Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

June 2026. Vol. 5. No. 3; e-ISSN: 2808-4519

Heart Disease Prediction Using a Comparison of Naïve Bayes and Random Forest Algorithms

Iman Abdurrachman[1]*, Asrul Abdullah[2], Syarifah Putri Agustini[3]

[1,2,3]*Faculty Teknik, Universitas Muhammadiyah Pontianak*
imanabdurachman0@gmail.com[1*], asrul.abdullah@ummuhpnk.ac.id[2],
agustini.putri@unmuhpnk.ac.id[3]

Abstract

Heart disease is one of the leading causes of death worldwide, making early detection essential. This study compares the performance of the Naive Bayes and Random Forest algorithms in predicting heart disease using clinical data. The dataset includes attributes such as chest pain type (cp), maximum heart rate achieved (thalach), and slope of the ST segment (slope). The research process consists of data preprocessing, feature selection, model training, and evaluation using accuracy, precision, recall, and F1-score metrics.
The results show that Random Forest outperformed Naive Bayes in heart disease prediction. Random Forest achieved an accuracy of 75%, precision of 69%, and recall of 86%, while Naive Bayes achieved an accuracy of 69%, precision of 66%, and recall of 72%. These findings indicate that Random Forest is more effective in handling the complexity of heart disease data and provides better predictive performance. This study demonstrates the potential of machine learning methods, particularly Random Forest, in supporting heart disease diagnosis and may serve as a reference for the development of medical decision support systems.

Keywords: *Heart Disease Prediction, Naïve Bayes, Random Forest*

1. Introduction

In In the healthcare sector, a large amount of data is available and can be utilized for research and educational purposes. However, existing datasets are generally used only as archives of laboratory examination results or patient medical records related to specific symptoms or diseases. These archives can provide significant benefits when processed and analyzed properly, as they can be used to study past events and support decision-making when similar cases occur in the future. By utilizing healthcare data effectively, disease detection can be performed earlier and more efficiently [1].

Heart disease consists of a range of disorders that affect the heart, including cardiovascular disease, coronary artery disease, congenital heart defects, arrhythmias, and heart muscle disorders. Heart disease remains one of the leading causes of death worldwide. Therefore, the medical field increasingly relies on computer-based systems to provide accurate, timely, and efficient diagnoses. The continuous growth of patient-related data creates opportunities to apply data mining techniques for knowledge extraction and prediction. As a result, heart disease prediction has become an important area of research in healthcare analytics [2].

Therefore, it is important to conduct performance evaluations in heart disease prediction systems to support future healthcare needs. This can be achieved by testing and comparing several classification algorithms to determine which method provides the highest predictive accuracy. Based on this objective, this study develops a heart disease prediction system using Python and Streamlit by comparing the performance of the Naïve Bayes and Random Forest algorithms. The dataset used in this research was obtained from Kaggle and consists of 1,025 records. Feature selection is performed based on correlation analysis to identify the most relevant variables for prediction [3].

This study not only compares the performance of two classification algorithms, as conducted in many previous studies, but also integrates Boundary Value Analysis (BVA) to evaluate the robustness of prediction models when handling extreme input values. BVA is commonly applied in software testing but is rarely used in the evaluation of heart disease classification models. Therefore, the incorporation of BVA provides an additional analytical perspective and contributes novelty to this research.The results of this study are expected to help determine the most accurate algorithm for heart disease prediction, support medical decision-making, and contribute to future research aimed at improving early detection of heart disease.

2. Theoritical Basis

This chapter discusses previous studies related to the discussion in the research conducted by the author.

2.1. Heart disease

Heart disease is one of the leading causes of death and requires early detection and prompt treatment due to its potentially sudden onset. Identifying risk factors at an early stage can help reduce the likelihood of heart attacks and support prevention through healthy lifestyle habits and regular exercise [4].

