HEART DISEASE PREDICTION SYSTEM

MINOR PROJECT REPORT

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

BACHELOR OF TECHNOLOGY

(Computer Science Engineering)

Submitted By:

Sparsh Gagneja (1706522)

Submitted To:

Mr. Kapil Sharma

CSE Department

Department of Computer Science & Engineering

Guru Nanak Dev Engineering College

Ludhiana-141006

ABSTRACT

Industrial training is one of the most important components in the fulfillment of any professional course conducted at any level and at any college. We have added advantages if we have a chance to come face to face with the tools and the processes we are being taught in our course. The main purpose of training program is to expose the trainees to the practical experience of the actual industrial conditions in which they are required to work in future.

The health care industries collect huge amounts of data that contain some hidden information, which is useful for making effective decisions. For providing appropriate results and making effective decisions on data, some advanced data mining techniques are used. In this study, an effective heart disease prediction system (EHDPS) is developed using neural network for predicting the risk level of heart disease. The system uses 15 medical parameters such as age, sex, blood pressure, cholesterol, and obesity for prediction. The EHDPS predicts the likelihood of patients getting heart disease. It enables significant knowledge, eg, relationships between medical factors related to heart disease and patterns, to be established. We have employed the multilayer perceptron neural network with backpropagation as the training algorithm. The obtained results have illustrated that the designed diagnostic system can effectively predict the risk level of heart diseases.

COMPANY CERTIFICATE

Auribises Technologies

Auribises is proud to award
certificate of completion
to

SPARSH GAGNEJA

for successfully completing training titled

Python Programming with AI
(Python, Data Analysis, Machine Learning)
Project - Heart Disease Prediction

for session

3rd June 2019 to 5th August 2019

given under our hand and seal on
this, 5th day of August 2019 at Ludhiana, India

JAS-ANZ IAF
An ISO 9001-2015 Company

AURIBISES TECHNOLOGIES PVT. LTD. LUDHIANA

Director

Registration No: AUR-2019516-1061

ACKNOWLEDGEMENT

I am highly grateful to the Dr. Sehijpal Singh, Principal, Guru Nanak Dev Engineering College (GNDEC), Ludhiana, for providing this opportunity to carry out the six-weeks industrial training at **AURIBISES, LUDHIANA**.

The constant guidance and encouragement received from Prof. G.S. SODHI Dean T&P, GNDEC Ludhiana has been of great help in carrying out the project work and is acknowledged with reverential thanks.

I would like to express a deep sense of gratitude and thanks profusely to **Mr. Ishant Kumar**, Director General of Company. Without the wise counsel and able guidance, it would have been impossible to complete the report in this manner.

The help rendered by Mr Jasvinder Singh and Mr. Gurpreet Singh Designation (Project Assistant, Department "Computer Interface Design") for experimentation is greatly acknowledged.

I express gratitude to other faculty members of Department of Computer Science & Engineering of GNDEC for their intellectual support throughout the course of this work.
Finally, I indebted to all whosoever have contributed in this report work and friendly stay at **AURIBISES, LUDHIANA**.

Sparsh Gagneja

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION TO COMPANY

Auribises offers a suite of mobile, web and software applications as a solution to industry. We were founded in November 2011. With unparalleled domain competencies in mobile and web, Auribises is poised to take on critical challenges that the industry manifests. Our culture is values based, and we assure the highest ethical standards of integrity, transparency and corporate governance. Our value system is driven by 3 D's, the acronym for our core values of DIVE, DEVELOP and DELIVER. At Auribises, we just not only develop, but also provide educational services that helps students to hone their technical skills. Auribises offers a unique combination of technical competencies with practical exposure on ongoing projects to its students. Auribises has a core vision for lightening thousands of candles with a single candle and hence we believe that it is our endeavor to continually share our learning's with the larger world.

Our objective is to surpass customer expectations consistently.

Auribises Technologies delivers Software Engineering Services and technology-driven Business Solutions that help meet the business objectives of our clients. We work constantly towards creating values in business for our customers, employees and partners.

OUR VISION

Delivering enduring value to customers, employees, investors and partners through totally fair, transparent and ethical practices.

OUR MISSION

To be a world class software company that enables its stakeholders unlock their value and realize their full potential by leveraging technology.

OUR VALUES

The values that drive us are based on integrity and honesty in all our dealings with all stakeholders. To grow and provide value consistently is the underlying driver.

CHAPTER 2: INTRODUCTION TO PROJECT

2.1 OVERVIEW

Heart disease describes a range of conditions that affect your heart. Diseases under the heart disease umbrella include blood vessel diseases, such as coronary artery disease, heart rhythm problems (arrhythmias) and heart defects you're born with (congenital heart defects), among others.

The term "heart disease" is often used interchangeably with the term "cardiovascular disease". Cardiovascular disease generally refers to conditions that involve narrowed or blocked blood vessels that can lead to a heart attack, chest pain (angina) or stroke. Other heart conditions, such as those that affect your heart's muscle, valves or rhythm, also are considered forms of heart disease.

Among various life-threatening diseases, heart disease has garnered a great deal of attention in medical research. The diagnosis of heart disease is a challenging task, which can offer automated prediction about the heart condition of patient so that further treatment can be made effective. The diagnosis of heart disease is usually based on signs, symptoms and physical examination of the patient. There are several factors that increase the risk of heart disease, such as body cholesterol level, family history of heart disease, high blood pressure, maximum heart rate achieved and lack of physical exercise.

2.2 EXISTING SYSTEM

Big medical companies who are spread in whole world use these types of prediction software to provide their users an easy way to make sure that if they have heart disease or not. Even if their exists such software, they do not have machine learning algorithms in their software. Machine learning can provide further help after managing data like in doing predictions. So, the existing up to date systems are either owned by bigger owners, outdated systems are owned by very few owners and most of the owners don't even have any systems yet.

2.3 USER REQUIREMENT ANALYSIS

A person health is most important thing in his life. This need is beautifully completed through the software with the help content stored and presented in form of tables and even through

graphical representations. The graphical view provides crisp and clear information and comparisons of their data. It also stores the data of the user in the database.

From a user's point of view, software should always have a simple and clean interface, the software must be easy to understand and should be easy to use without the dependency of any other person. This software is designed with all the points kept in mind of what software should be like. It has a quite clean and beautiful interface with both light and dark color blend, without being tedious and cluttered operations. It is easy to understand each and every objective of the software because every specific objective is given a special window and name when it has to be performed. The software is quite light weighted and does not require any high-end computer specifications to run it. It is quick to process and perform everything which it is designed to do and what a user is expecting it to do.

2.4 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase to put forth a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are:

1. Technical Feasibility
2. Economic Feasibility
3. Operational Feasibility

A feasibility study aims to objectively and rationally uncover the strengths and weaknesses of an existing business or proposed venture, opportunities and threats present in the natural environment, the resources required to carry through, and ultimately the prospects for success. In this project, we will look into all the possibilities and analyze whether it is profitable to work onto it or not.

2.4.1 TECHNICAL FEASIBILITY

This assessment is based on an outline design of system requirements, to determine whether the store has the technical expertise to handle completion of the project.

When writing the report, the following were under consideration:

- A brief description of the framework to assess more possible factors which could affect the study
- The part of the business being examined (marketing, economical)
- The human and economic factor
- The possible solutions to the problem within minimum time

The technical feasibility assessment is focused on gaining an understanding of the present technical resources of the store and their applicability to the expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the need of the proposed system.
After the above analysis, it was concluded that the system is technically feasible.

2.4.2 ECONOMICAL FEASIBILITY

The purpose of an **economic feasibility** study is to demonstrate the net benefit of a proposed project for accepting or disbursing electronic funds/benefits, taking into consideration the benefits and costs to the agency, other state agencies, and the general public as a whole.

The following costs were estimated:

- One-time development cost
- One-time H/W cost (if the user does not have a computer system)
- Benefits in reduced cost, error and saving will be made by reduction of present system expenses, time saving and increased accuracy

Once the application is developed, it would yield the user results if it is used effectively. If the offers are sent strategically, it will surely be profitable for the user. After the above analysis, it was concluded that the system is economically feasible.

2.4.3 OPERATIONAL FEASIBILITY

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

The operational feasibility assessment focuses on the degree to which the proposed development project fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture and existing business processes.

To ensure success, desired operational outcomes must be imparted during design and development. These include such design-dependent parameters as reliability, maintainability, supportability, usability, reducibility, disposability, sustainability, affordability and others.

Clients Supports: Client support for present system in the proposed system, as the current procedure used takes more time and effort than proposed system. No major training and new skills are required as it is based on DBMS model. It will help in the time saving and fast processing and dispersal of user request and application. New product will provide all the benefits of present system with better performance such as improved information, better management and collection of the reports. After the above analysis, it was concluded that the system is operationally feasible.

