A Project Report

on

Visualizing and Predicting Heart Diseases with an Interactive Dashboard

Under the guidance of

Industry Mentor

Mahidhar, Saumya

Faculty Mentor

Mrs. R.Mekala AP/IT

TEAM ID: PNT2022TMID16126

Submitted by,

ARAVINTH S (927619BIT4003)

JAGADEESH S (927619BIT4038)

KARTHIKEYAN P (927619BIT4045)

NACHIAPPAN P L (927619BIT4061)

DEPARTMENT OF INFORMATION TECHNOLOGY

M.KUMARASAMY COLLEGE OF ENGINEERING

Content

VISUALIZING AND PREDICTING HEART DISEASES WITH AN INTERACTIVE DASH BOARD

ABSTRACT

The Heart is one of the most vital structures in the human body. It is the center of the circulatory system. Heart disease is a main life intimidating disease that can origin either death or a severe long term disability. However, there is lack of effective tools to discover hidden relationships and trends in e-health data. Medical diagnosis is a complex task and plays a dynamic role in saving human lives so it needs to be executed accurately and efficiently. A suitable and precise computer based automated decision support system is required to reduce cost for achieving clinical tests. Health analytics have been proposed using ML to predict accurate patient data analysis. The data produced from health care industry is not mined. Data mining techniques can be used to build an intelligent model in medical field using data sets which involves risk factor of patients. The knowledge discovery in database (KDD) is startled with development of approaches and techniques for making use of data. This thesis provides an insight into deep learning and machine learning techniques used in diagnosing diseases. Numerous data mining classifiers have been conversed which has emerged in recent years for efficient and effective disease diagnosis. This thesis proposes a heart attack prediction system using Deep learning techniques, specifically Multi-Layer Perceptron (MLP) to predict the likely possibilities of heart related diseases of the patient. MLP is a very powerful classification algorithm that makes use of Deep Learning approach in Artificial Neural Network. The proposed model incorporates deep learning and data mining to provide the accurate results with minimum errors.

1. INTRODUCTION

1.1 PROJECT OVERVIEW

Deep learning algorithms have been compared, utilised for analysis for many different types of data science applications, and are already available. This research-based effort was primarily driven by the desire to investigate the methodologies for feature selection, data preparation, and processing used in machine learning training models. The problem we currently encounter with first-hand models and libraries is data, where in addition to their abundance and our cooked models, the accuracy we observe during training, testing, and actual validation has a higher variance.

1.2 PURPOSE

The goal of machine learning as a whole is to create an appropriate computer-based system and decision support that can help with the early detection of heart disease, in this project we have created a model that uses logistic regression to predict whether a patient will develop heart disease within ten years based on a variety of features (i.e., potential risk factors that can lead to heart disease). Therefore, the early diagnosis of cardiovascular illnesses might help in high risk patients' decision-making regarding lifestyle adjustments, which would ultimately lessen the problems. This can be a significant advancement in the field of medicine.

2. LITERATURE REVIEW

2.1 EXISTING PROBLEM

2.1.1 TITLE: Two-Stage Classification Model for the Prediction of Heart Disease Using IoMT and Artificial Intelligence, 2022

AUTHOR NAME: S. Manimurugan

The study proposal offers a categorization and prediction model for medical data that makes use of AI and machine learning methods. Datasets and wearable sensors are crucial parts of the suggested paradigm. The suggested approach runs in two stages, as was said. The first step involves classifying sensor data produced by medical sensors attached to a patient's body, and the second stage involves classifying echocardiography pictures. The two techniques of classification are used, and the classification outcomes are verified to forecast heart disease. The creation of a prediction model for heart disease is the primary driving force behind this endeavour. For classification and prediction, the majority of earlier research relied on sensor-based data (medical signals) or medical pictures. In the suggested concept, sensor data and picture data are merged as two phases of input rather than being used individually. Sensor data is utilised for prediction and categorization in the first stage. Based on the importance of the disease or the generated output, if the outcome is unsatisfactory, a second stage using medical image data will be used for precise disease classification and prediction. For the benefit of both patients and doctors, precise predictions can be made by putting this two-stage classification model into practice.

2.1.2 TITLE: Deep Learning in mHealth for Cardiovascular Disease, Diabetes, and Cancer: Systematic Review, 2022

AUTHOR NAME: Andreas Triantafyllidis

This study is to present a systematic review of studies that have used DL based on mHealth data for the diagnosis, prognosis, management, and treatment of major chronic diseases and advance our understanding of the progress made in this rapidly developing field. A search was conducted on the bibliographic databases Scopus and PubMed to identify papers with a focus on the deployment of DL algorithms that used data captured from mobile devices (eg, Smartphone, smart watches, and other wearable devices) targeting CVD, diabetes, or cancer. According to the target disease, the number of participants and their ages, the research period, the DL algorithm employed, the primary DL result, the data set utilised, the characteristics chosen, and the attained performance, the discovered studies were synthesised. The most frequent DL results were patient condition diagnosis for CVD investigations, blood glucose level prediction for diabetic studies, and early cancer detection. Convolutional neural networks and recurrent neural networks were the most common DL algorithms employed in research on CVD, cancer, and diabetes, respectively. Overall, it was determined that DL performed satisfactorily, with >84% accuracy in the majority of investigations. In practically all research that provided such comparative results, DL was shown to produce greater performance compared to standard machine learning methodologies. The explainability of DL results was not extensively discussed in the majority of the research. Utilizing mHealth data, DL can make it easier to diagnose, manage, and cure serious chronic illnesses. In order to show the value of applied DL in practical mHealth tools and therapies, prospective studies are now necessary.

2.1.3 TITLE: Transparency of deep neural networks for medical image analysis: A review of interpretability methods, 2021

AUTHOR NAME: Zohaib Salahuddin

AI technologies for medical imaging are primarily designed to support doctors in their decision-making by merging several variables into a model that produces an actionable result. The value of the model is constrained since it cannot reveal the limits, biases, or reasoning process without an explanation of this output. The interpretability of DL systems not only makes it possible to identify any flawed algorithmic processes but also makes it possible to find additional significant information in the imaging data that could otherwise go unreported. Understanding the DL systems' opaque nature is crucial for developing therapeutic trust and system troubleshooting in addition to being a legal and ethical obligation. Additionally, novel imaging biomarkers can be discovered using interpretability approaches in order to comprehend the particulars of the DL model. Modern medicine relies heavily on medical imaging because it enables the detailed non-invasive viewing of inside bodily structures and metabolic processes. By including potentially useful data in the form of patient-specific illness features, this supports disease diagnosis, treatment planning, and treatment follow-up. Specify how to use interpretability techniques and future approaches for deep neural network interpretability in medical image analysis.