2.2. Data Mining

Data mining is the process of extracting useful information and hidden patterns from large datasets using techniques from statistics, databases, artificial intelligence, and machine learning [5]. Its main purpose is to transform raw data into valuable knowledge that supports decision-making. Data mining operations are generally divided into two categories: prediction, which is used to forecast future outcomes, and discovery, which is used to identify hidden patterns and relationships within data. Common data mining techniques include clustering, regression, classification, and association rules. In this study, the classification technique is used to predict heart disease based on patient data.

2.3. Knowledge Discovery in Databases

Knowledge Discovery in Database (KDD) is the process of extracting useful knowledge and hidden patterns from large datasets. KDD encompasses the entire process of discovering information, while data mining is one of its main stages [6]. The KDD process consists of five stages: Data Selection, which involves choosing relevant data; Pre-processing and Cleaning, which removes errors, duplicates, and inconsistent data; Transformation, which prepares data for analysis; Data Mining, which applies algorithms to identify patterns and relationships; and Interpretation/Evaluation, which analyzes and presents the discovered knowledge in a meaningful form.

2.4. Prediction

Prediction is the process of estimating future outcomes based on historical and current data. Its purpose is to provide the most likely result while minimizing errors between predicted and actual outcomes. Prediction does not guarantee exact results but aims to produce estimates that are as accurate as possible. In decision-making and planning, prediction serves as an important tool for anticipating future conditions [7].

2.5. Naive Bayes

Naive bayes merupakan salah satu metode statistik untuk klasifikasi yang memungkinkan untuk menangkap ketidakpastian tentang suatu model dengan cara berprinsip pada mendefinisikan hasil probabilitas. Metode ini digunakan untuk menyelesaikan masalah diagnosa dan prediksi[8].

2.6. Random Forest

Random Forest is a machine learning algorithm that consists of multiple decision trees built from randomly selected samples and features. Each tree generates a prediction, and the final result is determined by combining the predictions from all trees, typically through majority voting or probability averaging. This approach improves classification accuracy and reduces the risk of overfitting. One important parameter in Random Forest is n_estimators, which determines the number of trees used in the model [9].

2.7. Python

Python is a high-level programming language developed by Guido van Rossum and first released in 1991. It is widely used in various fields, including machine learning and deep learning, due to its simple syntax, ease of use, and readability. Python scripts can be executed directly without a separate compilation process, making development more efficient [10].

2.8. Streamlit

Streamlit is a Python framework used to develop interactive web applications for data science and machine learning projects [9]. It enables developers to build user-friendly web applications with minimal web development effort. Streamlit integrates easily with popular libraries such as NumPy, Pandas, and Matplotlib, making it suitable for creating data visualizations and interactive interfaces [10]. In this study, Streamlit is used to develop a web-based application for heart disease prediction.

2.9. Visual Studio Code

Visual Studio Code (VS Code) is a lightweight yet powerful source-code editor developed by Microsoft. It provides built-in support for JavaScript, TypeScript, and Node.js, while also supporting various programming languages through extensions, including Python, C++, C#, and PHP [12]. VS Code offers an intuitive interface, integrated development tools, and extensive extension support, making it a popular choice for software and web application development.

3. Research Methods

The methodology used in this research can be seen in the flowchart used in Figure 1 below.

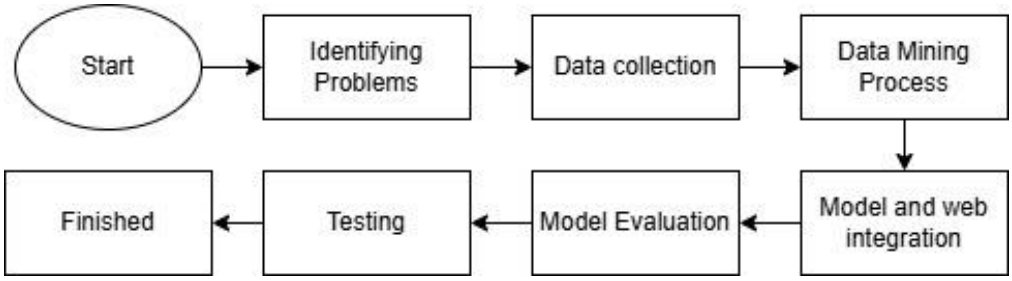

Fig. 1: Flow diagram

3.1. Defining the Problem

The problem addressed in this study is the difficulty of diagnosing heart disease without assistance from medical experts or doctors. Therefore, this research aims to develop an application that can help identify heart disease using machine learning methods. To support the study, relevant references and datasets were collected through a literature review of previous research and related publications.

