
        Output to console "4 " << n[4] << "  " << string((n[4] / 5), '*') newline;
        Output to console newline "Each Star Represents 5 Persons." newline;
        Output to console string(41, '-') newline;
End
```

3.10 Exang Function

```
void compExang(const vector<Records>& v)
start
        clear console
        Output stream outfile("Exang_Results.txt");
        int numYes → 0;         int Yes → 0;
        int numNo → 0;          int No → 0;
        int WithDis → 0;        int WoDis → 0;
```

```
int NoexDis → 0;        int NoexDisPerc → 0;
int exDis → 0;          int exDisPerc → 0;


Output to console newline "Exercised Induced Angina Summary" newline;
Output to console newline "All Patients: " vector size newline;


for (vector<Records>::size_type i = 0; i ≠ vector size; increment i)
        if (v[i].exang = 0 or v[i].exang = 1)
                continue;
        end if
        else


                Output to console  "line: "  i + 1  newline;
                throw domain_error("Incorrect Binary for Exercised Induced Angina Found");
        end else
end for loop
for (vector<Records>::size_type i = 0; i ≠ vector size; increment i)
        if (v[i].target = 0)
                increment WithDis;
        endif
        if (v[i].target = 1)
                increment WoDis;
        end if
        if (v[i].exang = 0)
```

```
            incrememt No;
        end if
        if (v[i].exang = 1)
            increment Yes;
        end if
        if (v[i].exang = 0 & v[i].target = 1)
            increment numNo;
        end if
        if (v[i].exang = 1 & v[i].target = 1)
            increment numYes;
        end if
        if (v[i].exang = 0 & v[i].target = 0)
            increment NoexDis;
        end if
        if (v[i].exang = 1 & v[i].target = 0)
            increment exDis;
        end if
    end for loop
    NoexDisPerc = round((NoexDis) / No) x 100);
    exDisPerc = round((exDis) / Yes) x 100);
    Output to console newline "Exercise Induced Angina Experienced:" newline;
    Output to console "No: " No newline;
    Output to console "Yes: " set width(1) Yes newline;
    Output to console string(36, '-') << newline;
```

```
    Output to console newline "Non Heart Disease Patients: "  WoDis newline;

    Output to console newline "Exercise Induced Angina Experienced:" newline;

    Output to console "No: " numNo newline;

    Output to console "Yes: " set width(1) numYes newline;

    Output to console string(36, '-') newline;

    Output to console newline "Heart Disease Patients: " WithDis newline;

    Output to console newline "Exercise Induced Angina Experienced:" newline;

    Output to console "No: " NoexDis " (" NoexDisPerc "%)" newline;

    Output to console "Yes: " exDis " (" exDisPerc "%)" newline;

    Output to console string(36, '-') newline;

    pause console

end
```

3.11 Chest Pain Function

```
void compPain(const vector<Records>& v)

start

    clear console

    Output stream outfile("ChestPain_Results.txt");

    int TypF → 0; int AtypF → 0; int NonAngF → 0; int AsympF → 0;

    int TypM → 0; int AtypM → 0; int NonAngM → 0; int AsympM → 0;

    int hdTypF → 0; int hdAtypF → 0; int hdNonAngF → 0; int hdAsympF → 0;

    int hdTypM → 0; int hdAtypM → 0; int hdNonAngM → 0; int hdAsympM → 0;

    int Typical → 0;

    int Atypical → 0;

    int Non_Anginal → 0;
```

```
int Asymptomatic → 0;

int HD → 0;  int NoHD → 0;

int hdAsymp → 0;

int hdAtyp → 0;

int hdTyp → 0;

int hdNonAng → 0;

Output to console newline "Chest Pain Summary" newline;

Output to console newline "Total Patients: " vector size  newline;


for (vector<Records>::size_type i = 0; i ≠ vector size; increment i)

        if (v[i].Chest_Pain = 0 or v[i].Chest_Pain = 1 or v[i].Chest_Pain = 2 or v[i].Chest_Pain = 3)

                continue;

        end if

        else

                Output to console "line: " i + 1 newline;

                throw domain_error("Incorrect Value for Chest Pain Found");

        end else

end for loop


for (vector<Records>::size_type i = 0; i ≠ vector size; increment i)

        if (v[i].target = 1)

                increment NoHD;

        end if

        if (v[i].Chest_Pain = 0 & v[i].target = 1)
```

```
        increment Asymptomatic;
end if
if (v[i].Chest_Pain = 0 & v[i].Sex = 1 & v[i].target = 1)
        increment AsympM;
end if
if (v[i].Chest_Pain = 0 & v[i].Sex = 0 & v[i].target = 1)
        increment AsympF;
end if
if (v[i].Chest_Pain = 1 & v[i].target = 1)
        increment Atypical;
end if
if (v[i].Chest_Pain = 1 & v[i].Sex = 1 & v[i].target = 1)

        increment AtypM;
end if
if (v[i].Chest_Pain = 1 & v[i].Sex = 0 & v[i].target = 1)
        increment AtypF;
end if
if (v[i].Chest_Pain = 2 & v[i].target = 1)
        increment Non_Anginal;
end if
if (v[i].Chest_Pain = 2 & v[i].Sex = 1 & v[i].target = 1)
        increment NonAngM;
end if
```

```
        if (v[i].Chest_Pain = 2 & v[i].Sex = 0 & v[i].target = 1)

            increment NonAngF;

        end if

        if (v[i].Chest_Pain = 3 & v[i].target = 1)

            increment Typical;

        end if

        if (v[i].Chest_Pain = 3 & v[i].Sex = 1 & v[i].target = 1)

            increment TypM;

        end if

        if (v[i].Chest_Pain = 3 & v[i].Sex = 0 & v[i].target = 1)

            increment TypF;

        end if

    end for loop

    Output to console newline "Number of Non Heart Disease Patients: "  newline;


    Output to console  "Chest Pain Type" set width(23) "Number of Patients"  set width(9) "Male" set
width(11) "Female" newline;


    Output to console  set width(7) "Typical" set width(22) Typical set width(17) TypM set width (9)
TypF newline;


    Output to console  set width(8) "Atypical" set width(21) Atypical set width(17) AtypM set width(9)
AtypF newline;
```

Output to console set width(4) "Non Anginal" set width(18) Non_Anginal set width(17) NonAngM set width(9) NonAngF newline;

Output to console set width(4) "Asymptomatic" set width(17) Asymptomatic set width(17 AsympM set width(9) AsympF newline;