2.5 OBJECTIVES OF PROJECT

For any software there is always some kind of aim or an idea which motivates a programmer to build the software. Every software developer has a goal to make the lives of their user easier and save their time. Therefore, every software has some objectives to perform so that its user can get the results they are looking for. The main objective of this project is to help user to:

1. To create a Machine Learning model which can predict the HEART DISEASE of an individual based on various parameters such as blood pressure, maximum heart rate achieved, blood sugar etc.
2. To create a user friendly interface of the Software.

CHAPTER 3: PRODUCT DESIGN

3.1 USER REQUIREMENTS

The user requirements of the project are quite simple and short. The user would feed the Logistic Regression the data which includes multiple factors such as Age, Sex, chest_pain, blood_pressure, cholestrol, blood_sugar, restecg, heart_rate, exercise_angina, depression, slope, vessels, thalassemia

The algorithm in return would predict i.e. either the user have heart disease or not. The user input and output is done via PyQT5.

Also the algorithm or the predictive model needs to be less commutative as well as accurate enough. These requirements were kept in mind during the course of the project.

3.2 FLOW CHART OF THE PROJECT

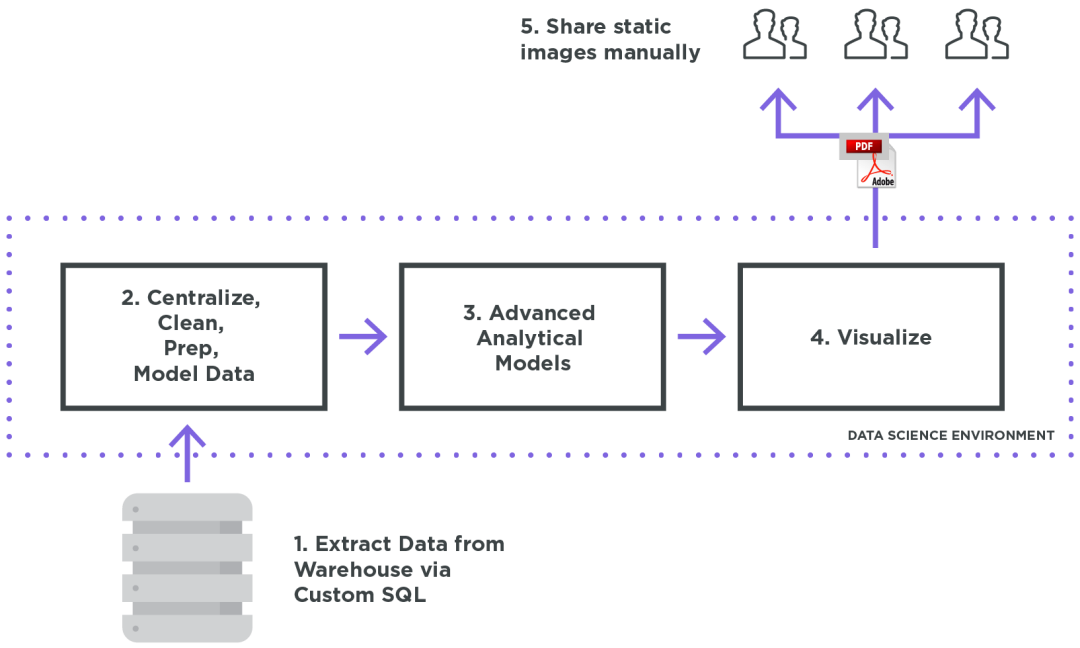


Figure 3. 1 depicts the workflow of a common data-science problem

Whenever we deal with any kind of data, we are dealing with the raw data in its raw format. There is no proper structure to the data. We need to clean the data, make it prepared for our next part for the training of our model.

The fig 2 shows below the project design, walking us through how the project implementation is carried out. The tuning part of our algorithm basically demonstrates cleaning of the data-set, balancing the different classes for our logistic predictive model to be more accurate, this can be achieved with the help of smote. Smote has been built in python for purposes like these only where we have to balance out our classes.

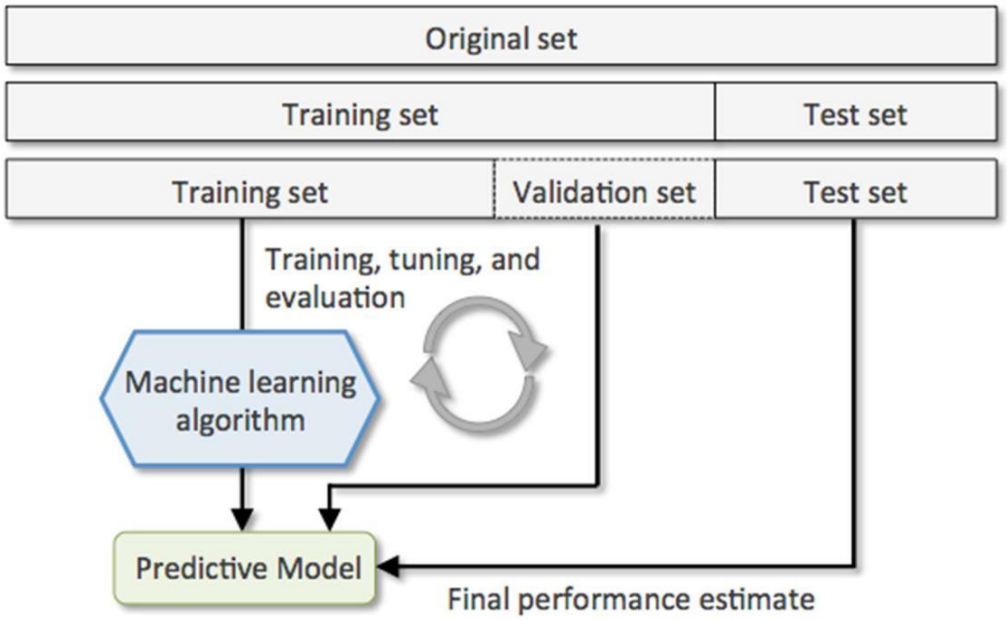


Figure 3. 2 Depicts the working of our project in a basic way

3.3 DATASET

A Data-set of 304 rows by 14 columns was provided as the data-set to train our model.

The attributes in the dataset are like Age, Education, Marital Status, Income, Gender, Native Country, Occupation etc.

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

Table 3. 1 The head of the data-set i.e. the first 5 rows of our data.

3.4 DATABASE DESIGN

The first step to designing any database in SQL is to identify what to include and what not to include. The next steps involve deciding how the included items relate to each other and then setting up tables accordingly.

In this software, database is designed in such a way that the every user's data is easily stored and can be accessed easily by simple SQL query.

Database contains 2 tables called PATIENT and PATIENT_COLUMNS. Every user who uses the Software have his details saved in the database.

Field	**Description**	**Range and Values**
Age	Age of the patient	0-100 in years
Sex	Gender of the patient	0-1 (1:Male 0:Female)
Chest Pain	Type of chest pain	1-4 (1: Typical Angina, 2: Atypical Angina, 3: Non-anginal, 4: Asymptotic)
Resting Blood Pressure	Blood pressure during rest	mm Hg
Cholesterol	Serum Cholesterol	mg / dl
Fasting Blood Sugar	Blood sugar content before food intake if >120 mg/dl	0-1 (0: False, 1: True)
ECG	Resting Electrocardiographic results	0-1 (0: Normal, 1: Having ST-T wave)
Max Heart Rate	Maximum heart beat rate.	Beats/min
Exercise Induced Angina	Has pain been induced by exercise	0-1 (0: No, 1: Yes)

Old Peak	ST depression induced by exercise relative to rest	0-4
Slope of Peak Exercise	Slope of the peak exercise ST segment	1-3 (1: Up sloping, 2: Flat, 3: Down sloping)
Ca	Number of vessels colored by fluoroscopy	0-3
Thal		3- normal 6-Fixed Defect 7- Reversible Defect
Num	Diagnostics of Heart Disease	0-1 (0: <50% Narrowing 1: >50% Narrowing)

Table 3. 2 Database Description

3.5 TABLE STRUCTURE

As mentioned above, there are 2 tables in the database named PATIENT and PATIENT_COLUMNS. PATIENT table has 7 columns i.e. Id which is a primary key, Name of the user, Age of the user, Email Id of the user, Gender of the user,Contact number of the user, Email and Address of the user.

PATIENT_COLUMNS has 14 columns i.e. Id which is a primary key, Age, gender, chest_pain, blood_pressure, cholestrol, blood_sugar, restecg, heart_rate, exercise_angina, depression, slope, vessels, thalassemia .

