A Project Report

Hybrid Technique For Predicting Heart Disease

Submitted in partial fulfillment of the requirements for the award of degree

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

by

K SHALINI (1601-16-733-013)

C SHIVANI (1601-16-733-014)

Department of Computer Science and Engineering,

Chaitanya Bharathi Institute of Technology (Autonomous),

(Affiliated to Osmania University, Hyderabad)
Hyderabad, TELANGANA (INDIA) –500 075

[2019-2020]

CERTIFICATE

This is to certify that the project titled “**Hybrid technique for predicting heart disease**” is the bonafide work carried out by **K Shalini (160116733013) and C Shivani (160116733014)**, a student of B.E.(CSE) of Chaitanya Bharathi Institute of Technology(A), Hyderabad, affiliated to Osmania University, Hyderabad, Telangana(India) during the academic year 2019-2020, submitted in partial fulfillment of the requirements for the award of the degree in **Bachelor of Engineering** (**Computer Science and Engineering**) and that the project has not formed the basis for the award previously of any other degree, diploma, fellowship or any other similar title.

Supervisor	**Head, CSE Dept.**
Smt G.Vanitha,	**Dr M.Swamy Das,**
Assistant Professor, CSE Dept.	**Professor.**

Place: Hyderabad
Date: 29-04-2020

DECLARATION

We hereby declare that the project entitled "**Hybrid technique for predicting heart disease**" submitted for the B.E (CSE) degree is my original work and the project has not formed the basis for the award of any other degree, diploma, fellowship or any other similar titles.

K Shalini

C Shivani

Place: Hyderabad

Date: 29-04-2020

ABSTRACT

Heart disease is one of the biggest causes of morbidity and mortality among the population of the world. It is a deadly disease that a large population of people around the world suffers with. When considering death rates and the large number of people who suffer from heart disease, it is revealed how important early diagnosis of heart disease is. Traditional way of diagnosis is not sufficient for such an illness. Prediction of cardiovascular disease is regarded as one of the most important subjects in the section of clinical data analysis.The amount of data in the healthcare industry is huge. Developing a medical diagnosis system based on machine learning for prediction of heart disease provides more accurate diagnosis than the traditional way. Machine learning (ML) has been shown to be effective in assisting i.e., making decisions and predictions from the large quantity of data produced by the healthcare industry. Various studies give only a glimpse into predicting heart disease with ML techniques. In this paper, we propose a method that aims at finding significant features by applying machine learning techniques resulting in improving the accuracy in the prediction of cardiovascular disease. The prediction model is introduced with different combinations of features and several known classification techniques. We produce an enhanced performance level with high accuracy through the prediction model for heart disease with the different techniques.

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without introducing the people who made it possible and whose constant guidance and encouragement crowns all efforts with success. They have been a guiding light and source of inspiration towards the completion of the project. We would like to express our sincere gratitude and indebtedness to our project guide, **Smt G.Vanitha,** Asst.Professor who has supported us throughout our project with patience and knowledge. We are also thankful to Head of the department **Dr M.Swamy Das** for providing excellent infrastructure and a conducive atmosphere for completing this project successfully. We are also extremely thankful to our Project coordinators **Dr K.Sagar**, Professor and **Smt E. Kalpana**, Asst. Professor, and Dept. of CSE, for their valuable suggestions and interest throughout the course of this project. We convey our heartfelt thanks to the lab staff for allowing us to use the required equipment whenever needed. We sincerely acknowledge and thank all those who gave directly or indirectly their support in completion of this work.

K.Shalini (160116733013)

C.Shivani(160116733014)

LIST OF FIGURES

Figure	Page Number
2.3.2 system architecture of base paper 1	10
3.1 Block Diagram	13
3.3.8 deep learning	30
3.3.9.1 gene	31
3.3.9.2 chromosomes as a list of 13 genes	32
3.3.9.3 population contains different chromosomes	32
4.1 Flow Chart	37
4.1.1.1 Use Case Diagram	38
4.1.1.2 Sequence Diagram	39
5.1 bar plot depicting accuracy of all algorithms	46

LIST OF TABLES

Table	Page Number
Table 5.1 Performance metrics of ML classifiers	46
Table 5.2 Results of various algorithms	47

LIST OF ABBREVIATIONS

CVD	Cardiovascular Diseases
NB	Naive Bayes
LR	Logistic Regression
DT	Decision Tree
SVM	Support Vector Machine
RF	Random Forest
HNB	Hidden Naive bayes
RFE	Recursive Feature Elimination
KNN	k-nearest neighbors
HRLFM	Hybrid Random Forest with Linear Model

TABLE OF CONTENTS

Topic **Page Number**

TITLE PAGE **i**

CERTIFICATE OF GUIDE **ii**

DECLARATION **iii**

ABSTRACT **iv**

ACKNOWLEDGEMENT **v**

LIST OF FIGURES **vi**

LIST OF TABLES **vii**

LIST OF ABBREVIATIONS **viii**

1. INTRODUCTION **1**

1.1 Problem Definition 1

1.2 Methodologies 2

1.3 Outline of the results 4

1.4 Scope of the project 5

1.5 Organization of the report 6

2. LITERATURE SURVEY **7**

2.1 Introduction to the problem domain terminology 7

2.2 Existing system 8

2.3 Related works 9

2.4 Tools 12

3. DESIGN OF THE PROPOSED SYSTEM 13

3.1 Block Diagram 13

3.2 Module Description 13

3.3 Theoretical Foundation/Algorithms 16

4. IMPLEMENTATION OF THE PROPOSED SYSTEM 37

4.1 Flowchart 37

4.1.1 UML Diagrams 37

4.2 Criteria 39

4.3 Algorithms / Pseudo Code 42

4.4 Data Sets description 43

5.RESULTS/OUTPUTS 46

6.CONCLUSION 48

6.1 Conclusion 48

6.2 Future Scope 48

REFERENCES

1. INTRODUCTION

1.1 Problem Definition

The term "heart disease" is often used interchangeably with the term "cardiovascular disease." Cardiovascular disease generally refers to conditions that involve narrowed or blocked blood vessels that can lead to a heart attack, chest pain (angina) or stroke. Heart related diseases or Cardiovascular Diseases (CVDs) are the main reason for a huge number of deaths in the world over the last few decades and has emerged as the most life-threatening disease, not only in India but in the whole world. So, there is a need for a reliable, accurate and feasible system to diagnose such diseases in time for proper treatment. Machine Learning algorithms and techniques have been applied to various medical datasets to automate the analysis of large and complex data.

Many researchers, in recent times, have been using several machine learning techniques to help the healthcare industry and the professionals in the diagnosis of heart related diseases. It is difficult to identify heart disease because of several contributory risk factors such as diabetes, high blood pressure, high cholesterol, abnormal pulse rate and many other factors. Various techniques in data mining and neural networks have been employed to find out the severity of heart disease among humans. .The nature of heart disease is complex and hence, the disease must be handled carefully. Not doing so may affect the heart or cause premature death.

Heart disease is predicted based on symptoms namely, pulse rate, sex, age, and many others. The main objective of this research is to improve the performance accuracy of heart disease prediction by combining two models which gives better accuracy. This paper presents comparative results of various models based on machine learning and deep learning algorithms and techniques and analyzes their performance. Models based on supervised learning algorithms such as SVM, Naive

Bayes, Decision Trees, Random Forest and ensemble models are found very popular among the researchers. Along with that we are also implementing deep learning and genetic algorithms. We implement the idea of hybrid technique i.e, combination of machine learning techniques using weighted average.

1.2 Methodologies

There are a lot of techniques used in heart disease prediction. The authors suggest using the following procedure to process the prediction accurately**.**

1. Import dataset from Kaggle or UCI
2. Exploratory Data analysis.
3. Preprocessing.
4. Feature extraction using RFE
5. Apply different classifiers on dataset
6. Calculate accuracy and classification error rate for different models
7. Combine models with low classification error rate using Hybrid method
8. Build a deep learning model and apply genetic algorithms also.
9. Comparison of results
10. Visualization

1.2.1 Data Pre-processing

Heart disease data is pre-processed after collection of various records. We are using two datasets Cleveland and Framingham datasets from kaggle. Cleveland dataset contains a total of 303 patient records, where 6 records are with some missing values.

Framingham dataset contains a total of 4240 patient records, where 645 records are with some missing values. Those 6 records and 645 records have been replaced with the mean of corresponding columns in the dataset.

The multiclass variable and binary classification are introduced for the attributes of the given dataset. The multi-class variable is used to check the presence or absence of heart disease. In the for instance if the patient has heart disease, the value is set to 1, else the value is set to 0 indicating the absence of heart disease in the patient. The pre-processing of data is carried out by converting medical records into diagnosis values.

The results of data pre-processing for 303 patient records indicate that 139 records show the value of 1 establishing the presence of heart disease while the remaining 164 reflected the value of 0 indicating the absence of heart disease.In case of framingham dataset , 3596 records show the value of 0 and 644 records show value of 1.

1.2.2 Feature selection and reduction [5]

From among the 13 attributes of the data set, two attributes pertaining to age and sex are used to identify the personal information of the patient. The remaining 11 attributes are considered important as they contain vital clinical records. Clinical records are vital to diagnosis and learning the severity of heart disease. As previously mentioned in this experiment, several(ML) techniques are used. The experiment was repeated with all the ML techniques using Recursive Feature Elimination (RFE) method as a feature selection approach.It works by recursively removing attributes and building a model on those attributes that remain. It uses the model accuracy to identify which attributes (and combination of attributes) contribute the most to predicting the target attribute.

1.2.3 Classification modelling

Different ML classifiers are applied to the dataset in order to estimate its performance. The best performing models are identified from the above results based on their low rate of error. The models with low error rate are combined using hybrid method using probabilities and weighted average.

