J[th] Component Project Report
ITA5007 - Data Mining and Business Intelligence
WINTER Semester 2022-23

TITLE

HEART DISEASE PREDICTION USING ML

Submitted by-

22MCA0137- KANISHK SHARMA

22MCA0174 - UDDASHAY KUMAR GUPTA

Submitted to-

PROF. EPHZIBAH E.P

Table of Contents

ABSTRACT

Machine Learning has found widespread applications in various industries worldwide, including the healthcare sector. With its potential to analyze large datasets and make accurate predictions, Machine Learning can be utilized in predicting the presence or absence of locomotor disorders, heart diseases, and other medical conditions. Early prediction of such diseases can provide valuable insights to healthcare professionals, enabling them to tailor their diagnoses and treatments on a per-patient basis. In this research project, our focus is on predicting possible heart diseases using Machine Learning algorithms. We conduct a comparative analysis of several popular classifiers, including decision tree, Naïve Bayes, logistic regression, support vector machine (SVM), and random forest. We also propose an ensemble classifier that combines strong and weak classifiers to improve accuracy and predictive performance. The proposed ensemble classifier can be trained and validated on multiple samples of data, which enhances its ability to make accurate predictions. Furthermore, we perform an in-depth analysis of existing classifiers, such as Ada-Boost and XG-Boost, to identify their strengths and weaknesses in predicting heart diseases. These classifiers are known for their ability to handle imbalanced datasets, and we investigate their performance in comparison to other classifiers. Our research aims to identify the most effective classifier for heart disease prediction, considering factors such as accuracy, predictive analysis, and potential for practical implementation in a healthcare setting. The results of our research have the potential to contribute to the field of cardiovascular health by providing insights into the performance of different classifiers and proposing an ensemble classifier that can enhance the accuracy of heart disease prediction. The findings of this study may have important implications for clinical decision-making and patient care, ultimately improving health outcomes for individuals at risk of heart diseases. Further research can be carried out to explore other Machine Learning techniques and optimize the ensemble classifier for real-world applications in healthcare settings.

Keywords: SVM; Naive Bayes; Decision Tree; Random Forest; Logistic Regression; XG-boost; confusion matrix;

INTRODUCTION

According to the World Health Organization, every year 12 million deaths occur worldwide due to heart disease. Heart disease is one of the biggest causes of morbidity and mortality among the population of the world. Prediction of cardiovascular disease is regarded as one of the most important subjects in the section of data analysis. The load of cardiovascular disease is rapidly increasing all over the world from the past few years. Many researches have been conducted in attempt to pinpoint the most influential factors of heart disease as well as accurately predict the overall risk. Heart Disease is even highlighted as a silent killer which leads to the death of the person without obvious symptoms. The early diagnosis of heart disease plays a vital role in making decisions on lifestyle changes in high-risk patients and in turn reduces the complications. Machine learning proves to be effective in assisting in making decisions and predictions from the large quantity of data produced by the health care industry. This project aims to predict future heart disease by analyzing data of patients which classifies whether they have heart disease or not using machine-learning algorithm. Machine Learning techniques can be a boon in this regard. Even though heart disease can occur in different forms, there is a common set of core risk factors that influence whether someone will ultimately be at risk for heart disease or not. By collecting the data from various sources, classifying them under suitable headings & finally analyzing to extract the desired data we can say that this technique can be very well adapted to do the prediction of heart disease.

Literature Survey

With growing development in the field of medical science alongside machine learning various experiments and researches has been carried out in these recent years releasing the relevant significant papers.

1) Heart disease prediction using machine learning algorithms

Publication Year: 2020

Author: Harshit Jindal1 , Sarthak Agrawal 1 , Rishabh Khera1 , Rachna Jain2 and Preeti Nag

Journal Name: IOP Conference Series: Materials Science and Engineering

Summary:

Data is directly retrieved from electronic records that reduce the manual tasks. Different algorithms of machine learning such as logistic regression and KNN are implemented to predict and classify the patient with heart disease.

The strength of the proposed model was quiet satisfying and was able to predict evidence of having a heart disease in a particular individual by using KNN and Logistic Regression which showed a good accuracy in comparison to the previously used classifier such as naive bayes etc

2) Prediction of Heart Disease Using Machine Learning Algorithms

Publication Year: 2019

Author: Santhana Krishnan J., Santhana Krishnan J., Geetha S.Nagra

Journal Name: IEEE

Summary:

Implemented decision tree and Naive Bayes algorithm for prediction of heart disease. In decision tree algorithm the tree is built using certain conditions which gives True or False decisions. The algorithms like SVM, KNN are results based on vertical or horizontal split conditions depends on dependent variables. But decision tree for a tree like structure having root node, leaves and branches base on the decision made in each of tree Decision tree also help in the understating the importance of the attributes in the dataset. They have also used Cleveland data set. Dataset splits in 70% training and 30% testing by using some methods. This algorithm gives 91% accuracy. The second algorithm is Naive Bayes, which is used for classification. It can handle complicated, nonlinear, dependent data so it is found suitable for heart disease dataset as this dataset is also complicated, dependent and nonlinear in nature.

3) Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques

Publication Year: 2019

Author: Senthilkumar Mohan; Chandrasegar Thirumalai; Gautam Srivastava

Journal Name: IEEE

Summary: Their main objective is to improve exactness in cardiovascular problems. The algorithms used are KNN, LR, SVM, NN to produce an improved exhibition level with a precision level of 88.7% through the prediction model for heart disease with hybrid random forest with linear model (HRFLM).