```
Output to console (string(58, '-') newline;

for (vector<Records>::size_type i = 0; i ≠ vector size; increment i)

    if (v[i].target = 0)

        increment HD;

    end if

    if (v[i].Chest_Pain = 0 & v[i].target = 0)

        increment hdAsymp;

    end if

    if (v[i].Chest_Pain = 0 & v[i].target = 0 & v[i].Sex = 1)

        increment hdAsympM;

    end if

    if (v[i].Chest_Pain == 0 & v[i].target = 0 & v[i].Sex = 0)

        increment hdAsympF;

    end if

    if (v[i].Chest_Pain = 1 & v[i].target = 0)

        increment hdAtyp;

    end if

    if (v[i].Chest_Pain = 1 & v[i].target = 0 & v[i].Sex = 1)

        increment hdAtypM;

    end if
```

```
        if (v[i].Chest_Pain = 1 & v[i].target = 0 & v[i].Sex = 0)
                increment hdAtypF;
        end if
        if (v[i].Chest_Pain = 2 & v[i].target = 0)
                increment hdNonAng;
        end if
        if (v[i].Chest_Pain = 2 & v[i].target = 0 & v[i].Sex = 1)
                increment hdNonAngM;
        end if
        if (v[i].Chest_Pain = 2 & v[i].target = 0 & v[i].Sex = 0)
                increment hdNonAngF;
        end if
        if (v[i].Chest_Pain = 3 & v[i].target = 0)
                increment hdTyp;
        end if
        if (v[i].Chest_Pain = 3 & v[i].target = 0 & v[i].Sex = 1)
                increment hdTypM;
        end if
        if (v[i].Chest_Pain = 3 & v[i].target = 0 & v[i].Sex = 0)
                increment hdTypF;
        end if
end for lop
Output to console newline "Number of Heart Disease Patients: "  HD newline;
```

Output to console newline "Chest Pain Type" set width(23) "Number of Patients" set width(9) "Male" set width(11) "Female" newline;

Output to console set width(7)"Typical" set width(22) hdTyp set width(17) hdTypM set width(9) hdTypF newline;

Output to console set width(8 "Atypical" set width(21) hdAtyp set width(17 hdAtypM set width(9) hdAtypF newline;

Output to console set width(4) "Non Anginal" set width(18) hdNonAng set width(17) hdNonAngM set width(9) hdNonAngF newline

Output to console set width(4) "Asymptomatic" set width(17) hdAsymp set width(17) hdAsympM set width(9) hdAsympF newline;

cout string(58, '-') newline;

pause console

end

3.12 Second Print Menu Function

int printMenu3()

start

int pick → 0;

Output to console "Pick an Option newline"

"(1) Diabetes newline"

"(2) High Blood Pressure newline"

"(3) Cholesterol newline"

"(4) Exit newline";

input to console set width(1) pick;

return pick;

end

3.13 Cholesterol Function

void compChol(const vector<Records>& v)

start

```
    clear console;
    Output stream outfile("Chol_Results.txt");
    int Yes → 0;          int YesPerc → 0;
    int No → 0;           int NoPerc → 0;
    int Ychol1 → 0;       int Ychol1Perc → 0;
    int Ychol0 → 0;       int Ychol0Perc → 0;
    int Nchol1 → 0;       int Nchol1Perc → 0;
    int Nchol0 → 0;       int Nchol0Perc → 0;
    int totalchol1 → 0;
    int totalchol0 →0;
    double Avgchol1 → 0;
    double Avgchol0 → 0;
    for (vector<Records>::size_type i = 0; i ≠ vector size; increment i)
        if (v[i].target = 0)
            increment Yes;
            totalchol1 += v[i].chol;
        end if
        if (v[i].target = 1)
            increment No;
            totalchol0 += v[i].chol;
        end if
```

```
        if ((v[i].target = 0) & (v[i].chol ≥ 240))
            increment Ychol1;
        end if
        if (v[i].target = 0 & v[i].chol < 240)
            increment Ychol0;
        end if
        if ((v[i].target = 1) & (v[i].chol ≥ 240))
            increment Nchol1;
        end if
        if (v[i].target = 1 & v[i].chol < 240)
            increment Nchol0;
        end if
    end for loop
    Avgchol1 = totalchol1 / Yes;
    Avgchol0 = totalchol0 / No;
    YesPerc = round((Yes) / vector size) x 100);
    NoPerc = round((No) / vector size) x 100);
    Ychol1Perc = round((Ychol1) / Yes) x 100);
    Ychol0Perc = round((Ychol0) / Yes) x 100);
    Nchol1Perc = round((Nchol1) / No) x 100);
    Nchol0Perc = round((Nchol0) / No) x 100);
    Output to console set width (30) "Heart Disease" newline;
    Output to console set width(16) "/" set width(16) "\\" newline;
    Output to console set width(15) "/" set width(18) "\\" newline;
```

```
    Output to console set width(14) "/" set width(20) "\\" newline;

    Output to console set width(14) "Yes" << set width(22) << "No" newline;

    Output to console set width(14) Yes set width(23) No newline;

    Output to console set width(13) YesPerc "%" set width(22) NoPerc "%" newline;

    Output to console set width(11) "/" set width(4) "\\" set width(19) "/" set width(4) "\\" newline;

    Output to console set width(10) "/" set width(6) "\\" set width(17) "/" set width(6) "\\" newline;

    Output to console set width(9) "/" set width(8) "\\" set width(15) "/" set width(8) "\\" newline;

    Output to console set width(8) "/" set width(10) "\\" set width(13) "/" set width(10) "\\" newline;

    Output to console set width(8) "chol>240" set width(15 "chol<240" set width(12) "chol>240" set
    width(12)"chol<240" newline;

    Output to console set width(6) Ychol1 set width(15) Ychol0 set width(12) Nchol1 set width(12)
    Nchol0 newline;

    Output to console set width(6) Ychol1Perc "%" set width(13) Ychol0Perc "%" set width(12
    Nchol1Perc "%" set width(10) Nchol0Perc "%" newline;

    Output to console newline;

    Output to console "Average Cholesterol For Heart Disease Patients: " << Avgchol1 newline newline;

    Output to console "Average Cholesterol For Non Heart Disease Patients: " << Avgchol0 newline

    Output to console string(56, '-') newline;

    pause console

end
```

3.14 CA Function

```
void compCA(const vector<Records>& v)

start

    clear console

    Output stream outfile("CA_Results.txt");
```

```
int vessel0 = 0;  int HDvessel0 = 0;

int vessel1 = 0;  int HDvessel1 = 0;

int vessel2 = 0;  int HDvessel2 = 0;

int vessel3 = 0;  int HDvessel3 = 0;

int HeartDisease = 0;

int zeroPercent = 0;       int twoPercent = 0;

int onePercent = 0;        int threePercent = 0;


for (vector<Records>::size_type i = 0; ; i ≠ vector size; increment i)

        if (v[i].fluorscopy = 0 or v[i].fluorscopy = 1 or v[i].fluorscopy = 2 or v[i].fluorscopy = 3 or v[i].fluorscopy = 4)

                continue;

        end if

        else

                Output to console "line: " i + 1 newline;

                throw domain_error("Incorrect Value for CA found");

        end else

end for loop

for (vector<Records>::size_type i = 0; i ≠ vector size; increment i)

        if (v[i].target = 0)

                increment HeartDisease;

        end if

        if (v[i].fluorscopy = 0)

