

PREDICTION OF HEART DISEASE USING MACHINE LEARNING TECHNIQUES

D. Sridevi¹, B. Sathyanarayanan², J. Sridharan³, S.R. Srimukunth⁴, S. Harshvardhan⁵

¹D. Sridevi Information Technology & SRM Valliammai Engineering college

²B. Sathyanarayanan Information Technology & SRM Valliammai Engineering college

³J. Sridharan Information Technology & SRM Valliammai Engineering college

⁴S.R. Srimukunth Information Technology & SRM Valliammai Engineering college

⁵S. Harshvardhan Information Technology & SRM Valliammai Engineering college

Abstract - The health care diligence collect huge quantities of data that contain some retired information, which is useful for making effective opinions. For furnishing applicable results and making effective opinions on data, some advanced data mining ways are used. In this study, a vaticination of Heart Disease is Using Machine literacy(PHDML) is developed using Random timber and Decision Tree algorithms for prognosticating the threat position of a heart complaint. The system uses 11 medical parameters similar as age, coitus, blood pressure, cholesterol, and rotundity for vaticination. The PHDML predicts the liability of cases getting a heart complaint. The opinion of a heart complaint is a grueling task, which can offer automated vaticination about the heart condition of a case so that farther treatment can be made effective. It enables significant knowledge.E.g. connections between medical factors related to a heart complaint and patterns, to be established. The attained results have illustrated that the designed individual system can effectively prognosticate the threat position of heart conditions. The major thing of this study is to ameliorate on former work by developing a new and unique fashion for creating the model, as well as to make the model applicable and easy to use in real- world situations.

Key Words: Machine learning, heart disease prediction, feature selection, prediction model, classification algorithms, cardiovascular disease (CVD).

1.INTRODUCTION

Among all fatal complaint, heart attacks conditions are considered as the most current. Medical interpreters conduct different checks on heart conditions and gather information of heart cases, their symptoms and complaint progression. Decreasingly are reported about cases with common conditions who have typical symptoms. In this presto moving world people want to live a veritably luxurious life so they work like a machine in order to earn lot of plutocrat and live a comfortable life thus in this race they forget to take care of themselves, because of this there food habits change their

entire life change, in this type of life they're more tensed they've blood pressure, sugar at a veritably youthful age and they don't give enough rest for themselves and eat what they get and they indeed don't bother about the quality of the food if sick the go for their own drug as a result of all these small negligence it leads to a major trouble that's the heart complaint The term ' heart complaint ' includes the different conditions that affect heart. The number of people suffering from heart complaint is on the rise(health motifs, 2010). The report from world health association shows us a large number of people that die every time due to the heart complaint each over the world. Heart complaint is also stated as one of the topmost killers in Africa. colorful machine literacy practices are employed in heart complaint vaticination. mortal heart is the top part of the mortal body. principally, it regulates blood inflow throughout our body. Any irregularity to heart can beget torture in other corridor of body. Any kind of disturbance to normal functioning of the heart can be classified as a Heart complaint. In moment's contemporary world, heart complaint is one of the primary reasons for circumstance of utmost deaths. Heart complaint may do due to unhealthy life, smoking, alcohol and high input of fat which may beget hypertension. According to the World Health Organization more than 10 million die due to Heart conditions every single time around the world. A healthy life and foremost discovery are only ways to help the heart affiliated conditions. The main challenge in moment's healthcare is provision of stylish quality services and effective accurate opinion. Indeed if heart conditions are set up as the high source of death in the world in recent times, they're also the bones that can be controlled and managed effectively. The whole delicacy in operation of a complaint lies on the proper time of discovery of that complaint. The proposed work makes an attempt to descry these heart conditions at early stage to avoid disastrous consequences. Records of large set of medical data created by medical experts are available for analysing and rooting precious knowledge from it. Data booby-trapping ways are the means of rooting precious and retired information from the large quantum of data available. substantially the medical database consists of separate information. Hence, decision timber using separate data becomes complex and tough task. Machine literacy(ML) which is subfield of data mining

handles large scale well- formatted dataset efficiently. In the medical field, machine literacy can be used for opinion, discovery and vaticination of colorful conditions. The main thing of this design is to give a tool for croakers to descry heart complaint as early stage. This in turn will help to give effective treatment to cases and avoid severe consequences. Machine literacy plays a veritably important part to descry the retired separate patterns and thereby assay the given data. After analysis of data ML ways help in heart complaint vaticination and early opinion. This design presents performance analysis of colorful ML ways similar as Naive Bayes, Decision Tree, Logistic Retrogression and Random Forest for prognosticating heart complaint at an early stage.