2.1.4 TITLE: Prediction of coronary heart disease based on combined reinforcement multitask progressive time-series networks, 2021

AUTHOR NAME: Wenqi Li

In this paper, a multitask progressive deep networks model is proposed to predict the degree of coronary artery occlusion by heart color Doppler echocardiography report, twenty-one items of blood biochemical tests and other basic body information about the patients(patients' gender, age, diabetes, blood pressure, blood sugar, heart rate, chest pain, city, family coronary heart disease history). The model takes each coronary artery occlusion prediction as a task and eight tasks are executed simultaneously. The model is divided into four parts. The first part is deep reinforcement learning (DRL) pretraining. Asynchronous advantage actor-critic (A3C) method is used to solve the problem efficiently. We use training data to optimize the recurrent neural network that parameterizes the stochastic policy. The second part is a soft parameter sharing layer composed of eight bidirectional LSTM, which ensures the similarity of model parameters by regularizing the distance of model parameters. The third part is a hard parameter sharing layer composed of a shared dropout layer, which reduces the risk of overfitting. And the fourth part is composed of a progressive networks of three dense layers. The progressive neural networks can transfer the prediction model of one blood vessel to the prediction of another blood vessel by storing the migration knowledge and extracting valuable features layer by layer, which can improve the prediction accuracy of the blood vessel with less data. The experimental results show that the model achieves better results than the general machine learning models, and is also superior to the single task model. According to the experimental results, we can draw the conclusions that the combined reinforcement multitask progressive time-series networks do facilitate mutual learning between tasks, thereby improving the overall performance of the model.

2.1.5 TITLE: Innovative feature selection and classification model for heart disease prediction, 2021

AUTHOR NAME: · A. Prathik

Heart disease is a complex disease that affects a large number of people worldwide. The timely and accurate detection of heart disease is critical in healthcare, particularly in the field of cardiology. In this article, we proposed a system for diagnosing heart disease that is both efficient and accurate, and it is based on machine-learning techniques. The diagnosis of heart disease is found to be a serious concern, so the diagnosis has to be done remotely and regularly to take the prior action. In the present world, finding the prevalence of heart disease has become a key research area for the researchers and many models have crown proposed in the recent year. The optimization algorithm plays a vital role in heart disease diagnosis with high accuracy. Important goal of this work is to develop a hybrid GCSA which represents a genetic-based crow search algorithm for feature selection and classification using deep convolution neural networks. From the obtained results, the proposed model GCSA shows increase in the classification accuracy by obtaining more than 94% when compared to the other feature selection methods. The feature selection technique is mainly used in the classification ensemble. While using the feature selection for the DCNN, it provides better results with the optimized features.

2.2 REFERENCES

[1]. S. Manimurugan, Two-Stage Classification Model for the Prediction of Heart Disease Using IoMT and Artificial Intelligence, 2022

[2]. Triantafyllid, Deep Learning in mHealth for Cardiovascular Disease, Diabetes, and Cancer: Systematic Review, 2022

[3]. Zohaib Salahuddin, Transparency of deep neural networks for medical image analysis: A review of interpretability methods, 2021

[4]. Wenqi Li, Prediction of coronary heart disease based on combined reinforcement multitask progressive time-series networks, 2021

[5]. A. Prathik, Innovative feature selection and classification model for heart disease prediction, 2021

2.3 PROBLEM STATEMENT DEFINITION

Present days one of the major application areas of machine learning algorithms is medical diagnosis of diseases and treatment. Machine learning algorithms also used to find correlations and associations between different diseases. Nowadays many people are dying because of sudden heart attack. Prediction and diagnosing of heart disease becomes a challenging factor faced by doctors and hospitals both in India and abroad. In order to reduce number of deaths because of heart diseases, we have to predict whether person is at the risk of heart disease or not in advance. Data mining techniques and machine learning algorithms play a very important role in this area. Many researchers are carrying out their research in this area to develop software that can help doctors to take decision regarding both prediction and diagnosing of heart disease. In this paper we focused on how data mining techniques can be used to predict heart disease in advance such that patient is well treated. An important task of any diagnostic system is the process of attempting to determine and/or identify a possible disease or disorder and the decision reached by this process

3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

3.2 IDEATION & BRAINSTORMING

3.3 PROPOSED SOLUTION

Heart disease diagnosis has become a difficult task in the field of medicine. This diagnosis depends on a thorough and accurate study of the patient's clinical tests data on the health history of an individual. The tremendous improvement in the field of machine learning aim at developing intelligent automated systems which helps the medical practitioners in predicting as well as making decisions about the disease. Such an automated system for medical diagnosis would enhance timely medical care followed by proper subsequent treatment thereby resulting in significant lifesaving. Incorporating the techniques of classification in these intelligent systems achieve at accurate diagnosis. Neural Networks has emerged as an important method of classification. Multi-layer Perceptron Neural Network with Back-propagation has been employed as the training algorithm in this work. This project proposes a diagnostic system for predicting heart disease with improved accuracy. The propagation algorithm has been repeated until minimum error rate was observed. And it is quite evident from the results presented in the previous section that the accuracy rate is maximized.

3.4 PROBLEM SOLUTION FIT

Medical diagnosis and treatment are currently one of the main application areas for machine learning algorithms. In order to discover relationships and interactions between various diseases, machine learning methods are also applied. Heart disease prediction and diagnosis has become a difficult task for medical professionals and facilities, both in India and overseas. We must be able to predict whether a person is at risk for heart disease or not in order to decrease the number of deaths caused by heart diseases. By incorporating classification techniques, these intelligent algorithms produce precise diagnoses. Neural networks have become a significant classification technique. This study uses a multi-layer perceptron neural network with back-propagation as its training technique. In this effort, a more precise diagnostic approach for heart disease prediction is proposed. Up until a minimal error rate was noticed, the propagation method was repeated. The findings shown in the previous section make it abundantly clear that the accuracy rate is maximised

4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

DATASETS ACQUISITION

A data set (or dataset, although this spelling is not present in many contemporary dictionaries like Merriam-Webster) is a collection of data. Most commonly a data set corresponds to the contents of a single database table, or a single statistical data matrix, where every column of the table represents a particular variable, and each row corresponds to a given member of the data set in question. The data set lists values for each of the variables, such as height and weight of an object, for each member of the data set. Each value is known as a datum. The data set may comprise data for one or more members, corresponding to the number of rows. The term data set may also be used more loosely, to refer to the data in a collection of closely related tables, corresponding to a particular experiment or event. In this module, we can upload the cardiovascular datasets related to heart diseases which includes the attributes such as age, gender, height, weight, systolic blood pressure, diastolic blood pressure, cholesterol, glucose, smoke, alcohol, active status, cardio labels.

PREPROCESSING

Data pre-processing is an important step in the [data mining] process. The phrase "garbage in, garbage out" is particularly applicable to data mining and machine learning projects. Data-gathering methods are often loosely controlled, resulting in out-of-range values, impossible data combinations, missing values, etc. Analyzing data that has not been carefully screened for such problems can produce misleading results. Thus, the representation and quality of data is first and foremost before running an analysis. If there is much irrelevant and redundant information present or noisy and unreliable data, then knowledge discovery during the training phase is more difficult. Data preparation and filtering steps can take considerable amount of processing time. In this module, we can eliminate the irrelevant values and also estimate the missing values of data. Finally provide structured datasets.

