HANIF
Journal of Information Systems
https://journal.ilmubersama.com/index.php/hanif

Machine Learning

Comparison of Logistic Regression and K-Nearest Neighbor (KNN) Algorithms in a Heart Failure Prediction Dataset

Julia Namira Nasution [1*], *Zainal Azis* [2]

[1] *Department of Information Technology, Faculty of Computer Science and Information Technology, Universitas Muhammadiyah Sumatera Utara, Medan, 20238, North Sumatra, Indonesia*
[2] *Department of Mathematics Education, Faculty of Teacher Training and Education, Universitas Muhammadiyah Sumatera Utara, Medan, 20238, North Sumatra, Indonesia*

ARTICLE INFORMATION

Received: Aug 04, 2025
Revised: Oct 16, 2026
Available Online: Jan 31, 2026

KEYWORDS

Heart Failure
K-Nearest Neighbor
Logistic Regression
Machine Learning
Prediction

CORRESPONDENCE (*)

Phone: +62 878-1309-8847
E-mail: julianamira33@gmail.com

A B S T R A C T

Heart failure is one of the leading causes of death worldwide. Early detection of heart failure risk is crucial to minimize its serious consequences. This study aims to compare the performance of two machine learning algorithms, namely Logistic Regression and K-Nearest Neighbor (KNN), in predicting heart failure using a dataset from the Kaggle platform. The research stages include data preprocessing, normalization, splitting into training and testing data, model implementation, and evaluation using a confusion matrix. Evaluation is based on accuracy, precision, recall, and F1-score metrics. The results show that Logistic Regression achieved an accuracy of 88.04% with an execution time of 0.022 seconds, while KNN achieved an accuracy of 85.51% with an execution time of 0.158 seconds. Logistic Regression outperformed in recall and F1-score, making it more effective for early detection of heart failure. Therefore, Logistic Regression is considered more optimal than KNN in the context of this study. However, Logistic Regression is not always superior to K-Nearest Neighbor, as prediction results highly depend on the characteristics of the specific case.

INTRODUCTION

Heart failure is a serious medical condition in which the heart cannot pump blood efficiently enough to meet the body's needs. This condition often develops gradually and can be caused by various factors such as coronary heart disease, high blood pressure, diabetes, and many other causes [1,2,3]. The heart is a vital organ that must function properly because it pumps blood throughout the body, delivering oxygen and nutrients. If the heart is not functioning properly, it can significantly disrupt the function of other organs and even lead to heart failure. In other words, cardiovascular disease, particularly heart disease, is one of the most deadly diseases in both developed and developing countries. Attention to this disease is crucial and essential.

With the advancement of technology, many things can be done to make things easier for people. Early detection of heart failure is crucial to prevent further complications and improve patient prognosis, as well as address diagnostic challenges. However, heart failure symptoms are often nonspecific, making clinical diagnosis challenging. Therefore, machine learning-based approaches are being used to assist in predicting the risk of heart failure by automatically analyzing medical data.

https://doi.org/10.56211/hanif.v3i1.53

Data-driven predictive models can help medical practitioners identify patients at high risk of heart failure earlier, allowing for more timely medical intervention [4,5]. Machine learning is the process by which computers can perform more accurately by collecting and learning from the data they are given.

This study uses a specific public dataset from Kaggle, namely heart failure prediction, which may not have been widely explored in previous research for heart failure prediction using comparative methods. The authors tried to use two different algorithms and compare them. This study focuses on identifying the most optimal algorithm in the context of heart failure prediction by using an evaluation model to measure the effectiveness of each algorithm. The Logistic Regression algorithm is a supervised learning method used for regression and classification problems. This method uses a logistic function to model the relationship between independent attributes and the classification probability of categorical data. Meanwhile, the K-Nearest Neighbor algorithm is a generalization algorithm for the nearest neighbor rule, its inductive offset is the class label of k-samples with the class label to be tested being the closest.

In this study, the authors will attempt to make predictions by comparing the accuracy obtained to provide a deeper understanding of the potential use of machine learning in supporting early detection of heart failure. The research was conducted through the stages of data collection, data preprocessing, implementation, and evaluation. Model evaluation used a Confusion Matrix with accuracy, precision, recall, and f1-score metrics.