2. OBJECTIVES

2.1. MAIN OBJECTIVES

The main ideal of this exploration is to develop a heart vaticination system. The system can discover and prize retired knowledge associated with conditions from a literal heart data set. Heart complaint vaticination system aims to exploit data mining ways on medical data set to help in the vaticination of the heart conditions.

2.2. SPECIFIC OBJECTIVES

Provides new approach to concealed patterns in the data. Helps avoid mortal biasness. To apply Naïve Bayes Classifier that classifies the complaint as per the input of the stoner. Reduce the cost of medical tests.

2.3. GOAL

prognosticate whether a case should be diagnosed with Heart Disease. This is a double outgrowth. Positive(+) = 1, patient diagnosed with Heart Disease. Negative(-) = 0, case not diagnosed with Heart Disease. trial with colorful Bracket Models & see which yields topmost delicacy. Examine trends & correlations within our data. Determine which features are most important to Positive/ Negative Heart complaint opinion. Use of the stylish Algorithm possible to get that delicacy and prognosticate the heart complaint at an early stage before it's too late.

3. CHALLENGES

Integration of clinical decision support with computer-grounded case records so that could reduce medical crimes, enhance patient safety, drop unwanted practice variation, and ameliorate patient outgrowth. Medical opinion is considered as a significant yet intricate task that needs to be carried out precisely and efficiently. The robotization of the same would be largely salutary. Clinical opinions are frequently made grounded on croaker 's suspicion and experience rather than

on the knowledge rich data hidden in the database. This practice leads to unwanted impulses, crimes and inordinate medical costs which affects the quality of service handed to cases. Machine literacy have the eventuality to induce a knowledge-rich terrain which can help to significantly ameliorate the quality of clinical opinions.

4. OVERVIEW OF METHOD AND RESULTS

4.1. IMPORT DATASETS AND LIBRARIES

In python, importing dataset and libraries were implemented by methods of Important of libraries, Import the libraries, Importing the Dataset Important of Libraries, here we use three main concepts. That they are Numpy: Package for scientific computing with python, MATPLOTLIB: Used for graphing, PANDAS: Data structures and Data Analysis Tool Import the libraries, by using 1.import numpy as np 2.import matplotlib.pyplot as plt 3.import pandas as pd as is used as shorthand to access the imported library Importing the dataset, we want to use the pandas library to import the dataset from a CSV file, using the .read_csv() function: dataset = pd.read_csv ('Data.csv') 1. Now we need to be able to distinguish between the data we just imported, the independent variables (matrix of feature's) and the dependent variable. To do this we need to tell python which columns are independent and which are dependent. Note: In python the index for columns and rows starts at zero. 2. To do this we take our variable dataset and use the .iloc[] method from pandas. The method takes two inputs (or 4). 3. iloc[:,] : The first input is the rows we wish to use. The : simply implies that we want to use all the rows. .iloc[rows, columns start : columns end but not including] : The left side takes columns and this part gets a little confusing so here are examples to explain: 1. iloc[rows, :-1] : All columns except the last one 2. iloc[rows, 2:13] : All columns from 2 until 12. 13 is NOT INCLUDED 4. we then use the .values function to retrieve the values from the column. Eg: X= dataset.iloc[:, :-1].values Y = dataset.iloc[:, 3]

4.2. DATA PREPROCESSING

Pre-processing refers to the changeovers applied to our data before feeding it to the algorithm. DataPre- processing is a fashion that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it's collected in raw format which is not realizable for the analysis. For achieving better results from the applied model in Machine knowledge projects the format of the data has to be in a proper manner. Some 16 specified Machine Learning model needs information in a specified format, for illustration, Random Forest algorithm does not

support null values, therefore to execute arbitrary timber algorithm null values have to be managed from the original raw data set. Another aspect is that the data set should be formatted in such a way that further than one Machine knowledge and Deep knowledge algorithm are executed in one data set, and swish out of them is chosen. preprocessing was executed in python by using the following styles 1. Importing the libraries 2. Importing the dataset 3. Handling of Missing data 4. Handling of Categorical data 5. ramifying the dataset into training set and testing set 6. point scaling We have formerly seen about the 1 and 2 module in importing dataset and libraries, now we have to know about, Handling of missing Data Well the first idea is to remove the lines in the obediences where there is some missing data. But that can be fairly dangerous because imagine this data set contains vital information. It would be fairly dangerous to remove an observation