FEATURES SELECTION

Feature selection refers to the process of reducing the inputs for processing and analysis, or of finding the most meaningful inputs. A related term, feature engineering (or feature extraction), refers to the process of extracting useful information or features from existing data. Filter feature selection methods apply a statistical measure to assign a scoring to each feature. The features are ranked by the score and either selected to be kept or removed from the dataset. The methods are often uni-variate and consider the feature independently, or with regard to the dependent variable. It can be used to construct the multiple heart diseases. In this module, select the multiple features from uploaded datasets. And train the datasets with various disease labels such as Coronary heart diseases, Cardiac arrest, High blood pressure, Arrhythmia and normal.

CLASSIFICATION

In this module implement classification algorithm to predict the heart diseases. And using deep learning algorithm such as Multi-layer perceptron algorithm to predict the diseases. A multilayer perceptron (MLP) is a feed forward artificial neural network model that maps sets of input data onto a set of appropriate outputs. It (MLP) consists of multiple layers of nodes in a directed graph, and each layer is fully connected to the next one. Each node is a neuron with a nonlinear activation function except for the input nodes. MLP utilizes a supervised learning technique called back propagation for training the network. MLP is a modified form of the standard linear perceptron and can distinguish data that are not linearly separable. If a multilayer perceptron (MLP) has a simple on-off mechanism i.e. linear activation function in all neurons ,to determine whether or not a neuron fires, then it is easily proved with linear algebra that any number of layers can be reduced to the standard two-layer input-output model. The gradient techniques are then applied to the optimization methods to adjust the weights to minimize the loss function in the network. Hence, the algorithm requires a known and a desired output for all inputs in order to compute the gradient of loss function. Usually, the generalization of MultiLayerd Feed Forward Networks is done using delta rule which possibly makes a chain of iterative rules to compute gradients for each layer. Back Propagation Algorithm necessitates the activation function to be different between the neurons. The ongoing researches on parallel, distributed computing and computational neuroscience are currently implemented with the concepts of MultiLayer Perceptron using a Back Propagation Algorithm. MLP Back Propagation Algorithm has also gained focus in pattern recognition domain. They are so convenient in research, because of their ability in solving complex problems, and also for

their fitness approximation results even with critical predictions. MLP is one of the Neural Network models, has the same architecture of Feed-Forward back Propagation for Supervised training. The multilayer perceptron is the most known and most frequently used type of neural network. User can provide the features and automatically predict the diseases.

DISEASE DIAGNOSIS

Medical decision support system is a decision-support program which is designed to assist physicians and other health professionals with decision making tasks, such as determining diagnosis of patients' data. In this module, provide the diagnosis information based on predicted heart diseases. Proposed system provides improved accuracy in heart disease prediction. Risk factors are conditions or habits that make a person more likely to develop a disease.

4.2 NON FUNCTIONAL REQUIREMENTS

Usability

The system shall allow the users to access the system with pc using web application. The system uses a web application as an interface. The system is user friendly which makes the system easy

Availability

The system is available 100% for the user and is used 24 hrs a day and 365 days a year. The system shall be operational 24 hours a day and 7 days a week.

Scalability

Scalability is the measure of a system's ability to increase or decrease in performance and cost in response to changes in application and system processing demands.

Security

A security requirement is a statement of needed security functionality that ensures one of many different security properties of software is being satisfied.

Performance

The information is refreshed depending upon whether some updates have occurred or not in the application. The system shall respond to the member in not less than two seconds from the time of the request submittal. The system shall be allowed to take more time when doing large processing jobs. Responses to view information shall take no longer than 5 seconds to appear on the screen.

Reliability

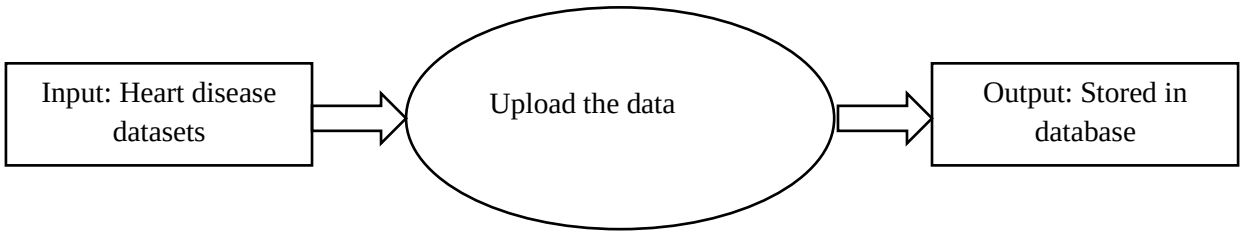
The system has to be 100% reliable due to the importance of data and the damages that can be caused by incorrect or incomplete data. The system will run 7 days a week. 24 hours a day.

5. PROJECT DESIGN

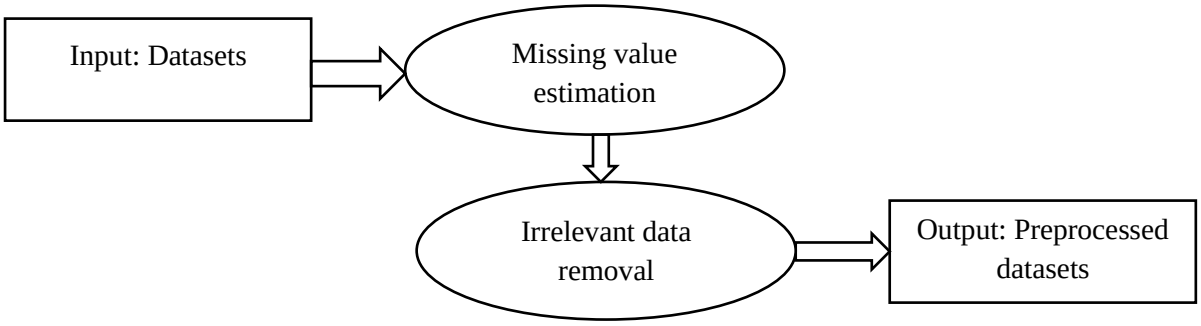
5.1 DATA FLOW DIAGRAMS

A data flow diagram is a two-dimensional diagram that explains how data is processed and transferred in a system. The graphical depiction identifies each source of data and how it interacts with other data sources to reach a common output. Individuals seeking to draft a data flow diagram must identify external inputs and outputs, determine how the inputs and outputs relate to each other, and explain with graphics how these connections relate and what they result in.

