Visualizing and predicting Heart Disease with an Interactive Dash Board

IBM-Project - 40392-1660628902

Nalaiya Thiran Project based learning on professional readlines for innovation, employment and entrepreneurship.

A Project Reported By

Team ID: PNT2022TMID41082

Nishanthini. R	612719104047
Ramya. G	612719104050
Savitha. C	612719104060
Shanthini.T	612719104061

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

THE KAVERY ENGINEERING COLLEGE, SALEM

(ANNA UNIVERSITY)

BONAFIDE CERTIFICATE

Certified that this project report "**Visualizing and predicting Heart Disease with an Interactive Dash Board"** is the bonafide record work done by **Ms Nishanthini R** (612719104047), **Ms Ramya G** (612719104050) , **Ms Savitha C** (612719104060), **Ms Shanthini T** (6127104061) for **IBM- NALAIYATHIRAN** in **VII** semester of **B.E.,** degree course **in Computer Science and Engineering** branch during the academic year of 2022 - 2023.

Staff-In charge
Ms SUDHA G

Evaluator
Ms VASUKI S

Head of the Department

Mr. M. BALAMURUGAN

ACKNOWLEDGEMENT

We are highly grateful to thank our Project coordinator MS SUDHA G and our Project Evaluator MS VASUKI S Department of Computer Science and Engineering, The Kavery Engineering College, Salem for the coordinating us throughout this Project.

We are very much indebted to thank all the faculty members of Department of Computer Science and Engineering in our Institute, for their excellent moral support and suggestions to complete our Project work successfully.

Nishanthini R (612719104047)
Ramya G (612719104050)
Savitha C (612719104060)
Shanthini T (612719104061)

Table of Content

1.Introduction:

1.1 Project overview

The proposed work predicts the chances of Heart Disease and classifies patient's risk level by implementing different data mining techniques such as Naive Bayes, Decision Tree, Logistic Regression and Random Forest. Patients' ECG samples must be obtained, and the extracted characteristics properties. Here, we provide a neural network-based method for a method that can identify irregular heartbeats with the objective of reduce the number of features needed for analysis. The cause of mortality from heart-related illnesses has been seen to be increasing quickly in the current situation. The study of ECG is crucial for the diagnosis of disorders with a link to the heart. So, one can forecast the normalcy and abnormality existing in the ECG waveform by using the segmentation approach.

The new strategy, which is based on deep learning and specifically on a CNN network, ensured great performance in the identification and, consequently, prevention of cardiovascular illnesses. This study introduces a robust classification technique as a consequence, and we'll discuss how to utilize Python to predict heart disease. In this study, a Heart Disease Prediction System(HDPS) is developed using Naives Bayes and Decision Tree algorithms 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 HDPS predicts the likelihood of patients getting heart disease. It enables significant knowledge. E.g. Relationships between medical factors related to heart disease and patterns, to be established. We have employed the multilayer perception neural network with back propagation as the training algorithm. The obtained results have illustrated that the designed diagnostic system can effectively predict the risk level of heart diseases

1.2. Purpose

Heart disease can be managed effectively with a combination of lifestyle changes, medicine and, in some cases, surgery. With the right treatment, the symptoms of heart disease can be reduced and the functioning of the heart improved. The predicted results can be used to prevent and thus reduce cost for surgical treatment and other expensive. The overall objective of my work will be to predict accurately with few tests and attributes the presence of heart disease. Attributes considered form the primary basis for tests and give accurate results more or less. Many more input attributes can be taken but our goal is to predict with few attributes and faster efficiency the risk of having heart disease. Decisions are often made based on doctors' intuition and experience rather than on the knowledge rich data hidden the data set and databases. This practice leads to unwanted biases, errors and excessive medical costs which affects the quality of service provided to patients. The main objective of this research is to develop a heart prediction system. The system can discover and extract hidden knowledge associated with diseases from a historical heart dataset. Heart disease prediction system aims to exploit machine learning techniques on medical dataset to assist in the prediction of the heart diseases. The major challenge in heart disease is its detection. There are instruments available which can predict heart disease but either it are expensive or are not efficient to calculate chance of heart disease in human. Early detection of cardiac diseases can decrease the mortality rate and overall complications. However, it is not possible to monitor patients everyday in all cases accurately and consultation of a patient for 24 hours by a doctor is not available since it requires more sapience, time and expertise. Since we have a good amount of data in today's world, we can use various machine learning algorithms to analyze the data for hidden patterns. The hidden patterns can be used for health diagnosis in medicinal data.

2. Literature Survey:

2.1 Existing Problem

Heart disease is even being highlighted as a silent killer which leads to the death of a person without obvious symptoms. The nature of the disease is the cause of growing anxiety about the disease & its consequences. Hence continued efforts are being done to predict the possibility of this deadly disease in prior. So that various tools & techniques are regularly being experimented with to suit the present-day healthneeds. Machine Learning techniques can be a boon in this regard. Even though heart disease can occur in different forms, there is a common set of core risk factors that influence whether someone will ultimately be at risk for heart disease or not. By collecting the data from various sources, classifying them under suitable headings & finally analysing to extract the desired data we can conclude. This technique can be very well adapted to the do the prediction of heart disease. As the well-known quote says "Prevention is better than cure", early prediction & its control can be helpful to prevent & decrease the death rates due to heart disease.

2.2. References

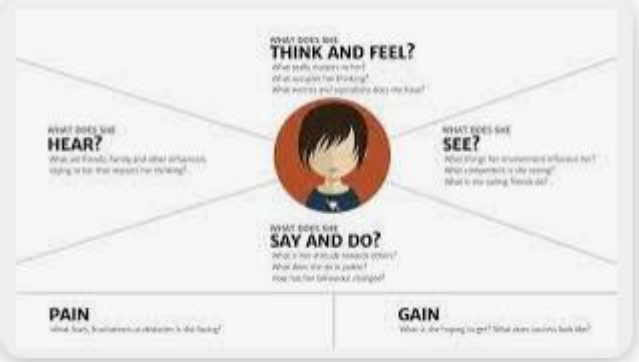
- ✓ Purushottam ,et ,al proposed a paper "Efficient Heart Disease Prediction System" using hill climbing and decision tree algorithms .They used Cleveland dataset andpreprocessing of data is performed before using classification algorithms. TheKnowledge Extraction is done based on Evolutionary Learning (KEEL), an open-source data mining tool that fills the missing values in the data set.A decision treefollows top-down order. For each actual node selected by hill-climbing algorithm anode is selected by a test at each level. The parameters and their values used areconfidence. Its minimum confidence value is 0.25. The accuracy of the system is about86.7%.
- ✓ Santhana Krishnan. J ,et ,al proposed a paper "Prediction of Heart Disease Using Machine Learning Algorithms" using decision tree and Naive Bayes algorithm forprediction of heart disease. In decision tree algorithm the tree is built using certainconditions which gives True or False decisions. The algorithms like SVM, KNN areresults based on vertical or horizontal split conditions depends on dependent variables.But decision tree for a tree like structure having root node, leaves and branches baseon the decision made in each of tree Decision tree also help in the understating theimportance of the attributes in the dataset. They have also used Cleveland data set.Dataset splits in 70% training and 30% testing by using some methods. This algorithmgives 91% accuracy. The second algorithm is Naive Bayes, which is used forclassification. It can handle complicated, nonlinear, dependent data so it is foundsuitable for heart disease dataset as this dataset is also complicated, dependent andnonlinear in nature. This algorithm gives an 87% accuracy.
- ✓ Sonam Nikhar et al proposed paper " Prediction of Heart Disease Using MachineLearning Algorithms" their research gives point to point explanation of Naïve Bayes and decision tree classifier that are used especially in the prediction of Heart Disease.
- ✓ Aditi Gavhane et al proposed a paper "Prediction of Heart Disease Using Machine Learning", in which training and testing of dataset is performed by using neuralnetwork algorithm multi-layer perceptron. In this algorithm there will be one input layer and one output layer and one or more layers are hidden layers between these two input and output layers. Through hidden layers each input node is connected to output layer. This connection is assigned with some random weights. The other input is calledbias which is assigned with weight based on requirement the connection between the nodes can be feedforwarded or feedback.
- ✓ Avinash Golande et al, proposed "Heart Disease Prediction Using Effective Machine Learning Techniques" in which few data mining techniques are used that support the doctors to differentiate the heart disease. Usually utilized methodologies are k-nearest neighbour, Decision tree and Naïve Bayes. Other unique characterization-based strategies utilized are packing calculation, Part thickness, consecutive negligible streamlining and neural systems, straight Kernel self arranging guide and SVM (Bolster Vector Machine).

2.3. Problem Statement Definition

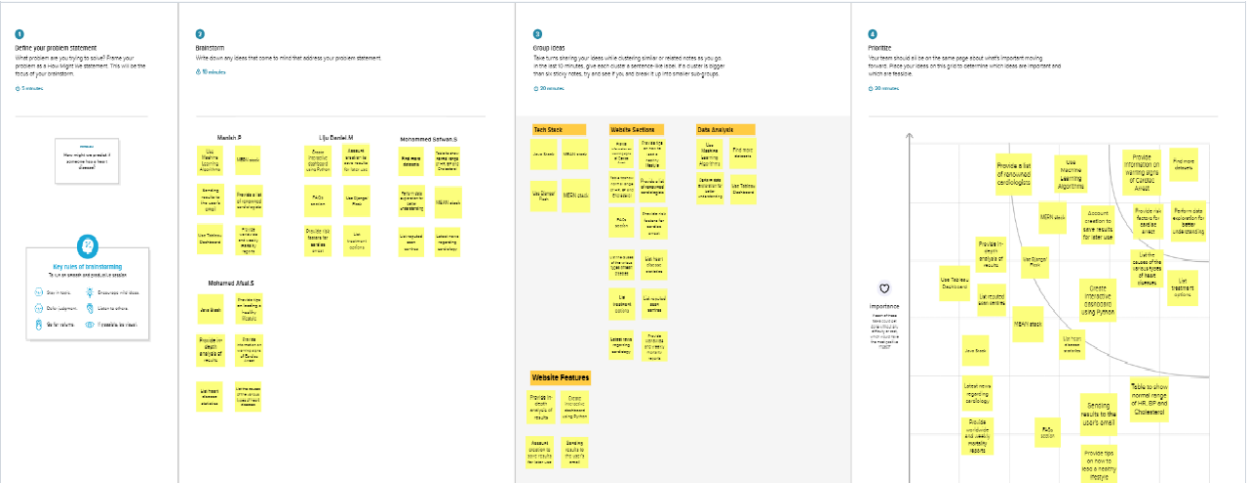
The major challenge in heart disease is its detection. There are instruments available which can predict heart disease but either it are expensive or are not efficient to calculate chance of heart disease in human. Early detection of cardiac diseases can decrease the mortality rate and overall complications. However, it is not possible to monitor patients everyday in all cases accurately and consultation of a patient for 24 hours by a doctor is not available since it requires more sapience, time and expertise. Since we have a good amount of data in today's world, we can use various machine learning algorithms to analyze the data for hidden patterns. The hidden patterns can be used for health diagnosis in medicinal data.