4.3. FEATURE SCALING

Consider the house price prediction dataset as an example of a real-world dataset with many features and a broad range of values. Numerous features, such as the number of bedrooms and the house's square footage, will be included. As you would have guessed, there will be between 1 and 5 bedrooms, but the square footage will range from 500 to 2000. The range of the two traits is drastically different in this way. Numerous machine learning algorithms that use Euclidean distance as a metric to evaluate similarity will fall short of giving a decent recognition to the smaller characteristic, in this case, the number of bedrooms, which in the real world can turn out to be an actually significant metric. Linear Regression, Logistic Regression, and KNN are a few examples. There are numerous approaches to feature scaling. I'll go through the top five most widely utilised feature scaling strategies. 1. Maximum Absolute Scaling 2. Minimum-Maximum Scaling 3. Standardization 4. Normalization 5. Stable Scaling. Maximum Absolute Scaling 1. Determine the feature's absolute maximum value in the dataset. 2. Divide all of the values in the column by the highest value. Minimum and Maximum Scaling In min-max, you remove the dataset's minimal value from all of the values and divide the result by the dataset's range (maximum-minimum). In this situation, your dataset will always be between 0 and 1, when it was before between -1 and +1. Again, this method is prone to outliers. Normalization Instead of utilising the min() value like in the previous situation, we will use the average() value in this case. Scaling modifies the range of your data, whereas normalisation modifies the shape of your data's distribution. Standardization We generate the z-value for each data point and replace them with these values during normalisation. This ensures that all of the characteristics are centred around the mean value and have a standard deviation of one. This is the ideal option if your feature, such as income or age, is regularly distributed. Scaling Capability In this procedure, you must

remove all of the data points from the median value and then divide the result by the Inter Quartile Range (IQR). The IQR is the difference between the 25th and 50th percentile points. This approach centres the median value at zero and is resistant to outliers.

4.4. SPLITTING DATASET INTO TRAINING DATASET AND TESTING DATASET

The train test split() method in the scikit-learn Python machine learning package implements the train-test split evaluation technique. The function accepts a loaded dataset as input and returns it divided into two subgroups. # divided into train and test sets test, train = train test split (dataset, ...) Ideally, you can separate your original dataset into input (X) and output (y) columns, then send both arrays to the method, which will split them suitably into train and test sections. ... # divided into train and test sets train test split = X train, X test, Y train, Y test (X, y, ...) The size of the split may be provided using the "test size" option, which accepts either a number of rows (integer) or a percentage (float) of the dataset size between 0 and 1. The latter is the most popular, with numbers such as 0.33 being used to allocate 33 percent of the dataset to the test set and 67 percent to the training set. # divided into train and test sets train test split(X, y, test size=0.33) X train, X test, y train, y test We can illustrate this with a 1,000-example synthetic classification dataset. The entire example is provided below. # divide a dataset from sklearn.datasets into train and test sets make blobs should be imported from sklearn.model selection. train test split import # create dataset X and y = make blobs(n samples=1000) X train, X test, y train, y test =train test split (X,y,test size=0.33) print(X train.shape, Y train.shape, Y test.shape) When the example is run, it divides the dataset into train and test sets and outputs the size of the new dataset. We can see that 670 instances (67%) were assigned to the training set and 330 examples (33%) were assigned to the test set, as required. (670, 2) (330, 2) (670,) (330,) The dataset may also be divided by giving the "train size" option, which can be either a number of rows (integer) or a percentage of the original dataset between 0 and 1, for as 0.67 for 67 percent. # train test split(X, y, train size=0.67) split into train test sets X train, X test, y train, y test = train test split(X, y, train size=0.67)

4.5. FEATURE EXTRACTION

The process of translating raw data into numerical features that may be processed while retaining the information in the original data set is referred to as feature extraction. It produces better outcomes than applying machine learning straight to raw data. Feature extraction can be done manually or

automatically: Manual feature extraction entails defining and specifying the characteristics that are important to a specific situation, as well as designing a method to extract such features. In many cases, having a thorough grasp of the backdrop or domain may assist make educated judgments about which features can be valuable. Engineers and scientists have spent decades developing feature extraction algorithms for photos, signals, and text. The mean of a window in a signal is an example of a basic characteristic.