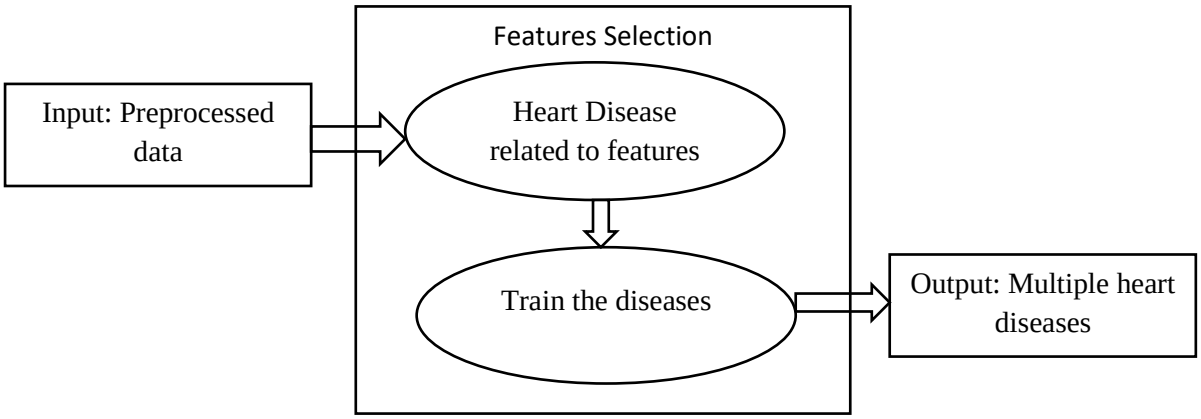
LEVEL 0



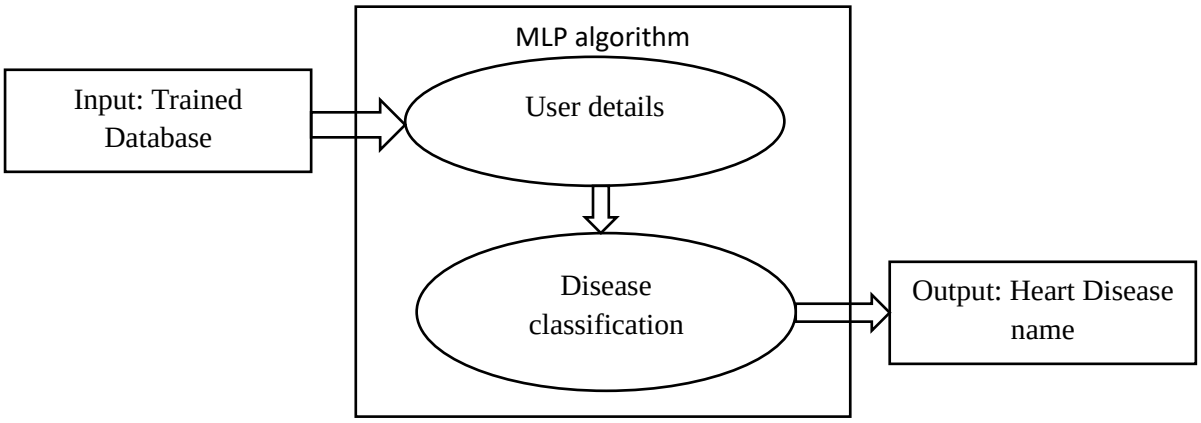
LEVEL 1



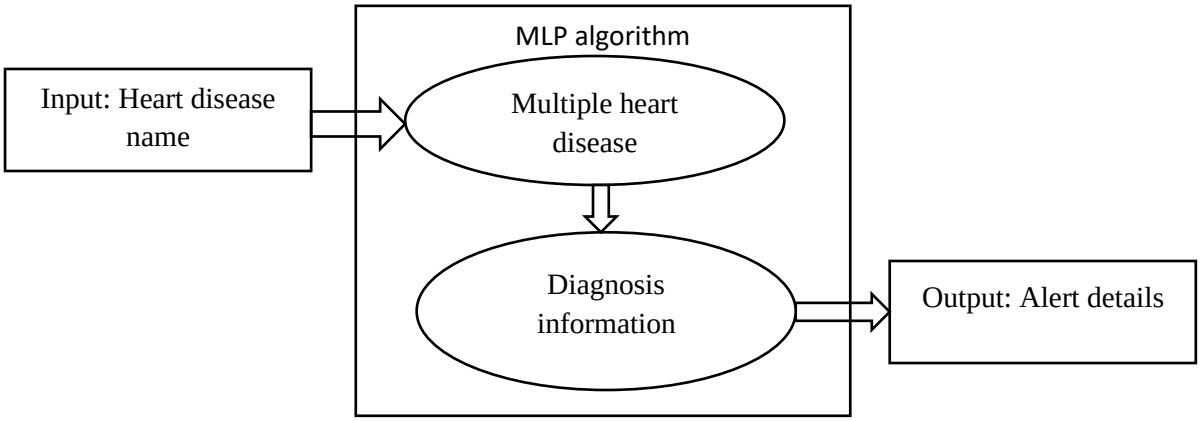
LEVEL 2



LEVEL 3

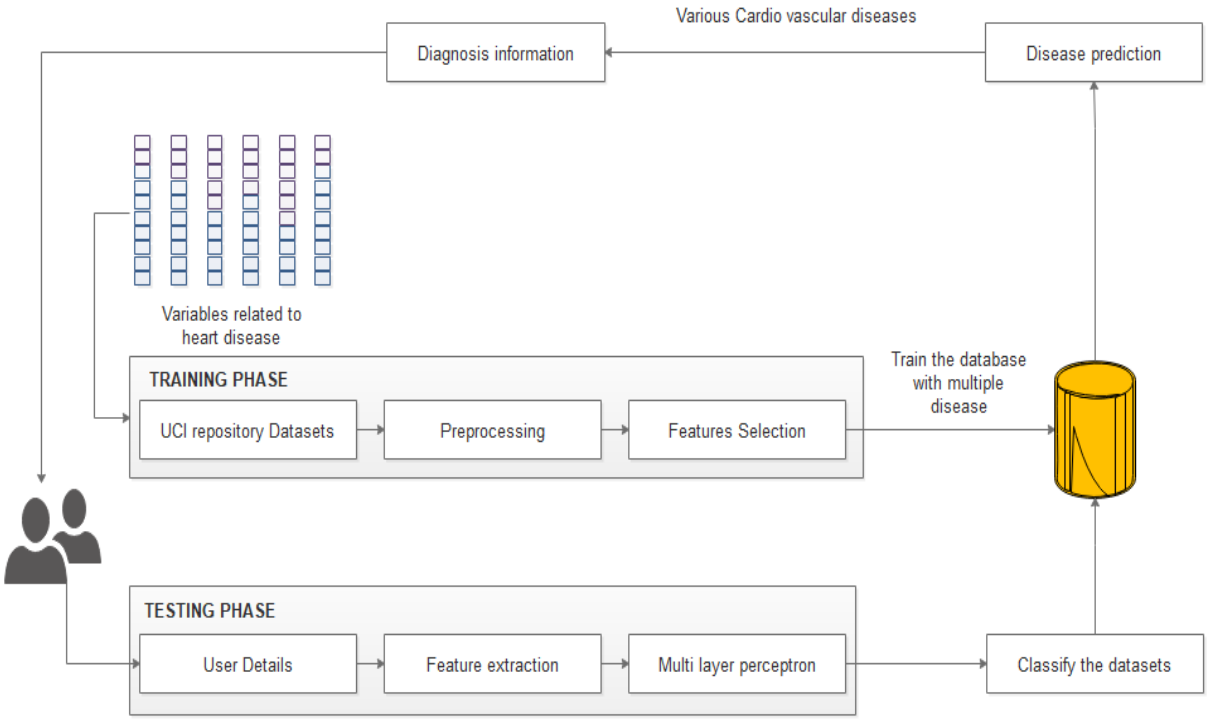


LEVEL 4



5.2 SOLUTION & TECHNICAL ARCHITECTURE

A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.