Serial No.	Attribute	Description
1	Sex	value 1: Male value 0: Female
2	Chest Pain Type	value 1: typical type 1 angina value 2: typical type angina value 3: non angina pain value 4: asymptomatic
3	Fasting Blood Sugar	value 1: > 120 mg/dl

		value 0:< 120 mg/dl
4	RestECG	resting electrographic results value 0 : normal value 1 : 1 having ST-T wave abnormality value 2 : showing probable/definite left ventricular hypertrophy
5	Exang	exercise induced angina value 1 : yes value 0 : no
6	Slope	the slope of the peak exercise ST segment value 1: unsloping value 2: flat value 3: downsloping
7	CA	number of major vessels colored by floursopy (value 0 – 3)
8	Thal	value 3: normal value 6: fixed defect value 7: reversible defect
9	Blood Pressure	(mm Hg on admission to the hospital)
10	Serum Cholesterol	(mg/dl)
11	Thalach	maximum heart rate achieved
12	Oldpeak	ST depression induced by exercise
13	Age	In years

Table 3. 3 Attributes of the proposed System

3.6 ASSUMPTIONS AND DEPENDENCIES

The data-set received was assumed to be just raw data and some cleaning of the data is required. Certain attributes might not be interfering the cause of their heart disease. In simple words certain attributes might be redundant and do not contribute towards the heart disease. So we have to check the heart disease dependencies and include only those attributes which contribute towards heart disease in our train data. This need to be done otherwise the results might not be fully accurate.

CHAPTER4:DEVELOPMENT AND IMPLEMENTATION

4.1 INTRODUCTION TO THE LANGUAGES AND SOFTWARE'S USED

4.1.1 CORE PYTHON

Introduction to the Language

Python is one of those rare languages which can claim to be both simple and powerful. You will find yourself pleasantly surprised to see how easy it is to concentrate on the solution to the problem rather than the syntax and structure of the language you are programming in.

Features

Python is a simple and minimalistic language. Reading a good Python program feels almost like reading English, although very strict English! This pseudo-code nature of Python is one of its greatest strengths. It allows you to concentrate on the solution to the problem rather than the language itself.

Easy to Learn

As you will see, Python is extremely easy to get started with. Python has an extraordinarily simple syntax, as already mentioned.

Free and Open Source

In simple terms, you can freely distribute copies of this software, read its source code, make changes to it, anduse pieces of it in new free programs. FLOSS is based on the concept of a community whichshares knowledge. This is one of the reasons why Python is so good it has been created and isconstantly improved by a community who just want to see a better Python.

High-level Language

When you write programs in Python, you never need to bother about the low-level details such as managing the memory used by your program, etc.

Portable

Due to its open-source nature, Python has been ported to (i.e. changed to make it work on) many platforms. All your Python programs can work on any of these platforms without requiring any changes at all if you are careful enough to avoid any system-dependent features. You can use Python on GNU/Linux, Windows, FreeBSD, Macintosh, Solaris, QNX, VMS, Psion, Acorn RISC OS, VxWorks, PlayStation, Windows CE and PocketPC! You can even use a platform like Kivy to create games for your computer and for iPhone, iPad, and Android.

Interpreted

When you run the program, the linker/loader software copies the program from hard disk to memory and starts running it. Python, on the other hand, does not need compilation to binary. You just run the program directly from the source code. Internally, Python converts the source code into an intermediate form called bytecodes and then translates this into the native language of your computer and then runs it.All this, actually, makes using Python much easier since you don't have to worry about compiling the program, making sure that the proper libraries are linked and loaded, etc.

Why is Python a language of choice for data scientists?

Ease of Learning — Advantage Python: Python is easier to learn. The syntax is intuitive. A lot of complexities are handled by Python internally

Library/ Module power: Python is equipped with modules like NumPy, Matplotlib, SciPy and pandas to name few amongst many which are very powerful for data visualization and analysis.

Minimalistic Approach: Python is a very simple and elegant language to use. Less complicated syntax unlike other traditional languages like C, C++ and JAVA; makes Python a very viable and feasible option for Data Scientists as well as Developers.

Scalability: Python's scalability lies in the flexibility that it gives to solve problems, as in the case of YouTube that migrated to Python

Python community: As Python extends its reach to the data science community, more and more volunteers are creating data science libraries.

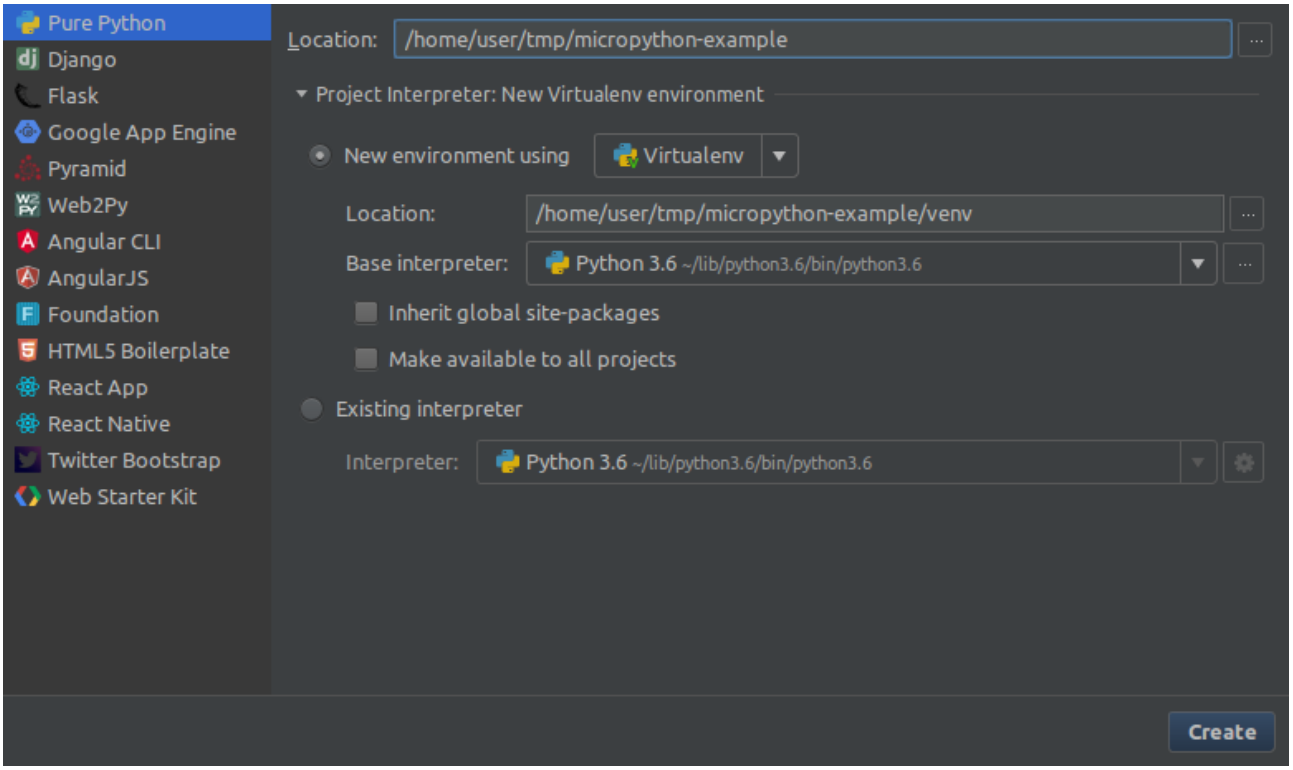
Graphics and visualization: Python come's with varied visualization options. Matplotlib provides the solid foundation around which other libraries like Seaborn, pandas plotting, and ggplot have been built.

4.1.2 PYCHARM

PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains. It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django as well as Data Science with Anaconda.

PyCharm is cross-platform, with Windows, macOS and Linux versions. The Community Edition is released under the Apache License, and there is also Professional Edition with extra features – released under a proprietary license

PyCharm provides API so that developers can write their own plugins to extend PyCharm features. Several plugins from other JetBrains IDE also work with PyCharm. There are more than 1000 plugins which are compatible with PyCharm.



4.2 PYTHON LIBRARIES

4.2.1 NUMPY

NumPy (or NumPy) is a Linear Algebra Library for Python, the reason it is so important for Data Science with Python is that almost all of the libraries in the PyData Ecosystem rely on NumPy as one of their main building blocks. NumPy is also incredibly fast, as it has bindings to C libraries

4.2.1.1 INSTALLATION INSTRUCTIONS:

It is highly recommended you install Python using the Anaconda distribution to make sure all underlying dependencies (such as Linear Algebra libraries) all sync up with the use of a conda install. If you have Anaconda, install NumPy by going to your terminal or command prompt and typing:

conda install numpy
into the anaconda prompt and press enter to install it.

Once you've installed NumPy you can import it as a library:

```
In [1]: import numpy as np
```

4.2.1.2 NUMPY ARRAYS

NumPy arrays are the main way we will use Numpy throughout the machine learning notebooks. Numpy arrays essentially come in two flavours: vectors and matrices. Vectors are strictly 1-d arrays and matrices are 2-d (but you should note a matrix can still have only one row or one column).

Let's begin our introduction by exploring how to create NumPy arrays.

