# Heart Disease Prediction Using Machine Learning Techniques


**Shekharesh Barik, Sambit Mohanty, Deepankar Rout, Subhra Mohanty, Akshaya Kumar Patra, and Alok Kumar Mishra**

**Abstract** Heart-related diseases or cardiovascular diseases are the primary purposes behind a large number of deaths on the planet in the course of the most recent couple of decades. It has risen as the most terrifying ailment around the world. Actually, in India, these issues are progressively awful; according to the Journal of the American College of Cardiology in India, the demise rate because of cardiovascular maladies increments around 34% in the middle of 1990–2016. Presently, we are in a time of data age where a huge quantity and variety of information is stored in different enterprises like retail, producing, medical clinic, and online networking. We can gather the information and break down the information to foresee the components and reasons for heart diseases so that safety measures can be taken to decrease the demise rate. There exists various types of information investigation instrument and procedure which requires an ideal informational collection; at that point, we can apply distinctive sort of machine learning strategies to anticipate whether the patient can be influenced by heart diseases or not by utilizing the recently gathered datasets. In this paper, we will exhibit how to utilize various kinds of machine learning models like K-nearest neighbor, decision tree classifier, and random forest classifier,

S. Barik (✉) · D. Rout · S. Mohanty
CSE Department, DRIEMS (Autonomous), Cuttack 754022, India
e-mail: shekharesh@gmail.com

D. Rout
e-mail: deepankarrout99@gmail.com

S. Mohanty
e-mail: subhramohanty008@gmail.com

S. Mohanty
Software Developer, SLFS Lab, Bhubaneswar 754001, India
e-mail: sambitmohanty778@gmail.com

A. K. Patra · A. K. Mishra
Department of Electrical and Electronics Engineering, Institute of Technical Education and Research, Siksha 'O' Anusandhan (Deemed to be University), Bhubaneswar 751030, India
e-mail: hiakp@yahoo.com

A. K. Mishra
e-mail: malok2010@gmail.com



G. Pradhan et al. (eds.), *Advances in Electrical Control and Signal Systems*,
Lecture Notes in Electrical Engineering 665,
https://doi.org/10.1007/978-981-15-5262-5_67

and furthermore make a presentation correlation among these models so that we can get accurate precision about a patient having heart disease (Chen et al. in 2011 Computing in Cardiology IEEE, 557–560, 2011, [1]), (Kishore et al. Heart attack prediction using deep learning, [2]).

**Keywords** Heart disease · Machine learning · Data analysis · K-nearest neighbor · Decision tree classifier · Random forest classifier

## 1 Introduction

The medical term for heart is cardio. So, heart diseases are also known as cardiovascular diseases (CVD). Heart disease is one of the most common diseases in our modern era, and it can prompt to reduce the life span of individuals these days. There are various reasons behind it, which leads it to increase day by day. Like in our everyday life, heavy routine and busy schedule increase work stress and the most significant reasons are people are addicted to having cigarette, tobacco, and habit to taking bad food impact to various heart diseases arising in human body. A report from World Health Organization (WHO) tells that 17.5 million people are dying every year because of heart diseases (Fig. 1).

The vital component of human body is heart. Due to various factors, blood clots in the vein. This may happen after a mild or severe heart attack. Clotting blocks the transmission of blood through the vein which is connected to heart. This may lead to a severe heart stroke or brain stroke resulting in loss of life. Parameter or factors for developing heart diseases are an important issue. High blood pressure is the biggest factor for heart problems. Diabetes is a very important parameter for heart disease.