3.2. Data collection

This study uses a heart disease dataset obtained from Kaggle due to the limited availability of medical data in Indonesia, which is restricted by patient confidentiality regulations. The dataset consists of 1,025 records with 14 attributes related to heart disease diagnosis. The collected data are used to develop and evaluate prediction models using the Naïve Bayes and Random Forest algorithms, with a comparison conducted to determine the most accurate method.

3.3. Data Mining Process

After obtaining the dataset, the data are processed using the Knowledge Discovery in Database (KDD) method. The process begins with Exploratory Data Analysis (EDA) to understand the characteristics of the dataset and analyze its features. Next, data preprocessing is performed to clean and prepare the data before training the prediction models, ensuring optimal performance and accurate results.

3.4. Model and Web Integration

In this phase, the developed prediction model is implemented using Python and saved in an.ipynb file. Streamlit is used as the web framework to build a web-based application that integrates and deploys the trained model for user interaction.

3.5. Model Evaluation

In this phase, the author evaluates the model that has been built to obtain an accuracy value from the data. Accuracy is used to identify how accurate the model is.

3.6. Testing

In this phase, the author tests the saved modeling results to see how good the model that has been built is and to display the labels of the results of the model that has been built and to see how accurate the results of the model are with the confusion matrix.

4. System Analysis and Design

4.1. Data Requirements Analysis

The data used in this study were sourced from Kaggle, comprising 1,025 raw heart disease datasets with 14 attributes. The attributes used in this study can be seen in Table 1 below.

Table 1: Method Raw Data Attributes

No	Atribut	Keterangan	Jenis Data
1	age	Usia pasien	Numerik
2	sex	Jenis kelamin (1 = Laki-laki, 0 = Perempuan)	Numerik
3	cp	Tipe nyeri dada (0 = Typical agnia, 1 = Atypical agnia, 2 = Non-anginal pain, 3 = Asymptomatic)	Numerik

A raw example of Heart Disease Prediction can be seen in Figure 2 below.

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1020	59	1	1	140	221	0	1	164	1	0.0	2	0	2	1
1021	60	1	0	125	258	0	0	141	1	2.8	1	1	3	0
1022	47	1	0	110	275	0	0	118	1	1.0	1	1	2	0
1023	50	0	0	110	254	0	0	159	0	0.0	2	0	2	1
1024	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0

Fig. 2: Raw Data for Heart Disease Prediction

4.2. Designing the Model Building Process

The model-building process begins by splitting the dataset into training and testing sets using an 80:20 ratio. Selected features, namely cp, thalach, and slope, are used as independent variables, while target serves as the dependent variable. The training data are then used to develop both the Naïve Bayes and Random Forest models. Model performance is evaluated using a confusion matrix, including metrics such as accuracy, precision, and recall. Finally, the trained models are saved in pkl format for deployment and future use.

4.3. Model Evaluation

The model evaluation stage aims to obtain accuracy results from training and testing data using the confusion matrix method. Accuracy is used to identify how accurate the model is.

4.4. Design of Model and Web Integration Process

This stage explains the processing of input data that has been modeled using Naïve Bayes and Random Forest in a web-based interface to produce output in the form of heart disease predictions that state whether the patient is at risk or not at risk of developing heart disease

4.5. Interface Design

User interface (UI) design is the communication mechanism between users and the system. Interface design aims to create a user experience that is easy to understand and use, and satisfying.

5. Results and Discussion

This chapter will explain how the data was collected and processed, how the two models were built and trained, and how they performed. The results will be presented in the form of a confusion matrix, including accuracy, precision, and recall.