AGE	Numeric [29 to 77;unique=41;mean=54.4;median=56]
SEX	Numeric [0 to 1;unique=2;mean=0.68;median=1]
CP	Numeric [1 to 4;unique=4;mean=3.16;median=3]
TESTBPS	Numeric [94 to 200;unique=50;mean=131.69;median=130]
CHOL	Numeric [126 to 564;unique=152;mean=246.69;median=241]
FBS	Numeric [0 to 1;unique=2;mean=0.15;median=0]
RESTECG	Numeric [0 to 2;unique=3;mean=0.99;median=1]
THALACH	Numeric [71 to 202;unique=91;mean=149.61;median=153]
EXANG	Numeric [0 to 1;unique=2;mean=0.33;median=0.00]
OLPEAK	Numeric [0 to 6.20;unique=40;mean=1.04;median=0.80]
SLOPE	Numeric [1 to 3;unique=3;mean=1.60;median=2]
CA	Categorical [5 levels]
THAL	Categorical [4 levels]
TARGET	Numeric [0.00 to 4.00;unique=5;mean=0.94;median=0.00]

Table -1: UCI dataset range and datatype

Attribute	Description	Type
Age	Patient's age in completed years	Numeric
Sex	Patient's Gender (male represented as 1 and female as 0)	Nominal
Cp	The type of Chest pain categorized into 4 values: 1. typical angina, 2. atypical angina, 3. non-anginal pain and 4. asymptomatic	Nominal
Trestbps	Level of blood pressure at resting mode (in mm/Hg at the time of admitting in the hospital)	Numeric
Chol	Serum cholesterol in mg/dl	Numeric
FBS	Blood sugar levels on fasting > 120 mg/dl; represented as 1 in case of true, and 0 in case of false	Nominal
Resting	Results of electrocardiogram while at rest are represented in 3 distinct values: Normal state is represented as Value 0, Abnormality in ST-T wave as Value 1, (which may include inversions of T-wave and/or depression or elevation of ST of > 0.05 mV) and any probability or certainty of LV hypertrophy by Estes' criteria as Value 2	Nominal
Thali	The accomplishment of the maximum rate of heart	Numeric
Exang	Angina induced by exercise. (0 depicting 'no' and 1 depicting 'yes')	Nominal
Oldpeak	Exercise-induced ST depression in comparison with the state of rest	Numeric
Slope	ST segment measured in terms of the slope during peak exercise depicted in three values: 1. unsloping, 2. flat and 3. downsloping	Nominal
Ca	Fluoroscopy coloured major vessels numbered from 0 to 3	Numeric
Thal	Status of the heart illustrated through three distinctly numbered values. Normal numbered as 3, fixed defect as 6 and reversible defect as 7.	Nominal
Num	Heart disease diagnosis represented in 5 values, with 0 indicating total absence and 1 to 4 representing the presence in different degrees.	Nominal

Table -1: UCI dataset attributes detailed information.

Automated feature extraction uses specialised algorithms or deep networks to extract features from signals or images automatically and without the need for human intervention. This strategy may be useful when you need to go quickly from raw data to machine learning algorithms. Wavelet scattering is used to demonstrate automated feature extraction. With the

emergence of deep learning, the earliest layers of deep networks have effectively replaced feature extraction - but mostly for image data. The first hurdle to overcome before constructing successful prediction models for signal and time-series applications is feature extraction.

4.6. CHECKING ACCURACY SCORE FOR RANDOM FOREST

Random forest is a supervised learning method. It is available in two flavours: one for classification problems and one for regression problems. It is one of the most versatile and user-friendly algorithms available. It builds decision trees from the data samples supplied, gets forecasts from each tree, then votes to select the best alternative. It's also a decent predictor of feature significance. The Random Forest approach combines several decision trees to form a forest of trees, hence the name. The random forest classifier's accuracy increases with the number of trees in the forest. There are two steps to random forest algorithm intuition. We generate the random forest in the first step by randomly selecting "k" features from a total of m features. In the first stage, we do the following: - 1. Choose k features at random from a total of m features, where k m. 2. Using the optimal split point among the k characteristics, compute the node d. 3. Using the best split, divide the node into daughter nodes. 4. Repeat steps 1–3 until the desired number of nodes is attained. 5. Build a forest by repeating steps 1–4 a n times to create a n number of trees. In the second stage, we use the taught random forest algorithm to produce predictions. 1. We take the test characteristics and predict the outcome using the rules of each randomly generated decision tree, and we save the anticipated outcome. 2. We then compute the votes for each anticipated target. 3. Finally, we use the highest voted projected target as the random forest algorithm's final prediction. We receive certain outcomes as floats or numerical digits from the voting mechanism by examining the values of the dataset as continuous and categorical values. The random forest uses a vote categorization approach to acquire various raw data from original resources. This ensemble classifier constructs numerous decision trees and combines them to achieve the best possible outcome. It mostly employs bootstrap aggregation or bagging for tree learning. For a given set of data, $X = x_1, x_2, x_3, \dots, x_n$ with replies $Y = y_1, y_2, y_3, \dots, y_n$, which repeats the bagging from $b = 1$