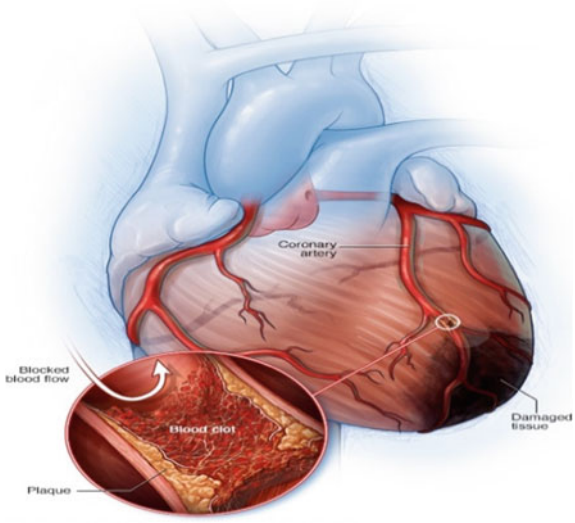


**Fig. 1** Diagram of blood clotting in heart

Person suffering from diabetes for long time is more likely to develop a heart disease. Nowadays, hypertension, bad food habit or lifestyle and stressful life are growing rapidly. These attributes are considered as one of the causes for heart disease. Age can also be considered as one of the factors for falling into heart diseases. Older people have more chance of getting into heart-related problems than younger ones. Other factors can be smoking, high cholesterol, obesity, etc. In this paper, we have taken some useful factors from dataset in order to predict the heart disease [3, 4].

The main challenge that lies in front of scientists and researchers is the accuracy in prediction of diseases and optimization of results. Nowadays, a large amount of data is available in various sectors. It can be social media, hospital, manufacturing, retail, etc. We can collect this data and analyze this data to predict the factors and causes of heart disease so that precautions can be taken to reduce the death rate. We can apply different kinds of machine learning techniques to predict whether the patient can be affected by heart diseases or not. We are using machine learning algorithms like K-nearest neighbor, decision tree classifier, and random forest classifier [5, 6].

## 2 Algorithm Used

In this paper, we have classified the patient's data to predict whether a patient has a heart disease or not. For this classification purpose, we have used Cleveland patient dataset available in UCI repository. The Cleveland dataset contains 303 numbers of patient records, and each record has 14 attributes. These 14 attributes are used for evaluating and predicting whether a patient has heart disease or not. If a patient has heart disease, then it is treated as 1, and if no heart disease, then treated as 0. We have used three classification algorithms for classifying and predicting whether a patient has heart disease or not. The machine learning algorithms are K-nearest neighbors, random forest classifier, and decision tree [7].

We have used Jupyter notebook which is a freely available software for performing machine learning operations. For machine learning, we need to import the sklearn module which contains all the essential algorithms and functions. We need to import Python NumPy module and Pandas module for data analysis purpose. To plot different graphs, we need to import Matplotlib module which contains all the methods related to plotting graphs. Our data is stored in a CSV file which needs to be imported to the notebook by using Python Pandas module as a data frame. After importing the data, we can apply various data analysis and machine learning algorithms for classification and prediction [1].

Before applying our machine learning algorithms on the dataset, we can see how the features of our dataset look like in plots; for this, we can use Matplotlib module (Fig. 2).