                increment vessel0;
```

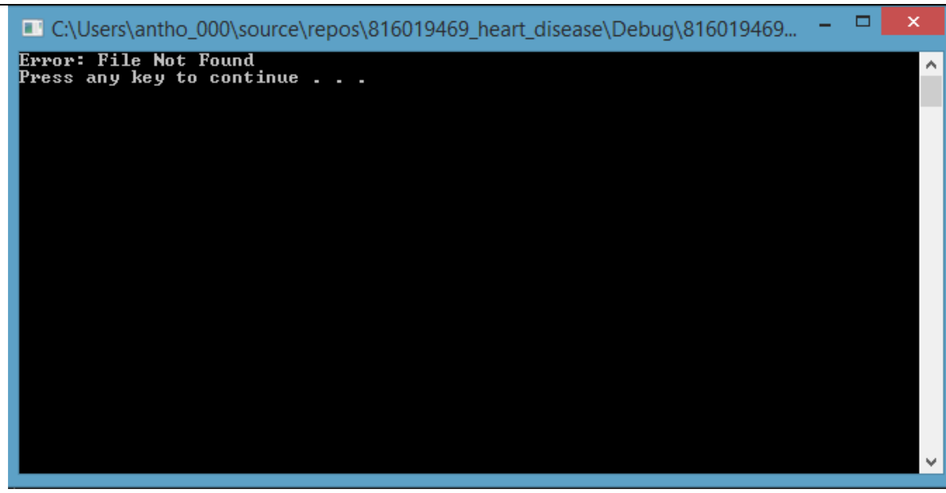
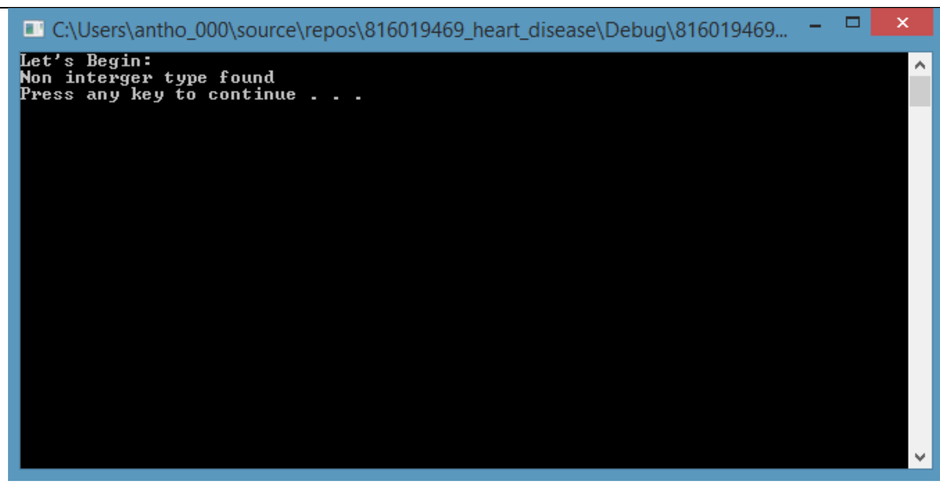
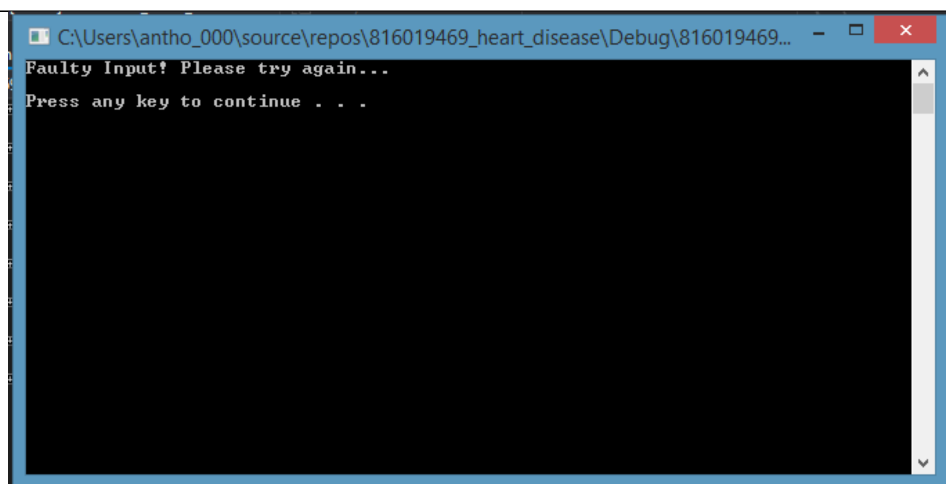
```
        end if
        if (v[i].fluorscopy = 1)
                increment vessel1;
        end if
        if (v[i].fluorscopy = 2)
                increment vessel2;
        end if
        if (v[i].fluorscopy = 3)
                increment vessel3;
        end if
        if (v[i].fluorscopy = 0 & v[i].target = 0)
                increment HDvessel0;
        end if
        if (v[i].fluorscopy = 1 & v[i].target = 0)
                increment HDvessel1;
        end if
        if (v[i].fluorscopy = 2 & v[i].target = 0)
                increment HDvessel2;
        end if
        if (v[i].fluorscopy = 3 & v[i].target = 0)
                increment HDvessel3;
        end if
end for loop
zeroPercent = round((HDvessel0) / vessel0) x 100);
```

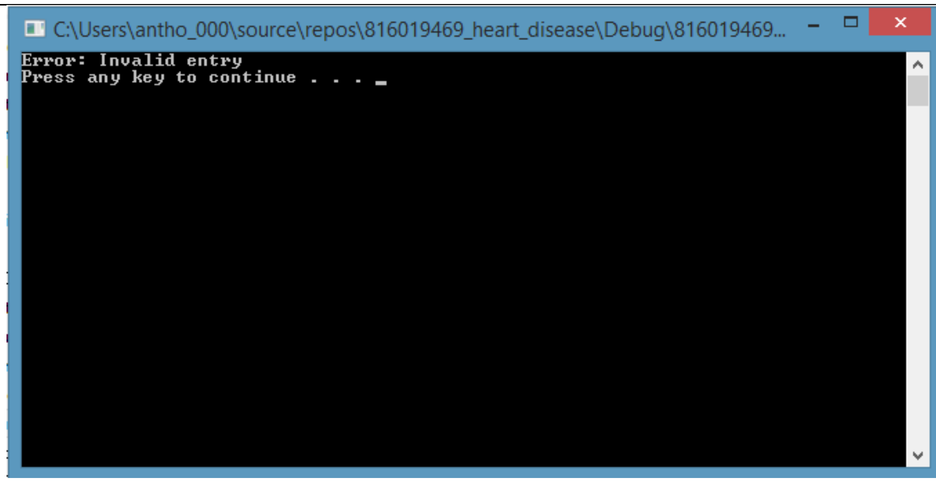
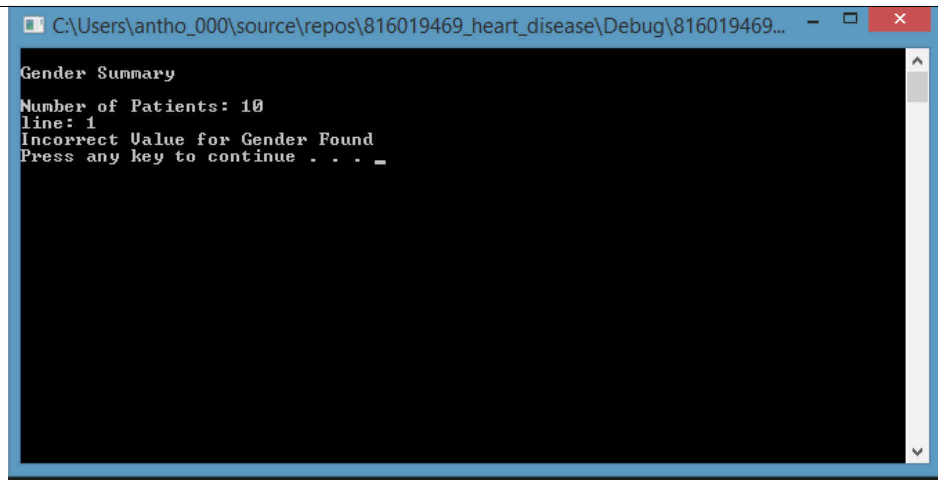
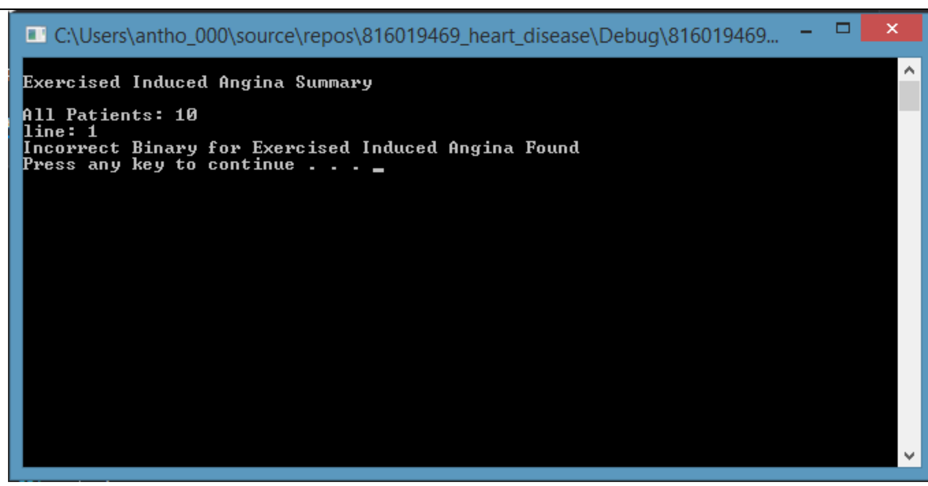
```
    onePercent = round((HDvessel1) / vessel1) x 100);