METHOD

Research Design

This study employed a quantitative approach. This quantitative approach was used to analyze data from individuals with heart disease using Logistic Regression and K-Nearest Neighbor (KNN) methods. The two methods were then compared to determine which method was more accurate in predicting heart disease based on their accuracy values.

Logistic Regression Algorithm

Logistic Regression is a classification algorithm used to predict the probability of a categorical dependent variable. The dependent variable in Logistic Regression is a binary variable that has a value of 1 (yes) or 0 (no). For binary classification, Logistic Regression is a statistical algorithm whose main goal is to predict the probability that an example will fall into one of two classes. In the context of heart failure prediction, Logistic Regression attempts to predict whether a patient is at risk of heart failure (positive) or not (negative). The sigmoid or logistic function limits the output to a value between 0 and 1. The sigmoid probability function converts linear input into probabilities and estimates parameters. To evaluate the relationship between a number of variables and binary or random variables, binary Logistic Regression is a common data analysis technique. The binary response variable (y) and the predictor variable (x) consist of two categories: success and failure, represented by the value y = 1 (success) and the value y = (failure) (Setyawan & Wakhidah, 2025).

K-Nearest Neighbor (KNN) Algorithm

K-Nearest Neighbors (KNN) is an instance learning method in supervised learning that is included in lazy learning techniques [6,7]. The KNN method is an embodiment-based learning method, where functions are locally approximated values and all calculations are postponed until the classification process. The training data is projected into a multidimensional space, where each dimension extracts features from the data. The proximity or distance of neighbors is usually calculated based on Euclidean distance [8,9].

Confusion Matrix

A confusion matrix is used to evaluate the effectiveness of a classification method in predicting data classes. This technique compares the actual class values with the predicted values [10,11]. An example of a confusion matrix table can be seen below.

Table 1. Confusion Matrix

	ACTUAL	
PREDICTED	*FALSE*	*TRUE*
FALSE	*TN (True Negative)*	*FP (False Positive)*
TRUE	*FN (False Negative)*	*TP (True Positive)*

RESULTS AND DISCUSSION

Importing Libraries

To predict and compare model performance, two algorithms were implemented: Logistic Regression and K-Nearest Neighbor (KNN) using the Python programming language with supporting libraries from Scikit-learn, Pandas, and Matplotlib.

```
1  import numpy as np
2  import pandas as pd
3  import matplotlib.pyplot as plt
4  import seaborn as sns
5  import time
6  from matplotlib import pyplot as plt
7  from sklearn.model_selection import train_test_split
8  from sklearn.preprocessing import MinMaxScaler
9  from sklearn.linear_model import LogisticRegression
10 from sklearn.neighbors import KNeighborsClassifier
11 from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
```

Figure 1. Library

Reading Data

After importing the library, the next step is to read the dataset into Python using the Pandas library. The dataset is saved in csv format and read using the read_csv() function.

	Age	Sex	ChestPainType	RestingBP	Cholesterol	...	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
0	40	M	ATA	140	289	...	172	N	0.0	Up	0
1	49	F	NAP	160	180	...	156	N	1.0	Flat	1
2	37	M	ATA	130	283	...	98	N	0.0	Up	0
3	48	F	ASY	138	214	...	108	Y	1.5	Flat	1
4	54	M	NAP	150	195	...	122	N	0.0	Up	0

Figure 2. Top Five Data

Data Preprocessing

Before model training, the dataset undergoes several stages:

1. One-Hot Encoding
   Categorical features such as Sex, ChestPainType, RestingECG, ExerciseAngina, and ST_Slope are converted into numeric representations using the One-Hot Encoding method so they can be processed by the machine learning algorithm.