3. Ideation and Proposed Solution

3.1. Empathy Map Canvas:



3.2 Ideation and Brainstorming



3.3 Proposed Solution

The working of the system starts with the collection of data and selecting the important attributes. Then the required data is preprocessed into the required format. The data is then divided into two parts training and testing data. The algorithms are applied and the model is trained using the training data. The accuracy of the system isobtained by testing the system using the testing data. This system is implemented using the following modules.

- ✓ Collection of Dataset
- ✓ Selection of attribute
- ✓ Data Pre-Processing
- ✓ Balancing of Data
- ✓ Disease Prediction

1. Collection of dataset

Initially, we collect a dataset for our heart disease prediction system. After the collection of the dataset, we split the dataset into training data and testing data. The training dataset is used for prediction model learning and testing data is used for evaluating the prediction model. For this project, 70% of training data is used and 30% of data is used for testing. The dataset used for this project is Heart Disease UCI. The dataset consists of 76 attributes; out of which, 14 attributes are used for the system.

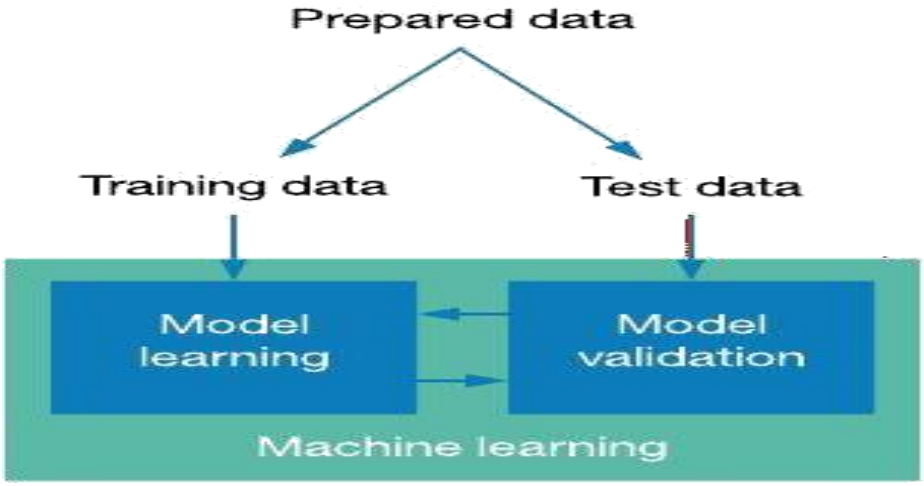


Figure: Collection of Data

2. Selection of attributes

Attribute or Feature selection includes the selection of appropriate attributes forthe prediction system. This is used to increase the efficiency of the system. Various attributes of the patient like gender, chest pain type, fasting blood pressure, serum cholesterol, exang, etc are selected for the prediction. The Correlation matrix is used for attribute selection for this model.

Figure: Correlation matrix

3. Pre-processing of Data

Data pre-processing is an important step for the creation of a machine learning model. Initially, data may not be clean or in the required format for the model which can cause misleading outcomes. In pre-processing of data, we transform data into our required format. It is used to deal with noises, duplicates, and missing values of the dataset. Data pre-processing has the activities like importing datasets, splitting datasets, attribute scaling, etc. Preprocessing of data is required for improving the accuracy of the model.

Figure: Data Pre-processing

4. Balancing of Data

Imbalanced datasets can be balanced in two ways. They are Under Samplingand Over Sampling

(a) Under Sampling:

In Under Sampling, dataset balance is done by the reduction of the size of the ampleclass. This process is considered when the amount of data is adequate.

(b) Over Sampling:

In Over Sampling, dataset balance is done by increasing the size of the scarce samples.This process is considered when the amount of data is inadequate.

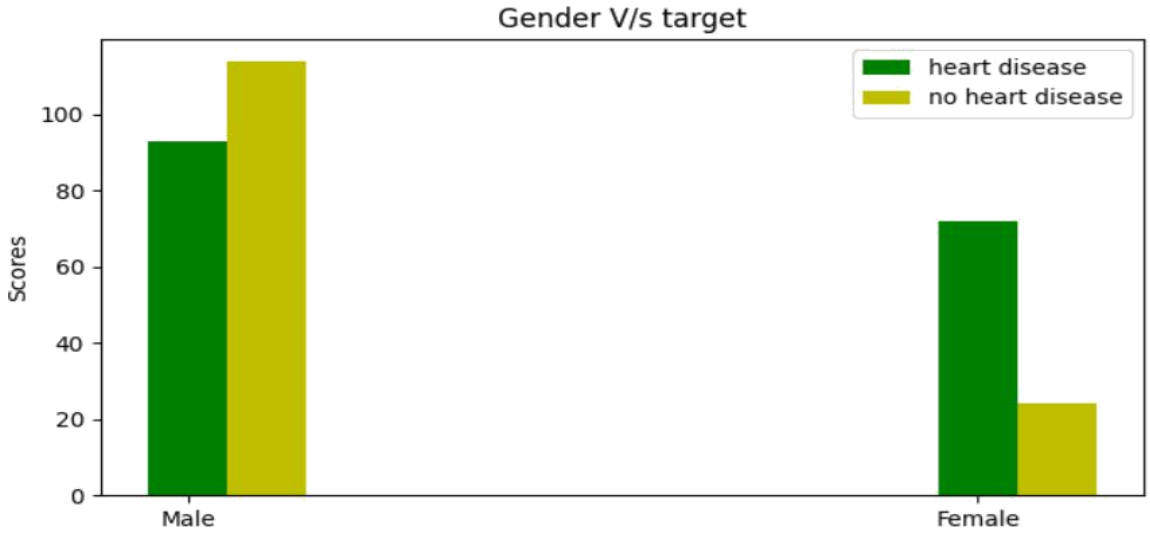


Figure: Data Balancing

5. Prediction of Disease

Various machine learning algorithms like SVM, Naive Bayes, Decision Tree, Random Tree, Logistic Regression, Ada-boost, Xg-boost are used for classification. Comparative analysis is performed among algorithms and the algorithm that gives the highest accuracy is used for heart disease prediction.

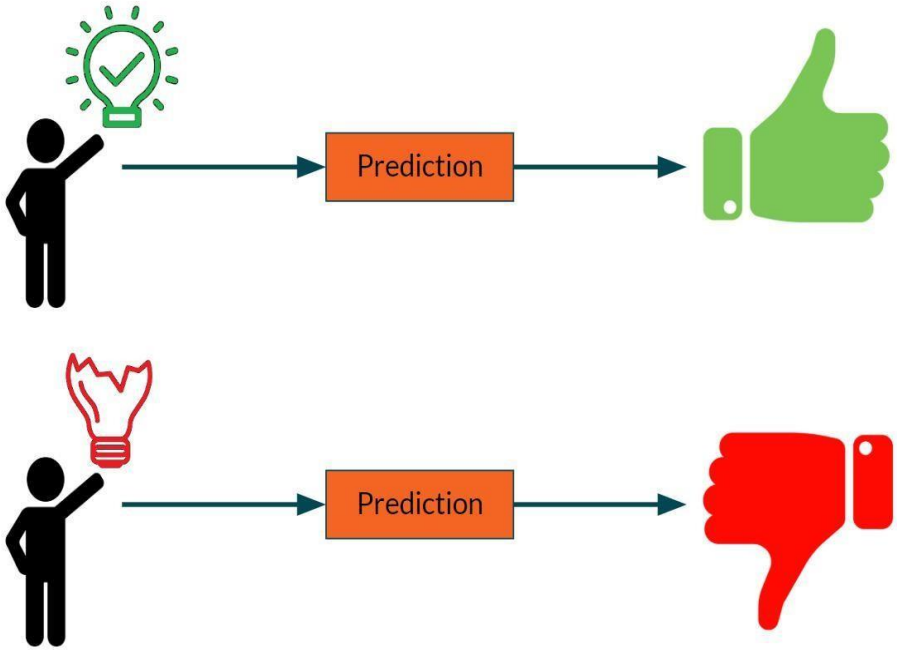


Figure: Prediction of Disease

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The leading cause of death is heart disease. Heart disease refers to several types of abnormalities in heart conditions. It is inconvenient for a common man to take ECG tests periodically. Also, lack of proper diagnostic tools and accurate results affect the treatment of cardiac patients Thus based on a patient's medical history, an expert's symptom analysis report, and physical laboratory results, invasive procedures are used to identify heart related problems. And so, there is a need for a replacement, which must be less complicated and reliable. The goal is to come up with a reliable prediction model so that the hospital can use this information to treat the patients at the starting state of the disease.
2.	Idea / Solution description	The solution is to provide an interactive dashboard for visualising and predicting cardiac problems. IBM Cognos platform is used to visualize the given data. Machine learning techniques like Support Vector Machine, Decision tree, Naive Bayes, Random forest, K-Nearest Neighbour, and Neural networks are used to predict cardiac disease. To achieve greater accuracy, fusion of these algorithms is done. Exploratory Data Analysis (EDA) is a method to analyse data using advanced techniques to expose hidden structure, enhance the insight into a given dataset, identify the anomalies and build parsimonious models to test the underlying assumptions. The parameters provided in the data set help hospitals identify the patient's heart condition. An informative and creative dashboard can be created to present the data and utilize it for further medications.

3.	Novelty / Uniqueness	The prime novelty of the solution is the fusion of highly efficient algorithms, that eliminates the disadvantage of every algorithm when employed individually and also provides a higher leb=vel of accuracy in the prediction. Another innovation is employed in the dashboard by providing diet and fitness related suggestions to the user based on his/her medical reports and history. In addition to it, the patient is given a list of hospitals closer to the patient's locality and severity of the disease.
4.	Social Impact / Customer Satisfaction	It helps with disease prediction at an early stage and alerts the user about his/her current health status. Heart disease can be cured by a mix of medication, lifestyle modifications, and occasionally, surgery. The system helps the user as well as the doctor to make better decisions. Complex questions related to heart diseases can be answered by extracting hidden knowledge, i.e., patterns and relationships from the heart disease database.
5.	Business Model (Revenue Model)	● This interactive dashboard for heart disease prediction can be installed in hospitals and healthcare facilities. Predicted outcomes can be utilized to avoid expensive surgeries. ● It can be used in educational institutions, industries and all types of workplaces to monitor the employees' health conditions and thereby helping them lead a healthier life.
6.	Scalability of the Solution	● The proposed solution works efficiently in both smaller and larger datasets. ● This predictive model can be used to detect diseases in other internal organs too.