4) Machine Learning Techniques for Heart Disease Prediction

Publication Year: 2020

Author: A. Lakshmanarao,Y.Swathi, P.Sri Sai Sundareswar

Journal Name: INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH

Summary: Lakshmana Rao proposed "Machine Learning Techniques for Heart Disease Prediction" in which the contributing elements for heart disease are more. So, it is difficult to distinguish heart disease. To find the seriousness of the heart disease among people different neural systems and data mining techniques are used.

5) Heart Disease Prediction Using Effective Machine Learning Techniques

Publication Year: 2019

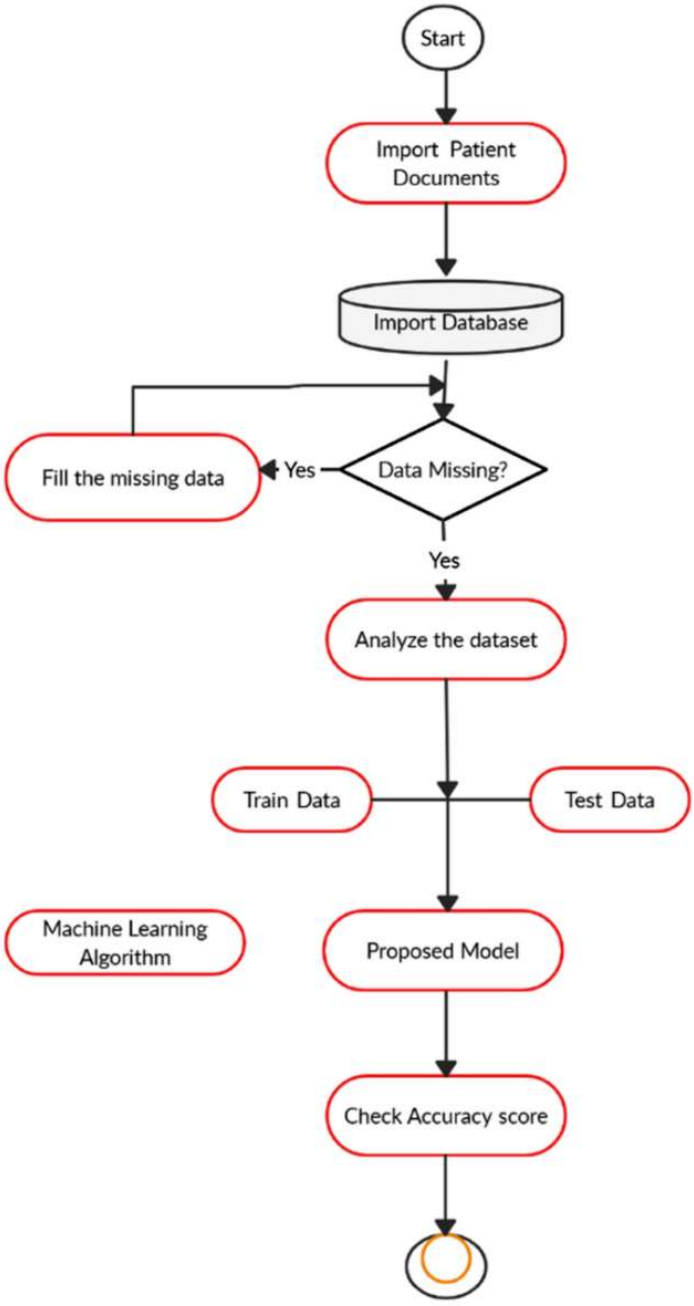
Author : Avinash Golande, Pavan Kumar T

Journal Name: International Journal of Recent Technology and Engineering (IJRTE)

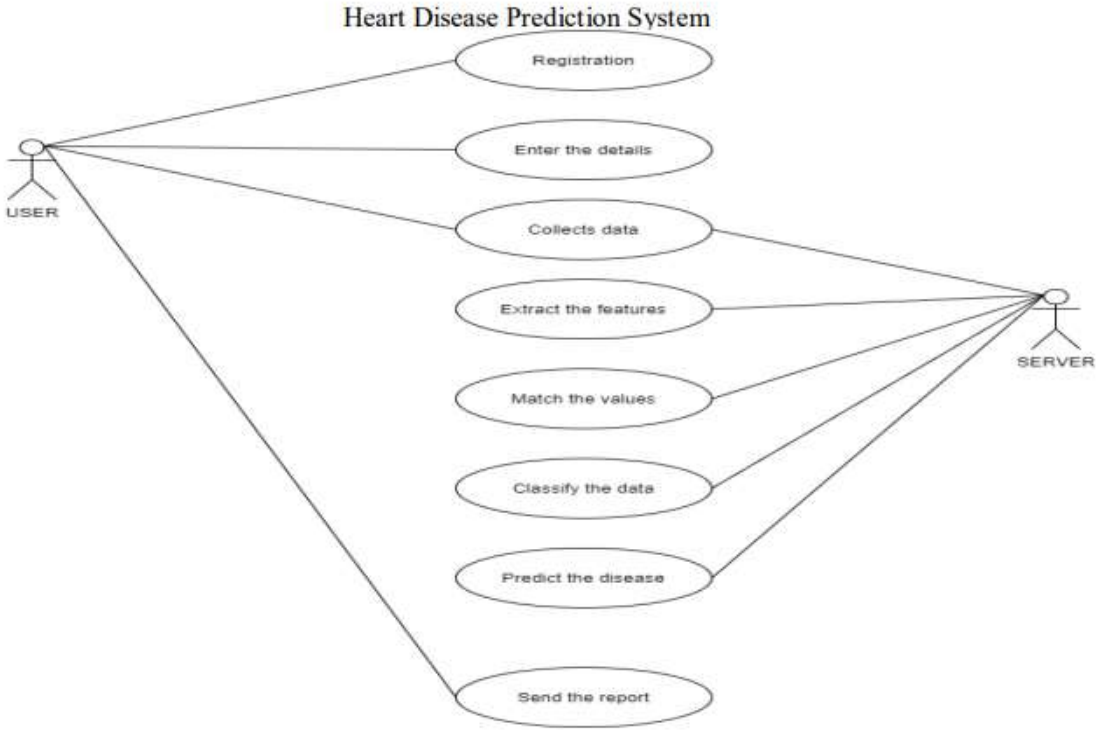
Summary:

In this paper, few data mining techniques are used that support the doctors to differentiate the heart disease. Usually utilized methodologies are k-nearest neighbour, Decision tree and Naïve Bayes. Other unique characterization-based strategies utilized are packing calculation, Part thickness, consecutive negligible streamlining and neural systems, straight Kernel selfarranging guide and SVM (Bolster Vector Machine).

SYSTEM DESIGN/ER DIAGRAM



USE CASE DIAGRAM



METHODOLOGY:

The working of the system starts with the collection of data and selecting the important attributes. Then the required data is preprocessed into the required format.

The data is then divided into two parts training and testing data. The algorithms are applied and the model is trained using the training data. The accuracy of the system is obtained by testing the system using the testing data. This system is implemented using the following modules.

1.) Collection of Dataset
2.) Selection of attributes
3.) Data Pre-Processing
4.) Balancing of Data
5.) Disease Prediction

3.1.1 Collection of dataset

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

2.1 DATASET DETAILS

• Of the 76 attributes available in the dataset,14 attributes are considered for the prediction of the output.

• Heart Disease UCI : https://archive.ics.uci.edu/ml/datasets/Heart+Disease

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target	
2	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1	
3	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1	
4	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1	
5	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1	
6	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1	
7	57	1	0	140	192	0	1	148	0	0.4	1	0	1	1	
8	56	0	1	140	294	0	0	153	0	1.3	1	0	2	1	
9	44	1	1	120	263	0	1	173	0	0	2	0	3	1	
10	52	1	2	172	199	1	1	162	0	0.5	2	0	3	1	
11	57	1	2	150	168	0	1	174	0	1.6	2	0	2	1	
12	54	1	0	140	239	0	1	160	0	1.2	2	0	2	1	
13	48	0	2	130	275	0	1	139	0	0.2	2	0	2	1	
14	49	1	1	130	266	0	1	171	0	0.6	2	0	2	1	
15	64	1	3	110	211	0	0	144	1	1.8	1	0	2	1	
16	58	0	3	150	283	1	0	162	0	1	2	0	2	1	
17	50	0	2	120	219	0	1	158	0	1.6	1	0	2	1	
18	58	0	2	120	340	0	1	172	0	0	2	0	2	1	
19	66	0	3	150	226	0	1	114	0	2.6	0	0	2	1	
20	43	1	0	150	247	0	1	171	0	1.5	2	0	2	1	
21	69	0	3	140	239	0	1	151	0	1.8	2	2	2	1	
22	59	1	0	135	234	0	1	161	0	0.5	1	0	3	1	
23	44	1	2	130	233	0	1	179	1	0.4	2	0	2	1	
24	42	1	0	140	226	0	1	178	0	0	2	0	2	1	
25	61	1	2	150	243	1	1	137	1	1	1	0	2	1	
26	40	1	3	140	199	0	1	178	1	1.4	2	0	3	1	
27	71	0	1	160	302	0	1	162	0	0.4	2	2	2	1	
28	59	1	2	150	212	1	1	157	0	1.6	2	0	2	1	

Figure: Dataset Attributes

Input dataset attributes

- Gender (value 1: Male; value 0 : Female)
- Chest Pain Type (value 1: typical type 1 angina, value 2: typical type angina, value 3: non-angina pain; value 4: asymptomatic)
- Fasting Blood Sugar (value 1: > 120 mg/dl; value 0:< 120 mg/dl)
- Exang – exercise induced angina (value 1: yes; value 0: no)
- CA – number of major vessels colored by fluoroscopy (value 0 – 3)
- Thal (value 3: normal; value 6: fixed defect; value 7:reversible defect)
- Trest Blood Pressure (mm Hg on admission to the hospital)
- Serum Cholesterol (mg/dl)
- Thalach – maximum heart rate achieved
- Age in Year
- Height in cms
- Weight in Kgs.

- Cholestrol
- Restecg

S. No.	Attribute	Description	Type
1	Age	Patient's age (29 to 77)	Numerical
2	Sex	Gender of patient(male-0 female-1)	Nominal
3	Cp	Chest pain type	Nominal
4	Trestbps	Resting blood pressure(in mm Hg on admission to hospital ,values from 94 to 200)	Numerical
5	Chol	Serum cholesterol in mg/dl, values from 126 to 564)	Numerical
6	Fbs	Fasting blood sugar>120 mg/dl, true-1 false-0)	Nominal
7	Resting	Resting electrocardiographics result (0 to 1)	Nominal
8	Thali	Maximum heart rate achieved(71 to 202)	Numerical
9	Exang	Exercise included agina(1-yes 0-no)	Nominal
10	Oldpeak	ST depression introduced by exercise relative to rest (0 to .2)	Numerical
11	Slope	The slop of the peak exercise ST segment (0 to 1)	Nominal
12	Ca	Number of major vessels (0-3)	Numerical
13	Thal	3-normal	Nominal
14	Targets	1 or 0	Nominal

TABLE2: Attributes of the dataset

3.1.2 Selection of attributes

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

Figure: Correlation matrix

3.1.3 Pre-processing of Data

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

Balancing of Data

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

(a) Under Sampling:

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

(b) Over Sampling:

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

Figure: Data Balancing

Prediction of Disease

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

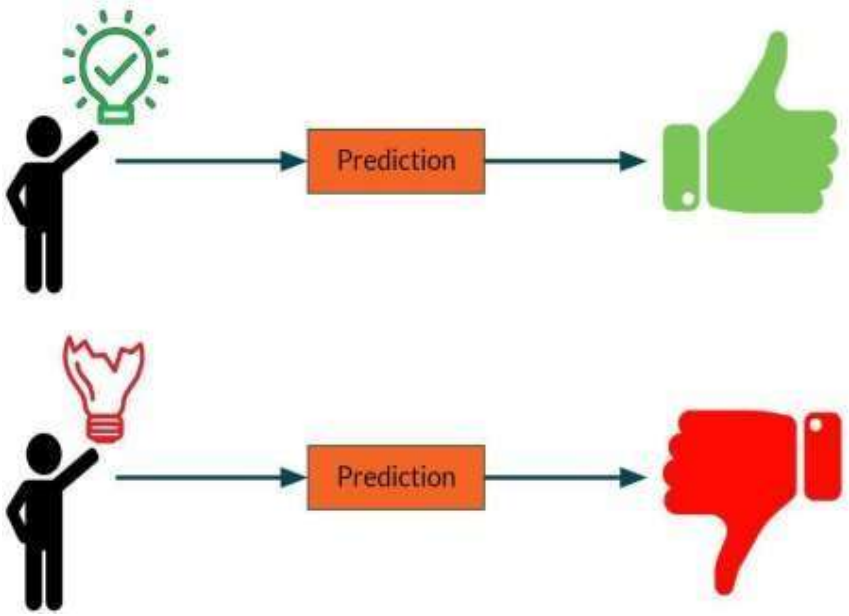


Figure: Prediction of Disease

WORKING OF SYSTEM

SYSTEM ARCHITECTURE

The system architecture gives an overview of the working of the system.

The working of this system is described as follows:
Dataset collection is collecting data which contains patient details. Attributes selection process selects the useful attributes for the prediction of heart disease. After identifying the available data resources, they are further selected, cleaned, made into the desired form. Different classification techniques as stated will be applied on preprocessed data to predict the accuracy of heart disease. Accuracy measure compares the accuracy of different classifiers.

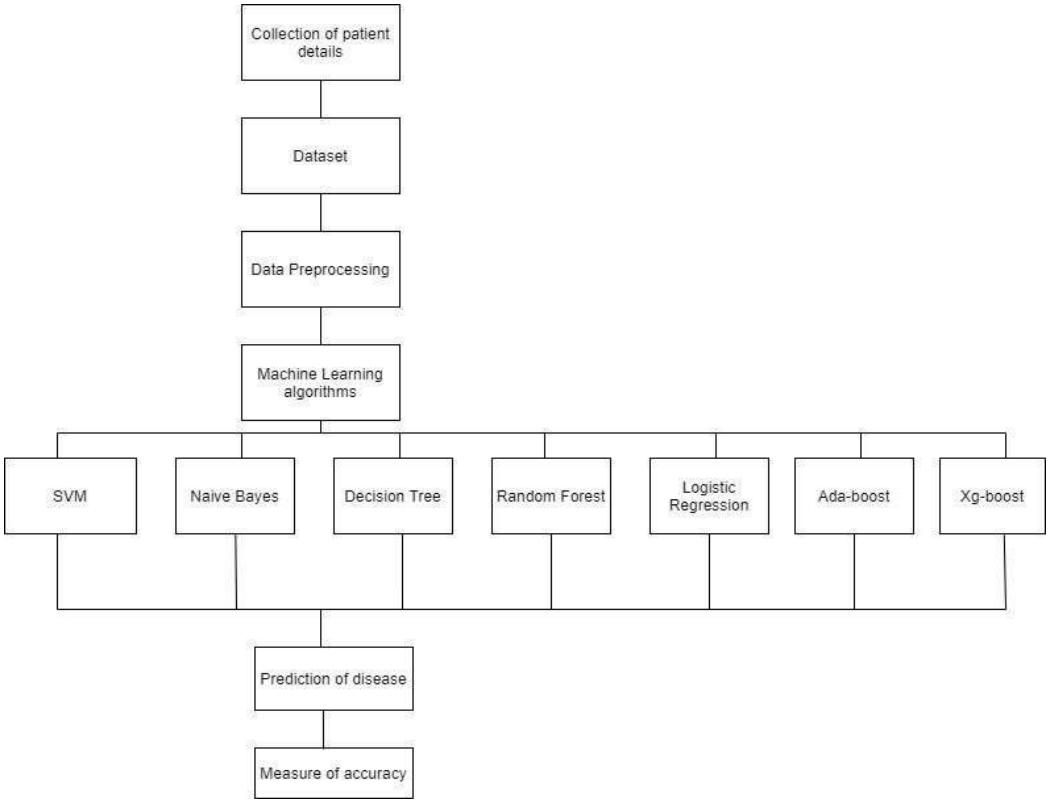


Figure:. SYSTEM ARCHITECTURE

ALGORITHMS

SUPPORT VECTOR MACHINE (SVM):

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called support vectors, and hence the algorithm is termed as Support Vector Machine.

Support vector machines (SVMs) are powerful yet flexible supervised machine learning algorithms which are used both for classification and regression. But generally, they are used

in classification problems. In the 1960s, SVMs were first introduced but later they got refined in 1990. SVMs have their unique way of implementation as compared to other machine learning algorithms. Lately, they are extremely popular because of their ability to handle multiple continuous and categorical variables.