    twoPercent = round((HDvessel2) / vessel2) x 100);

    threePercent = round((HDvessel3) / vessel3) x 100);


    Output to console "Coloured Fluoroscopy Summary: " newline newline;

    Output to console "All Patients: " << vector size() newline newline;

    Output to console "Number of Blood Vessels Highlighted: " newline newline;

    Output to console "0:" set width(3) vessel0 newline;

    Output to console "1:"  set width(3) vessel1 newline;

    Output to console "2:" set width(3)  vessel2 newline;

    Output to console "3:" set width(3) vessel3 newline newline;

    Output to console string(35, '-') newline;

    Output to console "Heart Disease Patients: " HeartDisease newline;

    Output to console "Number of Blood Vessels Highlighted: " newline;

    Output to console "0:" set width(3) HDvessel0 " (" zeroPercent "%)" newline;

    Output to console "1:" set width(3) HDvessel1 " (" onePercent "%)" newline;

    Output to console "2:" set width(3) HDvessel2 " (" twoPercent "%)" newline;

    Output to console "3:" set width(3) HDvessel3 " (" threePercent "%)" newline;

    Output to console string(35, '-') newline;

    pause console

end
```

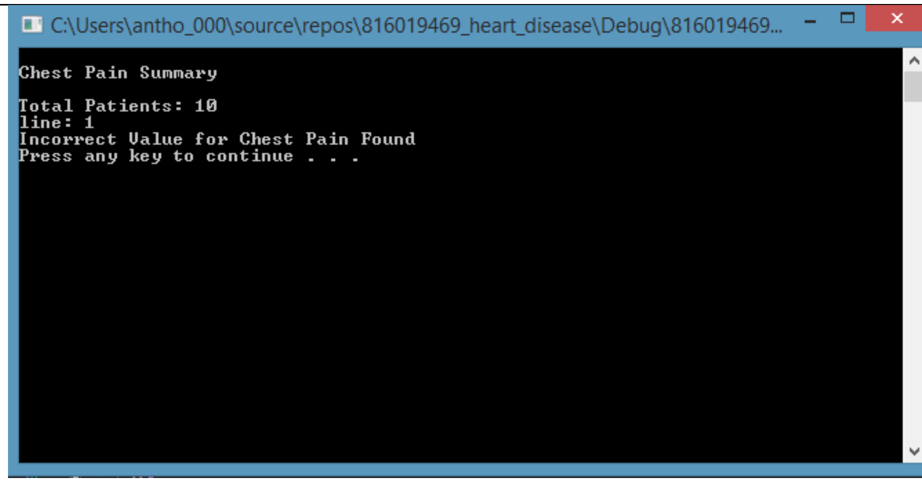
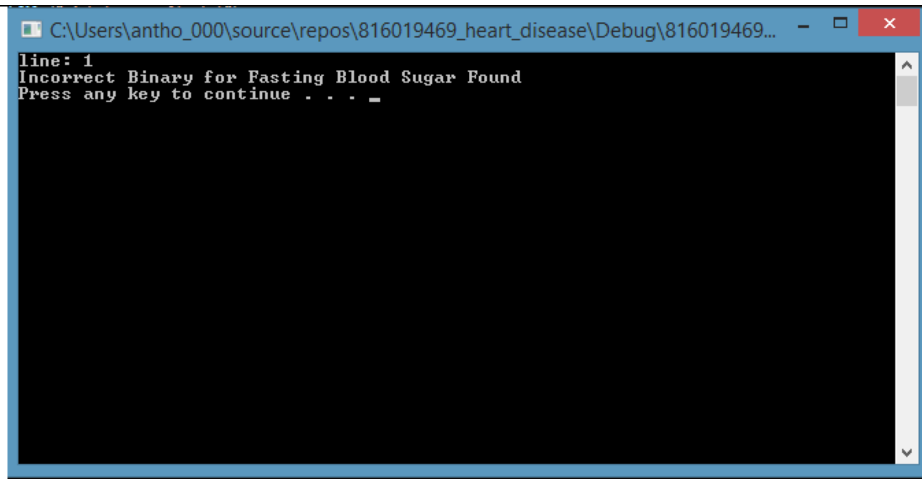
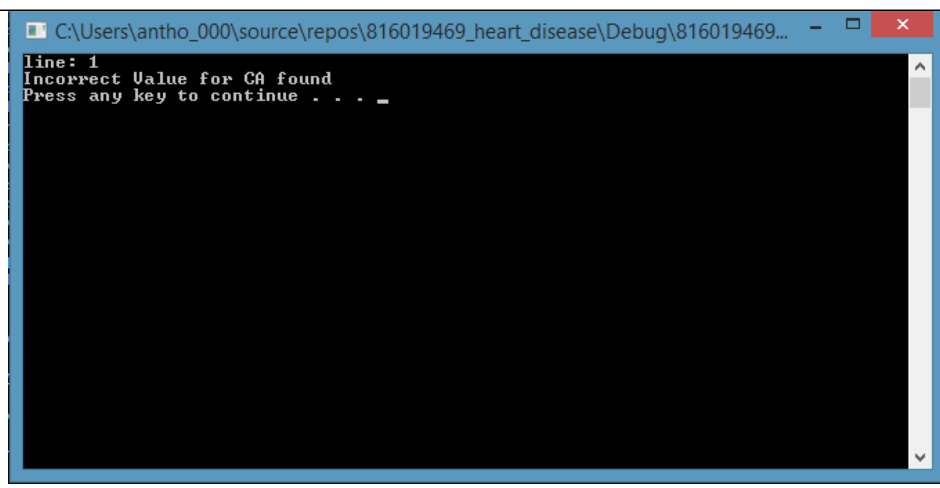
4 Test Cases (Invalid Input)

The program was checked to ensure that invalid inputs did not crash the program and that appropriate error messages were output to the console to inform the user. The results of each test are show below.

Table 1: Showing the program's responses to invalid input

<table>
<tr><th>Error Type</th><th>Error Message</th></tr>
<tr><td>Input file not located</td><td>
</td></tr>
<tr><td>Input file contained non-integer data point</td><td>
</td></tr>
<tr><td>Letter input in the menu</td><td>
</td></tr>
</table>

<table>
<tr><td>Multiple numbers input in menu</td><td>
</td></tr>