Creating NumPy Arrays

Arange

Return evenly spaced values within a given interval.

```
In [6]: np.arange(0,10)
Out[6]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

In [7]: np.arange(0,11,2)
Out[7]: array([ 0,  2,  4,  6,  8, 10])
```

Randint

Return random integers from low (inclusive) to high (exclusive).

```
In [19]: np.random.randint(1,100)
Out[19]: 89

In [20]: np.random.randint(1,100,10)
Out[20]: array([99, 75, 34, 56, 17,  1, 47, 14, 41, 59])
```

4.2.1.3 NUMPY INDEXING AND SELECTION

Bracket Indexing and Selection

The simplest way to pick one or some elements of an array looks very similar to python lists

```
In [36]: #Get a value at an index
         arr[8]
Out[36]: 8

In [37]: #Get values in a range
         arr[1:5]
Out[37]: array([1, 2, 3, 4])

In [38]: #Get values in a range
         arr[0:5]
Out[38]: array([0, 1, 2, 3, 4])
```

4.2.1.4 NUMPY OPERATIONS

Arithmetic

You can easily perform array with array arithmetic, or scalar with array arithmetic. Let's see some examples:

```
In [62]: arr = np.arange(0,10)

In [63]: arr + arr
Out[63]: array([ 0,  2,  4,  6,  8, 10, 12, 14, 16, 18])

In [64]: arr * arr
Out[64]: array([ 0,  1,  4,  9, 16, 25, 36, 49, 64, 81])

In [65]: arr - arr
Out[65]: array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

4.2.2PANDAS

4.2.2.1 SERIES

A Series is very similar to a NumPy array (in fact it is built on top of the NumPy array object). What differentiates the NumPy array from a Series, is that a Series can have axis labels, meaning it can be indexed by a label, instead of just a number location. It also doesn't need to hold numeric data; it can hold any arbitrary Python Object.

```
In [1]: import numpy as np
        import pandas as pd
```

Creating a Series

You can convert a list,numpy array, or dictionary to a Series:

```
In [2]: labels = ['a','b','c']
        my_list = [10,20,30]
        arr = np.array([10,20,30])
        d = {'a':10,'b':20,'c':30}
```

Using Lists

```
In [3]: pd.Series(data=my_list)
Out[3]: 0    10
        1    20
        2    30
        dtype: int64
```

NumPy Arrays

```
In [7]: pd.Series(arr)

Out[7]: 0    10
        1    20
        2    30
        dtype: int64
```

Dictionary

```
In [9]: pd.Series(d)

Out[9]: a    10
        b    20
        c    30
        dtype: int64
```

4.2.2.2 DATAFRAMES

DataFrames are the workhorse of pandas and are directly inspired by the R programming language. We can think of a DataFrame as a bunch of Series objects put together to share the same index.

```
In [20]: from numpy.random import randn
         np.random.seed(101)

In [21]: df = pd.DataFrame(randn(5,4),index='A B C D E'.split(),columns='W X Y Z'.split())

In [22]: df
```

Out[22]:

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057
E	0.190794	1.978757	2.605967	0.683509

4.2.2.2.1 SELECTION AND INDEXING

Let's learn the various methods to grab data from a DataFrame

```
In [23]: # Pass a list of column names
         df[['W','Z']]
```

Out[23]:

	W	Z
A	2.706850	0.503826
B	0.651118	0.605965
C	-2.018168	-0.589001
D	0.188695	0.955057
E	0.190794	0.683509

Creating a new column:

df['new'] = df['W'] + df['Y'] df['new'] = df['W'] + df['Y']

This creates a new column named as ‘new’ in the DataFrame.

Removing Columns:

Similar to creating columns, columns can be dropped:

df. drop('new',axis=1,inplace=**True**)

4.2.2.2.2 CONDITIONAL SELECTION

An important feature of pandas is conditional selection using bracket notation, very similar to numpy.

```
In [43]: df>0
```

Out[43]:

	W	X	Y	Z
A	True	True	True	True
B	True	False	False	True
C	False	True	True	False
D	True	False	False	True
E	True	True	True	True

4.2.3 DATA INPUT AND OUTPUT

This section is the reference code for getting input and output, pandas can read a variety of file types using its pd.readmethods. Let's take a look at the most common data types:

4.2.3.1 CSV

CSV Input

```
In [122]: df1 = pd.read_csv('example')
          df1
```

Out[122]:

	a	b	c	d
0	0	1	2	3
1	4	5	6	7
2	8	9	10	11
3	12	13	14	15

CSV Output

```
In [123]: df.to_csv('example',index=False)
```

4.2.3.2 EXCEL

Pandas can read and write excel files, keep in mind, this only imports data. Not formulas or images, having images or macros may cause this read_excel method to crash.

Excel Input

```
In [124]: pd.read_excel('Excel_Sample.xlsx',sheetname='Sheet1')
```

Out[124]:

	Group	Num	A	B
0	G1	1	0.302665	1.693723
1	NaN	2	-1.706086	-1.159119
2	NaN	3	-0.134841	0.390528
3	G2	1	0.166905	0.184502
4	NaN	2	0.807706	0.072960
5	NaN	3	0.638787	0.329646

Excel Output

```
In [90]: df.to_excel('Excel_Sample.xlsx',sheet_name='Sheet1')
```

4.3 DATA VISUALIZATION WITH MATPLOTLIB

Introduction

Matplotlib is the "grandfather" library of data visualization with Python. It was created by John Hunter. He created it to try to replicate MATLAB's (another programming language) plotting capabilities in Python. So, if you happen to be familiar with MATLAB, matplotlib will feel natural to you.

It is an excellent 2D and 3D graphics library for generating scientific figures.

Some of the major Pros of Matplotlib are:

- Generally easy to get started for simple plots
- Support for custom labels and texts
- Great control of every element in a figure
- High-quality output in many formats
- Very customizable in general

Installation

You'll need to install matplotlib first with either:

conda install matplotlib

or pip install matplotlib

Importing

Import the `matplotlib.pyplot` module under the name `plt` (the tidy way):

```
In [1]: import matplotlib.pyplot as plt
```

You'll also need to use this line to see plots in the notebook:

```
In [2]: %matplotlib inline
```

That line is only for jupyter notebooks, if you are using another editor, you'll use: **plt.show()** at the end of all your plotting commands to have the figure pop up in another window.

Basic Example

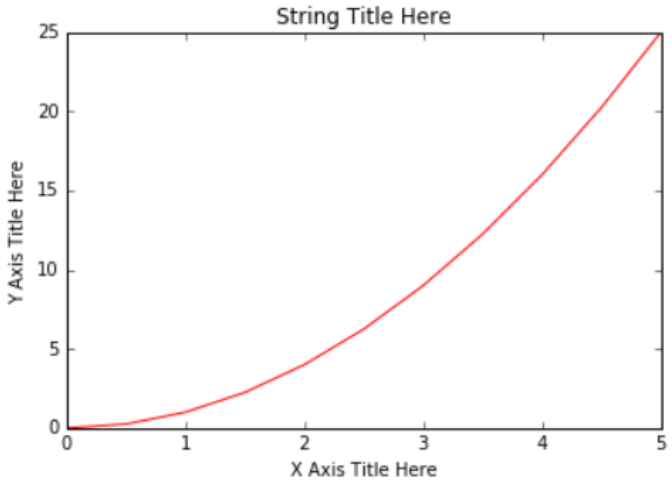
Let's walk through a very simple example using two numpy arrays. You can also use lists, but most likely you'll be passing numpy arrays or panda's columns (which essentially also behave like arrays). The data we want to plot:

```
In [7]: import numpy as np
        x = np.linspace(0, 5, 11)
        y = x ** 2

In [8]: x
Out[8]: array([ 0. ,  0.5,  1. ,  1.5,  2. ,  2.5,  3. ,  3.5,  4. ,  4.5,  5. ])

In [10]: y
Out[10]: array([  0.  ,   0.25,   1.  ,   2.25,   4.  ,   6.25,   9.  ,  12.25,
                 16.  ,  20.25,  25.  ])
```

Basic Matplotlib Commands

We can create a very simple line plot using the following (I encourage you to pause and use Shift+Tab along the way to check out the document strings for the functions we are using).