The followings are important concepts in SVM -

Support Vectors - Data Points that are closest to the hyperplane are called support vectors. Separating line will be defined with the help of these data points.

Hyperplane - As we can see in the above diagram, it is a decision plane or space which is divided between a set of objects having different classes.

Margin - It may be defined as the gap between two lines on the closest data points of different classes. It can be calculated as the perpendicular distance from the line to the support vectors. Large margin is considered as a good margin and small margin is considered as a bad margin.

NAIVE BAYES ALGORITHM:

Naive Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems.It is mainly used in text classification that includes a high-dimensional training dataset.

Naive Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions. It is a probabilistic classifier, which means it predicts on the basis of the probability of an object. Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles. It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

The Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

The Naive Bayes algorithm is comprised of two words Naive and Bayes, Which can be described as:

Naive: It is called Naive because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the basis of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.
Bayes: It is called Bayes because it depends on the principle of Bayes' Theorem.

Bayes's theorem:

Bayes' theorem is also known as Bayes' Rule or Bayes' law, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.

The formula for Bayes' theorem is given as:

$$P(A|B)=\frac{P(B|A)P(A)}{P(B)}$$

Where,

P(A|B) is Posterior probability:Probability of hypothesis A on the observed event B.

P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

P(A) is Prior Probability: Probability of hypothesis before observing the evidence. P(B) is

Marginal Probability: Probability of Evidence.

DECISION TREE ALGORITHM

Decision Tree is a Supervised learning technique that can be used for both classification and regression problems, but mostly it is preferred for solving classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. In a Decision Tree, there are two nodes, which are the Decision Node and Leaf Node.

Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches. The decisions or the test are performed on the basis of features of the given dataset. It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions. It is called a Decision Tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure. In order to build a tree, we use the CART

algorithm, which stands for Classification and Regression Tree algorithm. A Decision Tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees. The Decision Tree Algorithm belongs to the family of supervised machine learning algorithms. It can be used for both a classification problem as well as for a regression problem. The goal of this algorithm is to create a model that predicts the value of a target variable, for which the decision tree uses the tree representation to solve the problem in which the leaf node corresponds to a class label and attributes are represented on the internal node of the tree.

There are various algorithms in Machine learning, so choosing the best algorithm for the given dataset and problem is the main point to remember while creating a machine learning model. Below are the two reasons for using the Decision Tree:

- Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.
- The logic behind the decision tree can be easily understood because it shows a tree-like structure.

Working:

In a Decision Tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of the root attribute with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node.

For the next node, the algorithm again compares the attribute value with the other sub-nodes and moves further. It continues the process until it reaches the leaf node of the tree. The complete process can be better understood using the below algorithm:

- Step-1: Begin the tree with the root node, says S, which contains the complete dataset.
- Step-2: Find the best attribute in the dataset using Attribute Selection Measure (ASM).
- Step-3: Divide the S into subsets that contains possible values for the best attributes.
- Step-4: Generate the Decision Tree node, which contains the best attribute.

- Step-5: Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and call the final node as a leaf node.

RANDOM FOREST ALGORITHM

Random Forest is a supervised learning algorithm. It is an extension of machine learning classifiers which include the bagging to improve the performance of Decision Tree. It combines tree predictors, and trees are dependent on a random vector which is independently sampled. The distribution of all trees are the same. Random Forests splits nodes using the best among of a predictor subset that are randomly chosen from the node itself, instead of splitting nodes based on the variables. The time complexity of the worst case of learning with Random Forests is O(M(dnlogn)) , where M is the number of growing trees, n is the number of instances, and d is the data dimension.

It can be used both for classification and regression. It is also the most flexible and easy to use algorithm. A forest consists of trees. It is said that the more trees it has, the more robust a forest is. Random Forests create Decision Trees on randomly selected data samples, get predictions from each tree and select the best solution by means of voting. It also provides a pretty good indicator of the feature importance. Random Forests have a variety of applications, such as recommendation engines, image classification and feature selection. It can be used to classify loyal loan applicants, identify fraudulent activity and predict diseases. It lies at the base of the Boruta algorithm, which selects important features in a dataset.

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

Assumptions:

Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random forest classifier:

- There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.
- The predictions from each tree must have very low correlations.

Algorithm Steps:

It works in four steps:

- Select random samples from a given dataset.
- Construct a Decision Tree for each sample and get a prediction result from each Decision Tree.
- Perform a vote for each predicted result.
- Select the prediction result with the most votes as the final prediction.

Advantages:

- Random Forest is capable of performing both Classification and Regression tasks.
- It is capable of handling large datasets with high dimensionality.
- It enhances the accuracy of the model and prevents the overfitting issue.

Disadvantages:

Although Random Forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks.

LOGISTIC REGRESSION ALGORITHM

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.

Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1. Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas logistic regression is used for solving the classification problems.

In Logistic regression, instead of fitting a regression line, we fit an "S"shaped logistic function, which predicts two maximum values (0 or 1). The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.

Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.

Advantages:

Logistic Regression is one of the simplest machine learning algorithms and is easy to implement yet provides great training efficiency in some cases. Also due to these reasons, training a model with this algorithm doesn't require high computation power.

The predicted parameters (trained weights) give inference about the importance of each feature. The direction of association i.e. positive or negative is also given. So we can use Logistic Regression to find out the relationship between the features.

This algorithm allows models to be updated easily to reflect new data, unlike Decision Tree or Support Vector Machine. The update can be done using stochastic gradient descent.

Logistic Regression outputs well-calibrated probabilities along with classification results. This is an advantage over models that only give the final classification as results. If a training example has a 95% probability for a class, and another has a 55% probability for the same class, we get an inference about which training examples are more accurate for the formulated problem.