3.4 Problem Solution Fit:

1.CUSTOMER SEGMENT(S)

S

- Senior citizens
- Hospitals
- Pharmaceutical agencies
- Smokers
- Alcoholics
- Diabetes patients
- Hypercholesterolemia patients
- Hypertension patients
- Thrombosis patients
- Obese persons
- Peripheral artery disease patients
- Angina patients

6. CUSTOMER CONSTRAINTS

CC

- Instant network connectivity
- Presence of good-condition communication devices like smartphones and laptops
- Financial constraints to consult specialists
- Lack of awareness about heart disease
- Complex and expensive scanning methodologies
- Psychological problems
- Lack of hope in treatment

5. AVAILABLE S

- Manual data visualizatio are very tedi us
- Consult doctors (heart spe requires financial
- Quit smoking
- Restrain from alcohol
- Practice a healthy lifesty

exercises and a nutritious

- Take cholesterol tests

2.JOBS-TO-BE-DONE / PROBLEMS

J&P

- The data used for prediction should be accurate and reliable.
- If data is skewed, then the prediction is also skewed
- Predictions should be done based on various metrics such as blood pressure, cholesterol levels, heartbeat rates, etc. thatrequire complex integration
- Risk of lives depends on further medical support
- Timely alerts help in the prevention of the sudden onset of cardiac arrests

9.PROBLEM ROOT CAUSE

RC

- Difficulty in predicting heart disease at earlier stages
- Lack of awareness about physical fitness
- Genetic problems
- Lifestyle and eating habits
- A buildup of fatty plaques in the arteries is the most common cause of coronary artery disease.
- Obesity
- Alcohol and Smoking habits
- Stress, anxiety, depression and psychological problems

7.BEHAVIOUR

- Look up on the internet to fin
- Visit healthcare specialists
- Take advice from friends and
- Physical activity helps to low disease.
- Adopting a healthy diet can h blood pressure and cholestero risk of diabetes.
- Reduction of intake of alcoho
- Get quality sleep
- Prioritizing mental peace
- Develop unwanted mental tra about the aftermath of disease
- Falling into wrong assumptio instant solutions that have wo

3. TRIGGERS

- Insufficient ways to handle huge amounts of datasets
- Lives depending on medical support
- Symptoms such as chest pain, shortness of breath, etc.
- Lifestyle modifications
- Need to search for heart specialist at affordable price
- Need to apply for health insurance
- Anxiety and destructive curiosity
- Others getting treated due to earlier detection

4. EMOTIONS: BEFORE / AFTER

Before

- Fear of being attacked by diseases that don't have improved treatments
- Confusion and lack of clarity about one's health conditions
- The anxiety of being hospitalized and the financial stress

After

- Clarity about the disease and its severity
- Peace of mind due to earlier predictions
- Financial stress relief

10. YOUR SOLUTION

- The data is visualized with the aid of the IBM Cognos Analytics Tool for providing better insight into patients' health so that doctors could make better decisions
- With the notable technology of AI/ML and the given various metrics, heart diseases are predictedat an earlier stage and the same is displayed to theuser in an interactive dashboard
- Healthy lifestyle habits — such as eating a low-fat,low-salt diet, getting regular exercise and good sleep, and not smoking are user-specific suggestions are given
- Surgeries depend on the type of heart disease and the amount of damage to the heart, so suitable medical facility centers and specialized doctors are recommended

8. CHANNELS of BEHAVIOUR

CH

ONLINE

- Surfing the internet for disease-related information
- Using apps that provide fitness suggestions

OFFLINE

- Getting to know other people suffering from similar issues
- Visit doctors for a professional opinion
- Increasing the overall health conscious

4. Requirement Analysis:

4.1 Functional Requirements

Windows 7 or Higher

Windows Vista is a major release of the Windows NT operating system developed by Microsoft. It was the direct successor to Windows XP, which was released five years before, at the time being the longest time span between successive releases of Microsoft Windows desktop operating systems. Development was completed and over the following three months, it was released in stages to computer hardware and software manufacturers, business customers and retail channels. it was released internationally and was made available for purchase and download from the Windows Marketplace; it is the first release of Windows to be made available through a digital distribution platform.

SQL

SQL (S-Q-L Structured Query Language) is a domain-specific language used in programming anddesigned for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system (RDSMS). It is particularly useful in handling structured data, i.e. data incorporating relations among entities and variables.

SQL offers two main advantages over older read–write APIs such as ISAM or VSAM. Firstly, it introduced the concept of accessing many records with one single command. Secondly, it eliminates the need to specify how to reach a record, e.g. with or without an index.

Visual studio

Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silver light. It can produce both native code and managed code.

4.2 Non Functional Requirements:

Processor-i3

The Core i3 processor is available in multiple speeds, ranging from 1.30 GHz up to
3.50 GHz, and features either 3 MB or 4 MB of cache. It utilizes either the LGA 1150 or LGA 1155 socket on a motherboard. Core i3 processors are most often found as dual-core, having two cores. However, a select few high-end Core i3 processors are quad-core, featuring four cores.

Hard Disk-5GB

A computer hard disk drive (HDD) is a non-volatile data storage device. Non-volatile refers to storage devices that maintain stored data when turned off. All computers need a storage device, and HDDs are just one example of a type of storage device. HDDs are usually installed inside desktop computers, mobile devices, consumer electronics and enterprise storage arrays in data centers. They can store operating systems, software programs and other files using magnetic disks. Storage devices like hard disks are needed to install operating systems, programs and additional storage devices, and to save documents. Without devices like HDDs that can retain data after they have been turned off, computer users would not be able to store programs or save files or documents to their computers. This is why every computer needs at least one storage device to permanently hold data as long as it is needed.

Memory-1GB RAM

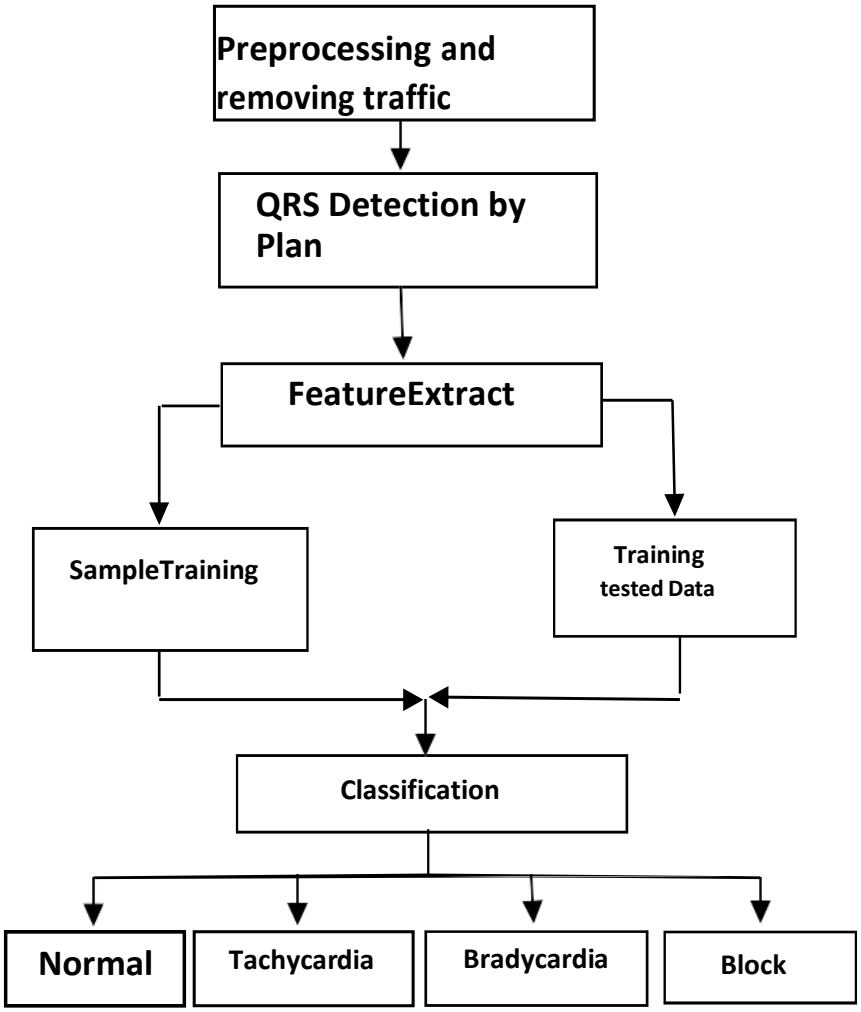
Computer random access memory (RAM) is one of the most important componentsin determining your system's performance. RAM gives applications a place to store and access data on a short-term basis. It stores the information your computer is actively using so that it can be accessed quickly. The more programs your system is running, the more you'll need. SSDs (solid state drives) are also importantcomponents and will help your system reach its peak performance. RAM allows the computer to perform many of its everyday tasks, such as loading applications, browsing the internet, editing a spreadsheet, or experiencing the latest game. As a rule, the more memory you have, the better.

Internet Connection

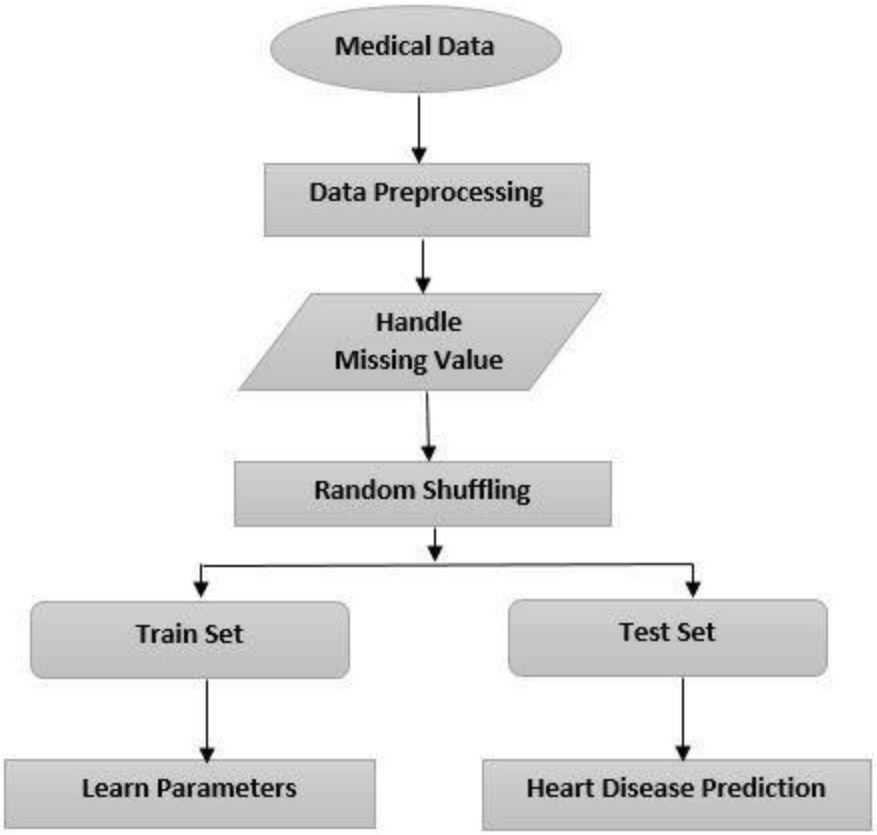
Internet Connection means a connection provided by an Internet Service Provider that enables individual computers or other hardware components, either individually or registered within a Local Area Network, to exchange Data over the public Internet. Internet access is often provided at home, schools, workplaces, public places, internet cafes, libraries and other locations. The internet began to gain popularity with dial-up internet access. In a relatively short time, internet access technologies changed, providing faster and more reliable options. Currently, broadband technologies such as cable internet and ADSL are the most widely used methods for internet access. The speed, cost, reliability and availability of internet access depends on the region, internet service provider and type of connection.