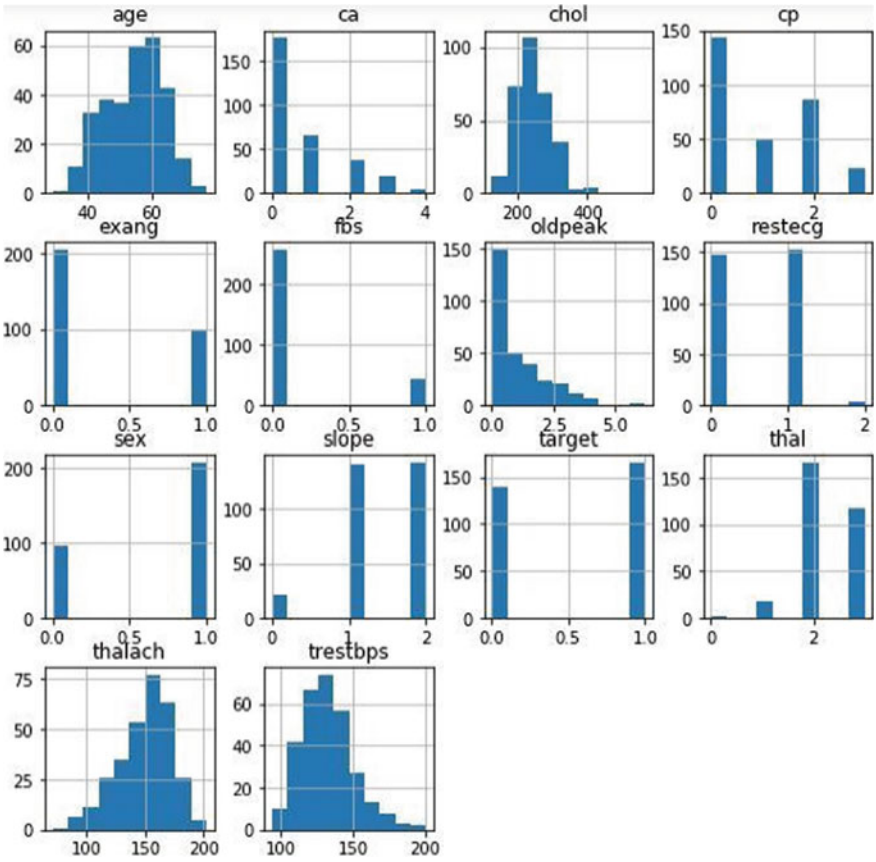


**Fig. 2** Features of datasets in plots

## *2.1 K-Nearest Neighbor (KNN) Algorithm*

KNN is a nonparametric supervised learning algorithm which creates the model structure from the given dataset. In KNN, $k$ is a number which decides how many nearest neighbors are used to provide the best result for the dataset. Let 'a' is point whose label is to be predicted then first we have to find the '$k$' closest point near to 'a'. Then, make a voting that how many nearest data points represent to a particular class. The class which gets highest voting, the point 'a' will belong to that class. For finding closet point near to our data point, we can use any of these distance calculation methods like Euclidean distance, Hamming distance, and Manhattan distance. Finding the best $k$ value for the dataset is a challenging task. For every dataset, $k$ value is different. However, some assumption says $k$ value is an odd number which provides the best results. We can say that the value of $k$ is the controlling factor for the problem. Research shows that if we take $k$ value a less number, then it may

lead to overfitting problem; i.e., noise has more impact on our prediction rather than actual prediction. If we take larger value of $k$, then it leads to expensive computation. So, we need to find an optimal value of $k$. We can find the best $k$ value by performing our operation on the dataset by using different $k$ values. The $k$ value which gives best result will be taken for that particular dataset [8].

To use KNN classifier in our code, we need to import a module named as sklearn module. In our dataset, there exist some categorical variables like sex, cp, fbs, restecg, exang, slope, ca, that. We need to convert these variables into dummy variables that means converting to numbers. For this purpose, we need to use standard scalar class from sklearn module. It will convert these categorical variables to dummy variables. Now, we can apply our KNN algorithm to this dataset. Before that, we have to separate the features and labels from the dataset. In $X$ variable, we have taken all the features of the dataset, and in $Y$ variable, we have taken the label, i.e., the target column of our dataset. We do not know which $k$ value is best suitable for our dataset. So, we will check every $k$ value on our model. In this case, we have taken $k$ value in a range of (1, 21); i.e., all values between this range are used as $k$ to give accurate score. For this purpose, we have used cross-validation techniques. It gives accurate measure of the performance of the machine learning model, that is, what we expect from our model. Here, we have used the cross-validation score for every value of $k$. The $k$ value which gives the highest score is chosen for further operation. To know all the cross-validation scores of all the values in the range (1, 21), we have plotted a graph by using Matplotlib (Fig. 3).