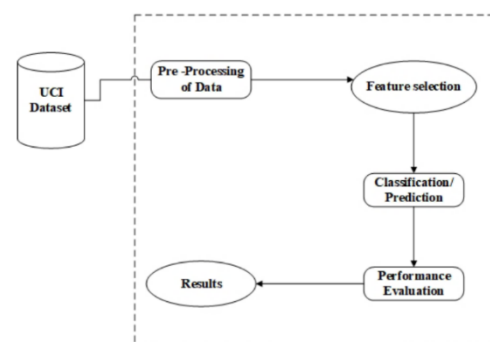


Fig -1: Random forest working structure

4.7. TRAINING MODEL FOR RANDOM FOREST

The final stage in data preparation is to divide the data into training and testing sets. We let the model 'see' the answers, in this case the real temperature, during training so it can learn how to forecast the temperature from the attributes. We anticipate that there will be some link between all of the characteristics and the goal value, and the model's duty throughout training will be to learn this relationship. When it comes time to evaluate the model, we ask it to make predictions on a testing set using only the features (not the responses)! Because we know the real test set answers, we can compare these predictions to the genuine number to see how accurate the model is. When training a model, we often divide the data into training and testing sets at random to obtain a representation of all data points (if we trained on the first nine months of the year and then used the final three months for prediction, our algorithm would not perform well because it has not seen any data from those last three months.) For repeatable results, I set the random state to 42, which means the results will be the same each time I run the split. With another single line, the following code divides the data sets: # Using Skicit-learn to split data into training and testing sets from sklearn.model_selection import train_test_split # Split the data into training and testing sets train_features, test_features, train_labels, test_labels = train_test_split(features, labels, test_size = 0.25, random_state = 42)

We can look at the shape of all the data to make sure we did everything correctly. We expect the training features number of columns to match the testing feature number of columns and the number of rows to match for the respective training and testing features and the labels

```
print('Training Features Shape:', train_features.shape)
print('Training Labels Shape:', train_labels.shape)
print('Testing Features Shape:', test_features.shape)
print('Testing Labels Shape:', test_labels.shape)
```

Training Features Shape: (261, 14)

Training Labels Shape: (261,)

Testing Features Shape: (87, 14)

Testing Labels Shape: (87,)

It looks as if everything is in order! Just to recap, to get the data into a form acceptable for machine learning we:

- 1) One-hot encoded categorical variables
- 2) Split data into features and labels
- 3) Converted to arrays
- 4) Split data into training and testing sets

Depending on the underlying data set, further effort such as eliminating outliers, imputing missing values, or turning temporal variables into cyclical representations may be required. These procedures may appear arbitrary at first, but if you understand the core approach, it will be similar for any machine learning task. It ultimately comes down to converting human-readable data into a format that a machine learning model can understand.

4.8. ESTABLISH BASELINE

We need to create a baseline, a reasonable metric that we aim to beat with our model, before we can make and assess predictions. If our model cannot outperform the baseline, it is a failure, and we should attempt an alternative model or concede that machine learning is not the correct solution for our situation. The historical maximum temperature averages can serve as the baseline projection in our scenario. To put it another way, our baseline is the inaccuracy we would obtain if we just projected the average maximum temperature for all days.

```
# The baseline predictions are the historical averages
baseline_preds = test_features[:, feature_list.index('average')]
# Baseline errors, and display average baseline error
baseline_errors = abs(baseline_preds - test_labels)
print('Average baseline error: ',
      round(np.mean(baseline_errors), 2))
Average baseline error: 5.06 degrees.
```