5.Project Design:

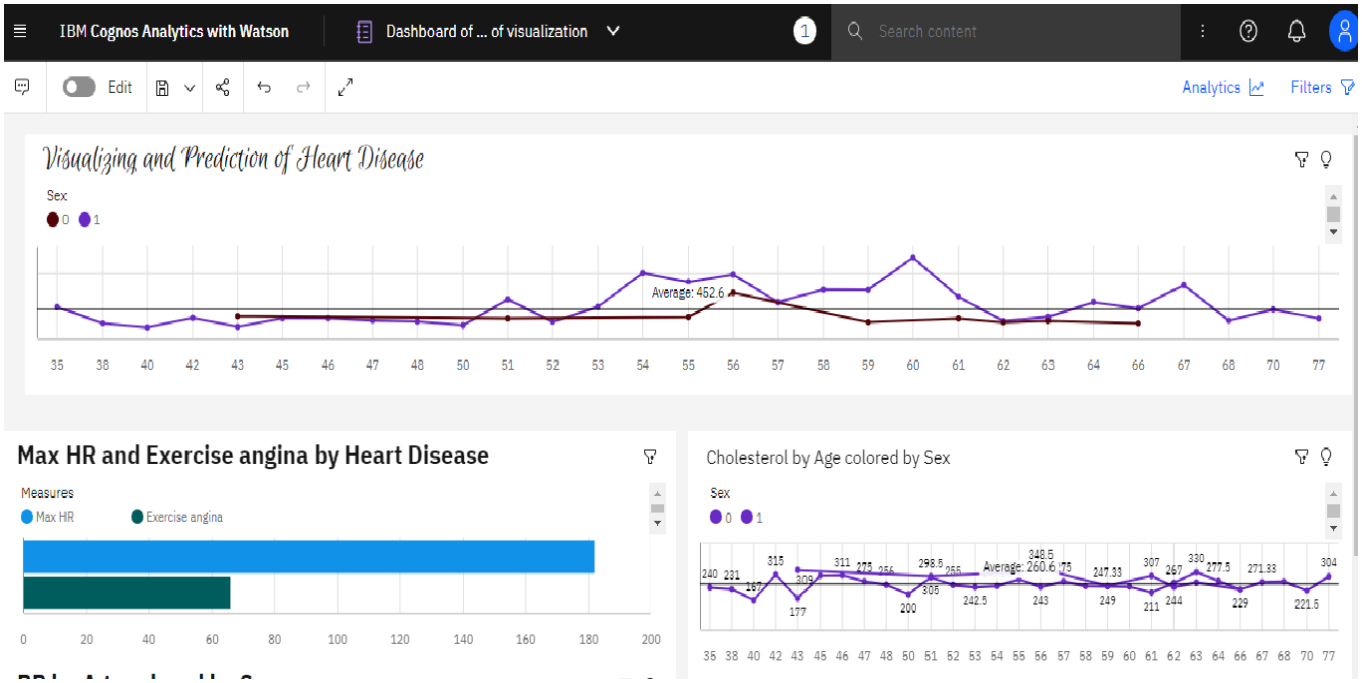
5.1 Data Flow Diagram



5.2. Solution Architecture



5.3. User stories



6.Project planning and Scheduling

6.1 Sprint planning and Estimation:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Ramya G Nishanthini R Savitha C Shanthini T
Sprint-1	Confirmation	USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Ramya G Nishanthini R Savitha C Shanthini T
Sprint-2		USN-3	As a user, I can register for the application through Facebook	2	Low	Savitha C
Sprint-1		USN-4	As a user, I can register for the application through Gmail	2	Medium	Ramya G
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	Ramya G
Sprint-1	User Interface	USN-6	As a user, I should not need any pre requisites to handle the UI	1	Medium	Shanthini T
Sprint-1	Dashboard		As a user, will use the templates and resources of the dashboard effectively	2	High	Savitha C

6.2. Sprint Delivery Schedule

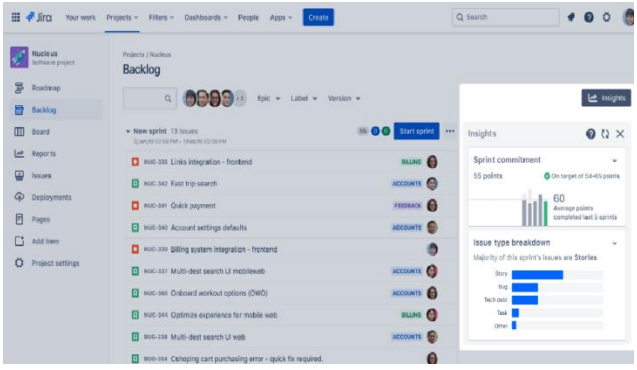
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	30	30 Oct 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	49	06 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	50	07 Nov 2022

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

6.3. Report from



7.Coding and Solution:

7.1 Feature -1

The datasets are collected from the "Kaggle". The datasets are collected with 14 columns such as Age, Sex, Blood Pressure, Cholestrol level, Maximum Heart Rate, etc…The data's in the datasets are to be read and the model has to be built to be plot a graph on behalf of using a matplotlib module. It is to be plot on the graphs(Bar, Pie)etc..

S. No.	Attribute	Description	Type
1	Age	Patient's age (29 to 77)	Numaric
2	Sex	Gender of patient(male-0female-1)	Nominal
3	Cp	Chest pain type	Nominal
4	Trestbps	Resting blood pressure(in mm Hg on admission to hospital ,values from 94 to 200)	Numerical
5	Chol	Serum cholesterol in mg/dl, values from 126 to564)	Numerical
6	Fbs	Fasting blood sugar>120 mg/dl, true-1 false-0)	Nominal
7	Resting	Resting electrocardiographic sresult (0 to 1)	Nominal
8	Thali	Maximum heart rate achieved(71 to 202)	Numerical
9	Exang	Exercise included agina(1-yes 0-no)	Nominal
10	Oldpeak	ST depression introduced by exercise relative to rest(0 to .2)	Numerical
11	Slope	The slop of the peak exercise ST segment (0 to 1)	Nominal
12	Ca	Number of major vessels (0-3)	Numerical
13	Thal	3-normal	Nominal
14	Targets	1 or 0	Nominal

A. Linear regression

It is the supervised learning technique. It is based on the relationship between independent variable and dependent variable as seen in Fig.5 variable "x" and "y" are independent and dependent variable and relation between them is shown by equation of line which is linear in nature that why this approach is called linear regression.

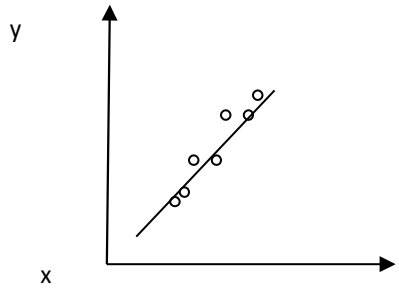


Fig.5 relation between x and y

It gives a relation equation to predict a dependent variable value "y" based on a independent variable value "x" as we cansee in the Fig.5 so it is concluded that linear regression technique give the linear relationship between x(input) and y(output).

B. Decision tree

On the other hand decision tree is the graphical representation of the data and it is also the kind of supervised machine learning algorithms.

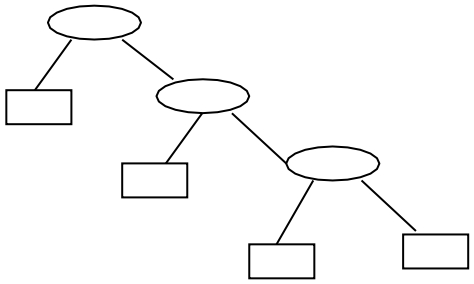


Fig.6 Decision tree

For the tree construction we use entropy of the data attributes and on the basis of attribute root and other nodes are drawn.

$$\text{Entropy} = -\sum Pij \log Pij \qquad (1)$$

In the above equation of entropy (1) Pij is probability of the node and according to it the entropy of each node is calculated. The node which have highest entropy calculation isselected as the root node and this process is repeated until all the nodes of the tree are calculated or until the tree constructed.

C. Preprocessing of data

Preprocessing needed for achieving prestigious result from themachine learning algorithms. For example Random forest algorithm does not support null values dataset and for this we have to manage null values from original raw data.

For our project we have to convert some categorized value by dummy value means in the form of “0”and “1” by using following code:

Classifier.py

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
sns.set_style('whitegrid')
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
data = pd.read_csv(r'C:\Users\DELL\Desktop\heart disease/Heart_Disease_Prediction.csv')
data.head()
data.info()
data.describe(include = 'all')
data.isnull().sum()
data.nunique()
data.columns
colm = ['Sex', 'Chest pain type','FBS over 120','EKG results','Exercise angina','Slope of ST', 'Number of vessels
fluro', 'Thallium', 'Heart Disease']
for col in colm:
  sns.countplot(data[col])
  plt.show()
  plt.figure(figsize=(12,10))
corr = data.corr()
sns.heatmap(corr, annot = True, linewidths= 0.2, linecolor= 'black', cmap = 'afmhot')
data.columns
X = data[['Age', 'Sex', 'Chest pain type', 'BP', 'Cholesterol', 'FBS over 120',
       'EKG results', 'Max HR', 'Exercise angina', 'ST depression',
       'Slope of ST', 'Number of vessels fluro', 'Thallium']]
y = data['Heart Disease']
print(X.shape,y.shape)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=42529)
print(X_train.shape, X_test.shape, y_train.shape, y_test.shape)
train_convert = {"Absence":0,"Presence":1}
y_train = y_train.replace(train_convert)
test_convert = {"Absence":0,"Presence":1}
y_test = y_test.replace(test_convert)
mms = MinMaxScaler()
X_train = mms.fit_transform(X_train)
X_test = mms.fit_transform(X_test)
rf = RandomForestClassifier()
rf.fit(X_train,y_train)
pred = rf.predict(X_test)
cm = confusion_matrix(y_test,pred)
print(classification_report(y_test,pred))
sns.heatmap(cm, annot = True, fmt = 'g', cbar = False, cmap = 'icefire', linewidths= 0.5, linecolor= 'grey')
plt.title('Confusion Matrix')
plt.ylabel('Actal Values')
plt.xlabel('Predicted Values')
print("Accuracy Score = {}".format(round(accuracy_score(y_test,pred),5)))
```

#multi classifier

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
df = pd.read_csv(r'C:\Users\DELL\Desktop\heart disease/Heart_Disease_Prediction.csv')
df.dtypes
df.head()
df.isnull().sum()
format(len(df[df.duplicated()]))
name = df.columns
num_var = ['Age', 'BP', 'Cholesterol', 'Max HR', 'Heart Disease']
cat_var = [item for item in name if item not in num_var]

num_var_data = df[df.columns & num_var]
num_var_data.describe()
num_var_data.corr()
sns.heatmap(num_var_data.corr(), cmap="YlGnBu", annot=True)
sns.pairplot(num_var_data)
num_var_data[num_var_data['Cholesterol'] > 500]
sns.pairplot(num_var_data, hue = 'Heart Disease')

x = df.drop(['Heart Disease'], axis = 1)
y = df['Heart Disease']
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.20)
model = LogisticRegression()
model.fit(X_train, y_train)
r_sq = model.score(x, y)
print(f"coefficient of determination: {r_sq}")
from sklearn.preprocessing import LabelEncoder
x_train_enc = X_train

le = LabelEncoder()
le.fit(y_train)
y_train_enc = le.transform(y_train)


from sklearn.inspection import permutation_importance


model.fit(x_train_enc, y_train_enc)

results = permutation_importance(model, x_train_enc, y_train_enc, scoring='neg_mean_squared_error')


importance = results.importances_mean


for i,v in enumerate(importance):
```

```
    print('Feature: %0d, Score: %.5f' % (i,v))
    # plot feature importance
plt.bar([x for x in range(len(importance))], importance)
plt.show()
df.columns
selected_feature = ['Sex','Max HR', 'Number of vessels fluro', 'Thallium']
print(selected_feature)
data = df[df.columns & selected_feature]

X_train, X_test, y_train, y_test = train_test_split(data, y, test_size=0.33)
model = LogisticRegression()
model.fit(X_train, y_train)
r_sq = model.score(data, y)
print(f"coefficient of determination: {r_sq}")

models = []
models.append(('LR', LogisticRegression()))
models.append(('LDA', LinearDiscriminantAnalysis()))
models.append(('KNN', KNeighborsClassifier()))
models.append(('CART', DecisionTreeClassifier()))
models.append(('NB', GaussianNB()))
models.append(('SVM', SVC()))

results = []
names = []
scoring = 'accuracy'

for name, model in models:
    kfold = KFold(n_splits=10, random_state=7, shuffle = True)
    cv_results = cross_val_score(model, data, y, cv=kfold, scoring=scoring)
    results.append(cv_results)
    names.append(name)
    msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
    print(msg)

fig = plt.figure()
fig.suptitle('Algorithm Comparison')
ax = fig.add_subplot(111)
plt.boxplot(results)
ax.set_xticklabels(names)
plt.show()
```