Disadvantages:

Logistic Regression is a statistical analysis model that attempts to predict precise probabilistic outcomes based on independent features. On high dimensional datasets, this may lead to the model being over-fit on the training set, which means overstating the accuracy of predictions on the training set and thus the model may not be able to predict accurate results on the test set. This usually happens in the case when the model is trained on little training data with lots of features. So on high dimensional datasets, Regularization techniques should be considered to avoid over- fitting (but this makes the model complex). Very high regularization factors may even lead to the model being under-fit on the training data.

Non linear problems can't be solved with logistic regression since it has a linear decision surface. Linearly separable data is rarely found in real world scenarios. So the transformation of non linear features is required which can be done by increasing the number of features such that the data becomes linearly separable in higher dimensions.

ADABOOST ALGORITHM

Adaboost was the first really successful boosting algorithm developed for the purpose of binary classification. Adaboost is short for Adaptive Boosting and is a very popular boosting technique which combines multiple “weak classifiers” into a single “strong classifier”

Algorithm:

1. Initially, Adaboost selects a training subset randomly.
2. It iteratively trains the Adaboost machine learning model by selecting the training set based on the accurate prediction of the last training.
3. It assigns the higher weight to wrong classified observations so that in the next iteration these observations will get the high probability for classification.
4. Also, it assigns the weight to the trained classifier in each iteration according to the

accuracy of the classifier. The more accurate classifier will get high weight.

5. This process iterates until the complete training data fits without any error or until reached to the specified maximum number of estimators.
6. To classify, perform a "vote" across all of the learning algorithms you built

Advantages:

Adaboost has many advantages due to its ease of use and less parameter tweaking when compared with the SVM algorithms. Plus Adaboost can be used with SVM though theoretically, overfitting is not a feature of Adaboost applications, perhaps because the parameters are not optimized jointly and the learning process is slowed due to estimation stage-wise. This link is useful to understand mathematics. The flexible Adaboost can also be used for accuracy improvement of weak classifiers and cases in image/text classification.

Disadvantages:

Adaboost uses a progressively learning boosting technique. Hence high-quality data is needed in examples of Adaboost vs Random Forest. It is also very sensitive to outliers and noise in data requiring the elimination of these factors before using the data. It is also much slower than the XG-boost algorithm.

XGBOOST ALGORITHM

XG-boost is an implementation of Gradient Boosted decision trees. It is a type of Software library that was designed basically to improve speed and model performance. In this algorithm, decision trees are created in sequential form. Weights play an important role in XG-boost. Weights are assigned to all the independent variables which are then fed into the decision tree which predicts results. Weight of variables predicted wrong by the tree is increased and these the variables are then fed to the second decision tree. These individual classifiers/predictors then assemble to give a strong and more precise model. It can work on regression, classification, ranking, and user-defined predict.

Regularization: XG-boost has in-built L1 (Lasso Regression) and L2 (Ridge Regression) regularization which prevents the model from overfitting. That is why, XG-boost is also called regularized form of GBM (Gradient Boosting Machine).

While using Scikit Learn libarary, we pass two hyper-parameters (alpha and lambda) to XG-boost related to regularization. alpha is used for L1 regularization and lambda is used for L2 regularization.

Parallel Processing: XG-boost utilizes the power of parallel processing and that is why it is much faster than GBM. It uses multiple CPU cores to execute the model. While using Scikit Learn libarary, nthread hyper-parameter is used for parallel processing. nthread represents number of CPU cores to be used. If you want to use all the available cores, don't mention any value for nthread and the algorithm will detect automatically.

Handling Missing Values: XG-boost has an in-built capability to handle missing values. When XG-boost encounters a missing value at a node, it tries both the left and right hand split and learns the way leading to higher loss for each node. It then does the same when working on the testing data.

Cross Validation: XG-boost allows user to run a cross-validation at each iteration of the boosting process and thus it is easy to get the exact optimum number of boosting iterations in a single run. This is unlike GBM where we have to run a grid-search and only a limited values can be tested.

Effective Tree Pruning: A GBM would stop splitting a node when it encounters a negative loss in the split. Thus it is more of a greedy algorithm. XG-boost on the other hand make splits upto the max_depth specified and then start pruning the tree backwards and remove splits beyond which there is no positive gain.

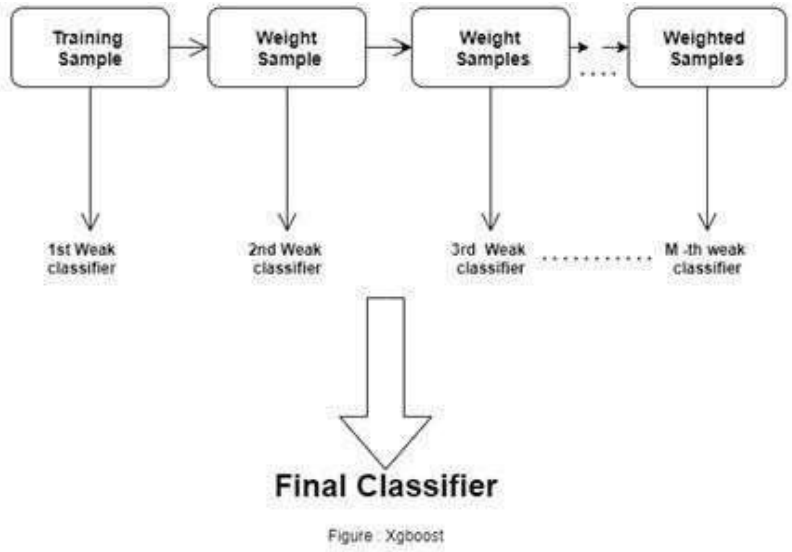


Figure : Xgboost

EXPERIMENTAL ANALYSIS

System Configuration:

Hardware requirements

Processer: Any Update Processer

RAM: Min 4GB

Hard Disk: Min 100GB

Software requirements

Operating System: Windows family

Technology: Python3.7

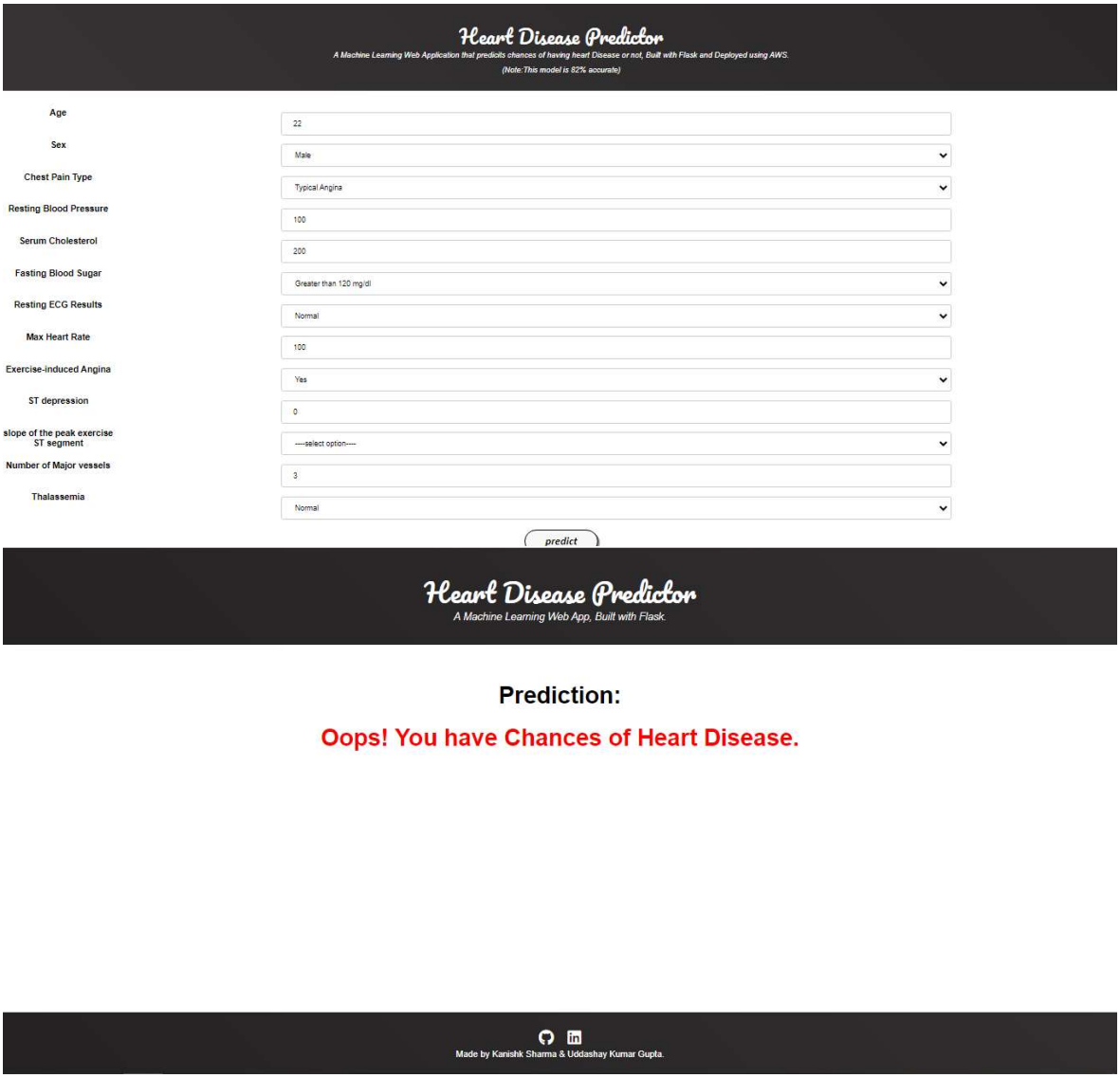
IDE: Jupiter notebook

Deployed Project:

Link: https://kanishksh4rma.github.io/heart-disease-prediction-ml//main.html

Prediction:

Great! You DON'T chances have Heart Disease.



PERFORMANCE ANALYSIS

In this project, various machine learning algorithms like SVM, Naive Bayes, Decision Tree, Random Forest, Logistic Regression, Adaboost, XG-boost are used to predict heart disease. Heart Disease UCI dataset, has a total of 76 attributes, out of those only 14 attributes are considered for the prediction of heart disease. Various attributes of the patient like gender, chest pain type, fasting blood pressure, serum cholesterol, exang, etc are considered for this project. The accuracy for individual algorithms has to measure and whichever algorithm is giving the best accuracy,that is considered for the heart disease prediction. For evaluating the experiment, various evaluation metrics like accuracy, confusion matrix, precision, recall, and f1-score are considered.

Accuracy- Accuracy is the ratio of the number of correct predictions to the total number of inputs in the dataset. It is expressed as:

$$\text{Accuracy} = (TP + TN) / (TP+FP+FN+TN)$$

Confusion Matrix- It gives us a matrix as output and gives the total performance of the system.