1.2.4 Performance measures

Several standard performance metrics such as accuracy, precision, specificity, sensitivity, f1 score and error in classification have been considered for the computation of performance efficiency 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 prediction in the positive class of the instances. Sensitivity is used to determine the proportion of actual positive cases, which got predicted correctly, Specificity is used to determine the proportion of actual negative cases, which got predicted correctly. The F1 score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. Classification error is defined as the percentage of accuracy missing or error available in the instances. To identify the significant features of heart disease, these 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. The performance of every classifier is evaluated individually and all results are adequately recorded for further investigation.

1.3 Outline of the results

In the first step, the UCI dataset is loaded and the data becomes ready for pre-processing. The subset of 13 attributes is selected from the pre-processed data set of heart disease. The machine learning models ,deep learning models and genetic algorithms for heart disease prediction are used to develop the classification. The evaluation of the model is performed with the confusion matrix. Totally, four outcomes are generated by a confusion matrix, namely TP (True Positive), TN (True Negative), FP (False Positive) and FN (False Negative). The following measures are used for the calculation of the accuracy, sensitivity, specificity.

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

Specificity = (TN) / (TN + FP)

Recall or Sensitivity = (TP)/(TP+FN)

Precision = (TP)/(TP+FP)

F1_score = (2*Precision*Recall)/(Precision+Recall)

In this paper, comparison of various machine learning methods is done for predicting the accurate risk of heart disease of the patients from their medical data.

1.4 Scope of the project

This project implements the various machine learning algorithms which were applied to the data set. It utilizes the data such as blood pressure, cholesterol, diabetes and then tries to predict. This may help in taking preventive measures and hence try to avoid the possibility of heart disease for the patient. So when a patient is predicted as positive for heart disease, then the medical data for the patient can be closely analyzed by the doctors. An example would be - suppose the patient has diabetes which may be the cause for heart disease in future and then the patient can be given treatment to have diabetes in control which in turn may prevent the heart disease. Also, the ensemble methods, deep learning and genetic algorithms are applied to the dataset along with hybrid method . The results are compared

1.5 Organization of the report

This introduction section is followed by the Literature Survey. The literature survey explains the current existing systems. It also introduces domain specific terminology which forms the background to understand this project. It discusses in depth about some existing solutions’ core aspect which also forms the basis for many other solutions. The section also discusses the drawbacks in all the solutions exhaustively. The literature survey section is followed by Design of the Proposed System section. This section discusses the evolution and design of the proposed solution. Next section discusses the implementation of the design discussed in the previous section. The Data Flow Diagrams and Flowcharts are discussed in this section of the project. The algorithm is also discussed in this section. The data set being used, the features of the data set, and their significance are mentioned. The testing process is also included. The next section deals with the result analysis. The system is executed over the test cases and the results are analyzed and discussed. The final section deals with conclusion and then references are mentioned.

2. LITERATURE SURVEY

2.1 Introduction to the problem domain terminology

In our day to day life, people are undergoing a routine and busy schedule which leads to stress and anxiety. In addition to this, the percentage of people who are obese and addicted to cigarette goes up drastically. This leads to diseases like heart disease, cancer, etc. The challenge behind these diseases is its prediction. Each person has different values of pulse rate and blood pressure. But medically proven, the pulse rate must be 60 to 100beats per minute and the blood pressure must be in the range of 120/80 to 140/90.

Heart disease is one of the major causes of death in the world. The number of people affected by heart disease increases irrespective of age in both men and women. But other factors like gender, diabetes, BMI also contribute to this disease. In this project, we have tried prediction and analysis of heart disease by considering the parameters like age, gender, blood pressure, heart rate, diabetes and so on. Since numerous factors are involved in heart disease, the prediction of this disease is challenging.

Many researchers, in recent times, have been using several machine learning techniques to help the healthcare industry and the professionals in the diagnosis of heart related diseases. It is difficult to identify heart disease because of several contributory risk factors such as diabetes, high blood pressure, high cholesterol, abnormal pulse rate and many other factors. Various techniques in data mining and neural networks have been employed to find out the severity of heart disease among humans. .The nature of heart disease is complex and hence, the disease must be handled carefully. Not doing so may affect the heart or cause premature death.

Heart disease is predicted based on symptoms namely, pulse rate, sex, age, and many others. So ML techniques are used.

The main objective of this research is to improve the performance accuracy of heart disease prediction. This paper presents comparative results of various models based on machine learning and deep learning algorithms and techniques and analyzes their performance. Models based on supervised learning algorithms such as SVM, Naive Bayes, Decision Trees, Random Forest and ensemble models are found very popular among the researchers. We are using a hybrid method to combine 2 models to improve accuracy of heart disease prediction. Along with that we are also implementing deep learning and genetic algorithms.

2.2 Existing System

Although many methods like decision trees ,logistic regression ,and naive bayes are used for predicting heart disease ,they did not provide the early prediction of heart disease with more accuracy. Almost all the existing models lacked in getting the most accurate predictions resulting in failure of techniques. Every existing model has issues like naive bayes algorithm can’t be used for large datasets and naive bayes considers that features are independent and fails at tasks to find relationship between features, in decision trees because of pruning it leads to reduced size of tree leading to poor accuracy, knn is a non parametric method which has no idea about underlying data etc. And also in some existing models classification is inaccurate leading to errors because the model did not fit well. So we are trying to combine the techniques to give the comparative predictive results in order to produce a prediction model using not only distinct techniques but also by relating two or more techniques with comparative results.

2.3 Related Works

2.3.1 Heart disease prediction system based on hidden naive bayes classifier[1]

Dataset used cleveland and Tool used weka 6.4 to apply HNB

HNB is a more accurate classification compared to naïve Bayes, with respect to attribute dependencies. HNB is equivalent to a Bayesian classifier which avoids the intractable complexity and takes the influence from all features into account. In hidden naïve bayes, a parent is created for each feature, which integrates the influences from other features.

Algorithm: Heart disease prediction using hidden naïve bayes

Input: Heart disease data set

Output: Classification whether a person is having heart Disease or not

Step 1: Heart data set is loaded

Step 2: Apply preprocessing filter discretization and inter quartile range (IQR)

Step 3: Partition the data sets into training and test set

Step 4: Heart disease data set is trained by HNB

Step 5: The test data set is given to HNB for testing

Step 6: Measure the accuracy of the HNB

Algorithm for hidden naïve bayes(HNB) is as follows

Input: A set of data base

Output: Hidden naïve bayes classifier

Step 1: For each value of c of class C

Step 2: Calculate probabilities P(C) from Database D

Step 3: For attributes Ai and Aj

Step 4: Compute P (ai|aj, c)from D

Step 5: Compute conditional mutual information MI=IP(Ai; Aj| C)and weights Wij from D

Drawback : HNB is a structure-extension-based algorithm and needs more training time .

2.3.2 Human heart disease prediction system using data mining techniques[2]

In this paper we are using three different data mining techniques namely – Decision Tree, Naïve Bayes and KNN. We are using KNN as a new method for heart disease prediction and we are comparing it with other two techniques.

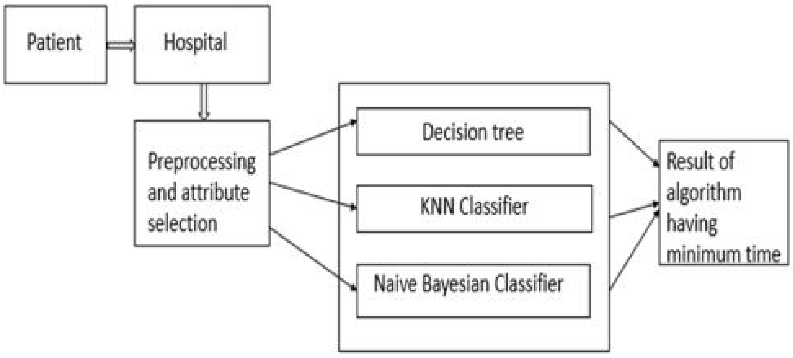

Patient
Hospital
Preprocessing and attribute selection
Decision tree
KNN Classifier
Naive Bayesian Classifier
Result of algorithm having minimum time

Figure 2.3.2 system architecture of base paper 1

Figure 2.3.2 is the system architecture of Human heart disease prediction system using data mining techniques

In recognition of patterns, the k-NN is a method which is non-parametric and used for classification and regression. In both of the cases, the input consists of the k closest training examples in the feature space. The output of pattern recognition depends on whether k-NN is used for classification or regression:

- In k-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor.

K-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The k-NN algorithm is among the simplest of all machine learning algorithms.

Algorithm

1.load and preprocess data

2. Feature Selection.

3. Apply knn,naive bayes and decision tree

4. Calculate accuracy

5. Find more accurate algorithm

2.3.3 Prediction of heart disease using neural network [3]

Dataset used here is Cleveland from uci repository.The proposed heart disease prediction system which uses a multilayer perceptron neural network was developed in MATLAB R2015a.

The designed ANN has three layers: namely an input layer, a hidden layer and an output layer.

- Input Layer was designed to contain 13 neurons. Number of neurons was decided to be equal to the number of attributes in the data set.
- Hidden Layer was designed to contain 3 neurons. This number was decided as a startup point. The number was changed increasing one by one until it reached the number of neurons of the input layer by comparing performance of them and then selecting the best one. This approach is based on one of machine learning best practices that the number of neurons of a hidden layer should be the mean of the number of the neurons of input and output layers.
- Output Layer was designed to contain 2 neurons. The designed NN is a classifier going running in Machine Mode which means returning a class label (e.g., "Disease Presence"/"Disease Absence"). Deciding 2 neurons is based on the idea that the output layer has one node per class label in the model.

Working of multilayer perceptron neural network in this research work:

Phase 1: Propagation

1) Forward propagation of a training pattern's input through the neural network in order to generate the propagation's output activations.

2) Back propagation of the propagation's output activations through the neural network using the training pattern's target in order to generate the deltas of all output and hidden neurons.

Phase 2: Weight update

For each weight-synapse:

1) Multiply its output delta and input activation to get the gradient of the weight.