5.1. Preprocessing Results

Preprocessing is the process of preparing raw data before it is analyzed and entered into a model. The purpose of data preprocessing is to prepare the data, check for missing values, remove duplicate data, and split the data.

5.1.1. Duplicate Data

Duplicate data occurs when identical entries are present in a dataset. This can impact data analysis and modeling because the model will read the data repeatedly. Here are the results after performing a duplicate data check.

```
0       False
1       False
2       False
3       False
4       False
        ...
1020     True
1021     True
1022     True
1023     True
1024     True
Length: 1025, dtype: bool
```

Fig. 3: Missing values check results

After checking for duplicate data and there is duplicated data, the duplicate data deletion stage is carried out.

```
Data setelah menghapus duplikat:
     age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  \
0     52    1   0       125   212    0        1      168      0      1.0
1     53    1   0       140   203    1        0      155      1      3.1
2     70    1   0       145   174    0        1      125      1      2.6
3     61    1   0       148   203    0        1      161      0      0.0
4     62    0   0       138   294    1        1      106      0      1.9
..   ...  ...  ..       ...   ...  ...      ...      ...    ...      ...
723   68    0   2       120   211    0        0      115      0      1.5
733   44    0   2       108   141    0        1      175      0      0.6
739   52    1   0       128   255    0        1      161      1      0.0
843   59    1   3       160   273    0        0      125      0      0.0
878   54    1   0       120   188    0        1      113      0      1.4

     slope  ca  thal  target
0        2   2     3       0
1        0   0     3       0
2        0   0     3       0
3        2   1     3       0
4        1   3     2       0
..     ...  ..   ...     ...
723      1   0     2       1
733      1   0     2       1
739      2   1     3       0
843      2   0     2       0
878      1   1     3       0

[302 rows x 14 columns]
```

Fig. 4: The results after deleting duplicate data

5.2. Naïve Bayes and Random Forest Modeling

After splitting the data with 80% training data and 20% testing data, the next step is to build the Naïve Bayes and Random Forest models, which will then be trained using the training data and make predictions using the testing data.

5.1.2. Naive Bayes Model

The Naïve Bayes model development and prediction process was carried out by importing the GaussianNB library from *scikit-learn* as the classification algorithm and accuracy_score to measure model accuracy, initializing the model using nb_model = GaussianNB(), training the model with the training data through nb_model.fit(X_train, y_train), making predictions on the testing data using nb_model.predict(X_test), and calculating the model accuracy by comparing the predicted results with the actual labels using accuracy_score(y_test, nb_predictions).

5.1.3. Random Forest Model

The Random Forest model development and prediction process was carried out by importing the RandomForestClassifier library from *scikit-learn*, initializing the model using rf_model = RandomForestClassifier(n_estimators=100, random_state=42), where n_estimators=100 indicates that the model uses 100 decision trees and random_state=42 ensures consistent results across runs, training the model with the training data through rf_model.fit(X_train, y_train), making predictions on the testing data using rf_model.predict(X_test), and calculating the model accuracy by comparing the predicted results with the actual labels using accuracy_score(y_test, rf_predictions).

5.3. Evaluasi Model

After carrying out Naïve Bayes and Random Forest modeling, the next step is to carry out a model evaluation which aims to find out how well these two methods work by obtaining accuracy results from the confusion matrix.

```
--- Naïve Bayes ---
Accuracy: 0.69
Precision: 0.66
Recall: 0.72
F1-Score: 0.69
Confusion Matrix:
[[21 11]
 [ 8 21]]

--- Random Forest ---
Accuracy: 0.75
Precision: 0.69
Recall: 0.86
F1-Score: 0.77
Confusion Matrix:
[[21 11]
 [ 4 25]]
```