5.3 USER STORIES

6 PROJECT PLANNING & SCHEDULING

6.3 SPRINT PLANNING & ESTIMATION

6.4 SPRINT DELIVERY SCHEDULE

6.5 REPORTS FROM JIRA

7 CODING & SOLUTION

7.3 FEATURE 1

7.4 FEATURE 2

7.5 DATABASE SCHEMA

8 TESTING

8.3 TEST CASES

A test case has components that describe input, action and an expected response, in order to determine if a feature of an application is working correctly. A test case is a set of instructions on “HOW” to validate a particular test objective/target, which when followed will tell us if the expected behavior of the system is satisfied or not.

Characteristics of a good test case:

- Accurate: Exacts the purpose.
- Economical: No unnecessary steps or words.
- Traceable: Capable of being traced to requirements.
- Repeatable: Can be used to perform the test over and over.
- Reusable: Can be reused if necessary.

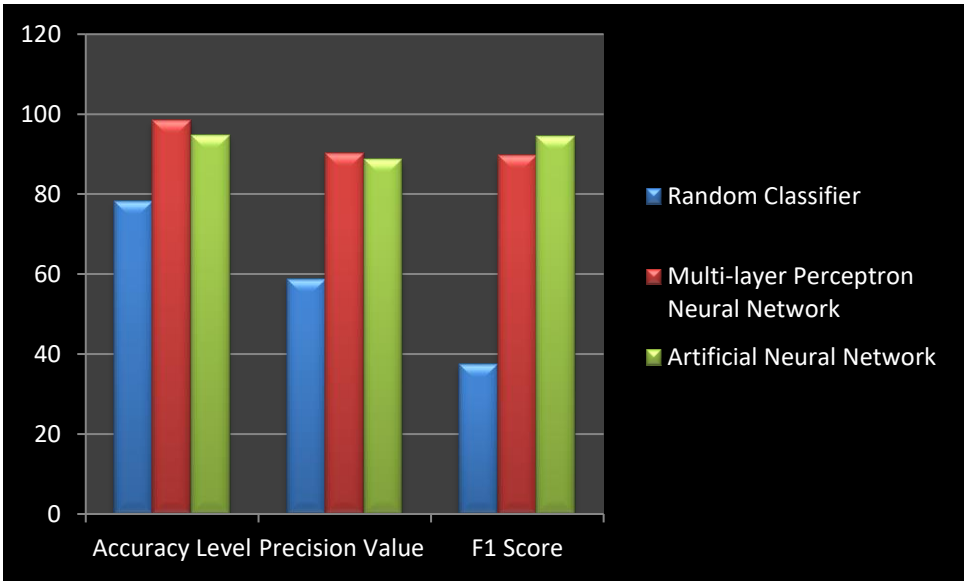
S.NO	FUNCTION	DESCRIPTION	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
1	Framework construction	Generate the GUI for admin and user	Individual page for admin and user	Individual page for admin and user	Success
2	Data preprocessing	Analysis missing values.	Comments in text format	Comments in text format	Success
3	Classification	Classify the datasets	Solving complex problems	Predicting the disease	Success

8.4 USER ACCEPTANCE TESTING

Acceptance testing can be defined in many ways, but a simple definition is the succeeds when the software functions in a manner that can be reasonable expected by the customer. After the acceptance test has been conducted, one of the two possible conditions exists. This is to fine whether the inputs are accepted by the database or other validations. For example accept only numbers in the numeric field, date format data in the date field. Also the null check for the not null fields. If any error occurs then show the error messages. The function of performance characteristics to specification and is accepted. A deviation from specification is uncovered and a deficiency list is created. User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

9 RESULTS

9.3 PERFORMANCE METRICS



10 ADVANTAGES & DISADVANTAGES

DISADVANTAGES

- Labeled data based disease classification
- Provide high number of false positive
- Binary classification can be occurred
- Computational complexity

ADVANTAGES

- Accuracy is high
- Parallel processing
- Multiple heart diseases are predicted
- Reduce number of false positive rate

11 CONCLUSION

In this project the problem of constraining and summarizing different algorithms of data mining used in the field of medical prediction are discussed. The focus is on using different algorithms and combinations of several target attributes for intelligent and effective heart disease prediction using data mining. Data mining technology provides an important means for extracting valuable medical rules hidden in medical data and acts as an important role in disease prediction and clinical diagnosis. There is an increasing interest in using classification to identify disease which is present or not. In the current study, have demonstrated, using a large sample of patients hospitalized with classification. Classification algorithm is very sensitive to noisy data. If any noisy data is present then it causes very serious problems regarding to the processing power of classification. It not only slows down the task of classification algorithm but also degrades its performance. Hence, before applying classification algorithm it must be necessary to remove all those attributes from datasets who later on acts as noisy attributes. In this research work, we can implement preprocessing steps and implemented the classification rule algorithms namely Multi-layer perceptron are used for classifying datasets which are uploaded by user. By analyzing the experimental results it is observed that the Multi-layer perceptron technique has yields better result than other techniques.

12 FUTURE SCOPE

In future we tend to improve efficiency of performance by applying other data mining techniques and algorithms.