2) Bring the weight in the opposite direction of the gradient by subtracting a ratio of it from the weight.

Repeat the phase 1 and 2 until the performance of the network is good enough.

2.4 Tools

Jupyter Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R.Jupyter ships with the IPython kernel, which allows you to write your programs in Python

3. DESIGN OF THE PROPOSED SYSTEM

3.1 Block Diagram

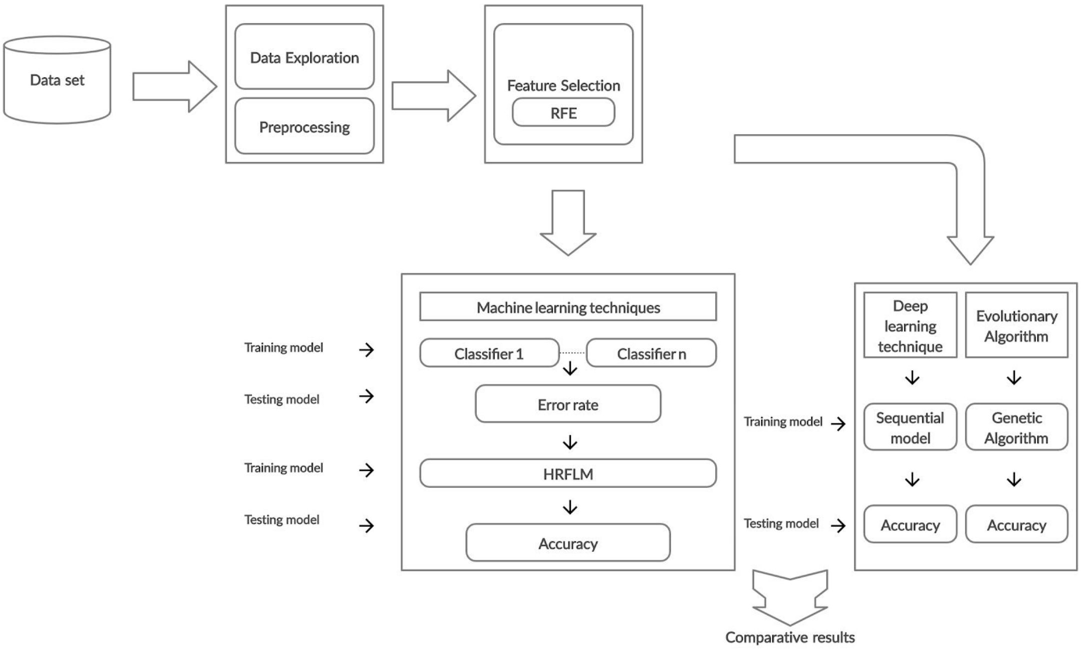


Figure 3.1 Block diagram

Figure 3.1 is a block diagram that gives a brief flow of algorithms in the project. The steps include preprocessing, feature extraction, classification and comparative results. Finally output of each accuracies of models are displayed.

3.2 Module Description

3.2.1 NUMPY

NumPy stands for ‘Numerical Python’ or ‘Numeric Python’. It is an open source module of Python which provides fast mathematical computation on arrays and matrices. Since arrays and matrices are an essential part of the Machine Learning ecosystem, NumPy along with Machine Learning modules like Scikit-learn, Pandas, Matplotlib, TensorFlow, etc. complete the Python Machine Learning Ecosystem.

NumPy provides the essential multi-dimensional array-oriented computing functionalities designed for high-level mathematical functions and scientific computation.

3.2.2 PANDAS

pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with structured (tabular, multidimensional, potentially heterogeneous) and time series data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open source data analysis / manipulation tool available in any language.

The two primary data structures of pandas, Series (1-dimensional) and DataFrame (2-dimensional), handle the vast majority of typical use cases in finance, statistics, social science, and many areas of engineering.

3.2.3 MATPLOTLIB

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. It is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

3.2.4 SCIKIT

Scikit-learn (formerly scikits.learn and also known as sklearn) is a free software machine learning library for the Python programming language. It is a Python module for machine learning built on top of SciPy and is distributed under the 3-Clause BSD license. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

Some popular groups of models provided by scikit-learn include:

- Clustering: for grouping unlabeled data such as KMeans.
- Cross Validation: for estimating the performance of supervised models on unseen data.
- Datasets: for test datasets and for generating datasets with specific properties for investigating model behavior.
- Dimensionality Reduction: for reducing the number of attributes in data for summarization, visualization and feature selection such as Principal component analysis.
- Ensemble methods: for combining the predictions of multiple supervised models.
- Feature extraction: for defining attributes in image and text data.
- Feature selection: for identifying meaningful attributes from which to create supervised models.
- Parameter Tuning: for getting the most out of supervised models.
- Manifold Learning: For summarizing and depicting complex multi-dimensional data.
- Supervised Models: a vast array not limited to generalized linear models, discriminate analysis, naive bayes, lazy methods, neural networks, support vector machines and decision trees.

3.2.5 TENSORFLOW

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks.It provides multiple APIs (Application Programming Interfaces).

TensorFlow is basically a software library for numerical computation using data flow graphs where:

nodes in the graph represent mathematical operations.

edges in the graph represent the multidimensional data arrays (called tensors) communicated between them. (Tensor is the central unit of data in TensorFlow).

3.3 Theoretical Foundation/Algorithms

3.3.1 Logistic Regression

Logistic Regression was used in the biological sciences in the early twentieth century. It was then used in many social science applications. Logistic Regression is used when the dependent variable(target) is categorical.

For example, To predict whether an email is spam (1) or (0)

Consider a scenario where we need to classify whether an email is spam or not. If we use linear regression for this problem, there is a need for setting up a threshold based on which classification can be done. Say if the actual class is malignant, predicted continuous value 0.4 and the threshold value is 0.5, the data point will be classified as not malignant which can lead to serious consequences in real time.

From this example, it can be inferred that linear regression is not suitable for classification problems. Linear regression is unbounded, and this brings logistic regression into picture. Their value strictly ranges from 0 to 1.

Logistic regression is a predictive analysis algorithm and based on the concept of probability. The hypothesis of logistic regression tends it to limit the cost function between 0 and 1. Therefore linear functions fail to represent it as it can have a value greater than 1 or less than 0 which is not possible as per the hypothesis of logistic regression.

It works by predicting the probability that Y belongs to a particular category by first fitting the data to a linear regression model, which is then passed to the sigmoid function (f(x)=1/1+e^−x), which then outputs a value between 0 and 1 and can be interpreted as the probability. If the probability is above a certain predetermined threshold (P(Yes) > 0.5), then the model will predict Yes.

Types of Logistic Regression

1. Binary Logistic Regression

The categorical response has only two 2 possible outcomes. Example: Spam or Not.

2. Multinomial Logistic Regression

Three or more categories without ordering. Example: Predicting which food is preferred more (Veg, Non-Veg, Vegan).

3. Ordinal Logistic Regression

Three or more categories with ordering. Example: Movie rating from 1 to 5.

Advantages of Logistic Regression

- Logistic Regression performs well when the dataset is linearly separable.
- Logistic regression is less prone to overfitting but it can overfit in high dimensional datasets. You should consider Regularization (L1 and L2) techniques to avoid over-fitting in these scenarios.

- Logistic Regression not only gives a measure of how relevant a predictor (coefficient size) is, but also its direction of association (positive or negative).
- Logistic regression is easier to implement, interpret and very efficient to train.

Disadvantages of Logistic Regression

- Main limitation of Logistic Regression is the assumption of linearity between the dependent variable and the independent variables. In the real world, the data is rarely linearly separable. Most of the time data would be a jumbled mess.
- If the number of observations are lesser than the number of features, Logistic Regression should not be used, otherwise it may lead to overfit.
- Logistic Regression can only be used to predict discrete functions. Therefore, the dependent variable of Logistic Regression is restricted to the discrete number set. This restriction itself is problematic, as it is prohibitive to the prediction of continuous data.

3.3.2Naive Bayes

Naive Bayes classifier is a straightforward and powerful algorithm for the classification task. Even if we are working on a data set with millions of records with some attributes, it is suggested to try Naive Bayes approach. Naive Bayes classifier gives great results when we use it for textual data analysis. Such as Natural Language Processing.

To understand the naive bayes classifier we need to understand the Bayes theorem.

Bayes theorem named after Rev. Thomas Bayes. It works on conditional probability. Conditional probability is the probability that something will happen, given that something else has already occurred. Using the conditional probability, we can calculate the probability of an event using its prior knowledge.

The formula for calculating the conditional probability: **P(H|E)=P(E|H)*P(H)/P(E)** where,

P(H) is the probability of hypothesis H being true. This is known as the prior probability.

P(E) is the probability of the evidence(regardless of the hypothesis).

P(E|H) is the probability of the evidence given that hypothesis is true.

P(H|E) is the probability of the hypothesis given that the evidence is there.

This naive bayes learning model applies Bayes rules through independent features. Every instance of data D is allotted to the class of highest subsequent probability. The model is trained through the Gaussian function with prior probability P(Xf) = priority∈(0:1)

$$
\begin{aligned}
& P\left(X_{f1}, X_{f2}, \ldots, X_{f_n} | c\right) \\
& \quad = \prod_{i=1}^{n} P\left(X_{fi} | c\right) \\
& P\left(X_f | c_i\right) \\
& \quad = \frac{P\left(c_i | X_f\right) P\left(X_f\right)}{P\left(c_i\right)} \quad c \in \{\ \mathit{benign},\ \mathit{malignant}\ \}
\end{aligned}
$$

The model predicts membership probabilities for each class such as the probability that given record or data point belongs to a particular class. The class with the highest probability is considered as the most likely class. This is also known as Maximum A Posteriori (MAP).

The Maximum A Posteriori for a hypothesis is:

MAP(H) = max(P(H|E))

= max((P(E|H)*P(H))/P(E))

= max(P(E|H)*P(H))

P(E) is evidence probability, and it is used to normalize the result.