D. Data Balancing

Data balancing is essential for accurate result because by data balancing graph we can see that both the target classes are equal. Fig.3 represents the target classes where “0” represents with heart diseases patient and “1” represents no heart diseases patient.

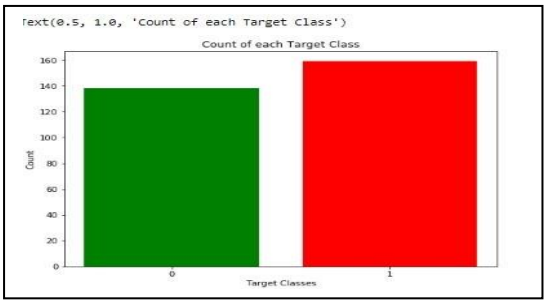


Fig.3 Target class view

E. Histogram of attributes

Histogram of attributes shows the range of dataset attributesand code which is used to create it.

```
dataset.hist()
```

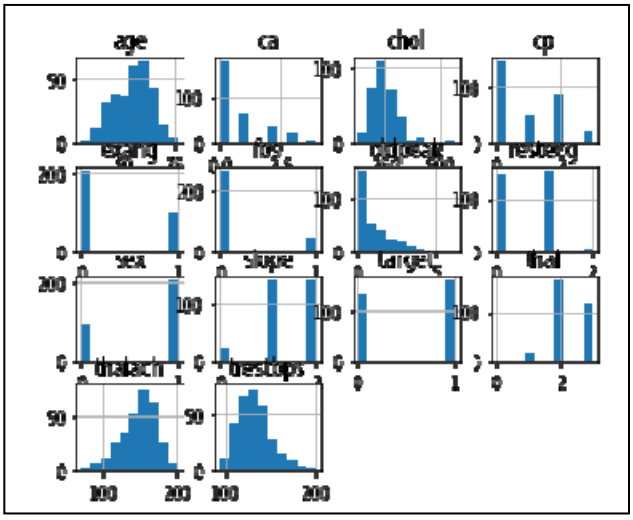


Fig.4 Histogram of attributes

7.2. Feature-2

A. About Jupyter Notebook

Jupiter notebook is used as the simulation tool and it is confortable for python programming projects. Jupytor notebook contains rich text elements and code also, which are figures, equations, links and many more. Because of the mixof rich text elements and code, these documents are perfect location to bring together an analysis description, and its results, as well as, they can execute data analysis in real time. Jupyter Notebook is an open-source, web-based interactive graphics, maps, plots, visualizations, and narrative text.

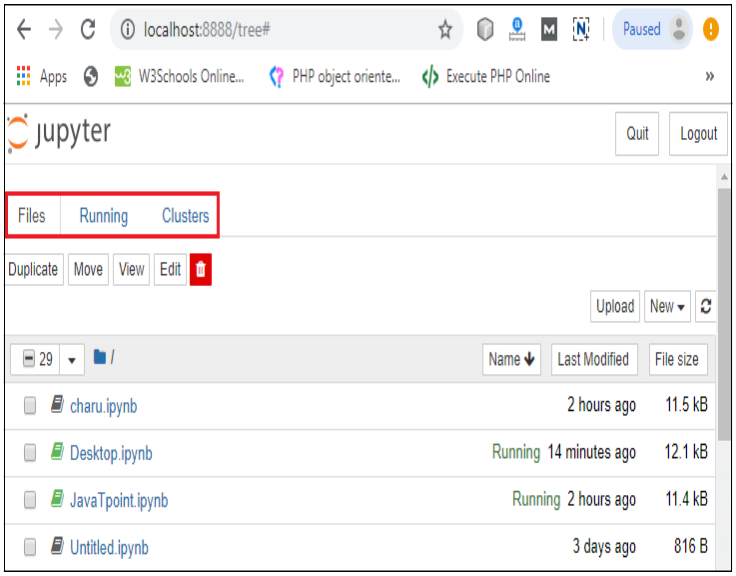


Fig.9 Jupyter Notebook

B. Accuracy calculation

Accuracy of the algorithms are depends on four values namely true positive(TP), false positive(FP), true negative(TN) and false negative(FN).

Accuracy= (FN+TP) / (TP+FP+TN+FN) (2)

The numerical value of TP, FP, TN, FN defines as:TP= Number of person with heart diseases

TN= Number of person with heart diseases and no heartdiseases

FP= Number of person with no heart diseases

FN= Number of person with no heart diseases and with heartdiseases

C. ***Support Vector Machine***

It is one category of machine learning technique which work on the concept of hyperplan means it classify the data bycreating hyper plan between them.

Training sample dataset is (Yi, Xi) where i=1,2,3,…….n and Xi is the ith vector, Yi is the target vector. Number of hyper plan decide the type of support vector such as example if a line is used as hyper plan then method is called linear support vector.

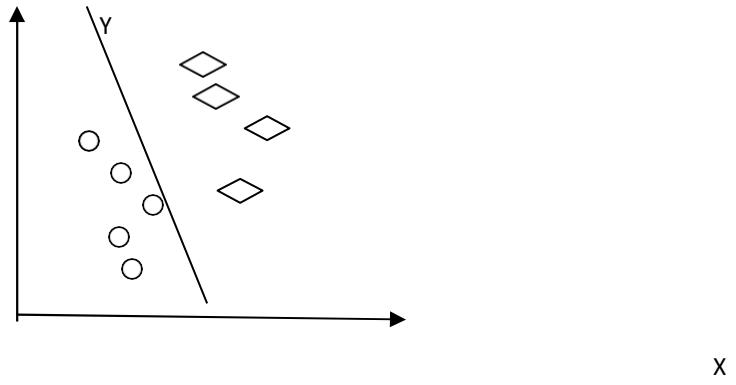


Fig.7 Linear Regression

D. K-nearest Neighbour

It work on the basis of distance between the location of data and on the basis of this distinct data are classified with each other. All the other group of data are called neighbor of each other and number of neighbor are decided by the user which play very crucial role in analysis of the dataset.

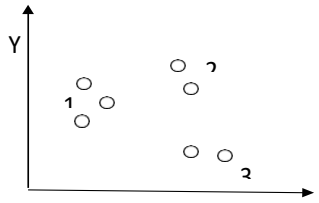


Fig.8 KNN where k=3

In the above Fig. k=3 shows that there are three neighbor that means three different type of data are there. Each cluster represented in two dimensional space whose coordinates are represented as (Xi,Yi) where Xi is the x-axis, Y represent y- axis and i= 1,2,3,….n.

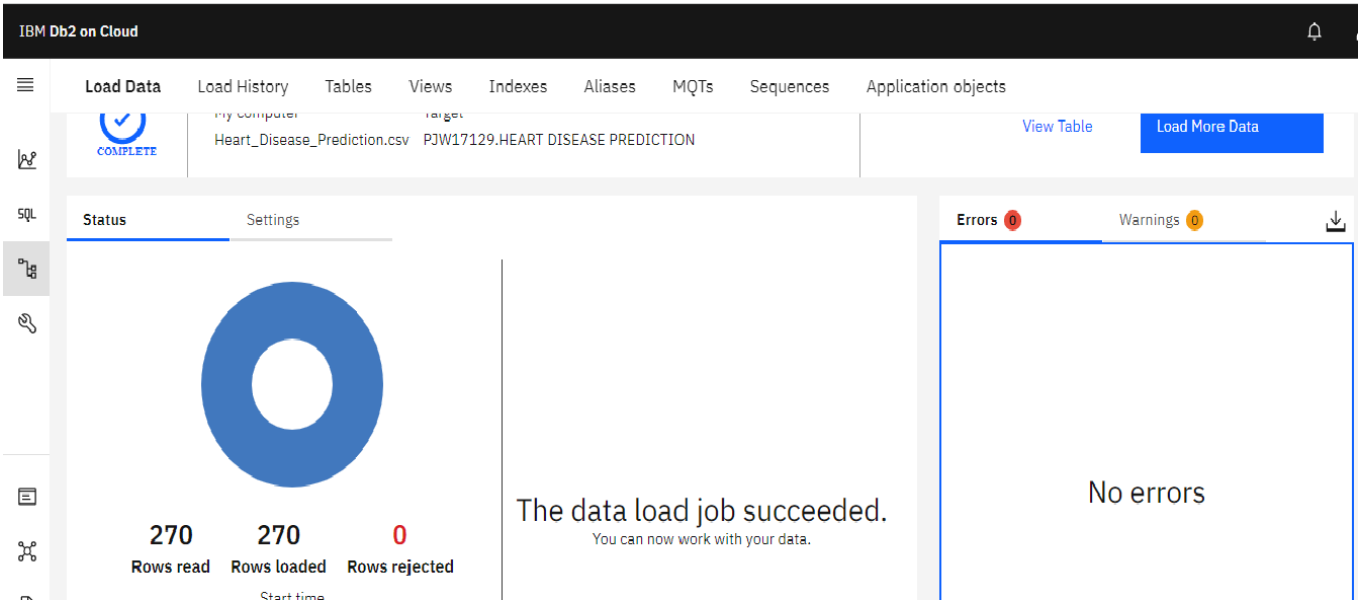
#model.py

```
# -*- coding: utf-8 -*-
"""
Created on Fri Nov 18 12:39:46 2022

@author: DELL
"""

import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as py
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
import pickle

df=pd.read_csv("Heart_Disease_Prediction.csv")
x=df.iloc[:,:-1].values
y=df.iloc[:,-1].values


std=StandardScaler()
x=std.fit_transform(x)

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)

model=RandomForestClassifier()
model.fit(x_train,y_train)
predictions=model.predict(x_test)
accuracy = accuracy_score(y_test,predictions)

def predict_heart_disease(parameter_list):
    return model.predict(parameter_list)[0]

pickle.dump(model, open('model.pkl','wb'))
```