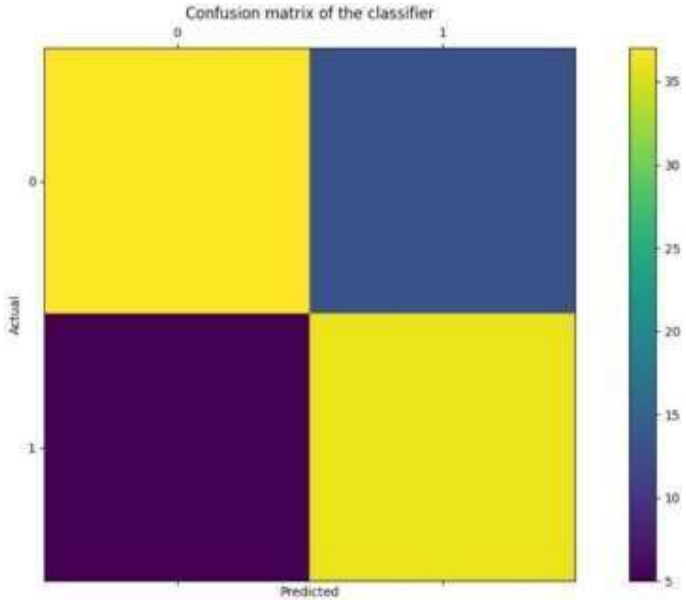


Figure: Confusion Matrix

Where

TP: True positive

FP: False Positive

FN: False Negative

TN: True Negative

Correlation Matrix: The correlation matrix in machine learning is used for feature selection. It represents dependency between various attributes.

Fig: Correlation matrix

Precision- It is the ratio of correct positive results to the total number of positive results predicted by the system.

It is expressed as:

Recall-It is the ratio of correct positive results to the total number of positive results predicted by the system.

It is expressed as:

F1 Score-It is the harmonic mean of Precision and Recall. It measures the test accuracy. The range of this metric is 0 to 1.

PERFORMANCE MEASURES
The highest accuracy is given by XG-boost.

```
Accuracy of svm: 0.8021978021978022
Accuracy of naive bayes: 0.7692307692307693
Accuracy of logistic regression: 0.7912087912087912
Accuracy of decision tree: 0.7582417582417582
Accuracy of random forest: 0.7912087912087912
```

```
Majority Voting accuracy score:  0.7912087912087912
Weighted Average accuracy score:  0.8131868131868132
Bagging accuracy score:  0.8021978021978022
Ada boost accuracy score:  0.7362637362637363
Gradient boosting accuracy score:  0.8131868131868132
```

RESULT

After performing the machine learning approach for training and testing we find that accuracy of the XG-boost is better compared to other algorithms. Accuracy is calculated with the support of the confusion matrix of each algorithm, here the number count of TP, TN, FP, FN is given and using the equation of accuracy, value has been calculated and it is concluded that extreme gradient boosting is best with 81% accuracy and the comparison is shown below.

TABLE: Accuracy comparison of algorithms Algorithm Accuracy

Algorithm	Accuracy
XG-boost	81.3%
SVM	80.2%
Logistic Regression	79.1%
Random Forest	79.1%
Naive Bayes	76.9%
Decision Tree	75.8%
Adaboost	73.6%

TABLE 2: Accuracy Table

CONCLUSION AND FUTURE WORK

Heart diseases are a major killer in India and throughout the world, application of promising technology like machine learning to the initial prediction of heart diseases will have a profound impact on society. The early prognosis of heart disease can aid in making decisions on lifestyle changes in high-risk patients and in turn reduce the complications, which can be a great milestone in the field of medicine. The number of people facing heart diseases is on a raise each year. This prompts for its early diagnosis and treatment. The utilization of suitable technology support in this regard can prove to be highly beneficial to the medical fraternity and patients. In this paper, the seven different machine learning algorithms used to measure the performance are SVM, Decision Tree, Random Forest, Naïve Bayes, Logistic Regression, Adaptive Boosting, and Extreme Gradient Boosting applied on the dataset.

The expected attributes leading to heart disease in patients are available in the dataset which contains 76 features and 14 important features that are useful to evaluate the system are selected among them. If all the features taken into the consideration then the efficiency of the system the author gets is less. To increase efficiency, attribute selection is done. In this n features have to be selected for evaluating the model which gives more accuracy. The correlation of some features in the dataset is almost equal and so they are removed. If all the attributes present in the dataset are taken into account then the efficiency decreases considerably.

All the seven machine learning methods accuracies are compared based on which one prediction model is generated. Hence, the aim is to use various evaluation metrics like confusion matrix, accuracy, precision, recall, and f1-score which predicts the disease

efficiently. Comparing all seven the extreme gradient boosting classifier gives the highest accuracy of 81%.

References

[1] Harshit Jindal1, Sarthak Agrawal1, Rishabh Khera1, Rachna Jain2 and Preeti Nagrath. Heart disease prediction using machine learning algorithms. International Journal of Computer Applications, 17(8), 43-8

[2] Dangare C S & Apte S S (2019). Improved study of heart disease prediction system using data mining classification techniques. International Journal of Computer Applications, 47(10), 44-8.

[3] Ordonez C (2020). Association rule discovery with the train and test approach for heart disease prediction. IEEE Transactions on Information Technology in Biomedicine, 10(2), 334-43.

[4] Shinde R, Arjun S, Patil P & Waghmare J (2019). An intelligent heart disease prediction system using k-means clustering and Naïve Bayes algorithm. International Journal of Computer Science and Information Technologies, 6(1), 637-9.

[5] Bashir S, Qamar U & Javed M Y (2020, November). An ensemble-based decision support framework for intelligent heart disease diagnosis. In International Conference on Information Society (i-Society 2014) (pp. 259-64). IEEE.