Research and Application of Heart Disease Prediction Model Based on Machine Learning

Yongli Bao

International College, Chongqing University of Posts and Telecommunications, Chongqing, 400065, China

Abstract. As heart disease has become the leading cause of death worldwide, early and accurate prediction is crucial to help doctors make initial judgments about patients and improve their survival rates. This study aims to improve the accuracy and efficiency of heart disease prediction through Machine learning (ML) methods to help medical diagnosis. A heart disease dataset was used in the study, and multiple ML models were used to analyze multiple key health features, and the model performance was verified through a test set. This paper concludes that Logistic regression and random forests perform well in this task and have high practical value. Future research can stack models and optimize data sources to improve the practical performance of the model. This study provides a basic framework for building an intelligent medical auxiliary diagnosis system, which helps to achieve early prevention and timely judgment of heart disease, thereby improving the overall efficiency of medical services.

1 Introduction

Worldwide, cardiovascular diseases (CVDs) are the cause of a relatively high percentage of people's deaths. According to the definition of the WHO, CVDs are a class of disorders affecting the heart and blood vessels. Their fundamental characteristic is the blockage of blood arteries, which prevents blood from reaching the heart and brain adequately and impairs the heart's and the brain's tissues' capacity to function normally [1]. According to the relevant epidemiologic and preventive statistical reports, it is evident that the lethality of CVDs is increasing globally. Compared with 1990, the mortality rate of CVDs such as ischemic heart disease has increased significantly [2]. This shows that cardiovascular disease is still one of the major diseases that endanger human health. There are many causes of cardiovascular disease, including diet, social environment, psychological factors, etc. These factors also increase the difficulty of diagnosing cardiovascular disease. Therefore, diagnosing cardiovascular disease is a complex process but has important significance.

However, diagnosing cardiovascular disease requires a lot of medical resources. Medical experts need to spend a lot of time reading the patient's medical history to help them make more accurate judgments. ML is one of the branches of artificial intelligence. The emergence of ML makes up for the limitations of traditional medical methods. ML can help doctors

Corresponding author: 2021215000@stu.cqupt.edu.cn

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

solve the problem of spending a lot of time reading patient records and making preliminary judgments. ML obtains clinical decisions made by a large number of doctors and medical records of millions of patients through a large amount of training. Currently, a variety of AI models have begun to be gradually put into use in the medical industry, such as prediction models for cancer in patients [3,4]. Different ML models have also been used in the health service industry [5]. Among them, ML has great advantages in predicting and diagnosing CVDs. ML can quickly process a large number of past cases to assist decision-makers in making diagnoses [6]. For example, Saba Bashir et al. used naive Bayes, decision tree, SVM and other methods to predict heart disease with an accuracy of 87. 37% [7]. Dutta used a convolutional neural network (CNN) to diagnose coronary heart disease [8]. V Krishnaiah et al. used the fuzzy K-NN method to establish a heart disease prediction model [9].

The purpose of this study is to assess how well several ML models—including individual models such as Decision Trees and logistic Regression—perform in predicting heart disease. By comparing their accuracy, precision, recall, and F1 score, their performance is evaluated.

2 Material and Method

2.1 Data source

The data set used in this article is heart disease from Cleveland UCI [10]. This data set is multivariate. It consists of 14 attributes. The model in this article can predict whether a patient has heart disease based on the 14 attributes.

Data preprocessing is a key step in ML. Its main function is to clean, transform and format the raw data to ensure data quality and consistency. After data preprocessing, the original data set can remove noise data, handle missing values, standardize or normalize data, and unify data in different formats into a processable format. This process is crucial to improving the accuracy of the model, reducing errors and speeding up training.

When conducting data preprocessing for this experiment, data cleaning is performed first to delete missing values and ensure data integrity during training.

Visualize the data distribution after that. The data set's label distribution is displayed in Figure 1. There are approximately 160 samples with the label "0" and 140 examples with the label "1". Although the number of samples of the two types of labels is different, they are relatively balanced overall. This ensures that the data categories are as balanced as possible.