<tr><td>Non-binary gender found in file</td><td>
</td></tr>
<tr><td>Non-binary value of exang found in file</td><td>
</td></tr>
</table>

Undefined chest pain value found in file	
Incorrect binary for fbs found in file	
Incorrect value of ca found in file	

5 Results

The data obtained from running the entire program using the Cleveland UCI heart disease dataset, is shown below.

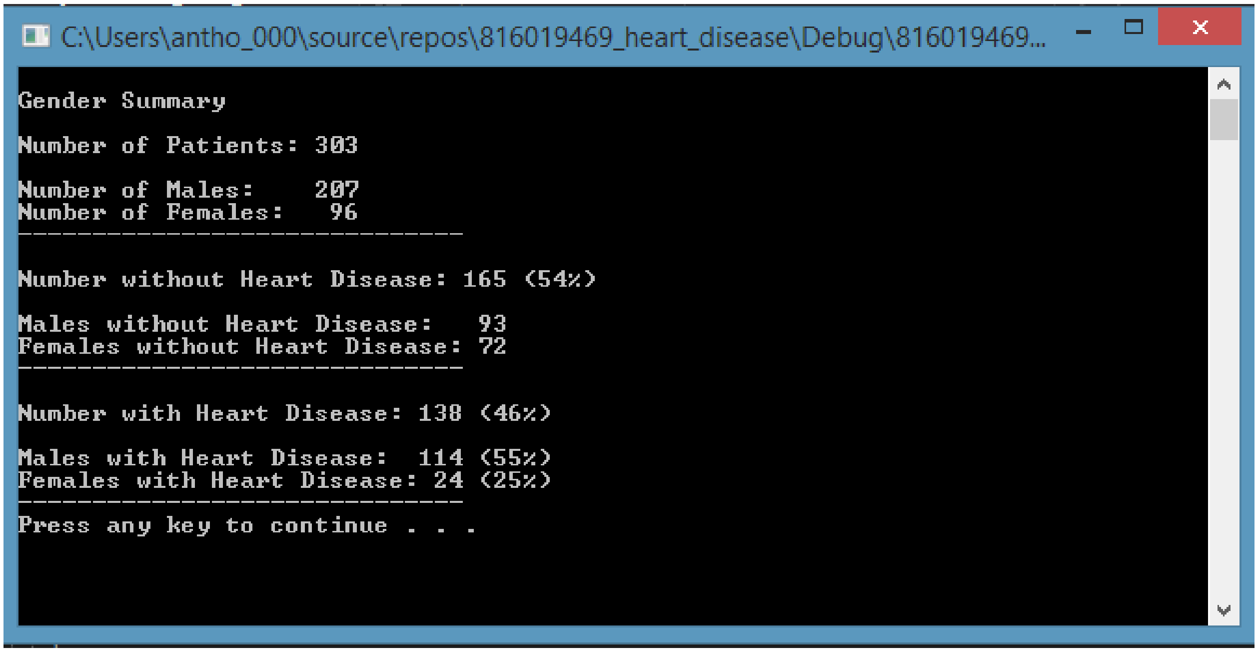


Figure 2: Showing Gender Correlation for Cleveland Dataset

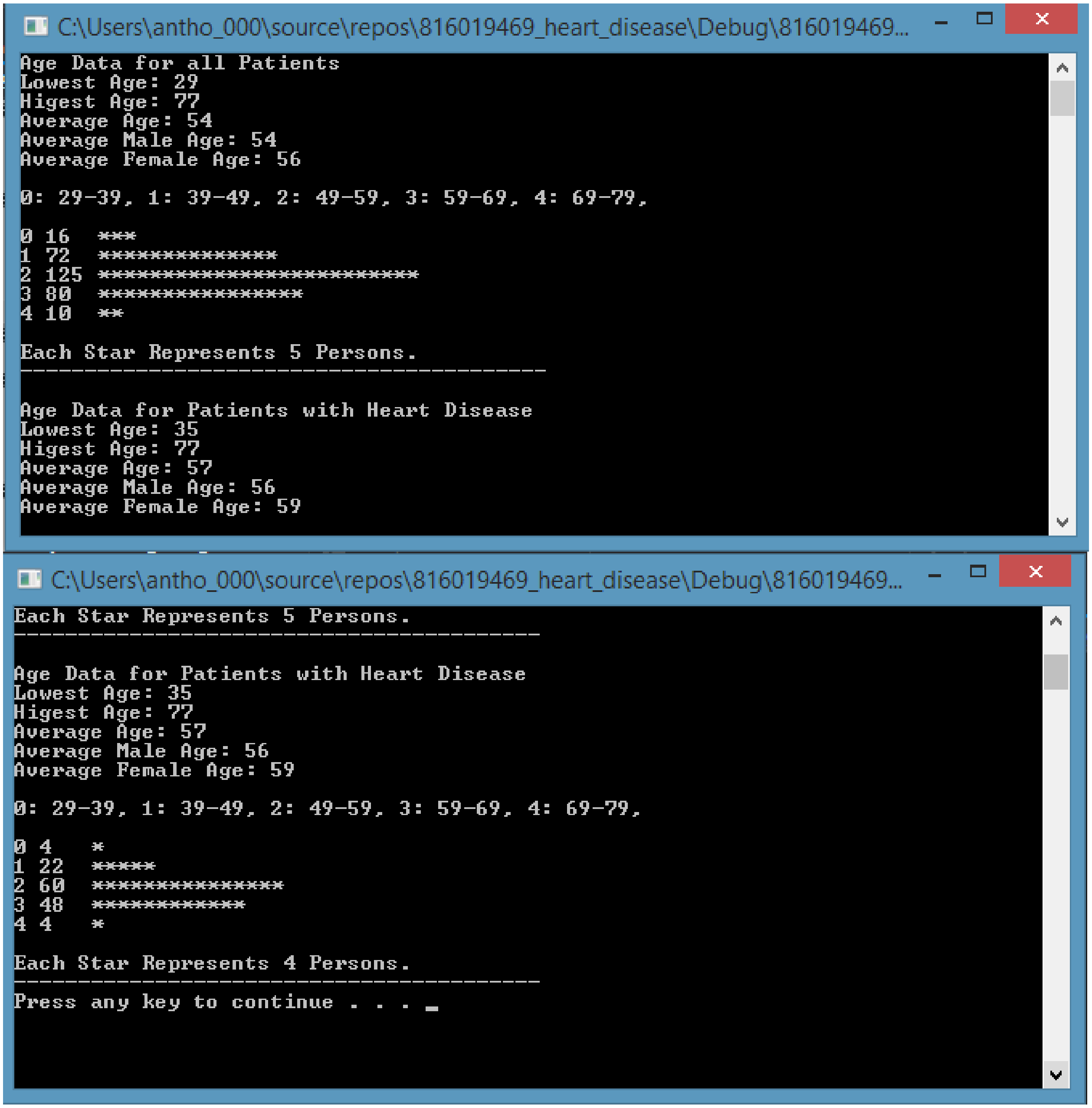


Figure 3:Showing the Most Common Age Groups and Average Ages

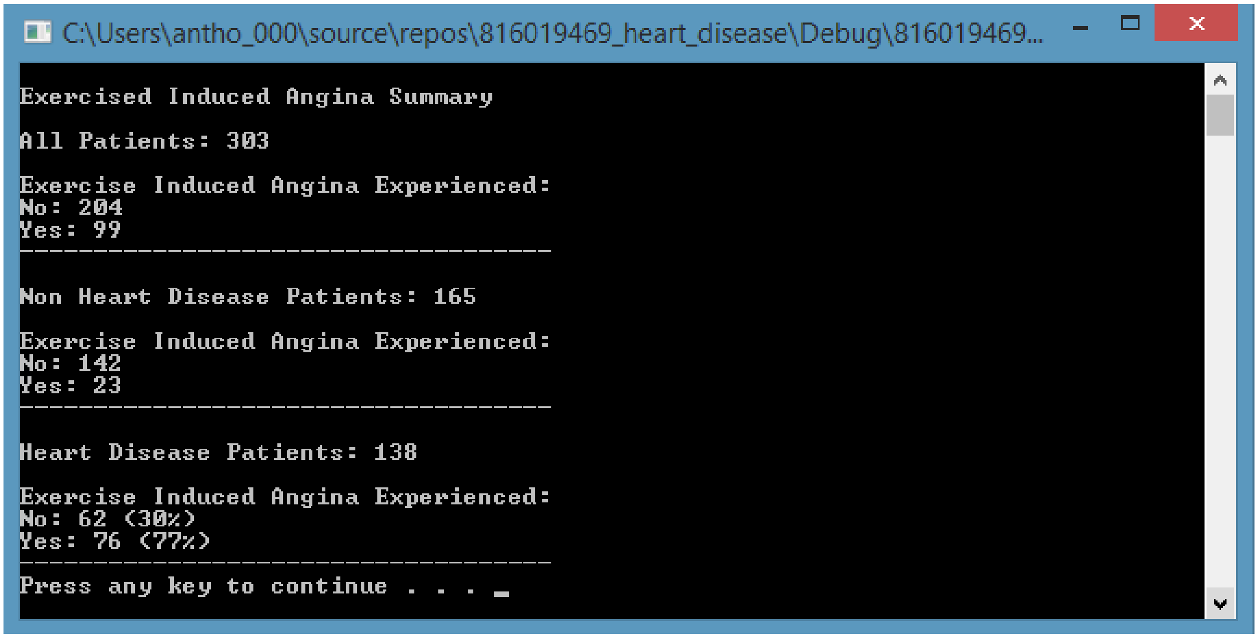


Figure 4: Showing the Percentage of Individuals Who Experienced Exercise Induced Angina

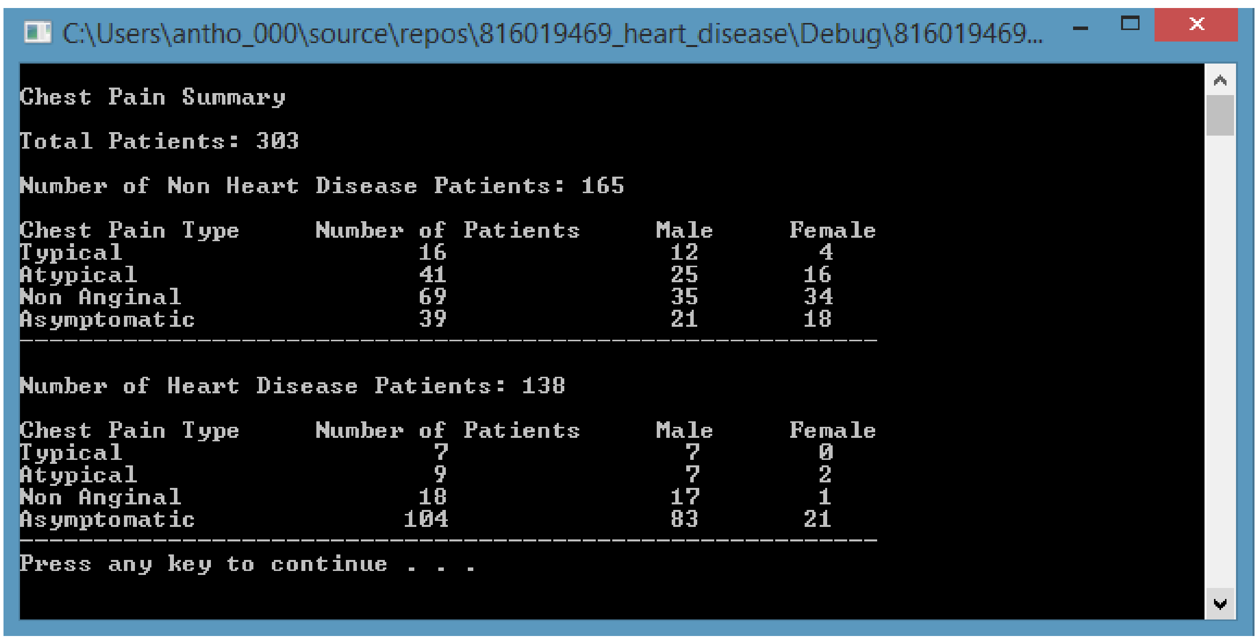


Figure 5: Showing the Classification of the Patients by Chest Pain Type

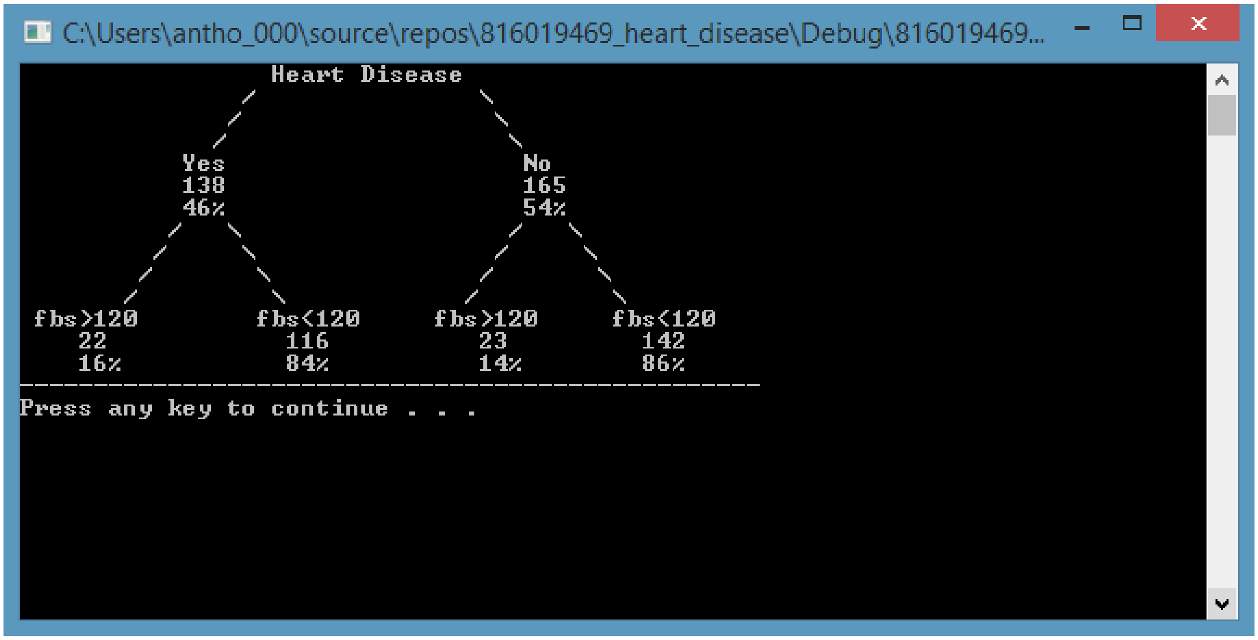


Figure 6: Showing Probability Tree for Diabetes

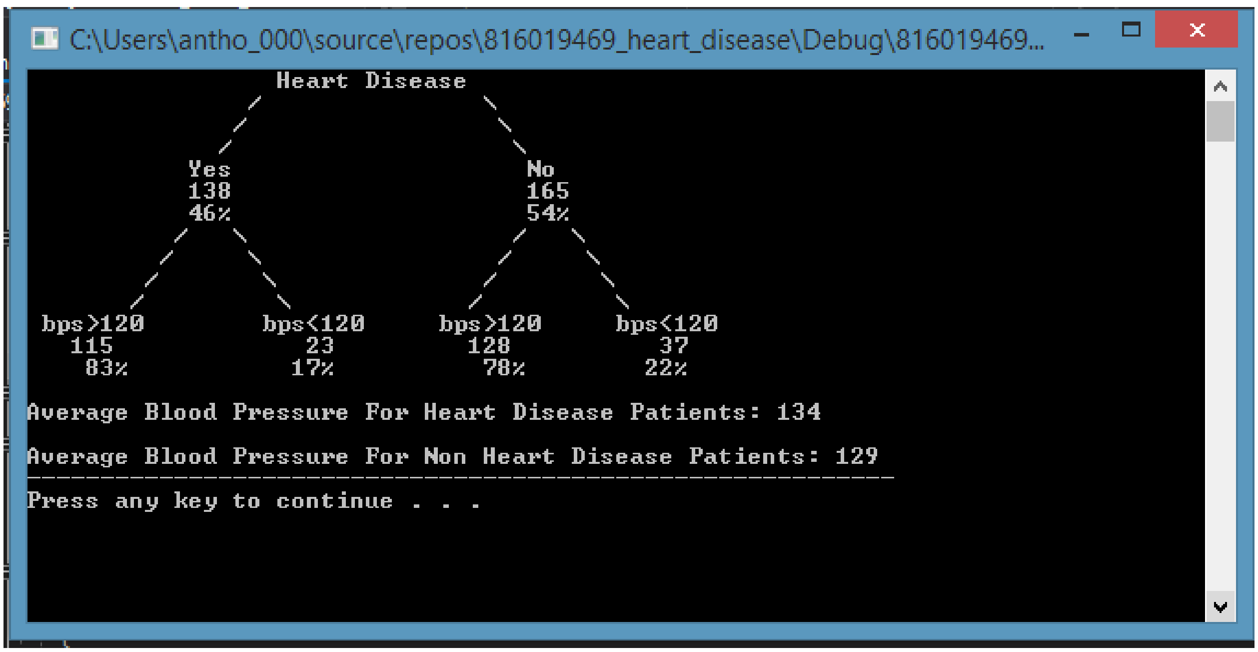


Figure 7: Showing Probability Tree for Hypertension

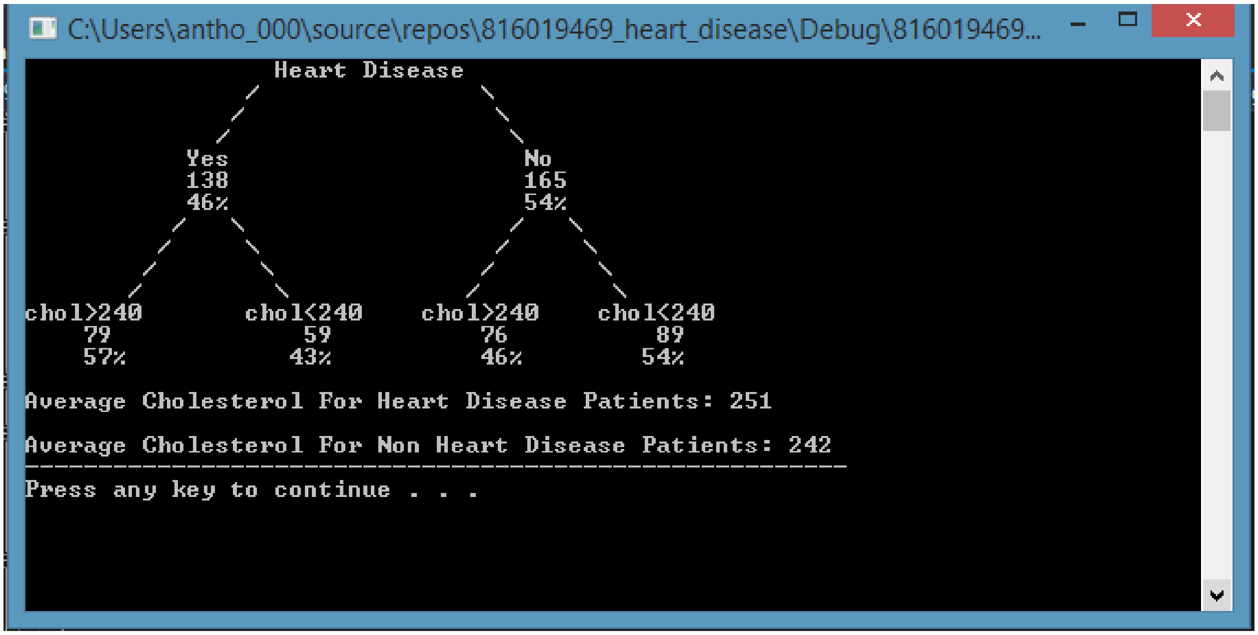


Figure 8: Showing the Probability Tree for High Cholesterol

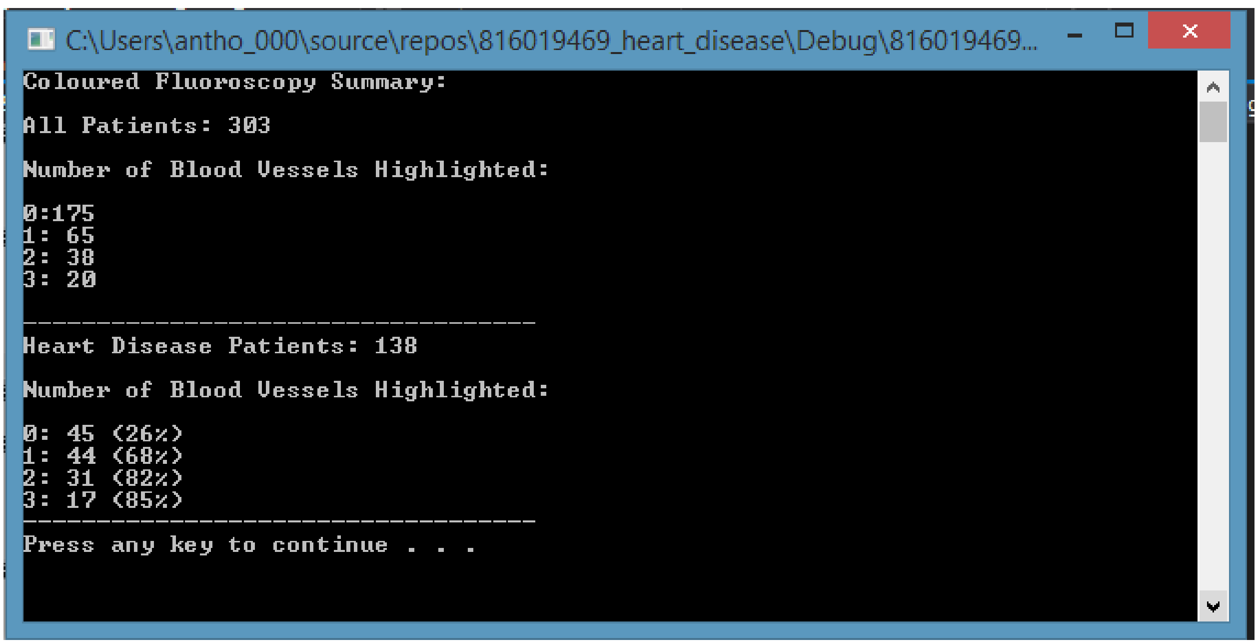


Figure 9: Showing the Fluoroscopy Data

All of the results shown in this section are for the Cleveland heart disease UCI dataset, however, the program can be applied to other datasets of the same layout. Furthermore, all deductions made from these results can be extrapolated to other patient groups.

6 Discussion

As seen in the results, a significantly larger number of males were diagnosed with heart disease than females. From the demographic of 207 males and 96 females, 114 or 55% of the males were diagnosed with heart disease compared to only 25% (24) of the females. This is in line with the NHLBI study, which stated that men were at a higher risk of developing heart disease than women. Therefore, gender can be used as an attribute in predicting heart disease.