Fig. 5: Model Evaluation Results

5.4. Website Page Results

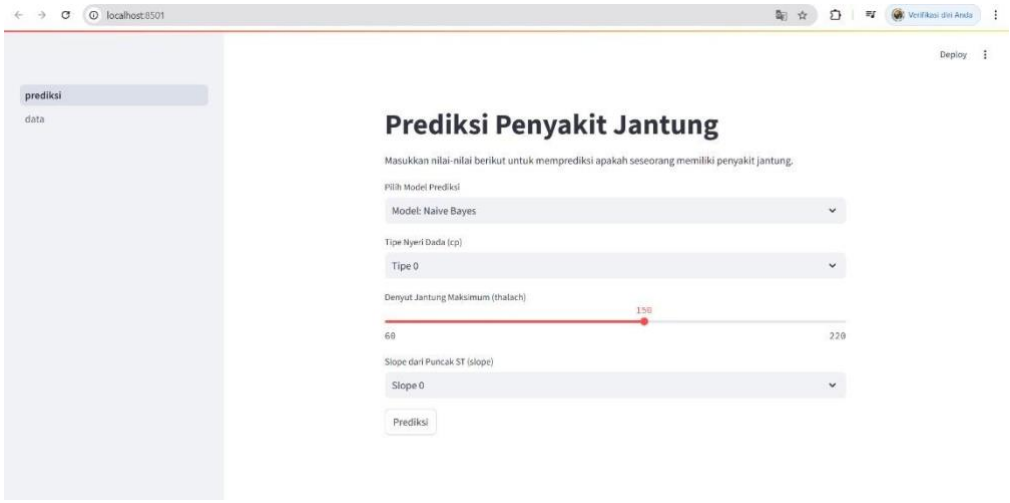

Fig. 6: Heart Disease Prediction Home Page

displays the application interface used to predict heart disease. Users can select a prediction model, enter the required parameters, and click the Predict button to obtain the prediction result.

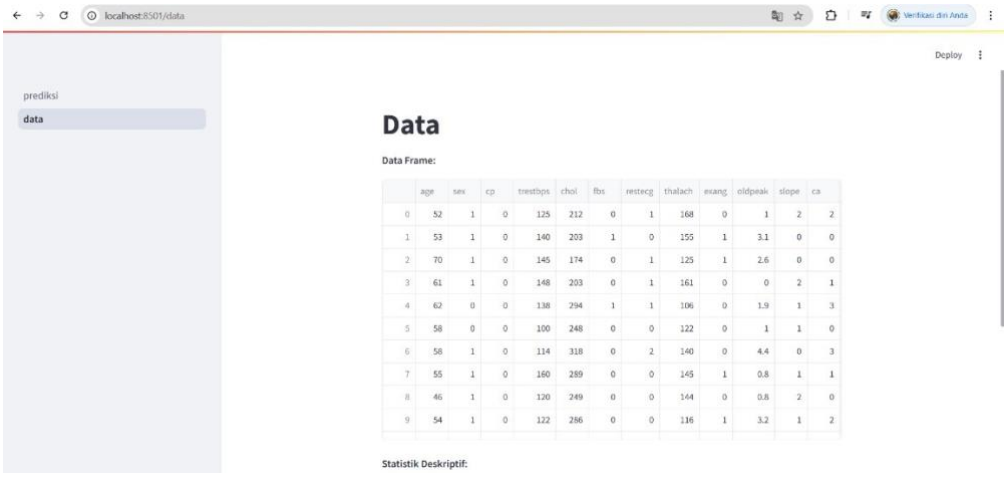

Fig. 7: Heart Disease Prediction Home Page

displays the heart disease dataset used in this study. This page contains patient data and the attributes used as variables in the training and testing processes of the prediction models.

6. Conclusion

Based on the research results, a web-based heart disease prediction application was successfully developed using the Naïve Bayes and Random Forest methods. The system predicts heart disease risk based on the cp, thalach, and slope features. Evaluation results show that Naïve Bayes achieved 69% accuracy, 66% precision, and 72% recall, while Random Forest achieved 75% accuracy, 69% precision, and 86% recall. Therefore, Random Forest performed better than Naïve Bayes, with an accuracy improvement of 6%.

7. Suggestion

Based on the results of this study, future research is expected to use larger and more diverse datasets and evaluate other methods to compare and improve prediction accuracy.