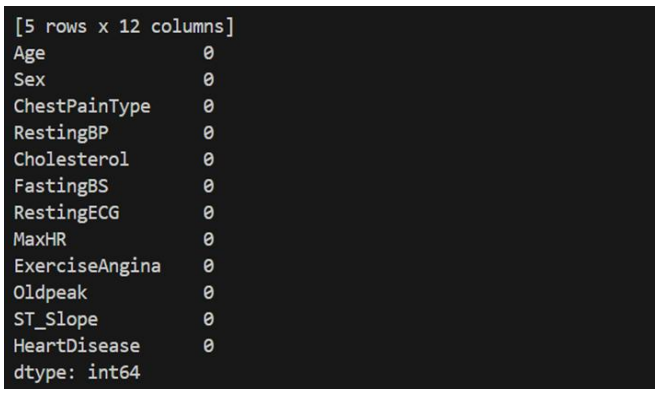

Figure 3. Number of Empty Data

2. Data Splitting
   The data was split into training and test data in a 70:30 ratio using train_test_split, resulting in 70% of the total training data and 30% of the test data.
3. Data Normalization
   Numerical features were normalized using a Min-Max Scaler to scale the values between 0 and 1, ensuring that the performance of algorithms like KNN was not biased towards large-scale features.

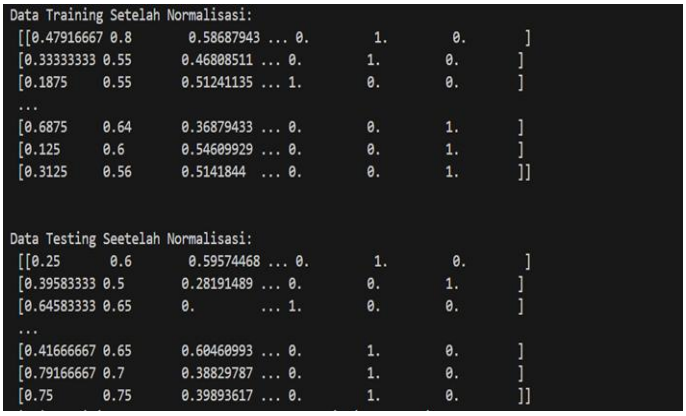

Figure 4. Training and Testing Data After Normalization

Algorithm Implementation

The Logistic Regression algorithm for heart failure prediction was implemented using the Scikit-learn module, which includes the Logistic Regression module. Import the Logistic Regression module and then create a classifier object using the LogisticRegression() function. Next, input the training data into the Logistic Regression function using the fit() function and make predictions on the testing data using the predict() function.

Comparative Evaluation

To evaluate the Logistic Regression and K-Nearest Neighbor algorithms, the author used the Confusion Matrix method. A Confusion Matrix is a matrix or table used to evaluate the performance of a classification model. To view the Confusion Matrix from the data, the confusion_matrix function was used. Then, the data was visualized using the seaborn and matplotlib libraries. To visualize this, the author used the heatmap() function.

The following is the Confusion Matrix of each algorithm:

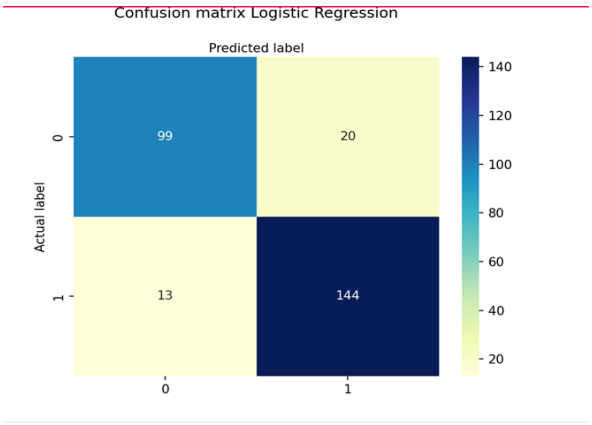

Figure 5. Confusion Matrix Logistic Regression Visualization

F1-Score is the harmonic mean between Precision and Recall, balancing between prediction quality and completeness.