13 APPENDIX

SOURCE CODE

```
from flask import Flask, render_template, flash, request, session

from flask import render_template, redirect, url_for, request

#from wtforms import Form, TextField, TextAreaField, validators, StringField, SubmitField

from werkzeug.utils import secure_filename


import mysql.connector

import smtplib

#from PIL import Image

import pickle



import numpy as np









app = Flask(__name__)

app.config.from_object(__name__)

app.config['SECRET_KEY'] = '7d441f27d441f27567d441f2b6176a'


app.config['DEBUG']
```

```
@app.route("/")
def homepage():
    return render_template('index.html')
@app.route("/Home")
def Home():
    return render_template('index.html')
@app.route("/AdminLogin")
def AdminLogin():
    return render_template('AdminLogin.html')
@app.route("/NewUser")
def NewUser():
    return render_template('NewUser.html')
@app.route("/UserLogin")
def UserLogin():
    return render_template('UserLogin.html')
@app.route("/UserHome")
def UserHome():
    return render_template('UserHome.html')
@app.route("/AdminHome")
def AdminHome():
    return render_template('AdminHome.html')
```

```
@app.route("/NewQuery1")
def NewQuery1():
    return render_template('NewQueryReg.html')


@app.route("/UploadDataset")
def UploadDataset():
    return render_template('ViewExcel.html')




@app.route("/adminlogin", methods=['GET', 'POST'])
def adminlogin():
    error = None
    if request.method == 'POST':
        if request.form['uname'] == 'admin' or request.form['password'] == 'admin':
            conn = mysql.connector.connect(user='root', password='', host='localhost', database='1heartdb')
            cursor = conn.cursor()
            cur = conn.cursor()
            cur.execute("SELECT * FROM register")
            data = cur.fetchall()
            return render_template('AdminHome.html', data=data)
```

```
        else:
            return render_template('index.html', error=error)


@app.route("/reg", methods=['GET', 'POST'])
def reg():
    if request.method == 'POST':
        n = request.form['name']


        address = request.form['address']
        age = request.form['age']
        pnumber = request.form['phone']
        email = request.form['email']
        zip = request.form['zip']
        uname = request.form['uname']
        password = request.form['psw']
        conn = mysql.connector.connect(user='root', password='', host='localhost', database='1heartdb')
        cursor = conn.cursor()
        cursor.execute(
            "INSERT INTO register VALUES ('" + n + "','" + age + "','" + email + "','" + pnumber + "','" + zip + "','" + address + "','" + uname + "','" + password + "')")
        conn.commit()
        conn.close()
```

```
        # return 'file register successfully'
        return render_template('UserLogin.html')


@app.route("/userlogin", methods=['GET', 'POST'])
def userlogin():
    error = None
    if request.method == 'POST':
        username = request.form['uname']
        password = request.form['password']
        session['uname'] = request.form['uname']


        conn = mysql.connector.connect(user='root', password='', host='localhost', database='1heartdb')
        cursor = conn.cursor()
        cursor.execute("SELECT * from register where uname='" + username + "' and psw='" + password + "'")
        data = cursor.fetchone()
        if data is None:
            return render_template('index.html')
            return 'Username or Password is wrong'
        else:
            print(data[0])
            session['uid'] = data[0]
            conn = mysql.connector.connect(user='root', password='', host='localhost', database='1heartdb')
```

```
        # cursor = conn.cursor()
        cur = conn.cursor()
        cur.execute("SELECT * FROM register where uname='" + username + "' and psw='" + password + "'")
        data = cur.fetchall()

        return render_template('UserHome.html', data=data )

@app.route("/newquery", methods=['GET', 'POST'])
def newquery():
    if request.method == 'POST':
        uname = session['uname']

        age = request.form['age']
        gender = request.form['gender']
        height = request.form['height']
        weight = request.form['weight']
        aphi = request.form['aphi']
        aplo = request.form['aplo']
        choles = request.form['choles']
        glucose = request.form['glucose']
        smoke = request.form['smoke']
        alcohol = request.form['alcohol']
```

```
    conn = mysql.connector.connect(user='root', password='', host='localhost', database='1heartdb')
    cursor = conn.cursor()
    cursor.execute(
      "INSERT INTO Querytb VALUES ('" + uname + "','" + age + "','" + gender + "','" + height + "','"+weight+"','"+aphi +"','"+ aplo
      +"','"+ choles +"','"+ glucose +"','"+ smoke +"','"+ alcohol +"','waiting','')")
    conn.commit()
    conn.close()
    # return 'file register successfully'
    conn = mysql.connector.connect(user='root', password='', host='localhost', database='1heartdb')
    # cursor = conn.cursor()
    cur = conn.cursor()
    cur.execute("SELECT * FROM Querytb where UserName='" + uname + "' and DResult='waiting '")
    data = cur.fetchall()
    return render_template('UserQueryInfo.html', data=data)


@app.route("/UQueryandAns")
def UQueryandAns():


   uname = session['uname']
```

```
    conn = mysql.connector.connect(user='root', password='', host='localhost', database='1heartdb')
    # cursor = conn.cursor()
    cur = conn.cursor()
    cur.execute("SELECT * FROM Querytb where UserName='" + uname + "' and DResult='waiting'")
    data = cur.fetchall()

    conn = mysql.connector.connect(user='root', password='', host='localhost', database='1heartdb')
    # cursor = conn.cursor()
    cur = conn.cursor()
    cur.execute("SELECT * FROM Querytb where UserName='" + uname + "' and DResult !='waiting'")
    data1 = cur.fetchall()


    return render_template('UserQueryAnswerinfo.html', wait=data, answ=data1 )

@app.route("/AdminQinfo")
def AdminQinfo():

    #uname = session['uname']
```

```
    conn = mysql.connector.connect(user='root', password='', host='localhost', database='1heartdb')
    # cursor = conn.cursor()
    cur = conn.cursor()
    cur.execute("SELECT * FROM Querytb where  DResult='waiting'")
    data = cur.fetchall()


    return render_template('AdminQueryInfo.html', data=data )


@app.route("/answer")
def answer():

    Answer = ''
    Prescription=''
    id =  request.args.get('lid')


    conn = mysql.connector.connect(user='root', password='', host='localhost', database='1heartdb')
    cursor = conn.cursor()
    cursor.execute("SELECT  *  FROM Querytb where  id='" + id + "'")
    data = cursor.fetchone()
```

```
if data:
    UserName = data[1]
    age = data[2]
    gender = data[3]
    height = data[4]
    weight = data[5]
    aphi = data[6]
    aplo = data[7]
    choles = data[8]
    glucose = data[9]
    smoke = data[10]
    alcohol = data[11]


else:
    return 'Incorrect username / password !'


age = float(age)
gender = float(gender)
height = float(height)
weight = float(weight)
aphi = float(aphi)
aplo = float(aplo)
choles = float(choles)
```

```
glucose = float(glucose)
smoke = float(smoke)
alcohol = float(alcohol)

filename = 'heart-prediction-rfc-model.pkl'
classifier = pickle.load(open(filename, 'rb'))

data = np.array([[age, gender, height, weight, aphi, aplo, choles, glucose, smoke, alcohol]])
my_prediction = classifier.predict(data)



if my_prediction == 1:
    #Answer = 'Heart'

    if(int(aphi) > 190 ):
        Answer = "Coronary Heart Disease";
        Prescription="Angiotensin-converting enzyme (ACE) inhibitors "

    elif(int(aphi) > 180 ):
        Answer = "Cardiac Arrest";

        Prescription="Coronary bypass surgery ";

    elif(int(aphi) > 170 ):
```

```
        Answer = "High Blood Pressure";
        Prescription= "Beta-blockers ";



    else:
        Answer = "Arrhythmia";
        Prescription="Procainamide (Procan, Procanbid) ";









    msg = 'Calculations, You have  '   +str(answer) +'Prescription : '+ str(Prescription)
    print('Hello:According to our Calculations, You have  Heart Disease')


else:
    Answer = 'You DON T have  Heart Disease'
    msg = 'Congratulations!!  You DON T have  Heart Disease'
    print('Congratulations!! You DON T have  Heart Disease')
    Prescription='Nill'


conn = mysql.connector.connect(user='root', password='', host='localhost',
database='1heartdb')
```

```
    cursor = conn.cursor()
    cursor.execute(
      "update Querytb set DResult='"+Answer+"', Prescription='" + Prescription +"'  where id='" + str(id) + "' ")
    conn.commit()
    conn.close()

    conn3 = mysql.connector.connect(user='root', password='', host='localhost', database='1heartdb')
    cur3 = conn3.cursor()
    cur3.execute("SELECT * FROM register where     uname='" + str(UserName) + "'")
    data3 = cur3.fetchone()
    if data3:
       phnumber = data3[4]
       print(phnumber)
       sendmsg(phnumber, msg)

    # return 'file register successfully'
    conn = mysql.connector.connect(user='root', password='', host='localhost', database='1heartdb')
    # cursor = conn.cursor()
    cur = conn.cursor()
    cur.execute("SELECT * FROM Querytb where  DResult !='waiting '")
    data = cur.fetchall()
    return render_template('AdminAnswer.html', data=data)
```