References

[1] H. M. Nawawi et al., "KOMPARASI ALGORITMA NEURAL NETWORK DAN NAÏVE BAYES," vol. 15, no. 2, pp. 189–194, 2019, doi: 10.33480/pilarv15i2.669.

[2] D. Derisma, "Perbandingan Kinerja Algoritma untuk Prediksi Penyakit Jantung dengan Teknik Data Mining," J. Appl. Informatics Comput., vol. 4, no. 1, pp. 84–88, 2020, doi: 10.30871/jaicv4i1.2152.

[3] D. H. Depari, Y. Widiastiwi, and M. M. Santoni, "Perbandingan Model Decision Tree, Naive Bayes dan Random Forest untuk Prediksi Klasifikasi Penyakit Jantung," Inform. J. Ilmu Komput., vol. 18, no. 3, p. 239, 2022, doi: 10.52958/iftkv18i3.4694.

[4] S. Sahar, "Analisis Perbandingan Metode K-Nearest Neighbor dan Naïve Bayes Clasiffier Pada Dataset Penyakit Jantung," Indones. J. Data Sci., vol. 1, no. 3, pp. 79–86, 2020, doi: 10.33096/ijodasv1i3.20.

[5] A. Riani, Y. Susianto, N. Rahman, and U. D. Ali, "Implementasi Data Mining Untuk Memprediksi Penyakit Jantung Mengunakan Metode Naive Bayes Data Mining Implementation to Predict Heart Disease using Naive Bayes Method," vol. 1, no. 01, pp. 25–34, 2019, doi: 10.35970/jinitav1i01.64.

[6] A. Riski, "Analisis Komparasi Algoritma Klasifikasi Data Mining Untuk Prediksi Penderita Penyakit Jantung," J. Tek. Inform. Kaputama, vol. 3, no. 1, pp. 22–28, 2019, [Online]. Available: https://jurnal.kaputama.ac.id/index.php/JTIK/article/view/141/156.

[7] M. R. S. Alfarizi, M. Z. Al-farish, M. Taufiqurrahman, G. Ardiansah, and M. Elgar, "Penggunaan Python Sebagai Bahasa Pemrograman untuk Machine Learning dan Deep Learning," Karya Ilm. Mhs. Bertauhid (KARIMAH TAUHID), vol. 2, no. 1, pp. 1–6, 2023.

[8] Rustam, Rustam, Sidik Rahmatullah, Supriyato Supriyato and Sri Wahyuni Sri Wahyuni. "PENERAPAN DATA MINING UNTUK PREDIKSI PENJUALAN PRODUK TRIPLEK PADA PT PUNCAK MENARA HIJAU MAS." Jurnal Informasi dan Komputer (2020): n. pag.https://doi.org/10.35959/jik.v8i2.186?sid=semanticscholar.

[9] Azhar, Yufis, Aidia Khoiriyah Firdausy, and Putri Juli Amelia. 2022. "Perbandingan Algoritma Klasifikasi Data Mining Untuk Prediksi Penyakit Stroke". SINTECH (Science and Information Technology) Journal 5 (2):191-97. https://doi.org/10.31598/sintechjournal.v5i2.1222.

[10] Cahyanti, F. L. D., Sarasati, F., Astuti, W., & Firasari, E. (2023). Klasifikasi Data Mining dengan Algoritma Machine Learning untuk Prediksi Penyakit Liver. Jurnal Ilmu dan Teknologi, 14(2), 134–139. https://ojs.uniska- bjm.ac.id/index.php/JIT

[11] M. Romzi and B. Kurniawan, "Pembelajaran Pemrograman Python dengan Pendekatan Logika Algoritma," Jurnal Teknik Informatika Mahakarya (JTIM), vol. 3, no. 2, pp. 37–44, 2020.

[12] A. Alhamad, A. I. S. Azis, B. Santoso, and S. Taliki, "Prediksi Penyakit Jantung Menggunakan Metode-Metode Machine Learning Berbasis Ensemble – Weighted Vote," Jurnal Edukasi dan Penelitian Informatika (JEPIN), vol. 5, no. 3, pp. 352–359, 2019