#app.py

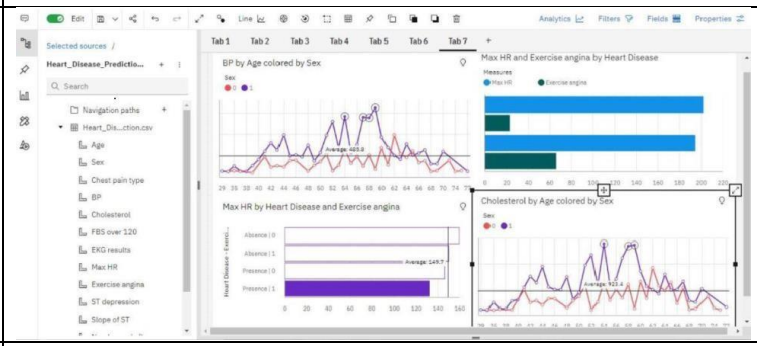
```
# -*- coding: utf-8 -*-

import numpy as np
import pickle
from flask import Flask, request, render_template

# Load ML model
model = pickle.load(open('model.pkl', 'rb'))

# Create application
app = Flask(__name__)

# Bind home function to URL
@app.route('/')
def home():
    return render_template('Heart Disease Classifier.html')

# Bind predict function to URL
@app.route('/predict', methods =['POST'])
def predict():

    # Put all form entries values in a list
    features = [float(i) for i in request.form.values()]
    # Convert features to array
    array_features = [np.array(features)]
    # Predict features
    prediction = model.predict(array_features)

    output = prediction

    # Check the output values and retrive the result with html tag based on the value
    if output == 1:
        return render_template('Heart Disease Classifier.html',
                               result = 'The patient is not likely to have heart disease!')
    else:
        return render_template('Heart Disease Classifier.html',
                               result = 'The patient is likely to have heart disease!')

if __name__ == '__main__':
#Run the application
    app.run()
```

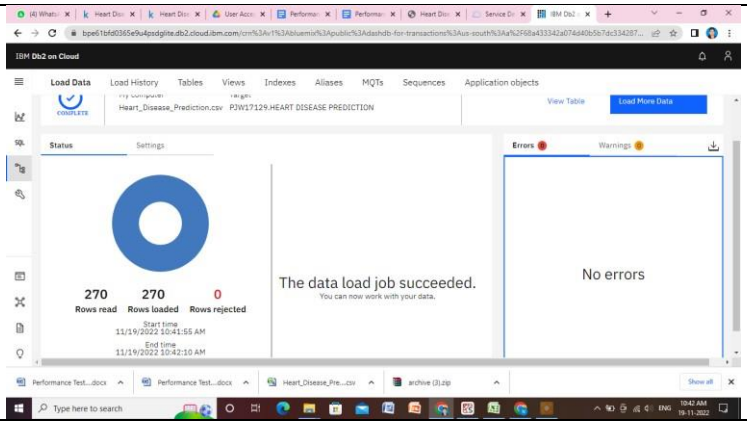
7.3. Database Schema



8.Testing

8.1 Testcase

S.No.	Parameter	Screenshot / Values
1.	Dashboard design	
2.	Data Responsiveness	Good and fast response