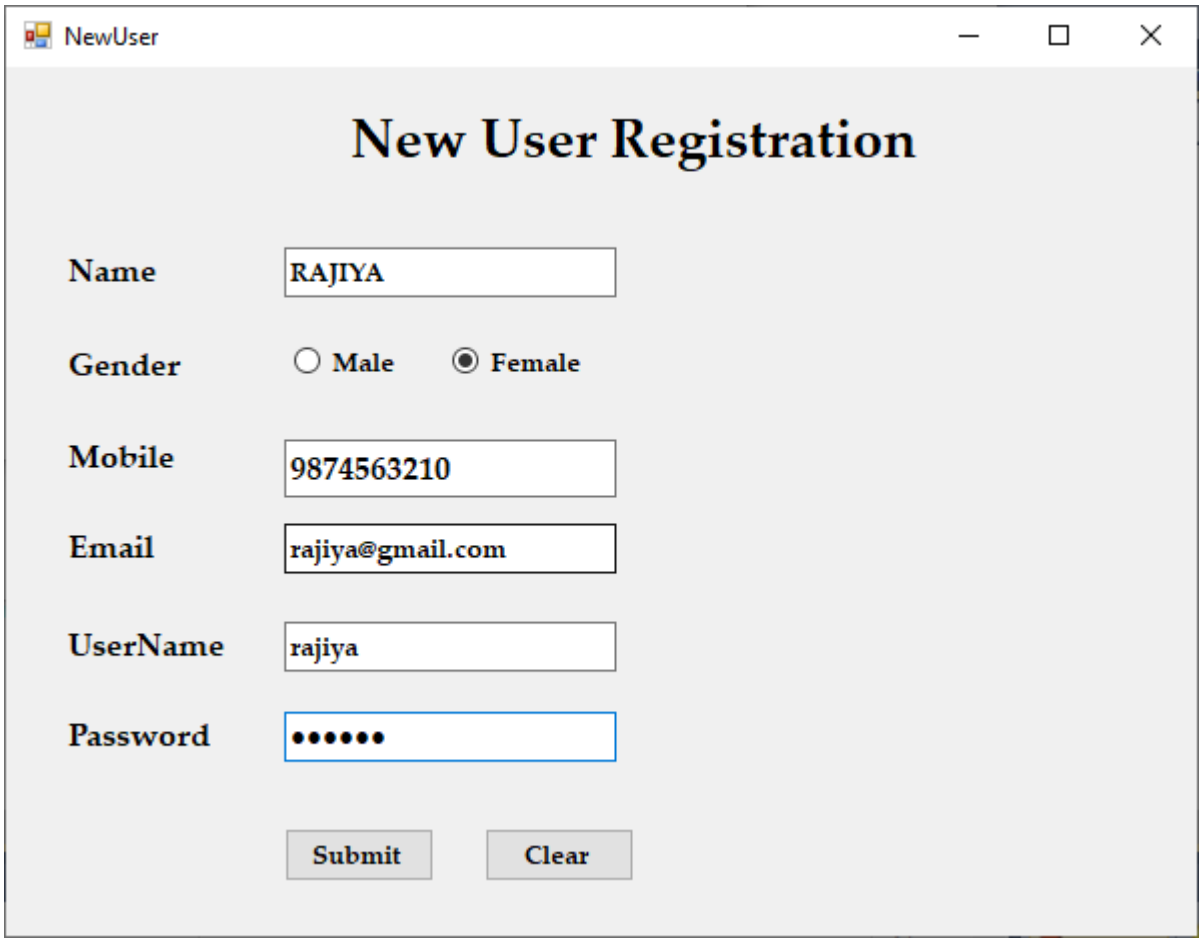
```
@app.route("/AdminAinfo")
def AdminAinfo():



    conn = mysql.connector.connect(user='root', password='', host='localhost', database='1heartdb')
    # cursor = conn.cursor()
    cur = conn.cursor()
    cur.execute("SELECT * FROM Querytb where  DResult !='waiting'")
    data = cur.fetchall()



    return render_template('AdminAnswer.html', data=data )



def sendmsg(targetno,message ):
    import requests
```

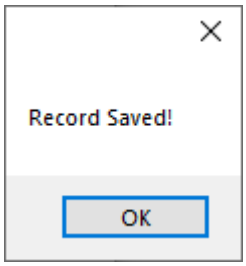
```
requests.post("http://smsserver9.creativepoint.in/api.php?username=fantasy&password=596692&to=" + targetno + "&from=FSSMSS&message=Dear user  your msg is " + message + " Sent By FSMSG FSSMSS&PEID=1501563800000030506&templateid=1507162882948811640")


@app.route("/excelpost", methods=['GET', 'POST'])
def uploadassign():
    if request.method == 'POST':



        file = request.files['fileupload']
        file_extension = file.filename.split('.')[1]
        print(file_extension)
        #file.save("static/upload/" + secure_filename(file.filename))

        import pandas as pd
        import matplotlib.pyplot as plt
        df = ''
        if file_extension == 'xlsx':
            df = pd.read_excel(file.read(), engine='openpyxl')
        elif file_extension == 'xls':
            df = pd.read_excel(file.read())
        elif file_extension == 'csv':
            df = pd.read_csv(file)
```

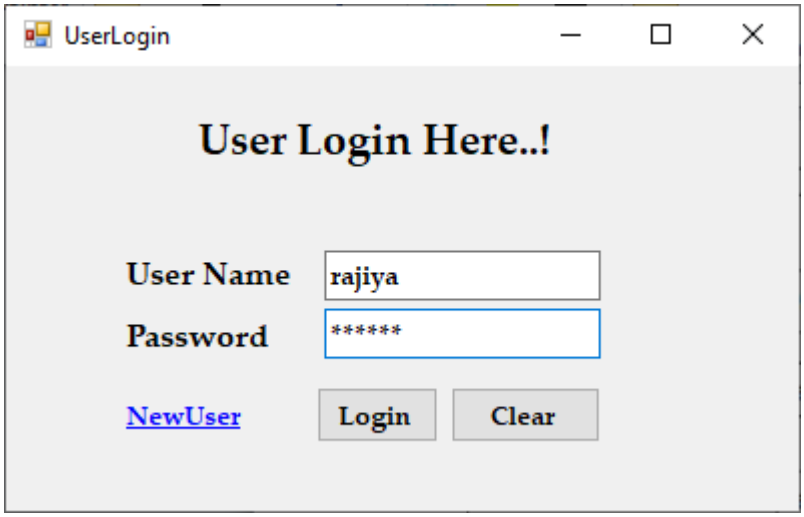
```
print(df)


import seaborn as sns
sns.countplot(df['active'], label="Count")
plt.savefig('static/images/out.jpg')
iimg = 'static/images/out.jpg'


#plt.show()
#df = pd.read_csv("./Heart/Heartnew.csv")
def clean_dataset(df):
    assert isinstance(df, pd.DataFrame), "df needs to be a pd.DataFrame"
    df.dropna(inplace=True)
    indices_to_keep = ~df.isin([np.nan, np.inf, -np.inf]).any(1)
    return df[indices_to_keep].astype(np.float64)
df = clean_dataset(df)
print("Preprocessing Completed")
print(df)




# import pandas as pd
import matplotlib.pyplot as plt
```