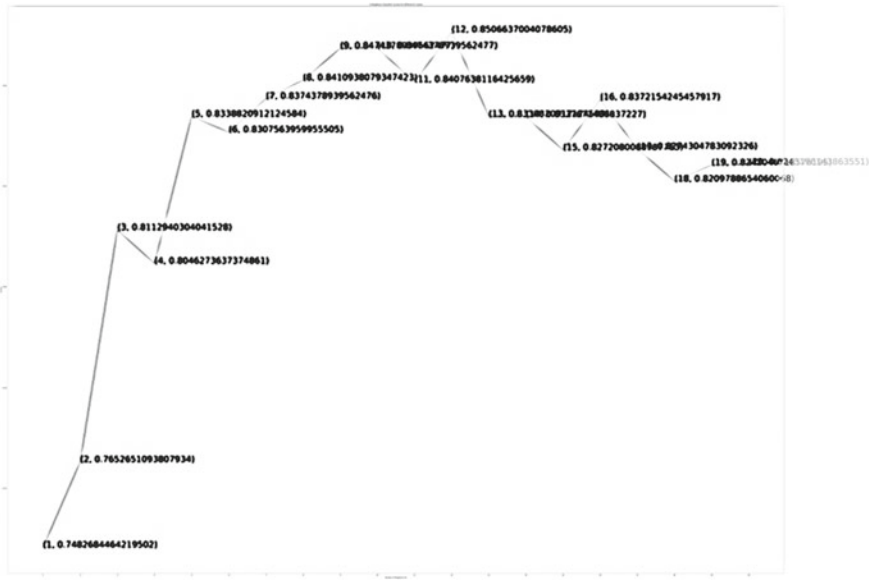


**Fig. 3** Finding the best value of '$k$' from a range of values

From this graph, it is clearly seen that at $k = 12$, the score is 0.85066 which is the highest among all the $k$ values in the range (1, 21). So, we have taken $k = 12$ for further operations. Then, we have applied this $k$ value to the KNN classifier and get the prediction value = 0.8506637004078605.

The snapshot of the code for getting the best value for '$k$' is given below:

```
knn_classifier = KNeighborsClassifier(n_neighbors = 12)
score=cross_val_score(knn_classifier,X,y,cv=10)
```

```
score.mean()
```

```
0.8506637004078605
```

## 2.2 Decision Tree

Decision tree is a tree-like structure. The internal node or non-leaf node represents the features, branch represents decision rules, and each leaf node represents the result. In the decision tree, the root of the tree is the topmost node. This root node learns to classify the dataset based on the features present in dataset. It partitions the dataset in a recursive manner. It gives the flowchart-like representation which is suitable for human understandable format and helps in understanding and taking decision. Decision tree is a white box-type machine learning algorithm model which gives detailed information of process behind it. It is a nonparametric method. Attribute selection measure process is used to break the dataset by using the features; it is a heuristic splitting rule to break the data points. Some of the popular selection measures are gain, gain ratio, Gini index, etc.

To use decision tree algorithm on our dataset, we need to import decision tree classifier from sklearn module. We need to divide the dataset into train and test. Train dataset is used for training our decision tree classifier, and test dataset is used to test the prediction values. We have used the cross-validation techniques to predict the score as it provides the most accurate results what we expect from the model. Decision tree problem is type of NP-complete problem. Practically, this algorithm uses heuristics which is a greedy algorithm. So, there exists local optimal solution at each node. That means, algorithm does not guarantee about returning global optimal tree. So, we need to use a random seed so that it can control the random choices; it can be any number. Here, we have taken random state = 7. In this paper, we have taken a parameter cv = 10 that is called cross-fold value. It indicates, we have divided the entire dataset to that many parts. Here, cv = 10 means the dataset is divided into ten parts between which nine parts are used for training the dataset and one part is used to test the dataset. Cross-validation technique gives better result in comparison with train and test methods; then, we calculate the mean value of the score to get accurate prediction value for the entire dataset. In this dataset, we get prediction value 0.755150. These values can be improved further by adjusting the parameters. The snapshot of the code is given below:

```
from sklearn.tree import DecisionTreeClassifier
dtree= DecisionTreeClassifier(random_state=7)
```

```
score=cross_val_score(dtree,X,y,cv=10)
```

```
score.mean()
```

```
0.7551501668520579
```