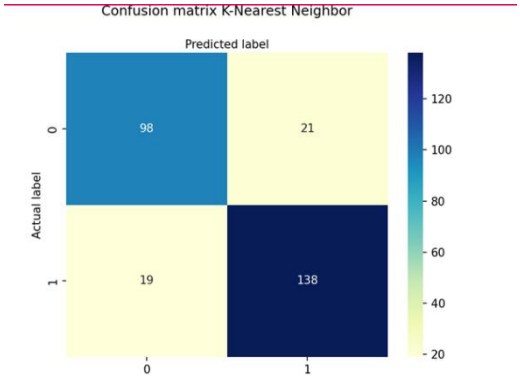

Figure 6. Confusion Matrix K-Nearest Neighbor Visualization

F1-Score is the harmonic mean between Precision and Recall, balancing between prediction quality and completeness.

```
Nilai Presisi, Recall, dan F1 Score Model Logistic Regression:
              precision    recall  f1-score   support

           0       0.88      0.83      0.86       119
           1       0.88      0.92      0.90       157

    accuracy                           0.88       276
   macro avg       0.88      0.87      0.88       276
weighted avg       0.88      0.88      0.88       276

Nilai Presisi, Recall, dan F1 Score Model K-Nearest Neighbor:
              precision    recall  f1-score   support

           0       0.84      0.82      0.83       119
           1       0.87      0.88      0.87       157

    accuracy                           0.86       276
   macro avg       0.85      0.85      0.85       276
weighted avg       0.85      0.86      0.85       276
```

Figure 6. Output Classification Report

The classification report above shows that the accuracy of Logistic Regression is higher than that of K-Nearest Neighbor (KNN), indicating that the Logistic Regression algorithm is superior to the K-Nearest Neighbor (KNN) algorithm. However, this assumption remains uncertain as further steps are needed.

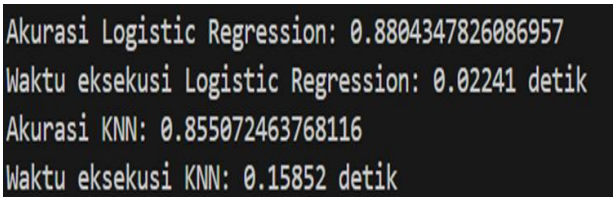

Figure 7. Accuracy and Execution Time

The execution time of each algorithm also shows that the Logistic Regression algorithm is more efficient than the K-Nearest Neighbor algorithm. The time difference between the two algorithms is 0.13611 seconds.

Table 2. Comparison of Algorithm Evaluation

Evaluation Criteria	**Logistic Regression**	**K-Nearest Neighbor (KNN)**
Accuracy	88,04%	85,51%
Precision	88%	87%
Recall	92%	88%
F1-Score	90% High (better)	87% Pretty good
Execution Time	0,022 second	0,158 second

CONCLUSION

Based on the results of the study entitled "Comparison of Logistic Regression and K-Nearest Neighbor (KNN) Algorithms in a Heart Failure Prediction Dataset," the following conclusions can be drawn: The Logistic Regression algorithm demonstrated excellent predictive performance with an accuracy rate of 88.04%. Logistic Regression also excelled in recall and f1-score, indicating greater sensitivity in detecting patients at risk of heart failure. The K-Nearest Neighbor (KNN) algorithm also demonstrated high performance with an accuracy of 85.51% and a slightly higher precision value than Logistic Regression, demonstrating its strength in minimizing false positives. In terms of time efficiency, Logistic Regression proved faster, with an execution time of approximately 0.022 seconds, compared to KNN's 0.158 seconds. The 0.136-second time difference indicates that Logistic Regression is more efficient for use on large data scales. Overall, based on the evaluation results using the confusion matrix, accuracy, precision, recall, and f1-score, the Logistic Regression algorithm is considered superior to KNN in the context of predicting heart failure in this dataset. However, the Logistic Regression algorithm is not always superior to K-Nearest Neighbor, because the prediction results are highly dependent on the type and pattern of the data. Therefore, the choice of algorithm must be adjusted to the characteristics of the case study.

REFERENCES

Buku

[1] Indah Purnama Sari. Algoritma dan Pemrograman. Medan: UMSU Press, 2023, pp. 290.

[2] Indah Purnama Sari. Buku Ajar Pemrograman Internet Dasar. Medan: UMSU Press, 2022, pp. 300.