3.	Amount Data to Rendered (DB2 Metrics)	

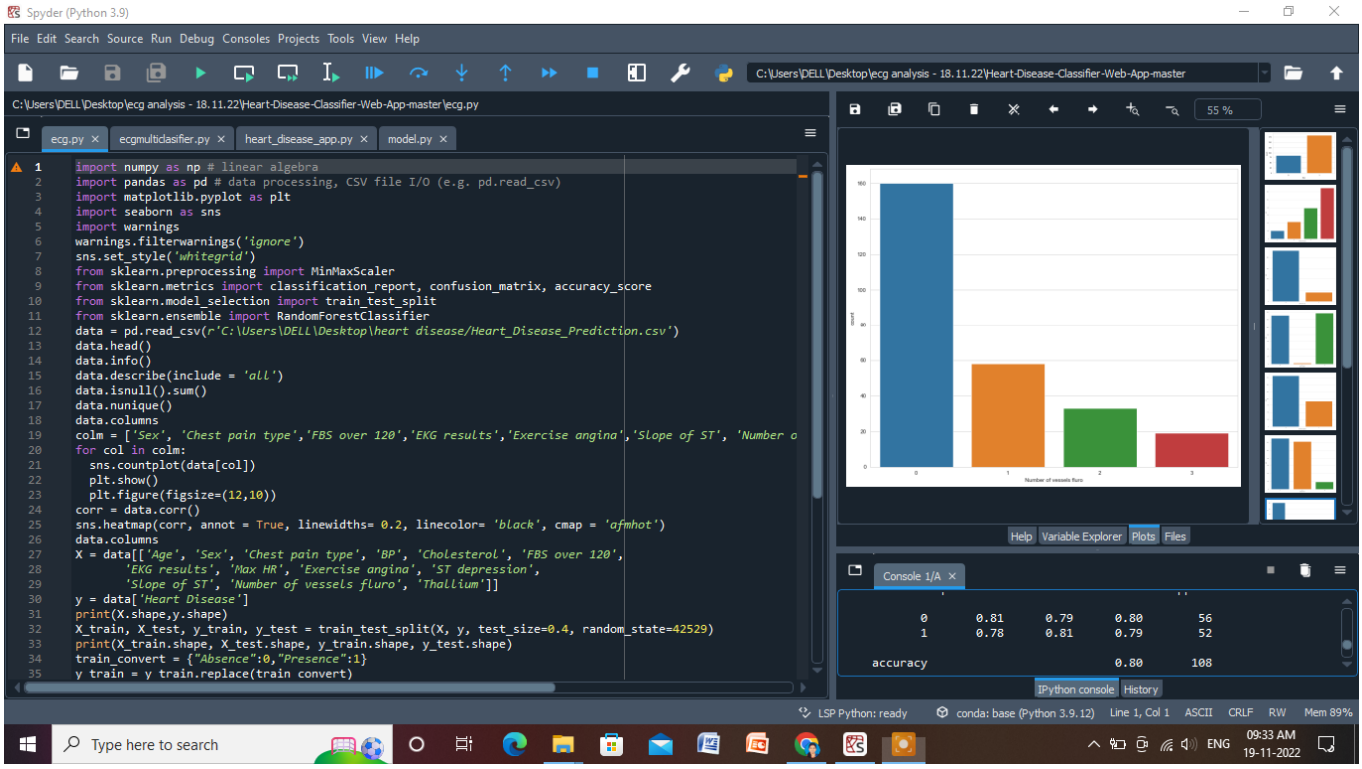
8.2 User acceptance testing

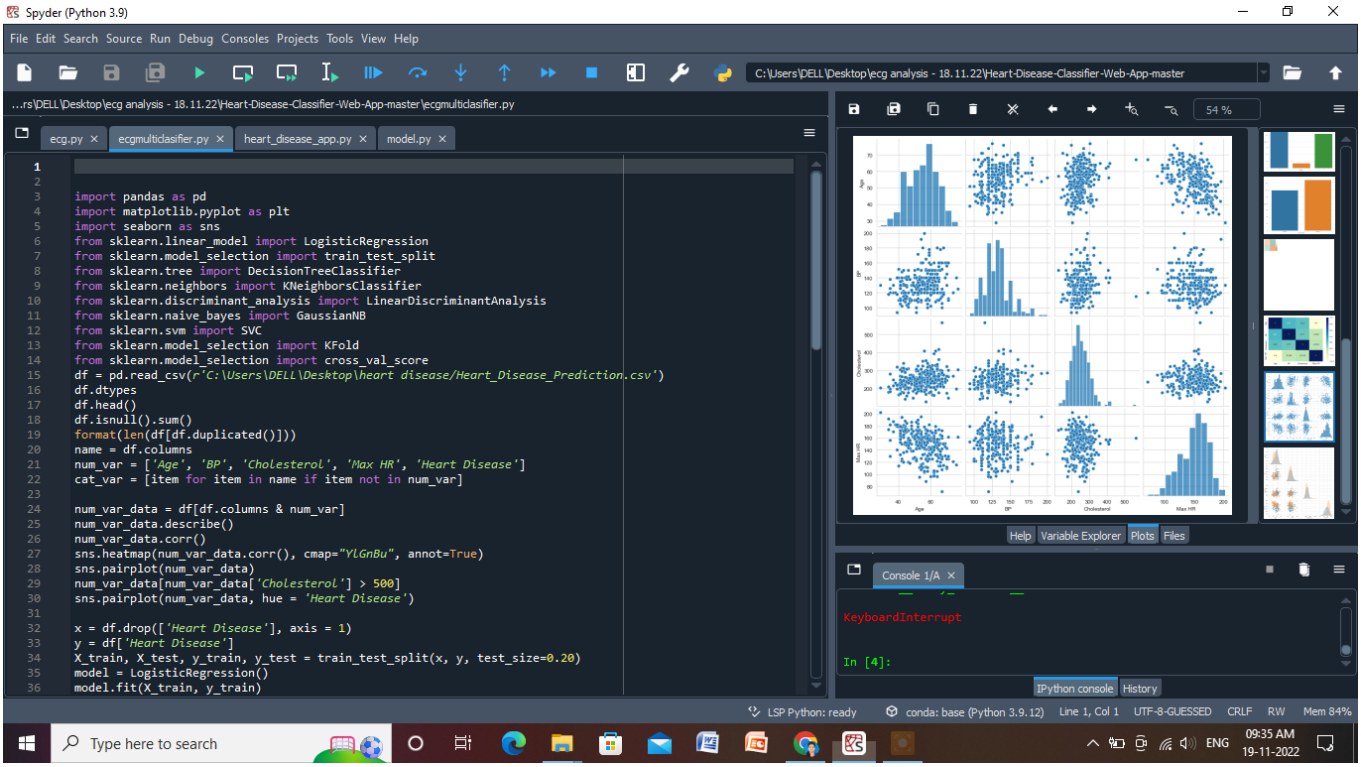
Defect Analysis

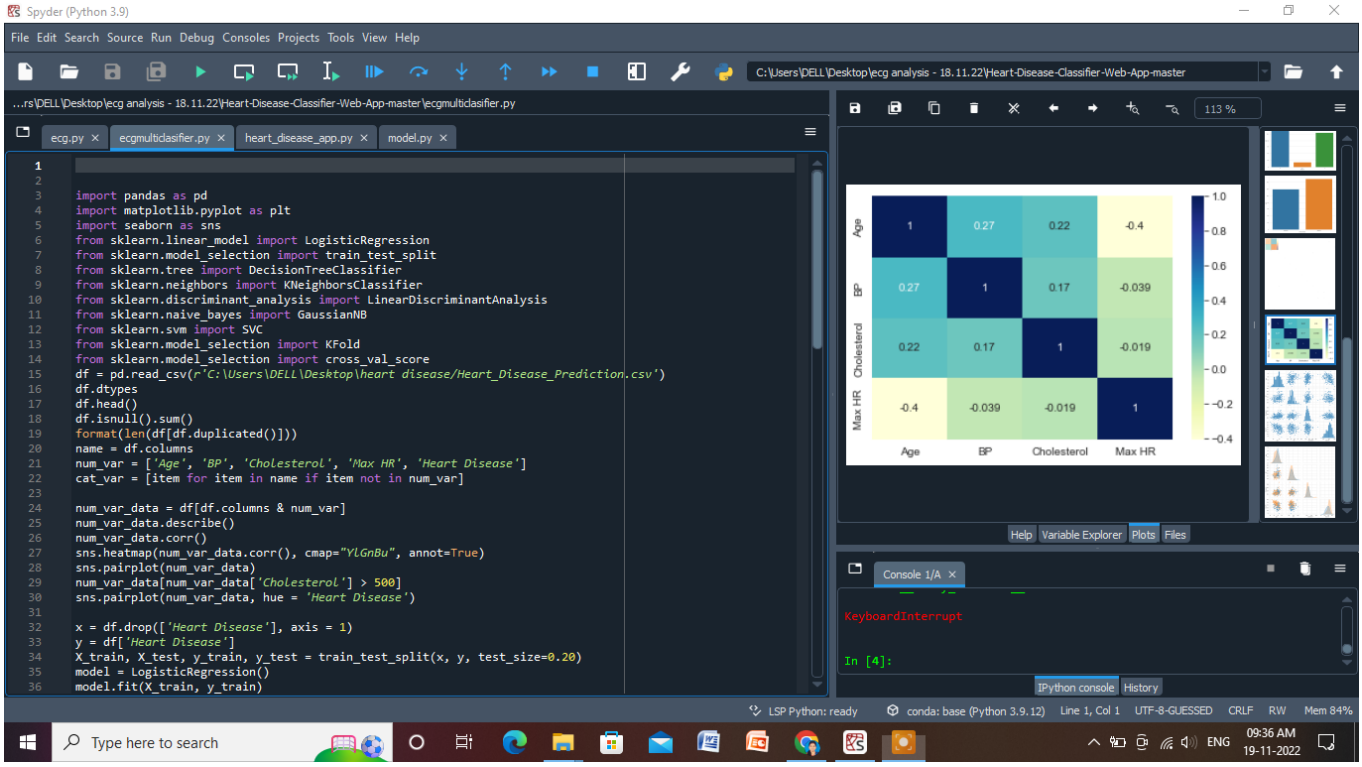
This report shows the number of resolved or closed bugs at each severity level, and howthey were resolved

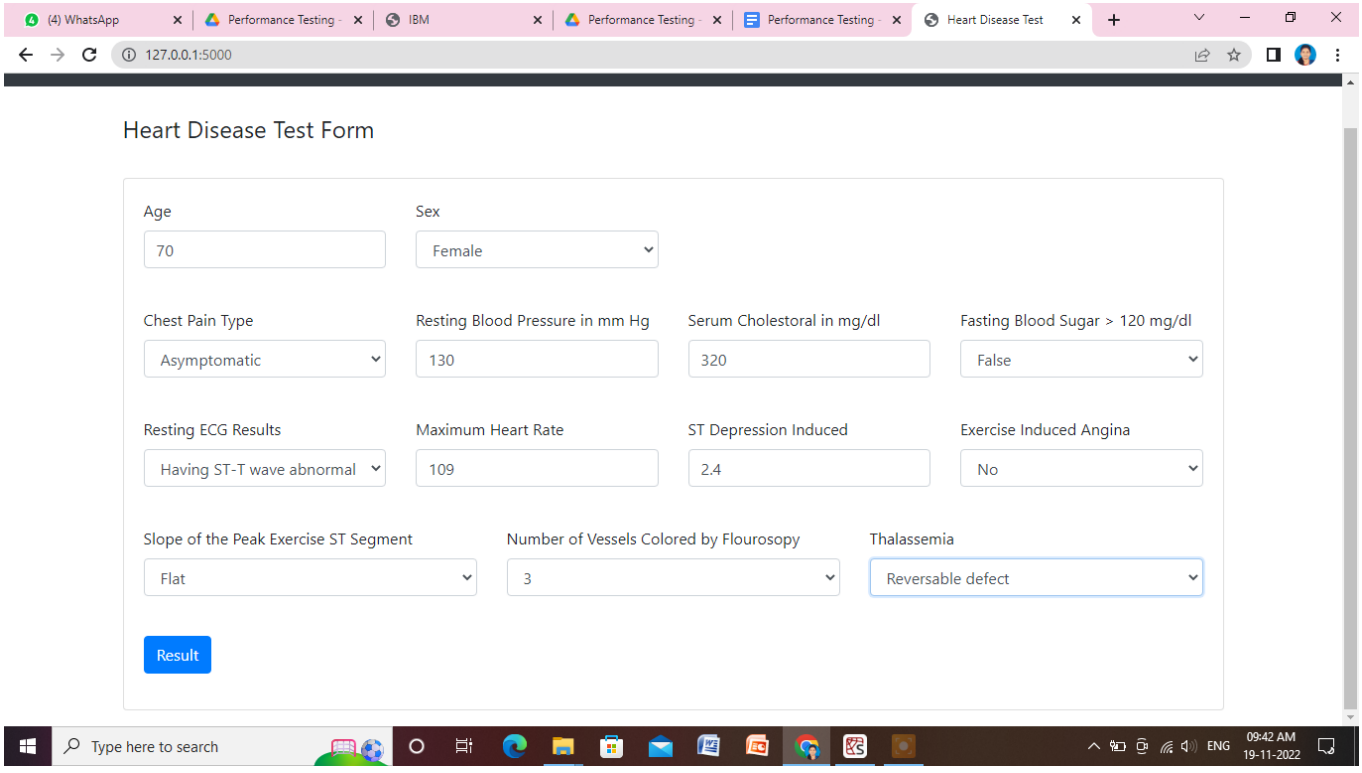
Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

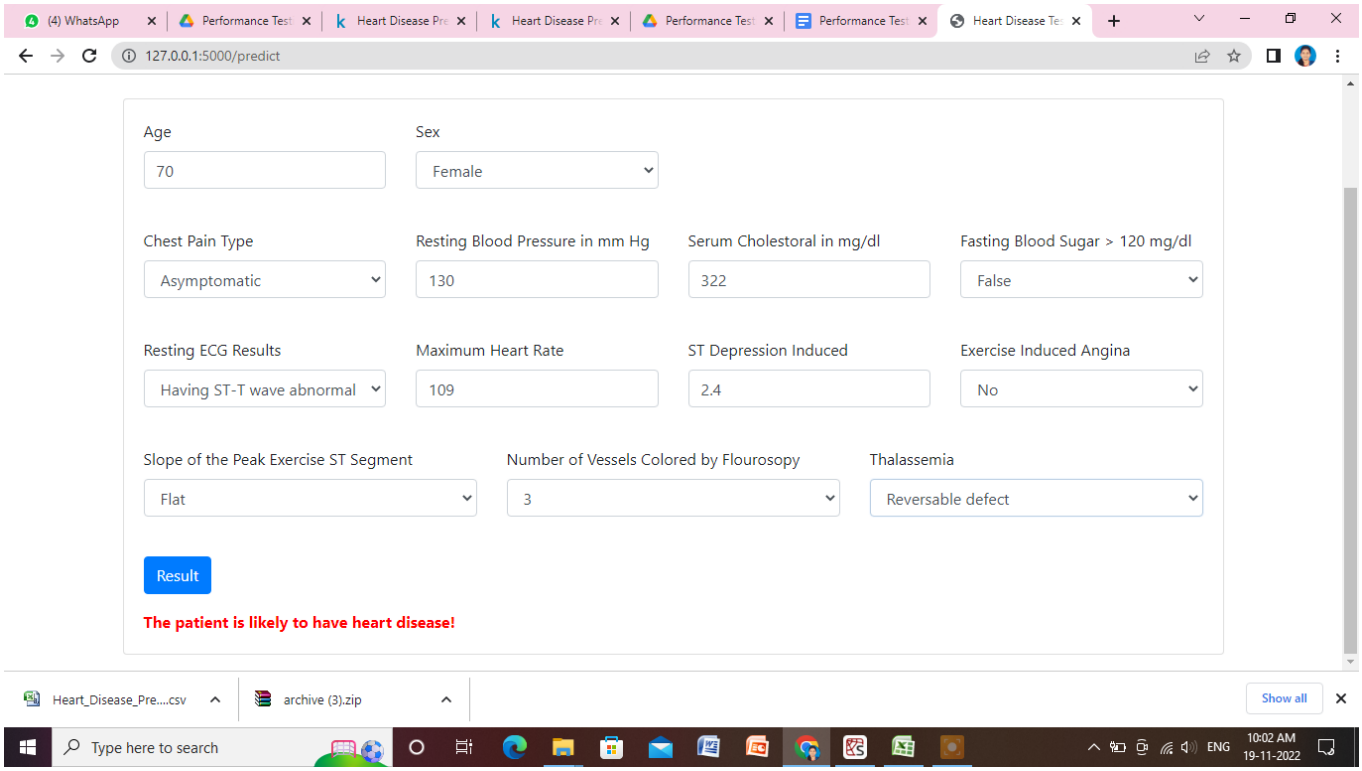
9.Results:


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\DELL\Desktop\ecg analysis - 18.11.22\Heart-Disease-Classifier-Web-App-master
ecg.py
ecgmulticlasifier.py
heart_disease_app.py
model.py
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
sns.set_style('whitegrid')
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
data = pd.read_csv(r'C:\Users\DELL\Desktop\heart disease/Heart_Disease_Prediction.csv')
data.head()
data.info()
data.describe(include = 'all')
data.isnull().sum()
data.nunique()
data.columns
colm = ['Sex', 'Chest pain type','FBS over 120','EKG results','Exercise angina','Slope of ST', 'Number o
for col in colm:
sns.countplot(data[col])
plt.show()
plt.figure(figsize=(12,10))
corr = data.corr()
sns.heatmap(corr, annot = True, linewidths= 0.2, linecolor= 'black', cmap = 'afmhot')
data.columns
X = data[['Age', 'Sex', 'Chest pain type', 'BP', 'Cholesterol', 'FBS over 120',
'EKG results', 'Max HR', 'Exercise angina', 'ST depression',
'Slope of ST', 'Number of vessels fluro', 'Thallium']]
y = data['Heart Disease']
print(X.shape,y.shape)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=42529)
print(X_train.shape, X_test.shape, y_train.shape, y_test.shape)
train_convert = {"Absence":0,"Presence":1}
y train = y train.replace(train convert)
Help Variable Explorer Plots Files
Console 1/A
0 0.81 0.79 0.80 56
1 0.78 0.81 0.79 52
accuracy 0.80 108
IPython console History
LSP Python: ready
conda: base (Python 3.9.12)
Line 1, Col 1
ASCII
CRLF
RW
Mem 89%
Type here to search
ENG
09:33 AM
19-11-2022