```
In [13]: plt.plot(x, y, 'r') # 'r' is the color red
         plt.xlabel('X Axis Title Here')
         plt.ylabel('Y Axis Title Here')
         plt.title('String Title Here')
         plt.show()
```



Saving figures

Matplotlib can generate high-quality output in a number of formats, including PNG, JPG, EPS, SVG, PGF and PDF.

To save a figure to a file we can use the save fig method in the Figure class:

```
In [68]: fig.savefig("filename.png")
```

Here we can also optionally specify the DPI and choose between different output formats:

```
In [69]: fig.savefig("filename.png", dpi=200)
```

Legends, labels and titles

Now that we have covered the basics of how to create a figure canvas and add axes instances to the canvas, let's look at how decorate a figure with titles, axis labels, and legends.

Figure titles

A title can be added to each axis instance in a figure. To set the title, use the set_title method in the axes instance:

```
ax.set_title("title");
```

Axis labels

Similarly, with the methods set_xlabel and set_ylabel, we can set the labels of the X and Y axes:

```
ax.set_xlabel("x")
ax.set_ylabel("y");
```

Legends

You can use the **label="label text"** keyword argument when plots or other objects are added to the figure, and then using the **legend** method without arguments to add the legend to the figure:

The **legend** function takes an optional keyword argument **loc** that can be used to specify where in the figure the legend is to be drawn.

The allowed values of **loc** are numerical codes for the various places the legend can be drawn. Some of the most common **loc** values are:

ax.legend(loc=1) # upper right corner

ax.legend(loc=2) # upper left corner

ax.legend(loc=3) # lower left corner

ax.legend(loc=4) # lower right corner

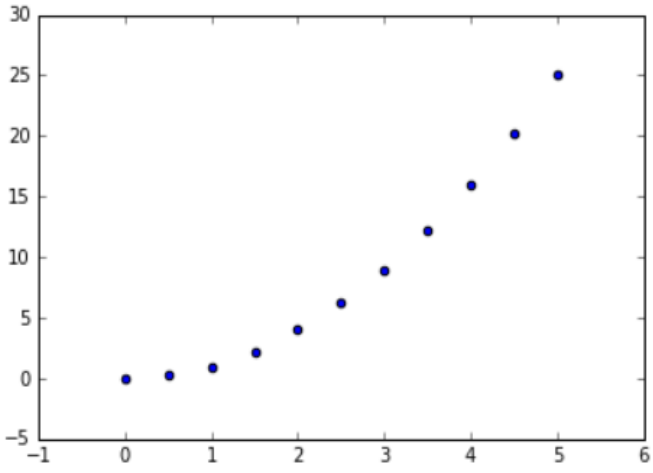
ax.legend(loc=0) # let matplotlib decide the optimal location

Special Plot Types

There are many specialized plots we can create, such as bar plots, histograms, scatter plots, and much more. Most of these types of plots we will actually create using seaborn, a statistical plotting library for Python. But here are a few examples of these type of plots:

Scatter-plot

```
In [60]: plt.scatter(x,y)
Out[60]: <matplotlib.collections.PathCollection at 0x1122be438>
```



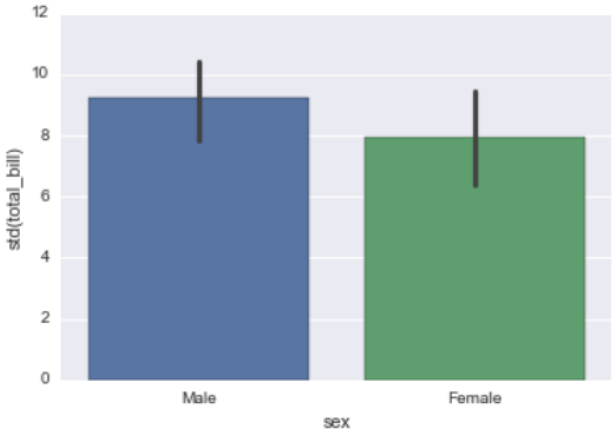
Barplot and countplot

These very similar plots allow you to get aggregate data off a categorical feature in your data. **barplot** is a general plot that allows you to aggregate the categorical data based off some function, by default the mean:

```
In [10]: import numpy as np
```

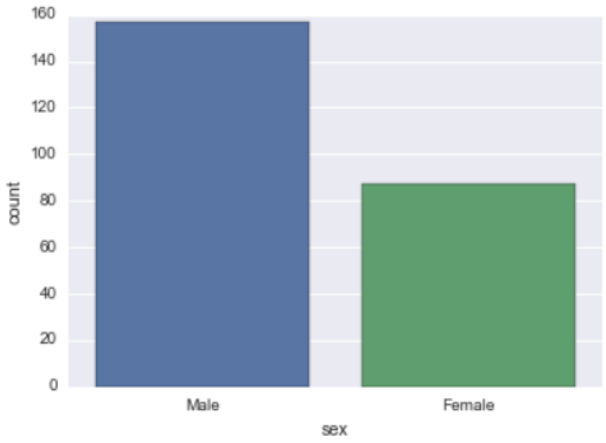
You can change the estimator object to your own function, that converts a vector to a scalar:

```
In [11]: sns.barplot(x='sex',y='total_bill',data=tips,estimator=np.std)
Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x11c9b00b8>
```



Countplot is essentially the same as barplot except the estimator is explicitly counting the number of occurrences. Which is why we only pass the x value:

```
In [13]: sns.countplot(x='sex',data=tips)
Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x1153276d8>
```

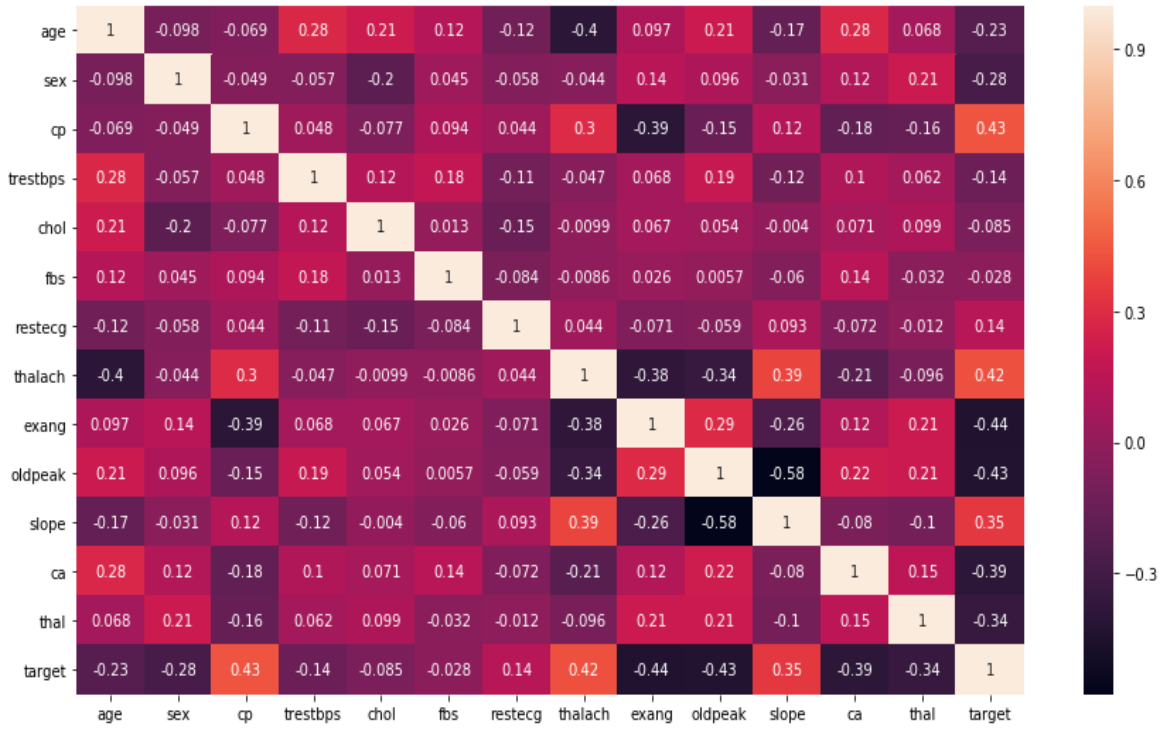


Heatmaps

A **heatmap** is a two-dimensional graphical representation of data where the individual values that are contained in a matrix are represented as colors. The seaborn **python** package allows the creation of annotated **heatmaps** which can be tweaked using **Matplotlib** tools as per the creator's requirement.

A heat map (or heatmap) is a graphical representation of data where the individual values contained in a matrix are represented as colors. "Heat map" is a newer term but shading matrices have existed for over a century.

In [11]:
```
fig = plt.gcf()
fig.set_size_inches(15, 8)
sns.heatmap(df.corr(), annot = True)
plt.show()
```



4.4 INTRODUCTION TO MACHINE LEARNING

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly

programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

Machine learning algorithms are often categorized as supervised or unsupervised.

- Supervised machine learning algorithms can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.

- In contrast, unsupervised machine learning algorithms are used when the information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system doesn't figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.

- Semi-supervised machine learning algorithms fall somewhere in between supervised and unsupervised learning, since they use both labeled and unlabeled data for training – typically a small amount of labeled data and a large amount of unlabeled data. The systems that use this method are able to considerably improve learning accuracy.

Machine learning enables analysis of massive quantities of data. While it generally delivers faster, more accurate results in order to identify profitable opportunities or dangerous risks, it may also require additional time and resources to train it properly. Combining machine learning with AI and cognitive technologies can make it even more effective in processing large volumes of information.

4.4.1LOGISTIC REGRESSION

Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes.

In simple words, the dependent variable is binary in nature having data coded as either 1 (stands for success/yes) or 0 (stands for failure/no).

Mathematically, a logistic regression model predicts P(Y=1) as a function of X. It is one of the simplest ML algorithms that can be used for various classification problems such as spam detection, Diabetes prediction, cancer detection etc.

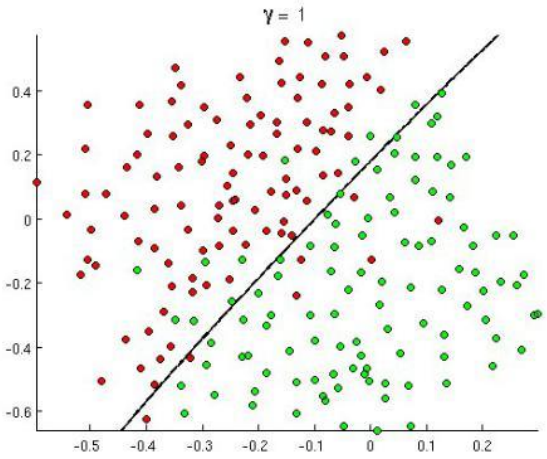


Figure 4. 1 demonstrates logistic regression

4.4.1.1 WHAT IS LOGISTIC REGRESSION ?

Logistic regression is a statistical method for analysing a dataset in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes). It is used to predict a binary outcome (1 / 0, Yes / No, True / False) given a set of independent variables.

You can also think of logistic regression as a special case of linear regression when the outcome variable is categorical, where we are using log of odds as dependent variable. In simple words, it predicts the probability of occurrence of an event by fitting data to a **logit** function.

Remember, there are some cases where dependent variables can have more than two outcomes, like married /unmarried/divorced such scenarios are classified as **multinomial logistic regression.** Though they work in the same manner to predict the outcome .
Some Familiar Example Of Logistics Regression :

Some prominent examples like:

- Email Spam Filter: Spam /No Spam
- Fraud Detection : Transaction is fraudulent, Yes/No
- Tumor: Benign/Malignant

4.4.1.2 HOW DOES LOGISTIC REGRESSION WORK?

As we are clear that logistics regression majorly makes predictions to handle problems which require a probability estimate as output, in the form of 0/1. Logistic regression is an extremely efficient mechanism for calculating probabilities. So you must be curious to understand how does it comes out with the value of 0 or 1 always. To understand more let's try to decode some math behind Logistic Regression.

4.4.1.3 LOGISTIC MODEL: SIGMOID FUNCTION

In order to map predicted values to probabilities, we use the Sigmoid function. The function maps any real value into another value between 0 and 1. In machine learning, we use sigmoid to map predictions to probabilities.
Let us try to understand logistic regression by understanding the logistic model. As in linear regression let's represent our hypothesis(Prediction Of Dependent Variable) in classification.
In classification our hypothesis representation which tries to predict the binary outcome of either o or 1, will look like,

hθ(x) = g(θ T x) = 1/ 1 + e −θ T x ,

Here g(z) = 1/(1 + e ^−z), is called the l**ogistic function or the sigmoid function:**

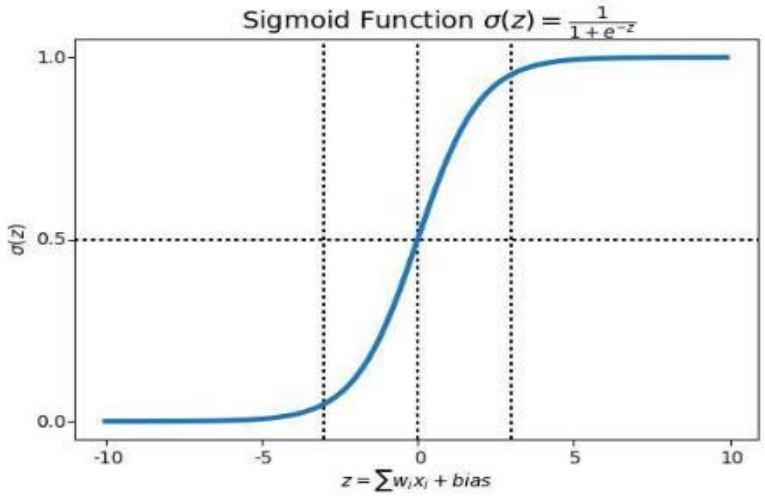


Figure 4. 2 depicts the sigmoid function traced on a graph

g(z): is a representation of logistic function, which we also call a Sigmoid function. From the above visual representation of sigmoid function, we can easily decipher how this curve describes many real-world situations, like population growth. In the initial stages it shows an exponential growth, but after some time, due to the competition for certain resources (bottle neck), the growth rate decreases until it gets to a stalemate and there is no growth.