[3] Indah Purnama Sari. Buku Ajar Rekayasa Perangkat Lunak. Medan: UMSU Press, 2021, pp. 228.

[4] Janner Simarmata Arsan Kumala Jaya, Syarifah Fitrah Ramadhani, Niel Ananto, Abdul Karim, Betrisandi, Muhammad Ilham Alhari, Cucut Susanto, Suardinata, Indah Purnama Sari, Edson Yahuda Putra. Komputer dan Masyarakat. Medan: Yayasan Kita Menulis, 2024, pp.162.

[5] Mahdianta Pandia, Indah Purnama Sari, Alexander Wirapraja Fergie Joanda Kaunang, Syarifah Fitrah Ramadhani Stenly Richard Pungus, Sudirman, Suardinata Jimmy Herawan Moedjahedy, Elly Warni, Debby Erce Sondakh. Pengantar Bahasa Pemrograman Python. Medan : Yayasan Kita Menulis, 2024, pp.180

[6] Zelvi Gustiana Arif Dwinanto, Indah Purnama Sari, Janner Simarmata Mahdianta Pandia, Supriadi Syam, Semmy Wellem Taju Fitrah Eka Susilawati, Asmah Akhriana, Rolly Junius Lontaan Fergie Joanda Kaunang. Perkembangan Teknologi Informatika. Medan: Yayasan Kita Menulis, 2024, pp.158

Jurnal

[7] Sari, I.P., Hariani, P.P., Al-Khowarizmi, A., Ramadhani, F., Sulaiman, O.K., Satria, A, & Manurung, A.A. (2024). CLUSTERING HIV/AIDS DISEASE USING K-MEANS CLUSTERING ALGORITHM. Proceeding International Seminar on Islamic Studies 5 (1), 1668-1676

[8] Sari, I.P., Ramadhani, F., Satria, A., & Sulaiman, O.K. Leukocoria Identification: A 5-Fold Cross Validation CNN and Adaboost Hybrid Approach. 2023 6th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), 486-491

[9] Manurung, A.A., Nasution, M.D., & Sari, I.P. (2023). Implementation of Fuzzy K-Nearest Neighbor Method in Dengue Disease Classification. 2023 11th International Conference on Cyber and IT Service Management (CITSM), 1-4

[10] Sari, I.P., Ramadhani, F., Satria, A., & Apdilah, D. (2023). Implementasi Pengolahan Citra Digital dalam Pengenalan Wajah menggunakan Algoritma PCA dan Viola Jones. Hello World Jurnal Ilmu Komputer 2 (3), 146-157

[11] Sari, I.P., Al-Khowarizmi, A, Sulaiman, O.K., & Apdilah, D. (2023). Implementation of Data Classification Using K-Means Algorithm in Clustering Stunting Cases. Journal of Computer Science, Information Technology and Telecommunication Engineering 4 (2), 402-412

[12] Sulaiman, O.K & Batubara, I.H. (2021). Implementation Data Mining For Level Analysis Traffic Violation By Algorithm Association Rule. Al'adzkiya International of Computer Science and Information Technology (AIoCSIT) Journal 2 (2), 128-135

[13] Sari, I.P., Batubara, I.H., & Al-Khowarizmi, A. (2021). Sensitivity Of Obtaining Errors In The Combination Of Fuzzy And Neural Networks For Conducting Student Assessment On E-Learning. International Journal of Economic, Technology and Social Sciences (Injects) 2 (1), 331-338

[14] Sari, I.P., Al-Khowarizmi, A., & Batubara, I.H. (2021). Cluster Analysis Using K-Means Algorithm and Fuzzy C-Means Clustering For Grouping Students' Abilities In Online Learning Process. Journal of Computer Science, Information Technology and Telecommunication Engineering 2 (1), 139-144

[15] Apdilah, D., & Sari, I.P. (2021). Optimization Of The Fuzzy C-Means Cluster Center For Credit Data Grouping Using Genetic Algorithms. Al'adzkiya International of Computer Science and Information Technology (AIoCSIT) Journal 2 (2), 156-163