The next attribute analysed was the age, to determine the most common age group for heart disease. For the entire dataset, the majority of individuals (125) were between 49-59 years and the average age was 54 years. Moreover, for the heart disease patients, the average was 56 years, with the average male and female ages being 56 year and 59 respectively. Furthermore, 60% of the patients in the 59-69 age group were diagnosed with heart disease and 48% in the 49-59 age group. This clearly showed that a greater proportion of heart disease patients lie within that age group. This finding corresponded with the WHO's findings, which stated that the majority of heart disease patients are under the age of 70.

Exercise induced angina, was the next attribute analysed to determine its relation to heart disease patients. In the entire demographic, 207 persons did not experience exercised induced angina, while 99 persons did. On further analysis, it was determined that 77% (76) of the persons who experienced exang, were diagnosed with heart disease while only 30% of those didn't experience exang were diagnosed. This showed that exercised induced angina is was good indicator to heart disease.

Unlike the other attributes, chest pain proved to have little correlation in diagnosing heart disease in this dataset. There were more non-heart disease patients who suffered from typical, atypical and non-anginal chest pain than heart disease patients who were dominantly asymptomatic. However, it is worthy to not that of the 7 heart disease patients who had typical pains, all were male and of the 9 who had atypical pains, 7 were also male.

The next 3 functions, in case 5 of the program analysed the fast blood sugar, resting blood pressure and cholesterol levels. It was found that only 16% of those diagnosed with heart disease were diabetic and 84% were not diabetic. This percentage showed that diabetes had very little correlation to heart disease. Unlike blood sugar, 83% of those diagnosed with heart disease had high blood pressure (systolic > 120), showing that it could be highly correlated. However, 78% of the non-heart disease patients also had high blood pressure. This made blood pressure unreliable as a possible predictor for heart disease. On the other hand, 57% of heart disease patients and 46% of non-heart disease patients had high cholesterol levels. This showed that between blood sugar, blood pressure and cholesterol, cholesterol had the highest correlation to heart disease.

The final attribute to be analysed, was ca, the number of blood vessels highlighted by the fluoroscopy. From this analysis, 85% of the heart disease patients had 3 vessels highlighted, 82% had 2 vessels highlighted and 68% had 1 vessel highlighted. This data showed that ca had the highest correlation to heart disease of all the attributes in the dataset and would be the most accurate in predicting heart disease.

7 Recommendations

Two recommendations to obtain better results from this data set are;

1. Utilize a combination of attributes to obtain better correlations to heart disease.
2. Use a combination of the three datasets available on the UCI machine learning website to have a larger demographic to analyse, which would yield more accurate results.

8 Conclusion

The main objectives, of the program was to determine, the relationship between 8 of the 14 attributes of the dataset and heart disease. Analysis of the data indicated that gender, patient age, exercised induced angina, cholesterol and ca had the greatest correlations to heart disease. Hence, they are the most suited useful features for predicting the presence of cardiac disease.

References

D. Mozaffarian et al., "AHA statistical Update", Hear. Dis. Stroke. Circ., 2015.