Naive Bayes classifier assumes that all the features are unrelated to each other. Presence or absence of a feature does not influence the presence or absence of any other feature.

Types of Naive Bayes Algorithm

Gaussian Naive Bayes

When attribute values are continuous, an assumption is made that the values associated with each class are distributed according to Gaussian i.e., Normal Distribution.

MultiNomial Naive Bayes

MultiNomial Naive Bayes is preferred to use on data that is multinomial distributed. It is one of the standard classic algorithms. Which is used in text categorization (classification). Each event in text classification represents the occurrence of a word in a document.

Bernoulli Naive Bayes

Bernoulli Naive Bayes is used on the data that is distributed according to multivariate Bernoulli distributions.i.e., multiple features can be there, but each one is assumed to be a binary-valued (Bernoulli, boolean) variable. So, it requires features to be binary valued.

Advantages of naive bayes classifier

- Naive Bayes Algorithm is a fast, highly scalable algorithm.
- Naive Bayes can be used for Binary and Multiclass classification.It is a simple algorithm that depends on doing a bunch of counts.
- Great choice for Text Classification problems. It’s a popular choice for spam email classification.
- It can be easily train on small dataset

Disadvantages of naive bayes classifier

It considers all the features to be unrelated, so it cannot learn the relationship between features. E.g., Let's say Remo is going to a part. While cloth selection for the party, Remo is looking at his cupboard. Remo likes to wear a white color shirt. In Jeans, he likes to wear a brown Jeans, But Remo doesn't like wearing a white shirt with Brown Jeans. Naive Bayes can learn individual features importance but can't determine the relationship among features.

3.3.3 Decision Trees

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. Decision trees learn from data to approximate a sine curve with a set of if-then-else decision rules. The deeper the tree, the more complex the decision rules and the fitter the model.

Decision tree build classification or regression models in the form of a tree structure. It breaks down a data set into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches. Leaf node represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data.

These trees are similar to flowcharts.For training samples of data D, the trees are constructed based on high entropy inputs. These trees are simple and fast constructed in a top down recursive divide and conquer(DAC) approach. Tree pruning is performed to remove the irrelevant samples on D.

$$Entropy = -\sum_{j=1}^{m} p_{ij} \log_2 p_{ij}$$

There are several steps involved in the building of a decision tree:

Step 1-Splitting

The process of partitioning the data set into subsets. Splits are formed on a particular variable.

Step 2-Pruning

The shortening of branches of the tree. Pruning is the process of reducing the size of the tree by turning some branch nodes into leaf nodes, and removing the leaf nodes under the original branch. Pruning is useful because classification trees may fit the training data well, but may do a poor job of classifying new values. A simpler tree often avoids overfitting.A pruned tree has less nodes and has less sparsity than an unpruned decision tree.

Step 3-Tree Selection

The process of finding the smallest tree that fits the data. Usually this is the tree that yields the lowest cross-validated error.

Factors of decision tree:

1. Entropy

A decision tree is built top-down from a root node and involves partitioning the data into subsets that contain instances with similar values (homogeneous). ID 3 algorithm uses entropy to calculate the homogeneity of a sample. If the sample is completely homogeneous the entropy is zero and if the sample is equally divided it has entropy of one.

2. Information Gain

The information gain is based on the decrease in entropy after a dataset is split on an attribute. Constructing a decision tree is all about finding the attribute that returns the highest information gain i.e., the most homogeneous branches.

Steps involved in working of decision tree:

Step 1: Calculate entropy of the target.

Step 2:

The dataset is then split on the different attributes. The entropy for each branch is calculated. Then it is added proportionally, to get total entropy for the split. The resulting entropy is subtracted from the entropy before the split. The result is the Information Gain, or decrease in entropy.

Step 3:

Choose an attribute with the largest information gain as the decision node, divide the dataset by its branches and repeat the same process on every branch.

Advantages of decision tree

Computationally cheap to use, easy for humans to understand results and it can deal with irrelevant features also.

Disadvantages of decision tree

Prone to Overfitting i.e, it refers to the process when models are trained on training data too well that any noise in testing data can bring negative impacts to performance of model.

3.3.4 Support Vector Machine

Support Vector Machine is a linear model for classification and regression problems. It can solve linear and non-linear problems and work well for many practical problems. The idea of it is simple: The algorithm creates a line or a hyperplane which separates the data into classes.

Let the training samples having dataset Data ={yi,xi};i = 1,2,...,n where xi ∈Rn represent the ith vector and yi ∈Rn represent the target item. The linear SVM finds the optimal hyperplane of the form f(x) = wTx + b where w is a dimensional coefficient vector and b is an offset. This is done by solving the subsequent optimization problem:

$$
\begin{aligned}
&Min_{w,b,\xi_i}\ \frac{1}{2}w^2 + C\sum_{i=1}^{n}\xi_i \\
&s.t.\ y_i\left(w^T x_i + b\right) \geq 1 - \xi_i,\ \xi_i \geq 0,\quad \forall_i \in \{1, 2, \ldots, m\}
\end{aligned}
$$

According to the algorithm it finds the points closest to the line or hyperplane from both the classes.These points are called support vectors. Now, it computes the distance between the line and the support vectors. This distance is called the margin. The model goal is to maximize the margin. The hyperplane for which the margin is maximum is the optimal hyperplane. Thus the support vector machine tries to make a decision boundary in such a way that the separation between the two classes is as wide as possible.

If the data is not linearly separable,it makes data linearly separable by converting data into higher dimension.

Advantages of support vector machine

- It is really effective in the higher dimension.
- Effective when the number of features are more than training examples.
- Best algorithm when classes are separable
- The hyperplane is affected by only the support vectors thus outliers have less impact.
- SVM is suited for extreme case binary classification.

Disadvantages of support vector machine

- For larger dataset, it requires a large amount of time to process.
- Does not perform well in case of overlapped classes.
- Selecting, appropriately hyperparameters of the SVM that will allow for sufficient generalization performance.
- Selecting the appropriate kernel function can be tricky.

3.3.5 Random Forest

Random forest is a supervised classification algorithm. It creates the forest with a number of trees.In general, the more trees in the forest the more robust the forest looks like. In the same way in the random forest classifier, the higher the number of trees in the forest gives the high accuracy results.This ensemble classifier builds several decision trees and incorporates them to get the best result. For tree learning, it mainly applies bootstrap aggregating or bagging. For a given data, X = {x1,x2, x3, ...,xn} with responses Y = {x1,x2,x3,...,xn}which repeats the bagging from b=1toB.

$$\sigma = \sqrt{\frac{\sum_{b=1}^{B}\left(fb\,(x') - \hat{f}\right)^2}{B-1}}$$

Random Forest pseudocode:

1.Randomly select “k” features from total “m” features where k << m.

2.Among the “k” features, calculate the node “d” using the best split point.

3.Split the node into daughter nodes using the best split.

4.Repeat 1 to 3 steps until “l” number of nodes has been reached.

5.Build forest by repeating steps 1 to 4 for “n” times to create “n” number of trees.

The beginning of a random forest algorithm starts with randomly selecting “k” features out of total “m” features. In the next stage, the model uses the randomly selected “k” features to find the root node by using the best split approach.The next stage model will be calculating the daughter nodes using the same best split approach. Model repeats the first 3 stages until it forms the tree with a root node and has the target as the leaf node.Finally, model repeats 1 to 4 stages to create “n” randomly created trees. This randomly created trees forms the random forest.

Random forest prediction pseudocode:

1.To perform prediction using the trained random forest algorithm uses the below pseudocode.

2. Takes the test features and use the rules of each randomly created decision tree to predict the outcome and stores the predicted outcome (target)

3. Calculate the votes for each predicted target.

4. Consider the high voted predicted target as the final prediction from the random forest algorithm.

Advantages of random forest algorithm

- The overfitting problem will never come when we use the random forest .
- The same random forest algorithm can be used for both classification and regression tasks.
- The random forest algorithm can be used for feature engineering which means identifying the most important features out of the available features from the training dataset.

Disadvantages of random forest algorithm

- It surely does a good job at classification but not as for regression problems as it does not give precise continuous nature prediction. In case of regression, it doesn't predict beyond the range in the training data, and that they may over fit data sets that are particularly noisy.
- Random forest can feel like a black box approach for statistical modelers; we have very little control on what the model does. You can at best try different parameters and random seeds.

3.3.6 Gradient Boosting

Gradient Boosting(an ensemble learner) is another technique for performing supervised machine learning tasks, like classification and regression. This means it will create a final model based on a collection of individual models.

The predictive power of these individual models is weak and prone to overfitting but combining many such weak models in an ensemble will lead to an overall much improved result.

Gradient boosting Algorithm involves three elements:

- A loss function to be optimized.
- Weak learner to make predictions.
- An additive model to add weak learners to minimize the loss function.

1. Loss Function

The loss function used depends on the type of problem being solved.It must be differentiable. Although, many standard loss functions are supported and you can define your own.

2. Weak Learner

- Generally used are decision trees as the weak learner in gradient boosting
- Specifically, use a regression tree that outputs real values for splits. And whose output can be added together. It allows next models outputs to be added and "correct" the residuals in the predictions.
- Trees need to be constructed in a greedy manner. It helps in choosing the best split points based on purity scores like Gini or to cut the loss.
- Initially, such as in the case of AdaBoost. Also, use very short decision trees that only had a single split, called a decision stump.
- Generally, use larger trees with 4 to 8 levels.
- It is common to constrain the weak learners in specific ways. Such as a maximum number of layers, nodes, splits or leaf nodes.
- This is to ensure that the learners remain weak, but can still need to construct in a greedy manner.

3. Additive Model

- Trees need to add one at a time, and existing trees in the model need not change.