## 2.3 *Random Forest Classifier*

It is a type of supervised learning algorithm. As the name says, forest basically contains trees, i.e., decision trees. Random forest gives its prediction value basing upon the prediction of decision trees used. In random forest, many decision trees are present, and from the dataset, few features and labels are given to each decision tree on random basis. Then, these decision trees perform their operation on the supplied data points and predict their score. As multiple decision trees are present, so random forest makes a voting for classification-type problems. Majority of voting result provided by decision trees will decide the class of given problem. Random forest can also be used as repressors. In this case, the output is a continuous value which is the mean or median of all predicted values of the decision tree [9] (Fig. 4).

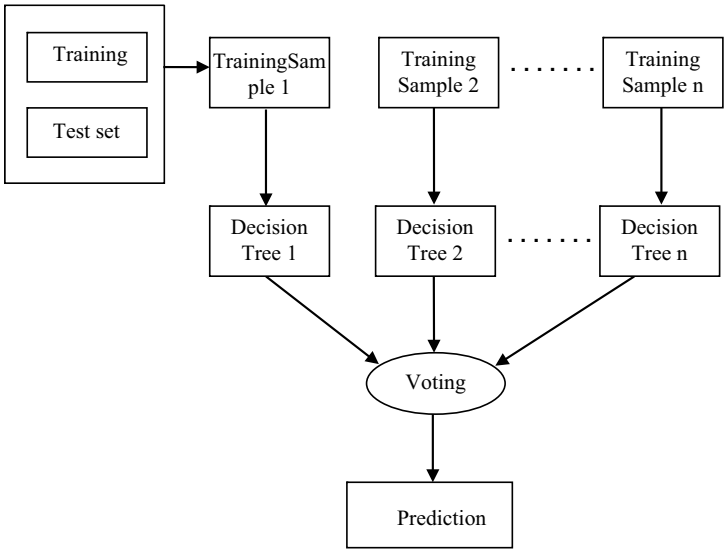


**Fig. 4** Control flow diagram for random forest

Random forest provides better results in comparison with decision trees as decision tree suffers from low bias and high variance problem so it may not give accurate prediction values. To overcome such type of problem in random forest, many number of decision trees are used so that the high variance problem can be reduced and it will provide more accurate prediction. In fact, random forest is the most popular machine learning algorithm used for both classification and regression problems. In real life, it is used for design of recommendation system, credit card fraud detection, classification of loyal loan applicants, disease prediction, and so on. Random forest may be slow in terms of prediction time as it uses many decision trees [7]. To use random forest classifier on our dataset, we need to import random forest classifier class from sklearn module. The snapshot of the code is given below:

```
from sklearn.ensemble import RandomForestClassifier
```

```
randomforest_classifier= RandomForestClassifier(n_estimators=10)

score=cross_val_score(randomforest_classifier,X,y,cv=10)
```

```
score.mean()
```

```
0.8012013348164627
```

Here, we have used a parameter **n_estimator** which is a controlling parameter of random forest algorithm. It indicates how many number of decision trees represent inside the random forest classifier. The more the number of value of n_estimator, i.e., the more the number of decision trees, then the more is the accuracy we can get.