The question here is, how does this **logit** (sigmoid function)helps us determine the probability of a classifying data into different classes. Let's try to understand how our logit function is calculated which will give us some clarity

4.4.1.4 MATHS BEHIND LOGISTIC FUNCTION:

Step 1: Classifying inputs to be in class zero or one.

First, we need to compute the probability that an observation belongs to class 1(we can also call it to be a positive class) using the Logistic Response Function. In this case, our z parameter is, as seen in the below given logit function.

$$P(y=1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n)}}$$

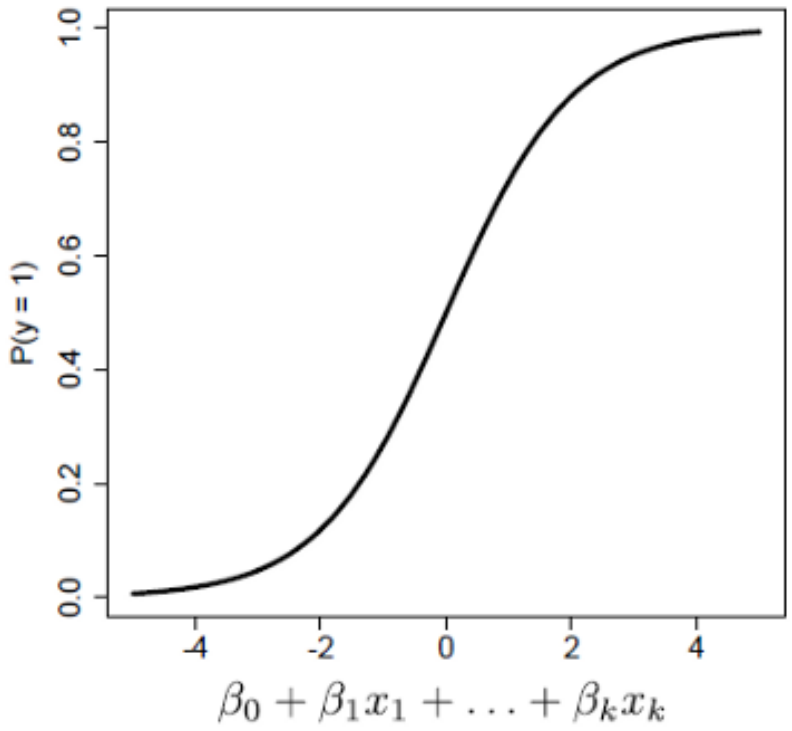


Figure 4. 3 Predicting a probability for observations belonging to class

- The coefficient beta 0, beta 1, beta K as shown in the fig above are selected to maximize the likelihood of predicting a high probability for observations belonging to class 1,
- And predicting a low probability for observations actually belonging to class 0. Above fig is depicting what we are trying to do with the coefficients**.**

Log Odds(Logit Function):

The above explanation can also be understood in terms of logarithmic odds, which is a kind of understanding the probabilities to classify elements into classes(1 or 0) is by using **ODDS**:

$$Odds = \frac{P(y=1)}{P(y=0)}$$

The Odds will be > 1 when there is a higher probability of predicting y = 1

The Odds will be < 1 when there is a higher probability of predicting y = 0

$$Odds = e^{\beta_0+\beta_1 x_1+\cdots+\beta_n x_n} \xrightarrow{applying\ log} \log(Odds) = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n$$

These Odds which resembles similarity to a linear regression is called the **logit.**
So, a logit is a log of odds and odds are a function of P. In logistic regression, we find
logit(P) = a + bX,

Step 2: Defining a boundary values for the odds

We now define a threshold boundary in-order to clearly classify each given input values into one of the classes.

We can choose a threshold value as per the business problem we are trying to solve , generally which is circled around 0.5. So if your probability values come out to be >0.5 we can classify such observation into class 1 type, and the rest into class 0. The choice of threshold value is generally based on error types, which are of two types , false positives, and false negatives.

A false positive error is made when the model predicts class 1, but the observation actually belongs to class 0. A false negative error is made when the model predicts class 0, but the observation actually belongs to class 1. The perfect model would classify all classes correctly: all 1´s (or trues) as 1´s, and all 0´s (or false) as 0´s. So we would have FN = FP = 0.

Threshold Values Impacts:

1. **Higher threshold value**

   Suppose if P(y=1) > 0.7. The model is being more restrictive when classifying as 1´s, and therefore, more False Negative errors will be made.

2. **Lower threshold value:**

Suppose, if P(y=1) > 0.3.

The model is now being less strict and we are classifying more examples as class 1, therefore, we are making more false positives errors.

4.4.1.5 CONFUSION MATRIX: A WAY TO CHOOSE AN EFFECTIVE THRESHOLD VALUE:

A confusion matrix also known as error matrix is a predictor of model performance on a classification problem. The number of correct and incorrect predictions is summarized with count values and broken down by each class. This lie at the core of the confusion matrix. The confusion matrix shows the ways in which your classification model is confused when it makes predictions on observations, it helps us to measure the type of error our model is making while classifying the observation into different classes.

		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Figure 4. 4 Confusion Matrix

Key Parts Of Confusion Matrix:

- **True Positive** (TP): This refers to the cases in which we predicted "**YES**" and our prediction was actually **TRUE**
  (Eg. Patient is actually having diabetes and you predicted it to be True)

- **True Negative** (TN): This refers to the cases in which we predicted "**NO**" and our prediction was actually **TRUE**
  (Eg. Patient is not having diabetes and you predicted the same.)

- **False Positive** (FP): This refers to the cases in which we predicted "**YES**", but our prediction turned out **FALSE**
  (Patient was not having diabetes, but our model predicted that s/he is diabetic)

- **False Negative** (FN): This refers to the cases in which we predicted "**NO**" but our prediction turned out **FALSE**

Key Learning Metrics From Confusion Matrix:

Confusion matrix helps us learn following metrics, helping us to measure logistic model performance.

Accuracy:

It answers:

Overall, how often is the classifier correct?
Accuracy = (TP+TN)/total No of Classified Item = (TP+TN)/ (TP+TN+FP+FN)

Precision:

It answers:

When it predicts yes, how often is it correct?

Precision is usually used when the goal is to limit the number of false positives(FP). For example, with the spam, filtering algorithm, where our aim is to minimize the number of reals emails that are classified as spam.