- Use a gradient descent procedure to minimize the loss when adding trees.
- Traditionally, use a gradient tree to cut a set of parameters. Such as the coefficients in a regression equation or weights in a neural network. After calculating error or loss, the weights need to be updated to minimize that error.
- Instead of parameters, have weak learner sub-models or more specifically decision trees. After calculating the loss, to perform the gradient descent procedure. Must add a tree to the model that reduces the loss.
- Do this by parameterizing the tree. Then change the parameters of the tree and move in the right direction by reducing the residual loss.

Advantages of gradient boosting trees

- Often provides predictive accuracy that cannot be beat.
- Lots of flexibility - can optimize on different loss functions and provides several hyperparameter tuning options that make the function fit very flexible.
- No data pre-processing required - often works great with categorical and numerical values.
- Handles missing data - imputation not required.

Disadvantages of gradient boosting trees

- Gradient boosting trees will continue improving to minimize all errors. This can overemphasize outliers and cause overfitting. Must use cross-validation to neutralize.
- Computationally expensive - Gradient boosting trees often require many trees (>1000) which can be time and memory exhaustive.
- The high flexibility results in many parameters that interact and influence heavily the behavior of the approach (number of iterations, tree depth, regularization parameters, etc.). This requires a large grid search during tuning.

- Less interpretable although this is easily addressed with various tools (variable importance, partial dependence plots, LIME, etc.).

3.3.7 K-nearest neighbors[12]

KNN algorithm uses ‘feature similarity’ to predict the values of new data points which further means that the new data point will be assigned a value based on how closely it matches the points in the training set. It is a non parametric and lazy algorithm. We can understand its working with the help of following steps −

we need to choose the value of K i.e. the nearest data points. K can be any integer.

For each point in the test data do the following −

- Calculate the distance between test data and each row of training data with the help of any of the methods namely: Euclidean, Manhattan or Hamming distance. The most commonly used method to calculate distance is Euclidean.
- Now, based on the distance value, sort them in ascending order.
- Next, it will choose the top K rows from the sorted array.
- Now, it will assign a class to the test point based on the most frequent class of these rows.

Voting Classifier[7]

A Voting Classifier is a machine learning model that trains on an ensemble of numerous models and predicts an output (class) based on their highest probability of chosen class as the output.

It simply aggregates the findings of each classifier passed into the Voting Classifier and predicts the output class based on the highest majority of voting. The idea is instead of creating separate dedicated models and finding the accuracy for each of them, it creates a single model which trains by these models and predicts output based on their combined majority of voting for each output class.

Voting Classifier supports two types of voting.

Hard Voting: In hard voting, the predicted output class is a class with the highest majority of votes i.e the class which had the highest probability of being predicted by each of the classifiers. Suppose three classifiers predicted the output class(A, A, B), so here the majority predicted A as output. Hence A will be the final prediction.

Soft Voting: In soft voting, the output class is the prediction based on the average of probability given to that class. Suppose given some input to three models, the prediction probability for class A = (0.30, 0.47, 0.53) and B = (0.20, 0.32, 0.40). So the average for class A is 0.4333 and B is 0.3067, the winner is clearly class A because it had the highest probability averaged by each classifier.

3.3.8 Deep Learning [9]

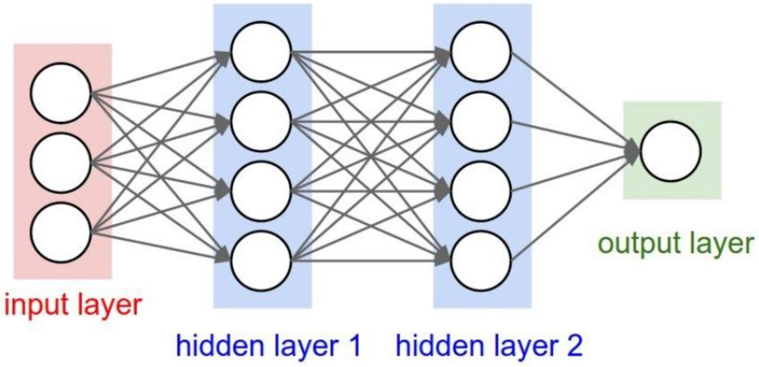


Figure 3.3.8 deep learning

Figure 3.3.8 is deep learning architecture with two hidden and one output layers.

Deep learning is an increasingly popular subset of machine learning. Deep learning models are built using neural networks. A neural network takes in inputs, which are then processed in hidden layers using weights that are adjusted during training. Then the model spits out a prediction. The weights are adjusted to find patterns in order to make better predictions. The user does not need to specify what patterns to look for — the neural network learns on its own. We use a sequential model of deep learning.Sequential is the easiest way to build a model in Keras. It allows you to build a model layer by layer. Each layer has weights that correspond to the layer that follows it. We use the 'add()' function to add layers to our model. We will add two layers and an output layer.

3.3.9 Genetic Algorithm [11]

Genetic algorithm is an optimization method inspired by the biological process of natural selection. It is based on the terms such as mutation, crossover and selection.The genetic algorithm is a random-based classical evolutionary algorithm. By random it means that in order to find a solution using the genetic algorithm, random changes applied to the current solutions to generate new ones.

Genetic Algorithm is based on Darwin's theory of evolution. It is a slow gradual process that works by making changes to the making slight and slow changes. Also, genetic algorithm, makes slight changes to their solutions slowly until getting the best solution.

Parameters used in genetic algorithm:

Genes

Using genetic algorithm nomenclature a feature is called a gene. It can be either included (1) or excluded (0) during the feature selection process.

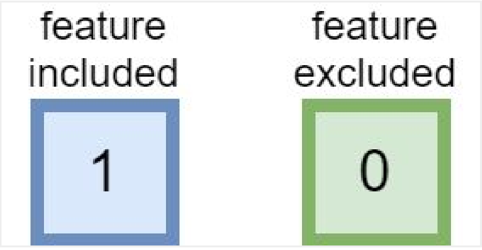
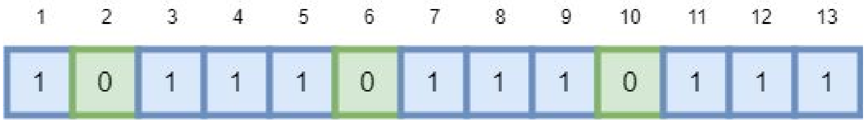


Figure 3.3.9.1 Gene

Figure 3.3.9.1 indicates whether features are included or not.If it is included it will be 1 otherwise 0.

Chromosome

A list of genes is called a chromosome. The chromosome contains information which feature is included and which is excluded.



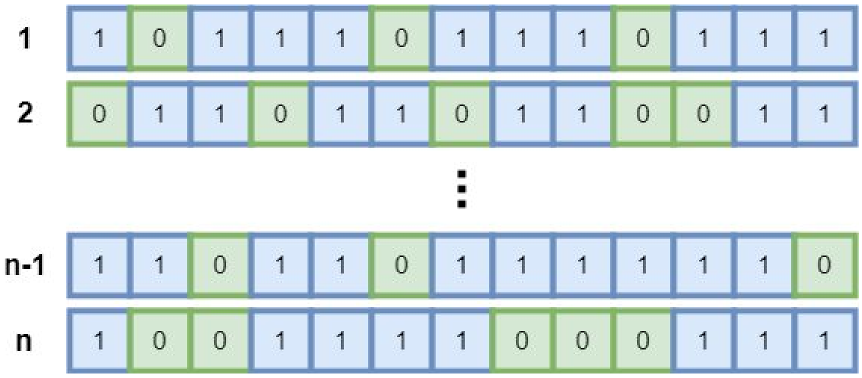
1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	1	1	1	0	1	1	1	0	1	1	1

Figure 3.3.9.2 Chromosome as a list of 13 genes.

Figure 3.3.9.2 shows chromosome structure where genes are combined as a list where some genes are included and some are not.

Population

Population contains several instances of different chromosomes. It is just a collection of different feature subsets.



1	1	0	1	1	1	0	1	1	1	0	1	1	1
2	0	1	1	0	1	1	0	1	1	0	0	1	1
							⋮						
n-1	1	1	0	1	1	0	1	1	1	1	1	1	0
n	1	0	0	1	1	1	1	0	0	0	1	1	1

Figure 3.3.9.3 Population contains several different chromosomes.

Figure 3.3.9.3 displays a population containing several instances of different chromosomes.

The first population of n chromosomes is created by random exclusion of features.

Working of genetic algorithm:

Genetic Algorithm works on a population consisting of some solutions where the population size is the number of solutions. Each solution is called individual. Each individual solution has a chromosome. The chromosome is represented as a set of parameters (features) that defines the individual. Each chromosome has a set of genes. Each gene is represented by somehow such as being represented as a string of 0s and 1s.

Also, each individual has a fitness value. To select the best individuals, a fitness function is used. The result of the fitness function is the fitness value representing the quality of the solution. The higher the fitness value the higher the quality the solution. Selection of the best individuals based on their quality is applied to generate what is called a mating pool where the higher quality individual has a higher probability of being selected in the mating pool.

The individuals in the mating pool are called parents. Every two parents selected from the mating pool will generate two offspring (children). By just mating high-quality individuals, it is expected to get a better quality offspring than its parents. This will kill the bad individuals from generating more bad individuals. By keeping selecting and mating high-quality individuals, there will be higher chances to just keep good properties of the individuals and leave out bad ones. Finally, this will end up with the desired optimal or acceptable solution.

But the offspring currently generated using the selected parents just have the characteristics of its parents and no more without changes. There is no new addition to it and thus the same drawbacks in its parents will actually exist in the new offspring. To overcome such problems, some changes will be applied to each offspring to create new individuals. The set of all newly generated individuals will be the new population that replaces the previously used old population. Each population created is called a generation. The process of replacing the old population by the new one is called replacement.

Methods in genetic algorithm:

Initialization

After getting how to represent each individual, next is to initialize the population by selecting the proper number of individuals within it.