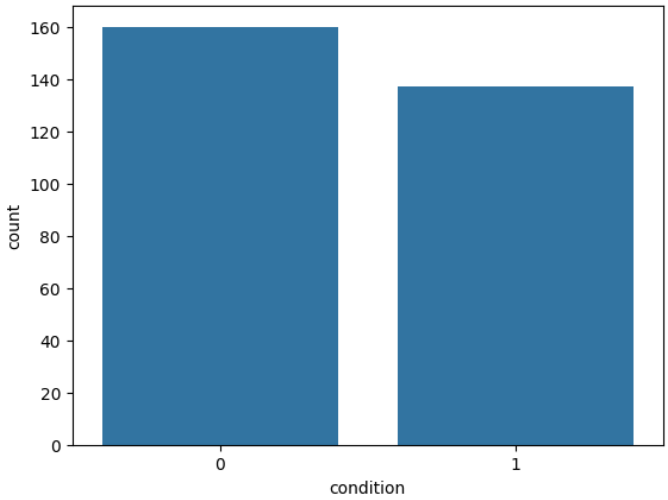

Fig. 1. Dataset label distribution (Photo/Picture credit : Original)

Simultaneously, the most predictive features are chosen, the model's complexity is decreased, and the model's capacity for generalization is enhanced by the application of recursive feature elimination and cross-validation (RFECV). The dimensions of these traits are then broken down into eight primary components using principal component analysis (PCA): 'age', 'cp', 'treetops', 'chol', 'thali', 'old peak', 'ca', and 'thal'. The purpose is to retain the main information of the data and reduce the computational complexity and the risk of overfitting. Lastly, split the test and training sets in order to get ready for the next model training.

2.2 Method

2.2.1 Logistic Regression

Logistic regression is a supervised ML method used to solve binary classification problems. Depending on the purpose, it can be used to complete multi-classification tasks or prediction and probability modelling tasks by linearly combining input values using sigmoid or logistic functions and coefficient values. Logistic regression maps the output values of the linear model to a range of 0-1 to predict the probability of an event occurring [11].

Import the Logistic Regression model from the sklearn library. To avoid overfitting of the model, apply L2 regularization and set the maximum number of iterations to 100.

2.2.2 Decision Tree

Decision tree is a supervised learning technique. It selects the best features from the data features as decision nodes. Based on the value of the attribute, the data is divided into several subsets and then the process is applied to other subsets.

Import the decision tree classifier DecisionTreeClassifier from the sklearn library to build a classification model.

When separating nodes, using the Gini impurity as the criterion. The impurity of the samples in the node is measured by this coefficient. Set the minimum number of samples for node splitting to two and the maximum depth to 300.

2.2.3 Random Forest

In order to enhance the performance of classification or regression, Random Forest builds numerous decision trees using the concept of ensemble learning and then applies ensemble learning to these trees. Random forest predicts by combining the output of several decision trees to produce the ultimate outcome. This method can effectively solve the defect of weak generalization ability of a single decision tree [12]. Therefore, decision trees perform well in processing high-dimensional data and preventing overfitting.

Use the random forest classifier RandomForestClassifier imported from the sklearn library to build 300 decision trees and use the Gini coefficient as the partition node, setting the maximum depth of each decision tree to 300.

2.2.4 SVM

A supervised learning approach for regression and classification problems is support vector machines. The primary objective is to create a linear decision plane by translating an instance's feature vector into a high-dimensional space. This plane has the greatest ability to

discriminate between the two cases that need to be classified. The high degree of generalization capacity of the model is guaranteed by this decision surface [13,14]. SVM can be used to solve nonlinear and high-dimensional issues.

To create a classification model, import the SVC support vector machine classifier from the sklearn package. In order to map the input to a high-dimensional space and make the data linearly separable in the high-dimensional space, the radial basis function (RBF) is selected as the kernel function. Then, the penalty coefficient is set to 1.0. Gamma='scale', degree=3, and the maximum number of iterations are set to zero.

2.2.5 KNN

K nearest neighbor (KNN) is a supervised ML algorithm and one of the simplest and most basic algorithms in ML algorithms. It determines the category or predicted value of the input sample by calculating the distance between samples. For example, when a new sample point is input, the category of the sample point is identified based on the categories of its nearest K sample points. The metric between the two nearest adjacent points usually uses the Euclidean distance or Manhattan distance.

Import K-nearest neighbor classifier KNeighborsClassifier from sklearn library to build a classification model. Set the number of neighboring instances considered for classification to 3. Set to automatically select the most appropriate algorithm to calculate the nearest neighbors. Possible choices include ball_tree, kd_tree, brute, or the best algorithm selected based on the data. The distance metric used is Euclidean distance.

2.2.6 Naive Bayes

A straightforward probabilistic classifier from supervised learning in ML, Naive Bayes is based on the Bayesian theorem. It assumes that each feature is independent of the other, that is, the impact of each feature on the category is independent. These features work together to obtain the final predicted probability. The Naive Bayes algorithm is simple, efficient and easy to implement, and is more suitable for processing high-dimensional data sets.