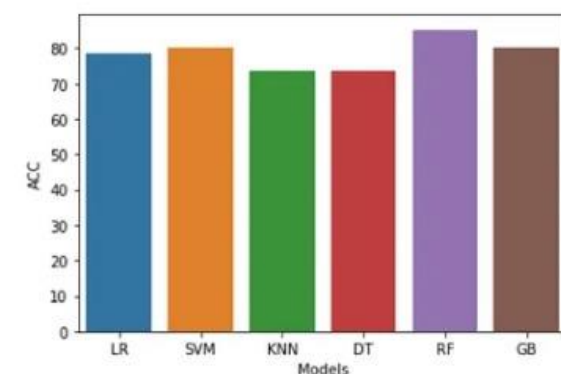
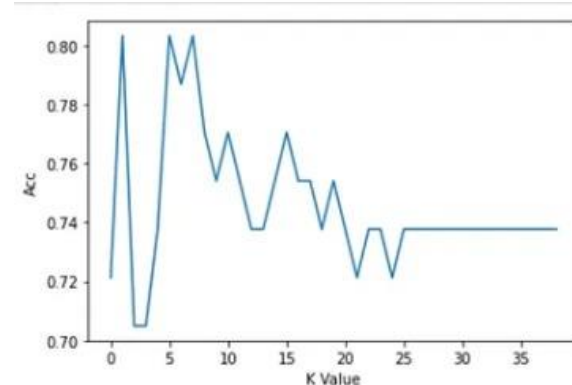
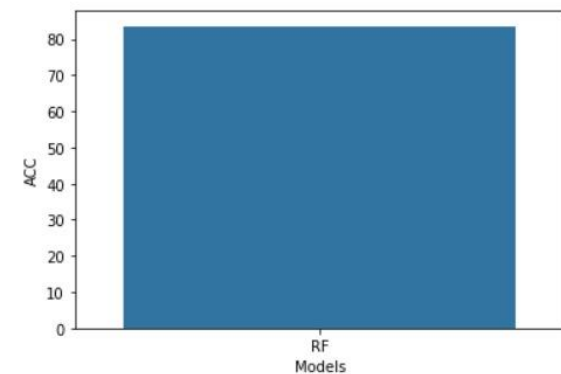


Fig -2: Different Models

5. ARCHITECTURAL DIAGRAM

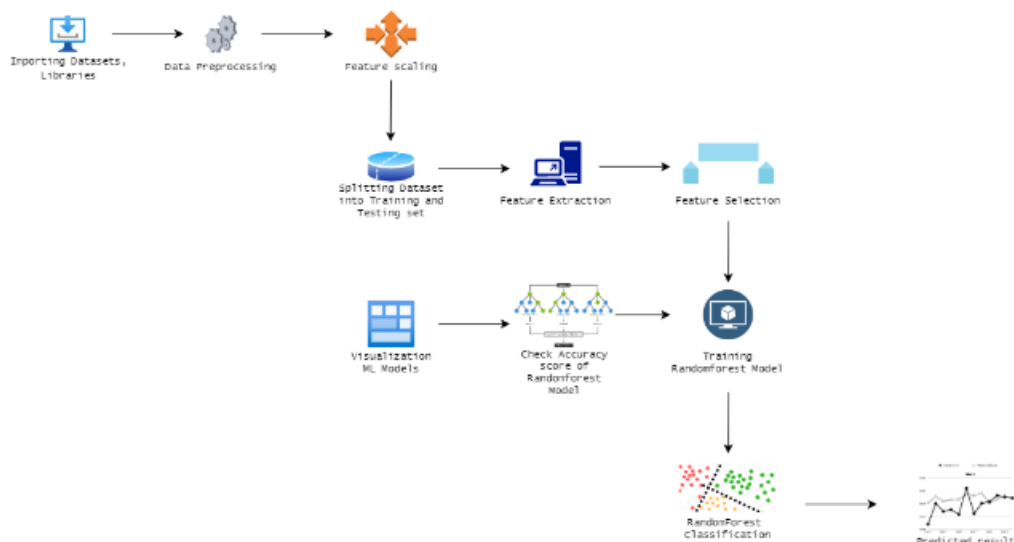


Fig -3: Architectural diagram for RF

6. PERFORMANCE MESAURES

Several standard performance metrics such as accuracy, precision and error in classification have been considered for the computation of performance efficacy of this model. Accuracy in the current context would mean the percentage of instances correctly predicting from among all the available instances. Precision is defined as the percentage of corrective in the positive class of the instances. Classification error is defined as the percentage of accuracy missing or error available in the instances. To identify the significant features of heart disease, three performance metrics are used which will help in better understanding the behavior of the various combinations of the feature-selection. ML technique focuses on the best performing model compared to the existing models. We introduce HRFLM, which produces high accuracy and less classification error in the prediction of heart disease. The performance of every classifier is evaluated individually and all results are adequately recorded for further investigation.