Selection

Next is to select a number of individuals from the population in the mating pool. Based on the previously calculated fitness value, the best individuals based on a threshold are selected. After that step, it will end selecting a subset of the population in the mating pool.

Variation Operators

Based on the selected individuals in the mating pool, parents are selected for mating. The selection of each two parents may be by selecting parents sequentially (1–2, 3–4, and so on). Another way is random selection of the parents.

For every two parents selected, there are a number of variation operators to get applied such as:

- Crossover (recombination)
- Mutation

Crossover

Crossover in genetic algorithm generates new generations the same as natural mutation. By mutating the old generation parents, the new generation offspring comes by carrying genes from both parents. The amount of genes carried from each parent is random. Sometimes the offspring takes half of its genes from one parent and the other half from the other parent and sometimes such percent changes. For every two parents, crossover takes place by selecting a random point in the chromosome and exchanging genes before and after such point from its parents. The resulting chromosomes are offspring. This operator is called single-point crossover.

Mutation

Next variation operator is mutation. For each offspring, select some genes and change its value. Mutation varies based on the chromosome representation.If the encoding is binary (i.e. the value space of each gene has just two values 0 and 1), then flip the bit value of one or more genes.

But if the gene value comes from a space of more than two values such as 1,2,3,4, and 5, then the binary mutation will not be applicable and we should find another way.To add new features to such offspring, mutation took place.

But because mutation occurs randomly, it is not recommended to increase the number of genes to be applied to mutation.The individual after mutation is called mutant.

Advantages of genetic algorithm

- It can find fit solutions in a very less time (fit solutions are solutions which are good according to the defined heuristic).
- The random mutation guarantees to some extent that we see a wide range of solutions.
- Coding them is really easy compared to other algorithms which do the same job.

Disadvantages of genetic algorithm

- It's really hard for people to come up with a good heuristic which actually reflects what we want the algorithm to do.
- It might not find the most optimal solution to the defined problem in all cases.
- It's also hard to choose parameters like number of generations, population size etc.

3.3.10 Proposed model - Hybrid Random Forest with Linear Model(HRFLM) Technique

The proposed hybrid HRFLM approach is used combining the characteristics of

Random Forest and Linear Method(Logistic Regression).HRFLM proved to be quite accurate in the prediction of heart disease.

There are three steps in performing the hybrid technique

1.Finding out the output probabilities of each model.To implement this we are using the pred_proba function which gives the probabilities for the target in array form. The number of probabilities for each row is equal to the number of categories in the target variable.

2.With the help of log loss function, finding the optimized weight that perfectly combines the two models which has low classification error rate.Log Loss function is a metric which takes into account the uncertainty of your prediction based on how much it varies from the actual label.

3.Using the optimized weight from the above step combining the two models with the help of weighted average and then performing the prediction.

The results of the hybrid classification method have proved a higher degree of accuracy and performance in prediction of heart disease compared to the other existing methods.

4. IMPLEMENTATION OF PROPOSED SYSTEM

4.1 Flowchart

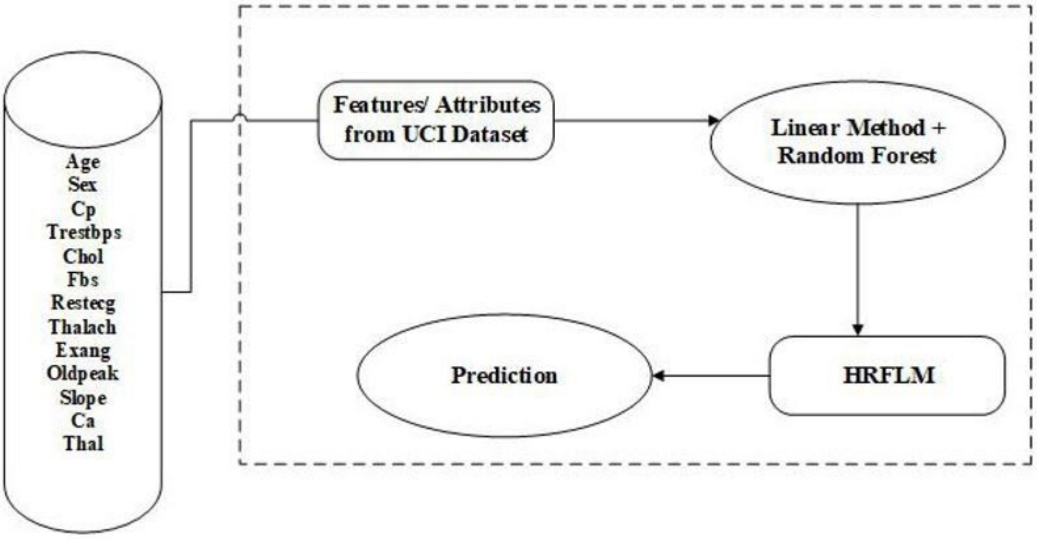


Figure 4.1 Flowchart

Figure 4.1 is a flowchart depicting a process, system or computer algorithm. Flowcharts are widely used in multiple fields to document, study, plan, improve and communicate complex processes in clear, easy-to-understand diagrams.

4.1.1 UML diagrams

UML stands for Unified Modelling Language. UML is a standardized general-purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The UML is a very important part of developing object oriented software and the software development process.

The UML uses mostly graphical notations to express the design of software projects.

Building blocks of the UML.

The vocabulary of the UML encompasses three kinds of building blocks.

1. Things.
2. Relationships
3. Diagrams.

4.1.1.1 Use Case Diagram

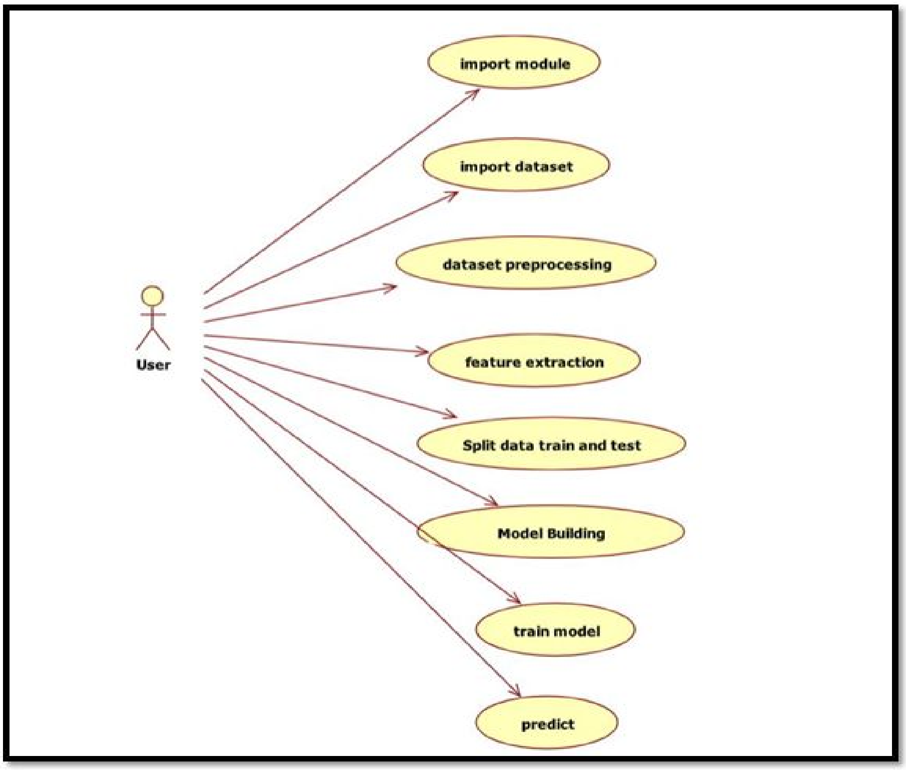


Figure 4.1.1.1 Use case Diagram

Figure 4.1.1.1 displays a use case diagram where it shows the functions of a user in this project.

A use case diagram is a dynamic or behavior diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform. In this context, a "system" is something being developed or operated, such as a web site. The "actors" are people or entities operating under defined roles within the system.

4.1.1.2 Sequence diagram

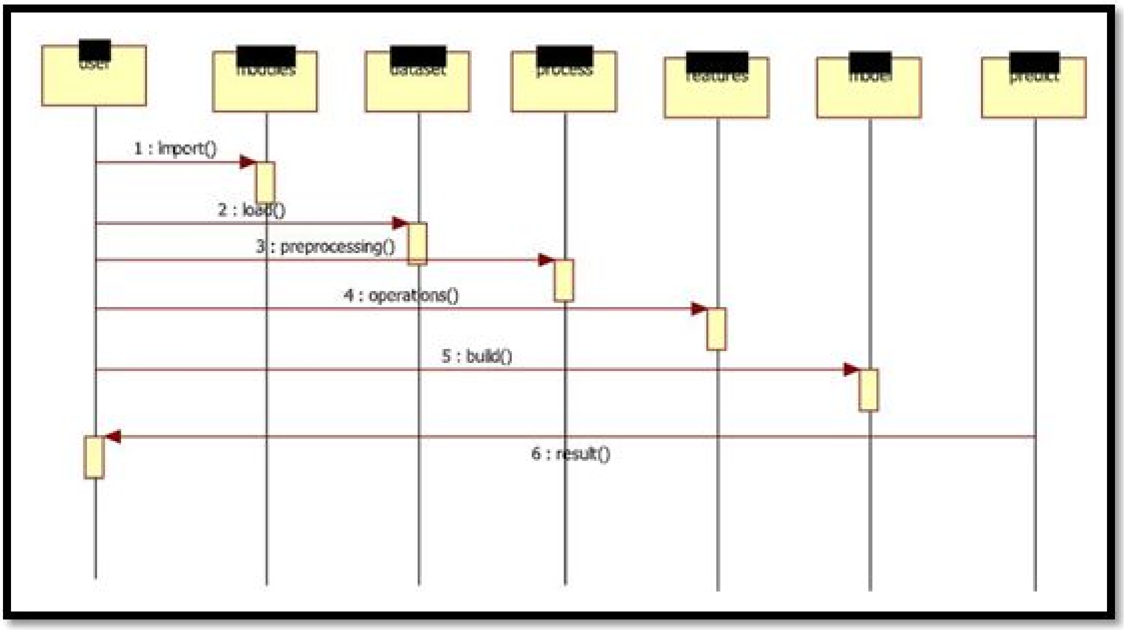


Figure 4.1.1.2 Sequence Diagram

Figure 4.1.1.2 displays a sequence diagrams that are time focused and they show the order of the interaction visually by using the vertical axis of the diagram to represent time, what messages are sent and when.