The Gaussian Naive Bayes classifier GaussianNB is imported from the sklearn library to build a classification task classifier suitable for continuous features, assuming that the value of each feature follows a Gaussian distribution. The parameter var_smoothing=1e-09 is set to add a constant to the variance of each feature to prevent the denominator from being 0 during calculation.

2.2.7 XGBoost

The eXtreme Gradient Boosting (XGBoost) is based on the traditional Gradient Boosting Decision Tree (GBDT). XGBoost has been improved in the following aspects compared with GBDT:

To keep the model from overfitting and to lower its complexity, an explicit regularization term is first added. Th, the second-order Taylor expansion (Hessian matrix) is used to approximate the loss function, making the model converge faster and perform better. Thirdly, a sparse-aware split search algorithm is proposed, which can effectively process sparse matrices by introducing a default direction to process the actual value.

Finally, it supports parallel and distributed computing and improves the training efficiency of the model by building a Column Block structure, so that the model can handle large-scale data sets well [15].

Import XGBClassifier from the xgboost library to build the XGBoost model. Set the maximum tree depth to three and the number of trees to one hundred. For every leaf node, set the minimum sum of weights to 2.

2.3 Model Evaluation

The four parameters commonly used for performance evaluation of ML models are True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). These parameters represent the number of positive and negative examples correctly predicted by the model and the number of negative and positive examples incorrectly predicted by the model.

Accuracy serves as a gauge for the model's overall performance by showing the percentage of accurate predictions the model made across the whole collection of data.

Precision reflects the accuracy of the model in identifying the positive class, that is, how much proportion of the model is correct when predicting the positive class. High precision means that the model has fewer false positives and can identify positive samples more accurately.

Recall shows how well the model can identify positive samples, or what percentage of all positive samples the model properly recognizes.

The F1 value is a measure of the balance between recall and precision, accounting for both. An extremely helpful evaluation indicator is the F1 value when the weights of recall and precision are equal.

3 Results Analysis

Table 1. Model Accuracy Comparison

Serial number	ML model	Accuracy
1	Logistic Regression	0. 833
2	Decision Tree	0. 717
3	Random Forest	0. 817
4	Support Vector Machine	0. 550
5	KNN	0. 483
6	Naive Bayes	0. 800
7	XGBoost	0. 767

From Table 1, Logistic Regression has the highest accuracy (0. 833), indicating that this model has the most accurate prediction on the entire dataset. Random Forest follows closely with an accuracy of 0. 817, which performs well.

However, SVM and KNN have lower accuracy, especially KNN (0. 483), indicating that they perform poorly on this task.

Table 2. Model Recall Comparison

Serial number	ML model	Recall

1	Logistic Regression	0. 821
2	Decision Tree	0. 677
3	Random Forest	0. 774
4	Support Vector Machine	0. 519
5	KNN	0. 444
6	Naive Bayes	0. 750
7	XGBoost	0. 750

According to Table 2, Logistic Regression has the highest recall rate of 0. 821, indicating that this model performs best in capturing positive samples. Random Forest has a recall rate of 0. 774, which performs well. Naive Bayes and XGBoost have a recall rate of 0. 750, which is weaker than the above models. SVM and KNN have lower recall rates of 0. 519 and 0. 444, respectively, indicating that these two models are weak in capturing positive samples.

Table 3. Model Precision Comparison

Serial number	**ML model**	**Precision**
1	Logistic Regression	0. 821
2	Decision Tree	0. 712
3	Random Forest	0. 814
4	Support Vector Machine	0. 509
5	KNN	0. 436
6	Naive Bayes	0. 800
7	XGBoost	0. 750

From Table 3, the precision of Logistic Regression and Random Forest is 0. 821 and 0. 814 respectively, indicating that they have fewer false positives when predicting the positive class. Naive Bayes and XGBoost also have high precision, 0. 800 and 0. 750 respectively, which perform well. SVM and KNN precision are also low, especially KNN (0. 436), indicating that these two models have more false positives when predicting the positive class.

Table 4. Model f1_score Comparison

Serial number	**ML model**	**f1_score**
1	Logistic Regression	0. 821
2	Decision Tree	0. 750
3	Random Forest	0. 857

4	Support Vector Machine	0. 500
5	KNN	0. 429
6	Naive Bayes	0. 857
7	XGBoost	0. 750

From Table 4, Random Forest and Naive Bayes have the highest F1 value (0. 857), indicating that they have found a good balance between precision and recall. Logistic Regression: The F1 value is 0. 821, which is also excellent. XGBoost and Decision Tree are mediocre. SVM and KNN: F1 values are 0. 500 and 0. 429 respectively, indicating that these two models perform poorly in balancing precision and recall.