## 3 Comparison Results

We have used three classification algorithms on our dataset to get the prediction for heart disease. For KNN, with $k$ value = 12, we get prediction value 0.85066. For decision tree classification, we get prediction value 0.7551. For random forest classification with ten decision trees, we get prediction value 0.8012. We have compared the above results with the results of another paper mentioned in the reference section list. We found that paper has the similar goal as of ours but with some other set of machine learning algorithms for heart disease prediction. It uses the machine learning algorithms like naïve Bayes classifier, support vector machine, decision tree, and K-nearest neighbor. The comparison result is matched with the results of our case. In the referred paper, we find that the prediction accuracy for naïve Bayes classifier is 83.4983, for support vector machine is 84.1584, for decision tree is 77.5578, and for K-nearest neighbor is 76.2376. Hence, it is clearly seen that KNN model predicts more accurately in our case (Fig. 5).

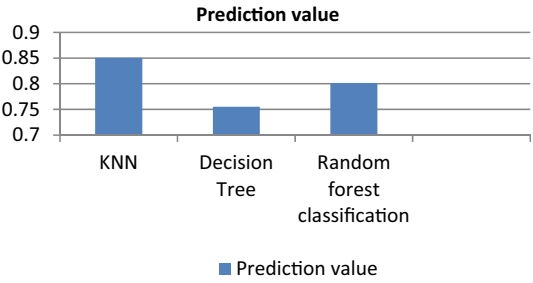


**Fig. 5** Comparison model for the three classification algorithms

## 4 Conclusion and Future Scope

The comparison results are matched with the results of another paper as mentioned in the reference list. It is found that KNN in our case gives the highest accuracy in comparison with all other machine learning algorithms. In this paper, we have applied only three machine learning algorithms on the dataset to predict the accuracy value but there exist other machine learning algorithms which can be applied on the dataset which may perform better. We can also improve the performance of these algorithms by adjusting tuning parameters and other adjustment parameters [10].

## References

1. Chen, A.H., Huang, S.Y., Hong, P.S., Cheng, C.H., Lin, E.J.: HDPS: Heart disease prediction system. In: 2011 Computing in Cardiology IEEE, pp. 557–560 (2011)
2. Kishore, A., Kumar, A., Singh, K., Punia, M., Hambir, Y.: Heart attack prediction using deep learning (2018)
3. Almarabeh, H., Amer, E.: A study of data mining techniques accuracy for healthcare. Int. J. Comput. Appl. **168**(3), 12–17 (2017)
4. Krishnaiah, V., Narsimha, G., Chandra, N.S.: Heart disease prediction system using data mining technique by fuzzy K-NN approach. In: Emerging ICT for Bridging the Future—Proceedings of the 49th Annual Convention of the Computer Society of India (CSI), vol. 1, pp. 371–384, Springer, Switzerland (2015)
5. Dinesh Kumar, G.: Prediction of cardiovascular disease using machine learning algorithms. In: Proceeding of 2018 IEEE International Conference on Current Trends Toward Converging Technologies, Coimbatore, India (2018)
6. Khourdifi, Y., Mohamed, B.: Heart disease prediction and classification using machine learning algorithms optimized by particle Swarm optimization and Ant Colony optimization. Int. J. Intell. Eng. Syst. **12**(1) (2019)
7. Rairikar, A., Kulkarni, V., Sabale, V., Kale, H., Lamgunde, A.: Heart disease prediction using data mining techniques. In: Intelligent Computing and Control (I2C2), IEEE International Conference on 2017 Jun, pp. 1–8 (2007)
8. Santhana Krishnan, J., Geetha, S.: Prediction of heart disease using machine learning algorithms. In: 1st International Conference on Innovations in Information and Communication Technology (ICIICT), IEEE (2019)

9. Xu, S., Zhang, Z., Wang, D., Hu, J., Duan, X., Zhu, T.: Cardiovascular risk prediction method based on CFS subset evaluation and random forest classification framework. In: 2017 IEEE 2nd International Conference on Big Data Analysis (2017)
10. Kaur, A., Arora, J.: Heart disease prediction using data mining techniques: A survey. Int. J. Adv. Res. Comput. Sci. **9**(2), 569–572. Mar-Apr 2018 (2018)