4.2 Criteria [8]

4.2.1 Confusion Matrix

A confusion matrix is a table that is often used to describe the performance of a classification model (or “classifier”) on a set of test data for which the true values are known. It allows the visualization of the performance of an algorithm.It is a summary of prediction results on a classification problem. The number of correct and incorrect predictions are summarized with count values and broken down by each class. It shows the ways in which your classification model is confused when it makes predictions. It gives us insight not only into the errors being made by a classifier but more importantly the types of errors that are being made.

True Positives (TP) - These are the correctly predicted positive values which means that the value of the actual class is yes and the value of predicted class is also yes. E.g. if actual class value indicates that this person has heart disease and predicted class tells you the same thing.

True Negatives (TN) - These are the correctly predicted negative values which means that the value of the actual class is no and value of predicted class is also no. E.g. if the actual class says the person has no heart disease and predicted class tells you the same thing.

False positives and false negatives, these values occur when your actual class contradicts with the predicted class.

False Positives (FP) – When actual class is no and predicted class is yes. E.g. if the actual class says the person has no heart disease but the predicted class tells you that the person has heart disease.

False Negatives (FN) – When actual class is yes but predicted class in no. E.g. if actual class value indicates the person has heart disease and predicted class tells you that person has no heart disease.

4.2.2 Accuracy

Accuracy is one metric for evaluating classification models. Informally, accuracy is the fraction of predictions our model got right.Accuracy has the following definition:

Accuracy = Total number of correct prediction/total number of predictions

For binary classification, accuracy can also be calculated in terms of positives and negatives as follows:

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

4.2.3 Specificity

Specificity is defined as the proportion of actual negatives, which got predicted as the negative (or true negative). This implies that there will be another proportion of actual negative, which got predicted as positive and could be termed as false positives. This proportion could also be called a false positive rate. The sum of specificity and false positive rate would always be 1.

Specificity = (True Negative)/(True Negative + False Positive)

4.2.4 Sensitivity or Recall

Sensitivity is a measure of the proportion of actual positive cases that got predicted as positive (or true positive). Sensitivity is also termed as Recall. This implies that there will be another proportion of actual positive cases, which would get predicted incorrectly as negative (and, thus, could also be termed as the false negative). This can also be represented in the form of a false negative rate. The sum of sensitivity and false negative rate would be 1.

Sensitivity = (True Positive)/(True Positive + False Negative)

4.2.5 Precision

Precision is defined as the number of true positives divided by the number of true positives plus the number of false positives**.**

Precision = (True Positive)/(True Positive + False Positive)

4.2.6 F1 Score

The F1 score can be interpreted as a weighted average of the precision and recall

F1 = 2*((precision*recall)/(precision+recall)).

4.2.7 Classification error rate

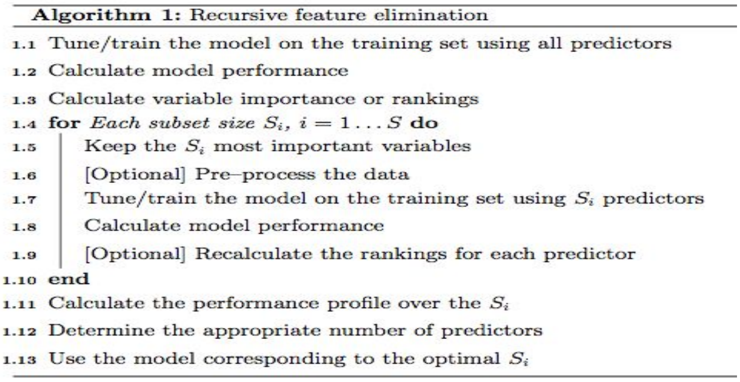
Classification error is defined as the percentage of accuracy missing or error available in the instances.It is nothing but the total number of wrong predictions out of all predictions i.e percentage of wrong predictions.

Classification error = 100-Accuracy

4.3 Algorithms / Pseudo Code

After loading the data and doing preprocessing then applying the feature selection algorithm as follows:

Algorithm 1



```
Algorithm 1: Recursive feature elimination
Tune/train the model on the training set using all predictors
Calculate model performance
Calculate variable importance or rankings
for Each subset size $S_i$, $i = 1 \dots S$ do
    Keep the $S_i$ most important variables
    [Optional] Pre–process the data
    Tune/train the model on the training set using $S_i$ predictors
    Calculate model performance
    [Optional] Recalculate the rankings for each predictor
end
Calculate the performance profile over the $S_i$
Determine the appropriate number of predictors
Use the model corresponding to the optimal $S_i$
```

Algorithm 2 Finding out the error rates for all models.

```
Require: Input: processed dataset
for ∀models find out  error rate from the input do
    error rate(dataset)
end for
Find out the two models with min(Error rate(dataset)) from the classifiers.
Output:Two models with less error.
```

Lastly,applying the hybrid random forest with a linear model.

4.4 Dataset description

4.4.1 Cleveland dataset

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 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. 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.

Attributes of Cleveland dataset:

age : age in years

sex : (1 = male; 0 = female)

cp : chest pain type -- Value 1: typical angina -- Value 2: atypical angina -- Value 3: non-anginal pain -- Value 4: asymptomatic

trestbps : resting blood pressure

chol : cholesterol

fbs : (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)

restecg : -- Value 0: normal -- Value 1: having ST-T wave abnormality -- Value 2: showing probable or definite left ventricular hypertrophy

thalach : maximum heart rate achieved

exang : exercise induced angina (1 = yes; 0 = no)

oldpeak : ST depression induced by exercise relative to rest

slope : the slope of the peak exercise ST segment -- Value 1: upsloping -- Value 2: flat -- Value 3: downsloping

ca : number of major vessels (0-3) colored by fluoroscopy

thal : 3 = normal; 6 = fixed defect; 7 = reversable defect

category : diagnosis of heart disease[0-4] (the predicted attribute)

4.4.2 Framingham dataset

The Framingham Heart Study dataset is dedicated to identifying common factors or characteristics that contribute to cardiovascular disease (CVD). In 1948, an original cohort of 5,209 men and women between 30 and 62 years old were recruited from Framingham, MA. An Offspring Cohort began in 1971, an Omni Cohort in 1994, a Third Generation Cohort in 2002, a New Offspring Spouse Cohort in 2004 and a Second Generation Omni Cohort in 2003. Core research in the dataset focuses on cardiovascular and cerebrovascular diseases. The data include biological specimens,molecular genetic data, phenotype data, samples, participant vascular functioning of the National Heart, Lung and Blood Institute and Boston University.

Attributes of Framingham dataset:

male : 0 = Female; 1 = Male

age : Age at exam time

education : 1 = Some High School; 2 = High School or GED; 3 = Some College or Vocational School; 4 = college

currentSmoker : 0 = nonsmoker; 1 = smoker

cigsPerDay : number of cigarettes smoked per day (estimated average)

BPMeds : 0 = Not on Blood Pressure medications; 1 = on Blood Pressure medications

prevalentStroke : 1 if a patient experienced a stroke during the 12 years and 0 otherwise.

prevalentHyp : 1 if the patient has hypertension, which is defined to be Systolic greater than 140mmHg or Diastolic greater than 90mmHg.

diabetes : 0 = No; 1 = Yes

totChol : mg/dL

sysBP : mmHg

diaBP : mmHg

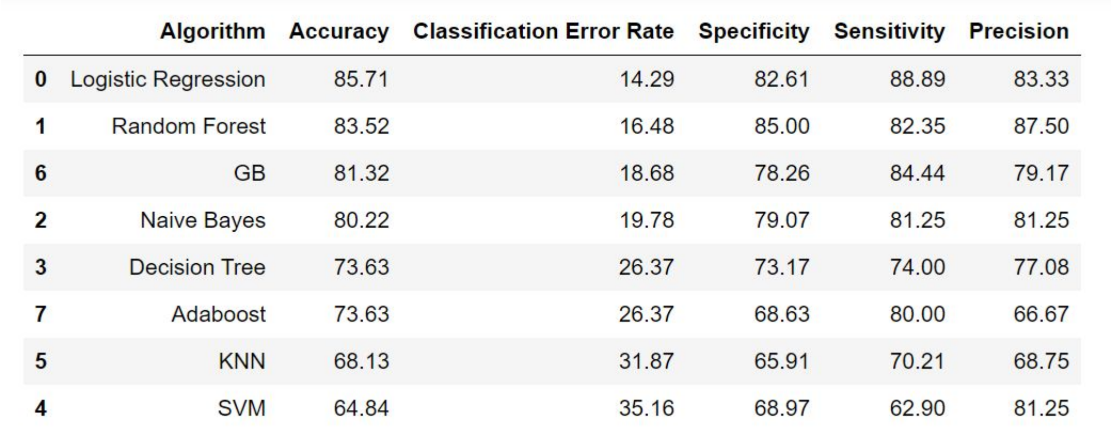
BMI : Body Mass Index calculated as: Weight (kg) / Height(meter-squared)

heartRate : Beats/Min (Ventricular)

glucose : mg/dL

5. RESULTS/ OUTPUTS

Table 5.1 Performance metrics of ML classifiers



	Algorithm	Accuracy	Classification Error Rate	Specificity	Sensitivity	Precision
0	Logistic Regression	85.71	14.29	82.61	88.89	83.33
1	Random Forest	83.52	16.48	85.00	82.35	87.50
6	GB	81.32	18.68	78.26	84.44	79.17
2	Naive Bayes	80.22	19.78	79.07	81.25	81.25
3	Decision Tree	73.63	26.37	73.17	74.00	77.08
7	Adaboost	73.63	26.37	68.63	80.00	66.67
5	KNN	68.13	31.87	65.91	70.21	68.75
4	SVM	64.84	35.16	68.97	62.90	81.25

Table 5.1 shows the machine learning classifiers sorted by classification error rate.Here we got Logistic regression and Random Forest with least classification error rate so we combined these two models using a hybrid method.