In general, Logistic Regression and Random Forest are the most balanced models in the four tables, with high accuracy, F1 value, recall rate and precision. SVM and KNN perform poorly in all four parameters and may not be suitable for the classification task of this dataset, or the stacking model method can be used to improve performance [16].

4 Conclusion

This study examines how well several ML models predict the development of heart disease. In order to assess how well several ML models—including Logistic Regression, Decision Trees, and other complex models—perform in handling feature complexity and sample imbalance, a heart disease dataset was processed utilizing feature selection and data preparation approaches. Finally, the performance of each model in the cardiovascular disease prediction task was evaluated using indicators such as accuracy, F1 value, recall, and precision. The results show that Logistic Regression and Random Forest performed best in comparison with other ML models. SVM and KNN models are not suitable for such tasks.

This study has a certain reference value for using ML models in the medical field. Accurate prediction models can help hospitals better manage medical resources. By using models to read case records, high-risk patients can be quickly identified and necessary examinations and treatments can be prioritized, which can improve the overall medical service efficiency of the medical system. Future research could superimpose simple models to improve the performance of the model. Alternatively, more effective models could be used to improve the efficiency of the model in this specific task. By collecting more information about heart disease from different populations and geographic regions, people can further improve the generalizability and applicability of the model. In addition, to provide the model with a more comprehensive source of information and improve the usefulness of the model in real-world applications, time series data, genetic data, and other types of data could be used.

Reference

1. World Health Organisation. Noncommunicable diseases [updated 11 June, 2021; cited 2024 6 October,]. Available from: https://www. who. int/zh/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)
2. S.S. Martin, A.W. Aday, Z.I. Almarzooq, C.A.M. Anderson, P. Arora, C.L. Avery, American Heart Association Council on Epidemiology and Prevention Statistics Committee and Stroke Statistics Subcommittee, 2024 Heart Disease and Stroke

Statistics: A Report of US and Global Data From the American Heart Association. Circulation 149, 8 (2024).

3. W. Li, S. Lin, Y. He, J. Wang, Y. Pan, Deep learning survival model for colorectal cancer patients (DeepCRC) with Asian clinical data compared with different theories. Arch. Med. Sci. 19, 264–269 (2023).
4. J.-O. Jung, N. Crnovrsanin, N.M. Wirsik, H. Nienhüser, L. Peters, F. Popp, et al., Machine learning for optimized individual survival prediction in resectable upper gastrointestinal cancer. J. Cancer Res. Clin. Oncol. 149, 1691–1702 (2023).
5. P. Doupe, J. Faghmous, S. Basu, Machine Learning for Health Services Researchers. Health 22, 808–815 (2019).
6. M. Mohammed, M.B. Khan, E.B.M. Bashier, Machine learning: algorithms and applications. CRC Press (2016).
7. S. Bashir, U. Qamar, F.H. Khan, A multicriteria weighted vote-based classifier ensemble for heart disease prediction. Comput. Intell. 32, 615–645 (2016).
8. A. Dutta, et al., An efficient convolutional neural network for coronary heart disease prediction. Expert Syst. Appl. 159, 113408 (2020).
9. V. Krishnaiah, G. Narsimha, N.S. Chandra, Emerging ICT for Bridging the Future—Proceedings of the 49th Annual Convention of the Computer Society of India (CSI) Volume 1 (2015).
10. Heart Disease Cleveland UCI, Kaggle (2019). https://www.kaggle.com/datasets/cherngs/heart-disease-cleveland-uci/code
11. R. Katarya, S.K. Meena, Machine Learning Techniques for Heart Disease Prediction: A Comparative Study and Analysis. Health Technol. 11, 87–97 (2021).
12. P.-N. Tan, M. Steinbach, V. Kumar, Introduction to Data Mining. Addison-Wesley Longman Publishing Co., Inc. (2005).
13. C. Cortes, V. Vapnik, Support-vector networks. Mach. Learn. 20, 273–297 (1995).
14. H. Taud, J.-F. Mas, Multilayer perceptron (MLP). In: Geomatic Approaches for Modeling Land Change Scenarios, pp. 451–455 (2018).
15. T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2016).
16. B. Pavlyshenko, Using stacking approaches for machine learning models. In: 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP), IEEE (2018).