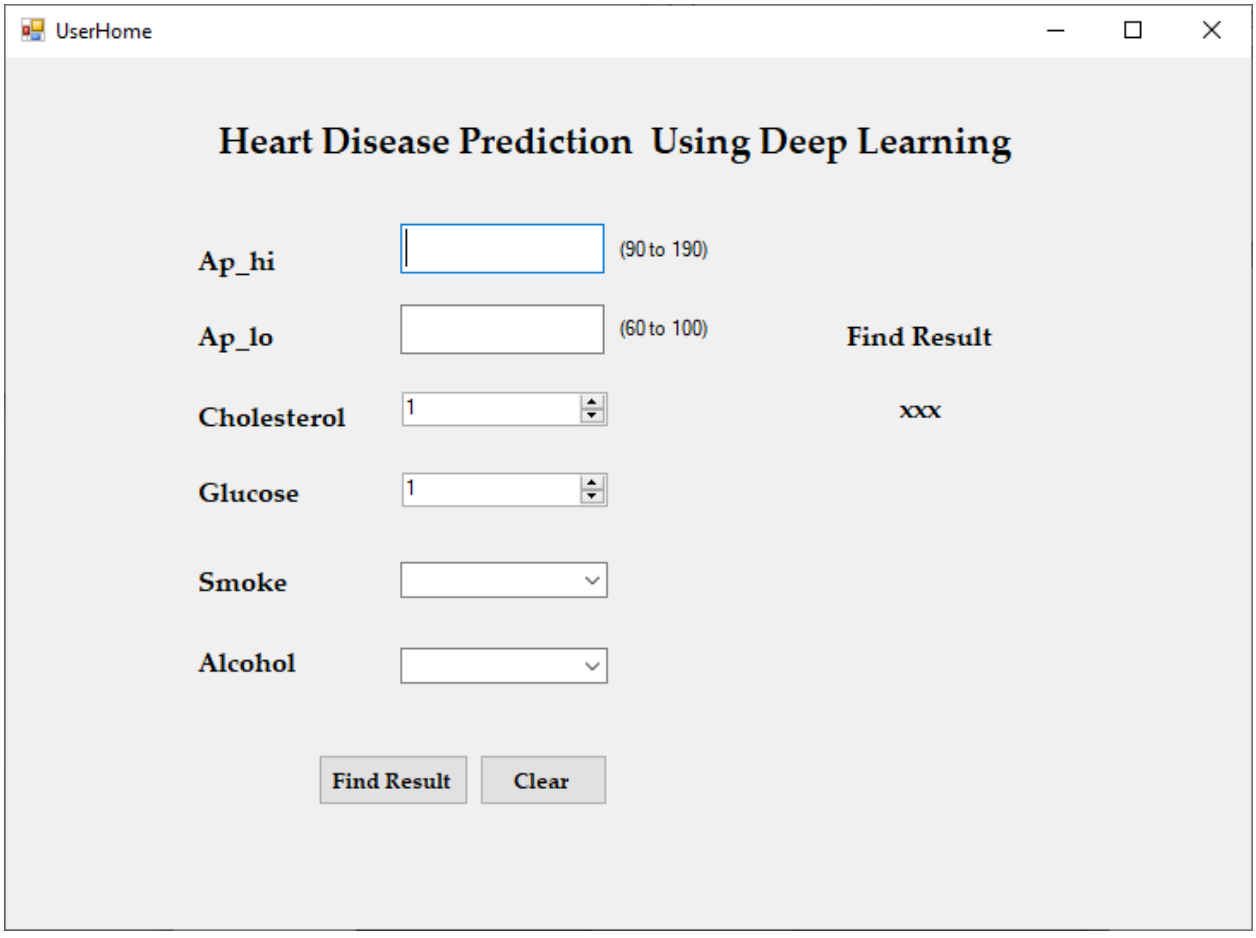
```
# read-in data
# data = pd.read_csv('./test.csv', sep='\t') #adjust sep to your needs
print("Feature Selection")

import seaborn as sns
sns.countplot(df['active'], label="Count")
plt.show()

# count occurences
# occurrences = df.loc[:, 'Outcome'].value_counts()

# plot histogram
# plt.bar(occurrences.keys(), occurrences)
# plt.show()

# Replacing the 0 values from ['Glucose','BloodPressure','SkinThickness','Insulin','BMI'] by NaN
df_copy = df.copy(deep=True)
df_copy[['gender', 'height', 'weight', 'ap_hi', 'ap_lo', 'cholesterol', 'gluc', 'smoke', 'alco']] = df_copy[
    ['gender', 'height', 'weight', 'ap_hi', 'ap_lo', 'cholesterol', 'gluc', 'smoke', 'alco']].replace(0, np.NaN)
```

```
        # Model Building
        from sklearn.model_selection import train_test_split
        df.drop(df.columns[np.isnan(df).any()], axis=1)
        X = df.drop(columns='active')
        y = df['active']
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=0)

        from sklearn.neural_network import MLPClassifier
        classifier = MLPClassifier(random_state=42)
        classifier.fit(X_train, y_train)

        # Creating a pickle file for the classifier
        filename = 'heart-prediction-rfc-model.pkl'
        pickle.dump(classifier, open(filename, 'wb'))
        print("Training process is complete Model File Saved!")




        return render_template('ViewExcel.html', data=df.to_html(), dataimg=iimg)

if __name__ == '__main__':
    app.run(debug=True, use_reloader=True)
```


NewUser
New User Registration
Name
RAJIYA
Gender
Male
Female
Mobile
9874563210
Email
rajiya@gmail.com
UserName
rajiya
Password
Submit
Clear


Record Saved!
OK


UserLogin
User Login Here..!
User Name
rajiya
Password
******
NewUser
Login
Clear


UserHome
Heart Disease Prediction Using Deep Learning
Ap_hi
(90 to 190)
Ap_lo
(60 to 100)
Find Result
Cholesterol
1
xxx
Glucose
1
Smoke
Alcohol
Find Result
Clear

GITHUB & PROJECT DEMO LINK

IBM-EPBL/IBM-Project-19147-1659693815: Visualizing and Predicting Heart Diseases with an Interactive Dash Board (github.com)