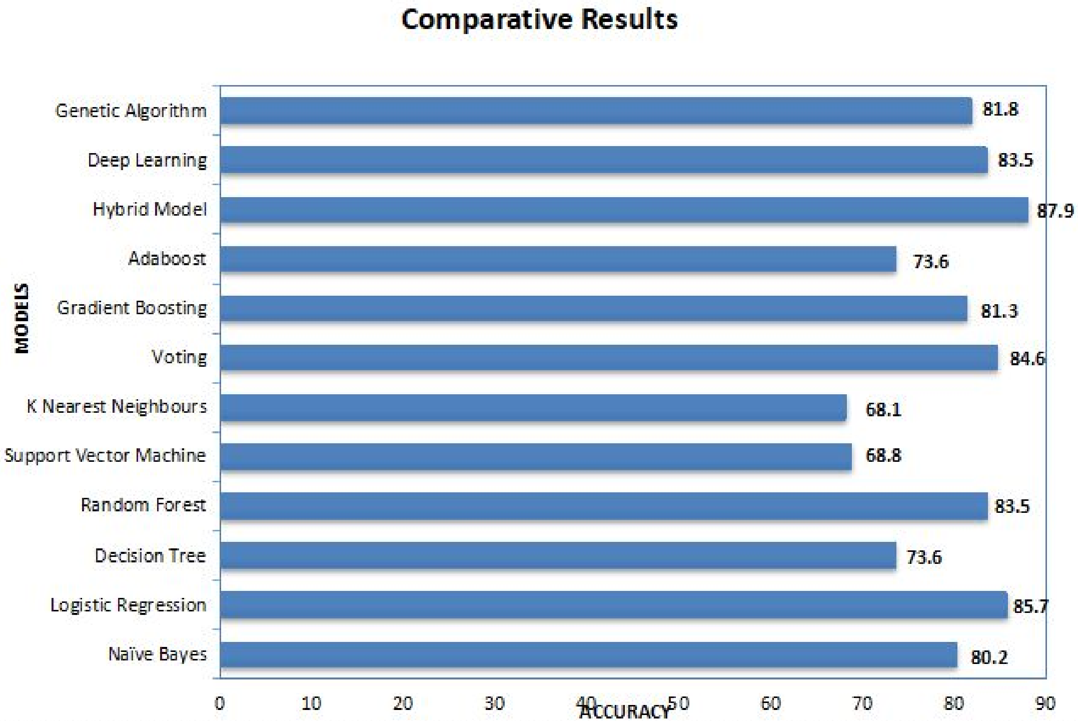
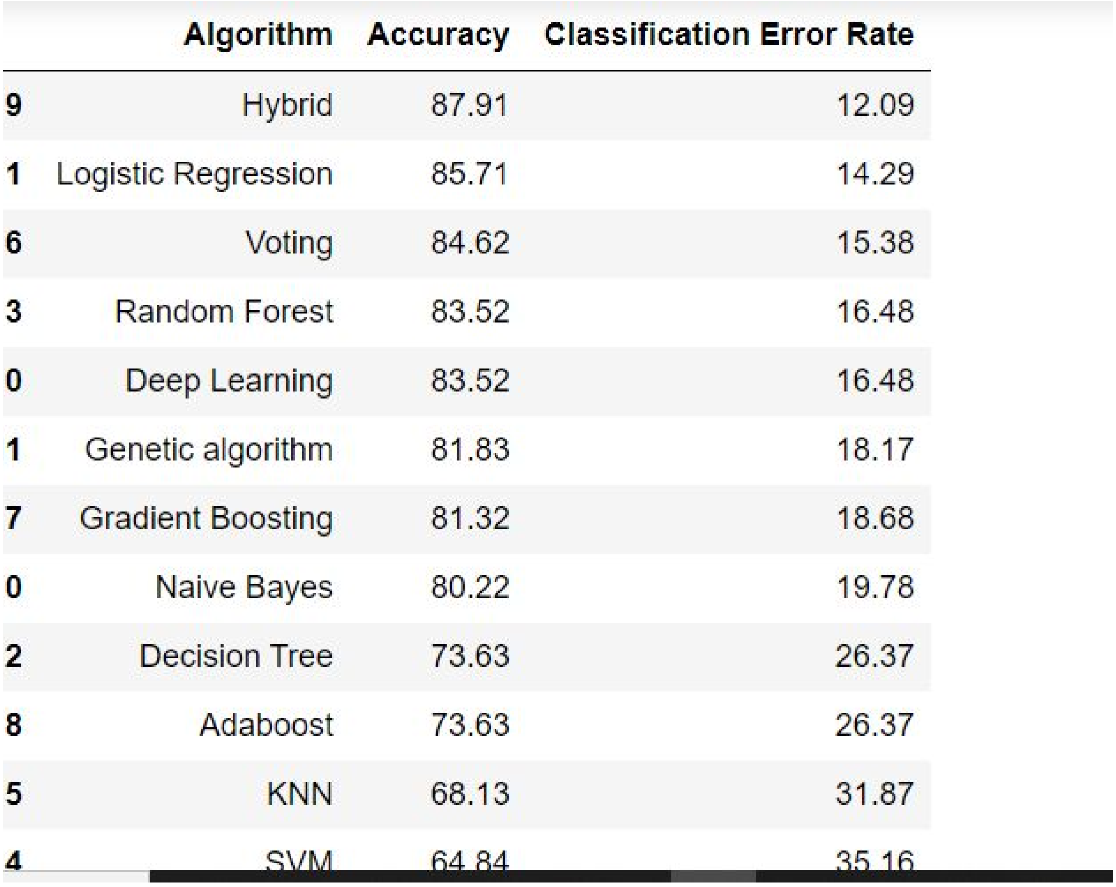


Figure 5.1 Bar plot depicting accuracy of all models

Figure 5.1 is a bar plot showing the accuracy of Naive Bayes, Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, KNN, Voting Classifier, Gradient Boosting, Adaboost , HRLFM, Deep Learning Model and Genetic Algorithm.

Table 5.2 Results of various algorithms



	Algorithm	Accuracy	Classification Error Rate
9	Hybrid	87.91	12.09
1	Logistic Regression	85.71	14.29
6	Voting	84.62	15.38
3	Random Forest	83.52	16.48
0	Deep Learning	83.52	16.48
1	Genetic algorithm	81.83	18.17
7	Gradient Boosting	81.32	18.68
0	Naive Bayes	80.22	19.78
2	Decision Tree	73.63	26.37
8	Adaboost	73.63	26.37
5	KNN	68.13	31.87
4	SVM	64.84	35.16

Table 5.2 shows the accuracy and error rate of all models applied on the dataset sorted in ascending order of classification error rate.Here we got HRLFM with highest accuracy.

6. CONCLUSION

6.1 Conclusion

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. After applying all the machine techniques we found out that logistic regression always stands out on top in terms of accuracy and so we combined logistic regression with random forest because random forest is a highly robust model. Because of the combination of those two models with the help of the hybrid method it resulted in the increase in accuracy in predicting heart disease.

6.2 Future scope

- Further extension of this study is highly desirable to direct the investigations to real-world raw data.
- 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.

REFERENCES

[1] M. A. Jabbar , Shirina Samreen , " Heart disease prediction system based on hidden naïve bayes classifier " , *https://ieeexplore.ieee.org/document/8053261*

[2] Abhishek Rairikar, Vedant Kulkarni, Vikas Sabale, Harshavardhan Kale, Anuradha Lamgunde " Heart disease prediction using data mining techniques", *2017 International Conference on Intelligent Computing and Control (I2C2), IEEE, Coimbatore, India, March 2017*

[3] Tulay Karayilan , Ozkan Kilic , " Prediction of heart disease using neural network", *https://www.researchgate.net/publication/320829299_Prediction_of_heart_disease_using_neural_network*

[4] A.T.Sayad, P. P. Halkarnikar, "Diagnosis of heart disease using neural network approach", *M. Tech, Fourth Semester, Department of Technology, Shivaji University, Kolhapur, India , Volume- 2, Issue-3, July-2014.*

[5] " Feature Selection using RFE ", *https://scikit-learn.org/stable/modules/generate/sklearn.feature_selection.RFE.html*

[6] Avinash Golande, Pavan Kumar T, " Heart Disease Prediction Using Effective Machine Learning Techniques ", *International Journal of Recent Technology and Engineering (IJRTE), ISSN: 2277-3878, Volume-8, Issue-1S4, June 2019*

[7] "Ensemble Voting Classifier", *https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.VotingClassifier.html*

[8] "Performance Metrics", *https://www.geeksforgeeks.org/confusion-matrix-machine-learning/*

[9] Eijaz Allibhai, " Deep learning model using Keras " *https://towardsdatascience.com/building-a-deep-learning-model-using-keras-1548ca149d37*

[10] Manas Narkar," Heart disease prediction using keras,deep learning " , *https://medium.com/@manasnarkar/heart-disease-prediction-using-keras-deep-learning-960a1b7b98ee*

[11] "Genetic Algorithm in machine learning", *https://dkopczyk.quantee.co.uk/genetic - algorithm/*

[12] "KNN", *https://www.tutorialspoint.com/machine_learning_with_python/machine_ learning_with_python_knn_algorithm_finding_nearest_neighbors.htm*