Precision = TP/ (TP+FP)

Recall:

It answers:

When it is actually the positive result, how often does it predict correctly?

It is calculated as,

Recall = TP/ (TP+FN), also known as sensitivity.

f1-score:

It is just the harmonic mean of precision and recall:

It is calculated as,

f1-score = 2*((precision*recall)/(precision+recall))

So when you need to take both precision and recall into account this f1 score is a useful metric to measure. If you try to only optimize recall, your algorithm will predict most examples to belong to the positive class, but that will result in many false positives and, hence, low precision. Also, if you try to optimize precision, your model will predict very few examples as positive results (the ones which highest probability), but recall will be very low. So it can be insightful to balance out and consider both and see the result.

4.4.1.6 TYPES OF LOGISTIC REGRESSION

Generally, logistic regression means binary logistic regression having binary target variables, but there can be two more categories of target variables that can be predicted by it. Based on those number of categories, Logistic regression can be divided into following types –

- **Binary logistic regression*:*** It has only two possible outcomes. Example- yes or no
- **Multinomial logistic regression**: It has three or more nominal categories.Example- cat, dog, elephant.
- **Ordinal logistic regression**- It has three or more ordinal categories, ordinal meaning that the categories will be in a order. Example- user ratings(1–5).

Implementing Logistic Regression :

We are going to cover this model building exercise in the following steps:

- Reading Data
- Analyzing Data(Basic EDA/Descriptive analysis)
- Train and Test(Break the sample data into two sets)
- Accuracy Report(Measuring the model performance using confusion matrix we discussed above)

Main Objective : To predict heart disease using Logistic Regression Classifier.

4.5 IMPLEMENTATION AND TESTING

After understanding the problem, what needs to be implemented came the implantation part. What needs to be implemented was sound and clear, we had to create a predictive model for the employee attrition. Some other things which were in user requirements were the fact that user wanted it to be a little interactive with inputs and output. So we need to use and lastly but most important, our model should be accurate.

The implementation and the testing of the project were done side by side keeping in mind to get the most accurate predictive model for our problem.

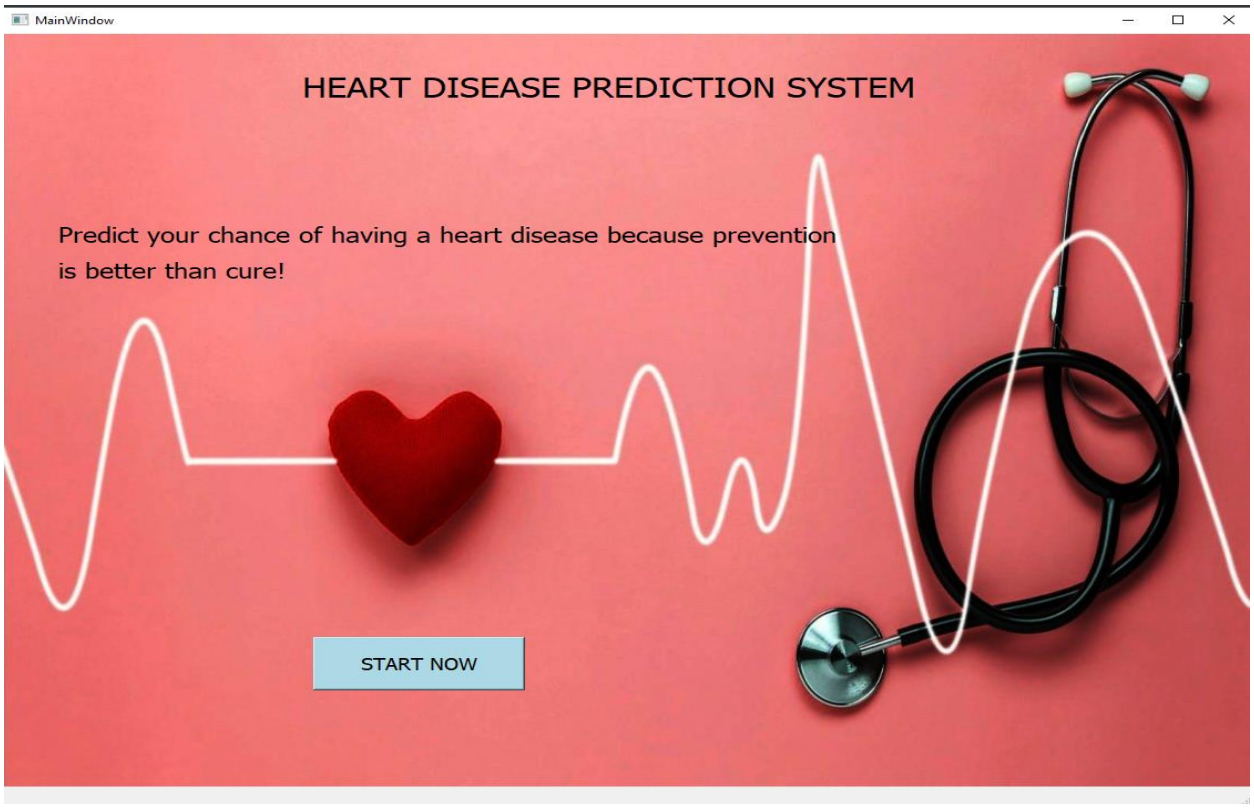


Figure 4. 5 Main Page

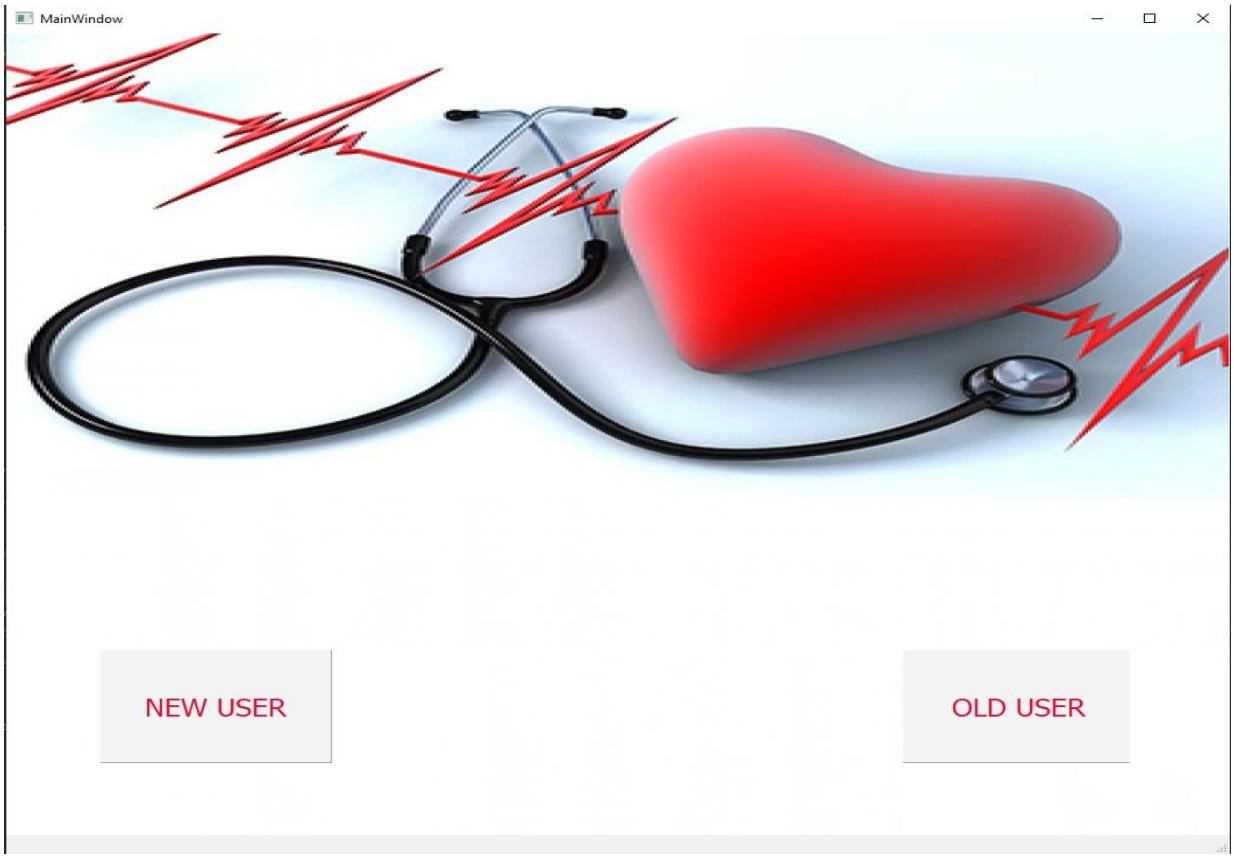


Figure 4. 6 Register Page

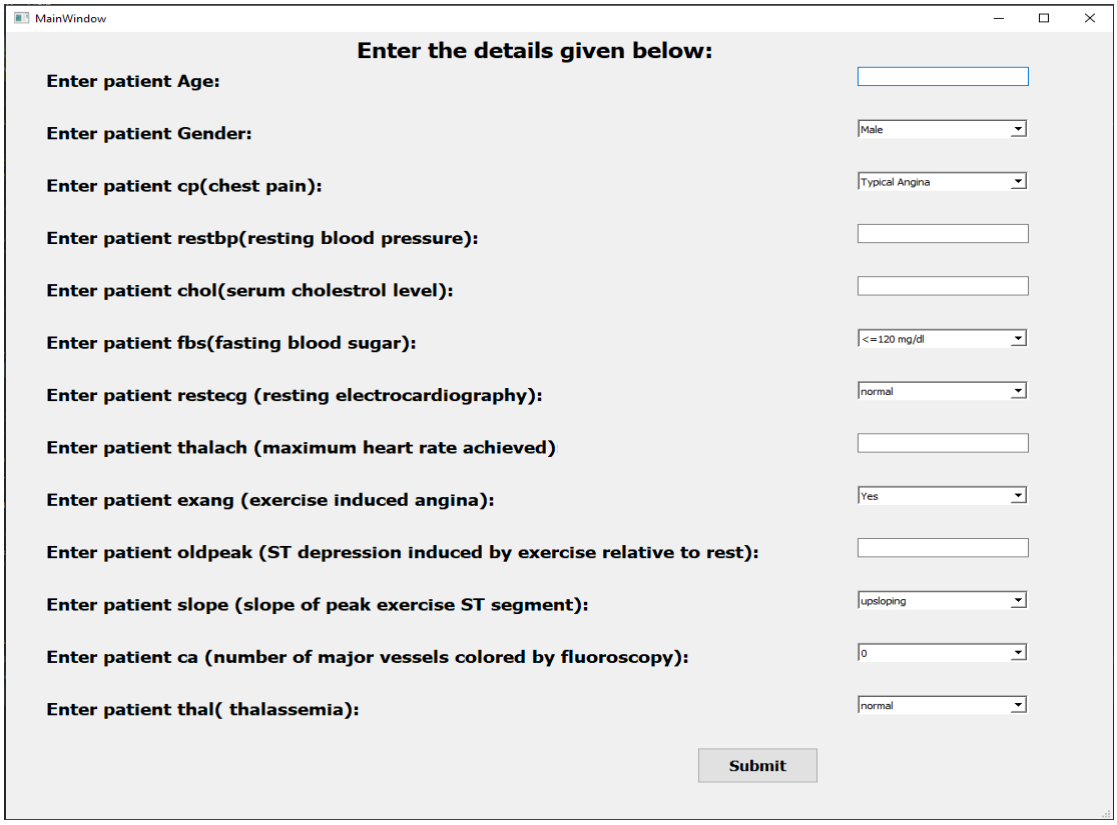


Figure 4. 7 Enter Details

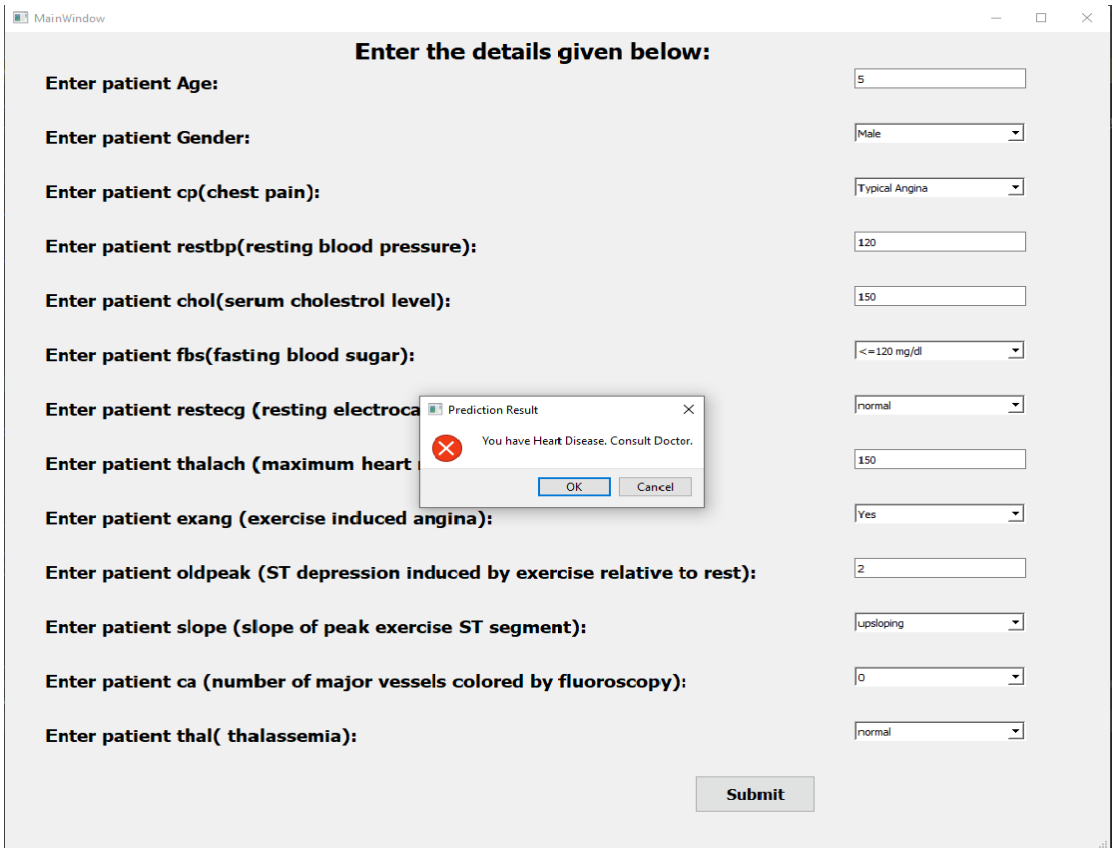


Figure 4. 8 Result Page

4.5.1 REDUCING REDUNDANCY/ CLEANING DATA

After importing the data-set, just to visualize what is present in the raw data, we simply create a heatmap. The heatmap is produced by the correlation matrix of the different diabetes attributes present in the data-set. As expected from a data in raw from, there are certain attributes which do not have any relation with the diabetes (these are represented by a lighter colour on the heatmap).

Since these do not contribute towards our goal and is basically redundant piece of information, we should remove these attributes as they might hamper our accuracy.

CHAPTER 5: CONCLUSION AND FUTURE SCOPE

5.1 CONCLUSION

This project is designed to meet the requirements set by the user for the prediction of employee attrition. Overall during the course of the project I got to know about Machine Learning, Data Science and other important things such as class imbalance while dealing with classification problems. All in all, this project turned out to be a good learning for me.

The project itself is self a good starting point for people as it is dealing with elementary topics regarding machine learning and the accuracy it yields is good enough (80-85%).

5.2 FUTURE SCOPE

The future scope of the project seems good, as data science is a growing field and its implementation and machine learning is a hot topic in each field. The project is self sufficient to the root level but certain changes can be made to make it more users friendly and viable.

Future Recommendations (Development):

- The accuracy of the model can vary with which attributes are taken into consideration. Taking a large no of attributes like done in the project might decrease the accuracy a little bit.
- The GUI could be developed for the same project or a mobile application can work the same, where user/HR fills in the credentials or selects a certain category and receives the result. Although this would be only for a certain organisation but a better interface would help out upcoming users in the future.

REFERENCES

- https://github.com/ishantk
- https://www.tensorflow.org/learn
- https://towardsdatascience.com/
- https://www.kaggle.com/uciml/pima-indians-diabetes-database
- https://www.geeksforgeeks.org/python-programming-language/
- https://www.coursera.org/learn/machine-learning?