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\DELL\Desktop\ecg analysis - 18.11.22\Heart-Disease-Classifier-Web-App-master
...rs\DELL\Desktop\ecg analysis - 18.11.22\Heart-Disease-Classifier-Web-App-master\ecgmulticlasifier.py
ecg.py
ecgmulticlasifier.py
heart_disease_app.py
model.py
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
df = pd.read_csv(r"C:\Users\DELL\Desktop\heart disease/Heart_Disease_Prediction.csv")
df.dtypes
df.head()
df.isnull().sum()
format(len(df[df.duplicated()]))
name = df.columns
num_var = ['Age', 'BP', 'Cholesterol', 'Max HR', 'Heart Disease']
cat_var = [item for item in name if item not in num_var]
num_var_data = df[df.columns & num_var]
num_var_data.describe()
num_var_data.corr()
sns.heatmap(num_var_data.corr(), cmap="YlGnBu", annot=True)
sns.pairplot(num_var_data)
num_var_data[num_var_data['Cholesterol'] > 500]
sns.pairplot(num_var_data, hue = 'Heart Disease')
x = df.drop(['Heart Disease'], axis = 1)
y = df['Heart Disease']
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.20)
model = LogisticRegression()
model.fit(X_train, y_train)
Help Variable Explorer Plots Files
Console 1/A
KeyboardInterrupt
In [4]:
IPython console History
LSP Python: ready
conda: base (Python 3.9.12)
Line 1, Col 1
UTF-8-GUESSED
CRLF
RW
Mem 84%
Type here to search
ENG
09:35 AM
19-11-2022


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
ecg.py
ecgmulticlasifier.py
heart_disease_app.py
model.py
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
df = pd.read_csv(r'C:\Users\DELL\Desktop\heart disease/Heart_Disease_Prediction.csv')
df.dtypes
df.head()
df.isnull().sum()
format(len(df[df.duplicated()]))
name = df.columns
num_var = ['Age', 'BP', 'Cholesterol', 'Max HR', 'Heart Disease']
cat_var = [item for item in name if item not in num_var]
num_var_data = df[df.columns & num_var]
num_var_data.describe()
num_var_data.corr()
sns.heatmap(num_var_data.corr(), cmap="YlGnBu", annot=True)
sns.pairplot(num_var_data)
num_var_data[num_var_data['Cholesterol'] > 500]
sns.pairplot(num_var_data, hue = 'Heart Disease')
x = df.drop(['Heart Disease'], axis = 1)
y = df['Heart Disease']
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.20)
model = LogisticRegression()
model.fit(X_train, y_train)
Age
BP
Cholesterol
Max HR
Console 1/A
KeyboardInterrupt
In [4]:
IPython console
History
LSP Python: ready
conda: base (Python 3.9.12)
Line 1, Col 1
UTF-8-GUESSED
CRLF
RW
Mem 84%
Type here to search
ENG
09:36 AM
19-11-2022


(4) WhatsApp
Performance Testing
IBM
Performance Testing
Performance Testing
Heart Disease Test
127.0.0.1:5000
Heart Disease Test Form
Age
70
Sex
Female
Chest Pain Type
Asymptomatic
Resting Blood Pressure in mm Hg
130
Serum Cholestoral in mg/dl
320
Fasting Blood Sugar > 120 mg/dl
False
Resting ECG Results
Having ST-T wave abnormal
Maximum Heart Rate
109
ST Depression Induced
2.4
Exercise Induced Angina
No
Slope of the Peak Exercise ST Segment
Flat
Number of Vessels Colored by Flourosopy
3
Thalassemia
Reversable defect
Result
Type here to search
ENG
09:42 AM
19-11-2022



10.Advantages and Disadvantages:

Advantages:

- ✓ Easy to predict the heart disease
- ✓ No cost
- ✓ Easy to use

Disadvantages:

- ✓ Sometimes server problem may occurs
- ✓ No doctor Consultancy

11. Conclusion:

Heart is one of the essential and vital organ of human body and prediction about heart diseases is also important concern for the human beings so that the accuracy for algorithm is one of parameter for analysis of performance of algorithms. Accuracy of the algorithms in machine learning depends upon the dataset that used for training and testing purpose. When we perform the analysis of algorithms on the basis of dataset whose attributes are shown and on the basis of confusion matrix, we find KNN is best one.

12.Future Scope:

For the Future Scope more machine learning approach will be used for best analysis of the heart diseases and for earlier prediction of diseases so that the rate of the death cases can be minimized by the awareness about the diseases.

13.Appendix:

13.1 coding

#app.py

```
# -*- coding: utf-8 -*-

import numpy as np
import pickle
from flask import Flask, request, render_template

# Load ML model
model = pickle.load(open('model.pkl', 'rb'))

# Create application
app = Flask(__name__)

# Bind home function to URL
@app.route('/')
def home():
    return render_template('Heart Disease Classifier.html')

# Bind predict function to URL
@app.route('/predict', methods =['POST'])
def predict():

    # Put all form entries values in a list
    features = [float(i) for i in request.form.values()]
    # Convert features to array
    array_features = [np.array(features)]
    # Predict features
    prediction = model.predict(array_features)

    output = prediction

    # Check the output values and retrive the result with html tag based on the value
    if output == 1:
        return render_template('Heart Disease Classifier.html',
                               result = 'The patient is not likely to have heart disease!')
    else:
        return render_template('Heart Disease Classifier.html',
                               result = 'The patient is likely to have heart disease!')

if __name__ == '__main__':
#Run the application
    app.run()
```

Classifier.py

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
sns.set_style('whitegrid')
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
data = pd.read_csv(r'C:\Users\DELL\Desktop\heart disease/Heart_Disease_Prediction.csv')
data.head()
data.info()
data.describe(include = 'all')
data.isnull().sum()
data.nunique()
data.columns
colm = ['Sex', 'Chest pain type','FBS over 120','EKG results','Exercise angina','Slope of ST', 'Number of vessels
fluro', 'Thallium', 'Heart Disease']
for col in colm:
  sns.countplot(data[col])
  plt.show()
  plt.figure(figsize=(12,10))
corr = data.corr()
sns.heatmap(corr, annot = True, linewidths= 0.2, linecolor= 'black', cmap = 'afmhot')
data.columns
X = data[['Age', 'Sex', 'Chest pain type', 'BP', 'Cholesterol', 'FBS over 120',
       'EKG results', 'Max HR', 'Exercise angina', 'ST depression',
       'Slope of ST', 'Number of vessels fluro', 'Thallium']]
y = data['Heart Disease']
print(X.shape,y.shape)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=42529)
print(X_train.shape, X_test.shape, y_train.shape, y_test.shape)
train_convert = {"Absence":0,"Presence":1}
y_train = y_train.replace(train_convert)
test_convert = {"Absence":0,"Presence":1}
y_test = y_test.replace(test_convert)
mms = MinMaxScaler()
X_train = mms.fit_transform(X_train)
X_test = mms.fit_transform(X_test)
rf = RandomForestClassifier()
rf.fit(X_train,y_train)
pred = rf.predict(X_test)
cm = confusion_matrix(y_test,pred)
print(classification_report(y_test,pred))
sns.heatmap(cm, annot = True, fmt = 'g', cbar = False, cmap = 'icefire', linewidths= 0.5, linecolor= 'grey')
plt.title('Confusion Matrix')
plt.ylabel('Actal Values')
plt.xlabel('Predicted Values')
print("Accuracy Score = {}".format(round(accuracy_score(y_test,pred),5)))
```

#multi classifier

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
df = pd.read_csv(r'C:\Users\DELL\Desktop\heart disease/Heart_Disease_Prediction.csv')
df.dtypes
df.head()
df.isnull().sum()
format(len(df[df.duplicated()]))
name = df.columns
num_var = ['Age', 'BP', 'Cholesterol', 'Max HR', 'Heart Disease']
cat_var = [item for item in name if item not in num_var]

num_var_data = df[df.columns & num_var]
num_var_data.describe()
num_var_data.corr()
sns.heatmap(num_var_data.corr(), cmap="YlGnBu", annot=True)
sns.pairplot(num_var_data)
num_var_data[num_var_data['Cholesterol'] > 500]
sns.pairplot(num_var_data, hue = 'Heart Disease')

x = df.drop(['Heart Disease'], axis = 1)
y = df['Heart Disease']
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.20)
model = LogisticRegression()
model.fit(X_train, y_train)
r_sq = model.score(x, y)
print(f"coefficient of determination: {r_sq}")
from sklearn.preprocessing import LabelEncoder
x_train_enc = X_train

le = LabelEncoder()
le.fit(y_train)
y_train_enc = le.transform(y_train)


from sklearn.inspection import permutation_importance


model.fit(x_train_enc, y_train_enc)

results = permutation_importance(model, x_train_enc, y_train_enc, scoring='neg_mean_squared_error')


importance = results.importances_mean


for i,v in enumerate(importance):
    print('Feature: %0d, Score: %.5f' % (i,v))
    # plot feature importance
plt.bar([x for x in range(len(importance))], importance)
```

```
plt.show()
df.columns
selected_feature = ['Sex','Max HR', 'Number of vessels fluro', 'Thallium']
print(selected_feature)
data = df[df.columns & selected_feature]

X_train, X_test, y_train, y_test = train_test_split(data, y, test_size=0.33)
model = LogisticRegression()
model.fit(X_train, y_train)
r_sq = model.score(data, y)
print(f"coefficient of determination: {r_sq}")

models = []
models.append(('LR', LogisticRegression()))
models.append(('LDA', LinearDiscriminantAnalysis()))
models.append(('KNN', KNeighborsClassifier()))
models.append(('CART', DecisionTreeClassifier()))
models.append(('NB', GaussianNB()))
models.append(('SVM', SVC()))

results = []
names = []
scoring = 'accuracy'

for name, model in models:
    kfold = KFold(n_splits=10, random_state=7, shuffle = True)
    cv_results = cross_val_score(model, data, y, cv=kfold, scoring=scoring)
    results.append(cv_results)
    names.append(name)
    msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
    print(msg)

fig = plt.figure()
fig.suptitle('Algorithm Comparison')
ax = fig.add_subplot(111)
plt.boxplot(results)
ax.set_xticklabels(names)
plt.show()
```

```
# -*- coding: utf-8 -*-

"""
Created on Fri Nov 18 12:39:46 2022

@author: DELL
"""

import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as py
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
import pickle

df=pd.read_csv("Heart_Disease_Prediction.csv")
x=df.iloc[:,:-1].values
y=df.iloc[:,-1].values
```

```
std=StandardScaler()
x=std.fit_transform(x)

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)

model=RandomForestClassifier()
model.fit(x_train,y_train)
predictions=model.predict(x_test)
accuracy = accuracy_score(y_test,predictions)

def predict_heart_disease(parameter_list):
    return model.predict(parameter_list)[0]

pickle.dump(model, open('model.pkl','wb'))
```

13.2.Github Link:

https://github.com/IBM-EPBL/IBM-Project-40392-1660628902

Demo Vedio Link:

https://icecreamapps.com/v/d2nfvb2