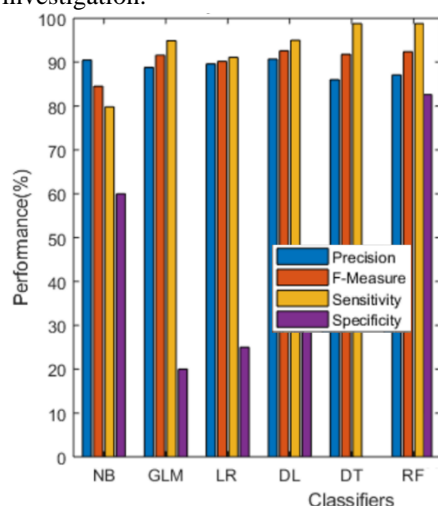


Fig -4 : Random forest working structure

7. EXPERIMENTAL ENVIRONMENT DATASETS

Heart disease data was collected from the UCI machine learning repository. There are four databases (i.e. Cleveland, Hungary, Switzerland, and the VA Long Beach). The Cleveland database was selected for this research because it is a commonly used database for ML researchers with comprehensive and complete records. The dataset contains 303 records. Although the Cleveland dataset has 76 attributes, the data set provided in the repository furnishes information for a subset of only 14 attributes. The data source of the Cleveland dataset is the Cleveland Clinic Foundation. Table 1 depicts the description and type of attributes. There are 13 attributes that feature in the prediction of heart disease, where only one attribute serves as the output or the predicted attribute to the presence of heart disease in a patient. The Cleveland dataset contains an attribute named num to show the diagnosis of heart disease in patients on different scales, from 0 to 4. In this scenario, 0 represents the absence of heart disease and all the values from 1 to 4 represent patients with heart disease, where the scaling refers to the severity of the disease (4 being the highest). Figure 1 shows the distribution of the num attribute among the identified 303 records.

Data Split	Overall error rate			Best Model	Overall classification error rate			Best Model
	DT	RF	LM		DT	RF	LM	
1	14.9	4	6.7	RF	14.9	14.9	16.2	DT /RF
2	34.9	12.2	22.6	RF	39.6	37.7	38.7	RF
3	50	11.1	16.6	RF	50	27.8	50	RF
4	62.5	20.9	29.2	RF	62.5	54.1	54.2	RF
5	60	13.3	13.3	RF/ LM	60	53.4	53.3	LM
6	54.6	12	18.1	RF	60.6	57.6	54.6	LM
7	57.1	0	28.5	RF	57.1	28.5	42.8	RF
8	36.4	18.2	9.1	LM	36.4	27.3	27.3	RF/ LM

Fig -5: result rate

8. GRAPHICAL USER INTERFACE CREATION

Choose a programming language and GUI toolkit: There are many programming languages and GUI toolkits available, but some popular options for machine learning projects include Python with PyQt, Tkinter, or wxPython.

1. Collect and preprocess data: You'll need to collect data on heart disease and preprocess it for use in your machine learning model. This might involve cleaning the data, transforming it into a suitable format, and selecting the features to use.

2. Train a machine learning model: Next, you'll need to train a machine learning model on your preprocessed data. You might choose to use a supervised learning algorithm, such as logistic regression or decision trees, to predict whether a patient has heart disease or not.

3. Create a GUI interface: Once you have a trained model, you can create a GUI interface that allows users to input patient data and receive a prediction of whether the patient has heart disease. You might include fields for inputting information such as age, sex, blood pressure, cholesterol level, and whether the patient smokes or not.

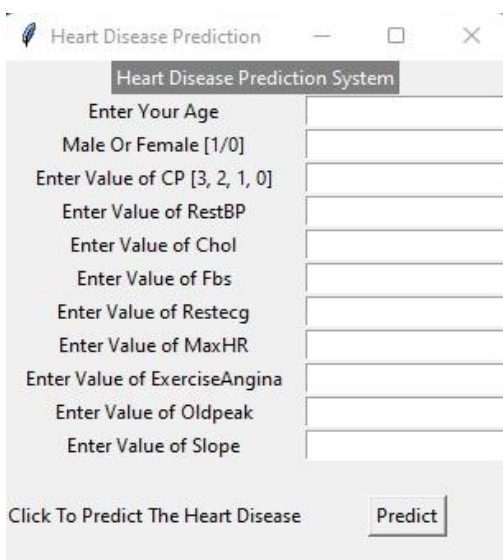


Fig -6: GUI Model 1 output

4. Integrate the machine learning model into the GUI: Finally, you'll need to integrate the machine learning model into the GUI so that it can make predictions based on user input. This might involve writing code that takes the user's input, preprocesses it in the same way as the training data, and then passes it to the machine learning model to generate a prediction.

5. Test and refine: Once you have a working GUI, you should test it thoroughly to ensure that it's accurate and user-friendly. You may need to refine the GUI or the machine learning model based on feedback from users.

Overall, creating a GUI for a heart disease prediction using machine learning project requires a combination of programming skills, data preprocessing expertise, and machine learning knowledge. It's important to approach the project systematically, testing each component thoroughly and refining as needed.