World Health Organization. “Cardiovascular Diseases (CVDs).” World Health Organization. World Health Organization, May 17, 2017.

https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)

“Coronary Heart Disease.” National Heart Lung and Blood Institute. U.S.

Department of Health and Human Services. Accessed May 21, 2020.

https://www.nhlbi.nih.gov/health-topics/espanol/enfermedad-coronaria.

Karthick, D., and B. Priyadharshini. “Predicting the Chances of Occurrence of Cardio Vascular Disease (CVD) in People Using Classification Techniques within Fifty Years of Age.” 2018 2nd International Conference on Inventive Systems and Control (ICISC), 2018. https://doi.org/10.1109/icisc.2018.8398990.

Rawat, Shubhankar. “Heart Disease Prediction.” Medium. Towards Data Science,December 9, 2019.

https://towardsdatascience.com/heart-disease-prediction-73468d630cfc.

Dominguez, Carlos. “A Detail Description of the Heart Disease Dataset.” Kaggle. Kaggle, April 20, 2020.

https://www.kaggle.com/carlosdg/a-detail-description-of-the-heart-disease-dataset.

Sajeev, Shelda, Anthony Maeder, Stephanie Champion, Alline Beleigoli, Cheng Ton, Xianglong Kong, and Minglei Shu. “Deep Learning to Improve Heart Disease Risk Prediction.” Machine Learning and Medical Engineering for Cardiovascular Health and Intravascular Imaging and Computer Assisted Stenting Lecture Notes in Computer Science, 2019, 96–103.

https://doi.org/10.1007/978-3-030-33327-0_12.

Anthony Meade May 25th, 2020

Milane, Aline, Jad Abdallah, Roy Kanbar, and Michella Ghassibe-Sabbagh. “Association of Hypertension with Coronary Artery Disease Onset in the Lebanese Population,” 2014. https://doi.org/10.1186/2193-1801-3-533.

“Recommendation of Attributes for Heart Disease Prediction Using Correlation Measure.” International Journal of Recent Technology and Engineering Special Issue 8, no. 2S3 (2019): 870–75. https://doi.org/10.35940/ijrte.b1163.0782s319.

Anthony Meade May 25th, 2020