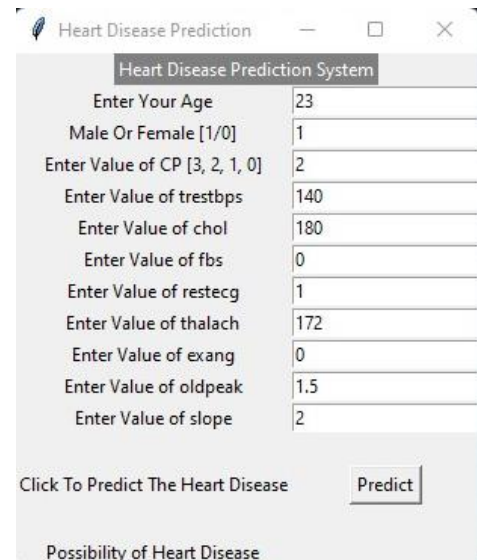


Fig -7: GUI Model 2 output

9. CONCLUSIONS

Identifying the processing of raw healthcare data of heart information will help in the long term saving of human lives and early detection of abnormalities in heart conditions. Machine learning techniques were used in this work to process raw data and provide a new and novel discernment towards heart disease. Heart disease prediction is challenging and very important in the medical field. However, the mortality rate can be drastically controlled if the disease is detected at the early stages and preventative measures are adopted as soon as possible. PHDML proved to be quite accurate in the prediction of heart disease. The future course of this research can be performed with diverse mixtures of machine learning techniques to better prediction techniques. Furthermore, new feature selection methods can be developed to get a broader perception of the significant features to increase the performance of heart disease prediction.

ACKNOWLEDGEMENT

We are grateful to our internal guide Dr.D. Sridevi, MCA, M.E, Ph.D., Assistant Professor (Sr.G) for her invaluable guidance and encouragement, this project would not have been successful.

REFERENCES

1. C. A. Devi, S. P. Rajamhoana, K. Umamaheswari, R. Kiruba, K. Karunya, and R. Deepika, "Analysis of neural networks based heart disease prediction system," in Proc. 11th Int. Conf. Hum. Syst. Interact. (HSI), Gdansk, Poland, Jul. 2018, pp. 233–239.

2. L. Baccour, "Amended fused TOPSIS-VIKOR for classification (ATOVIC) applied to some UCI data sets," *Expert Syst. Appl.*, vol. 99, pp. 115–125, Jun. 2018. doi: 10.1016/j.eswa.2018.01.025.
3. B. S. S. Rathnayakc and G. U. Ganegoda, "Heart diseases prediction with data mining and neural network techniques," in *Proc. 3rd Int. Conf. Conver. Technol. (I2CT)*, Apr. 2018, pp. 1–6.
4. M. S. Amin, Y. K. Chiam, K. D. Varathan, "Identification of significant features and data mining techniques in predicting heart disease," *Telematics Inform.*, vol. 36, pp. 82–93, Mar. 2019. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0736585318308876>
5. T. Vivekanandan and N. C. S. N. Iyengar, "Optimal feature selection using a modified differential evolution algorithm and its effectiveness for prediction of heart disease," *Comput. Biol. Med.*, vol. 90, pp. 125–136, Nov. 2017.
6. S. Radhimeenakshi, "Classification and prediction of heart disease risk using data mining techniques of support vector machine and artificial neural network," in *Proc. 3rd Int. Conf. Comput. Sustain. Global Develop. (INDIACom)*, New Delhi, India, Mar. 2016, pp. 3107–3111.
7. R. Wagh and S. S. Paygude, "CDSS for heart disease prediction using risk factors," *Int. J. Innov. Res. Comput.*, vol. 4, no. 6, pp. 12082–12089, Jun. 2016.
8. O. W. Samuel, G. M. Asogbon, A. K. Sangaiah, P. Fang, and G. Li, "An integrated decision support system based on ANN and Fuzzy_AHP for heart failure risk prediction," *Expert Syst. Appl.*, vol. 68, pp. 163–172, Feb. 2017.
9. S. Zaman and R. Toufiq, "Codon based back propagation neural network approach to classify hypertension gene sequences," in *Proc. Int. Conf. Elect., Comput. Commun. Eng. (ECCE)*, Feb. 2017, pp. 443–446.
10. W. Zhang and J. Han, "Towards heart sound classification without segmentation using convolutional neural network," in *Proc. Comput. Cardiol. (CinC)*, vol. 44, Sep. 2017, pp. 1–4.