Classification of Heart Disease Diagnoses Using Gaussian Naïve Bayes

Ibnu Akil[1*], **Indra Chaidir**[2]

[1,2] Department of Information System, Faculty of Information Technology, Universitas Bina Sarana Informatika, Jakarta, 10450, Indonesia

Abstract

Machine learning, which is part of artificial intelligence, has been widely applied in various fields, especially the medical field. Machine learning helps doctors make more accurate diagnoses. Heart disease is one of the highest causes of death in the world, so the need for accurate diagnosis is absolute for this disease. There are many algorithms that have been applied in machine learning to classify and detect heart disease, such as Linear Discriminant Analysis [1], KNN, Decision Tree, Random Forest [2], and Logistic Regression [3]. One classification algorithm that has not been implemented is Gaussian Naive Bayes. So, in this research the Gaussian Naive Bayes algorithm will be tested on the cardio health risk assessment dataset. From the research results of applying the Gaussian Naive Bayes algorithm to cardio health risk assessment data, accuracy was 0.87%, precision was 0.88%, recall was 0.90%, and f1-score was 0.89%.

Keywords: *artificial intelligence; machine learning; gaussian naïve bayes; heart disease diagnoses; classification algorithm*

1. Introduction

The application of artificial intelligence (AI) in the medical world has brought about a major revolution in diagnosis, treatment, data management and research. The heart of healthcare is being profoundly and fundamentally transformed every day by AI-driven medical technologies that are altering how we identify, treat, and comprehend diseases [4]. Here are some examples of applications of AI in the medical world: Medical Diagnosis, Personalized Care, Data Management, Patient Monitoring, Medical Robotics, Virtual Assistants, and Disease Prediction. In addition, recent years have seen an unprecedented growth in the quantity and complexity of quantitative data, which can be used for machine learning datasets [5]. The application of AI in medicine continues to advance rapidly, bringing hope to improve the quality of healthcare, reduce costs, and save lives. However, it is also important to remember that the use of AI in a medical context also requires serious ethical and security considerations. In carrying out medical diagnoses, AI can help increase diagnostic accuracy, speed up the diagnosis process, and reduce doctors' workload.

One of the diseases that can be diagnosed with AI is heart disease. The cardiovascular disease that causes the highest death rate in the world is coronary heart disease. According to WHO, almost 85% of deaths in the world are caused by strokes and heart attacks [6]. Most of the sudden deaths that occur are caused by heart disease. Sudden death is often equated with sudden natural unexpected death, namely death that is not caused by disease, not due to trauma or accident [7]. If sudden death caused by the heart can be detected early, prevention can be done.

Many researches have been carried out to detect heart disease based on machine learning, one of which is research conducted by Dewi [2]. Dewi uses the KNN, Decision Tree and Random Forest algorithms. From the results of implementation with KNN, the accuracy was 64%, Decision Tree 90%, and Random Forest 87%.

Another research to test machine learning algorithms for heart disease detection was also carried out by Abdar et al who tested several algorithms, namely: C5.0, Neural Network, SVM, KNN, Logistic Regression, and Decision Tree, where the results were won by Decision Tree with a level of accuracy of

93% [8]. In his paper, Abdar does not consistently mention that the algorithm being tested, namely Neural Network and Logistic Regression, is often confused, so it cannot be concluded whether Logistic

[*] **Corresponding author**. *E-mail address:* ibnu.ial@bsi.ac.id

Received: 23 June 2024, Accepted: 26 July 2024 and available online 31 July 2024
DOI: https://doi.org/10.33751/komputasi.v21i2.5260

Regression was tested or not.

Abdar's research was continued by Jefri Junifer who tried to test the accuracy of the Logistic Regression algorithm on heart disease datasets. The results of this research show that the Logistic Regression algorithm still has lower accuracy than the KNN algorithm [3]. Unfortunately, this research tested an algorithm that had been tested previously on the same data, namely Logistic Regression.

In another study, Isnanto et al tried to apply the Linear Discriminant Analysis (LDA) algorithm to the same case, namely the classification of heart disease. From the results of this research, it was found that two output targets were more accurate than five output targets. Accuracy for the two output classes is 81% [1].

In this research, the Gaussian Naive Bayes algorithm will be tested on the Cardio Health Risk Assessment dataset. Gaussian NB is included in the Supervised Learning algorithm for the Classification type. It is suspected that this algorithm has a high level of accuracy for predicting heart disease.

2. Methods

2.1. Research Design

This research is experimental research with a research framework as in Figure 1. Experimental research has the characteristic that the researcher can control the independent variables. Researchers determine and design and organize the treatment of experimental groups and control groups [9]. Meanwhile, the stages used are adopted from the data mining process stages with a slight simplification.

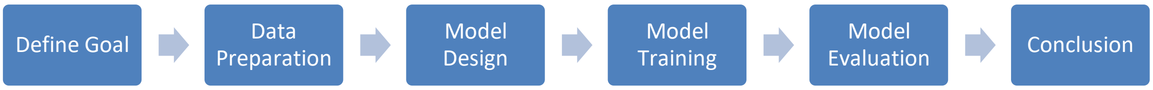
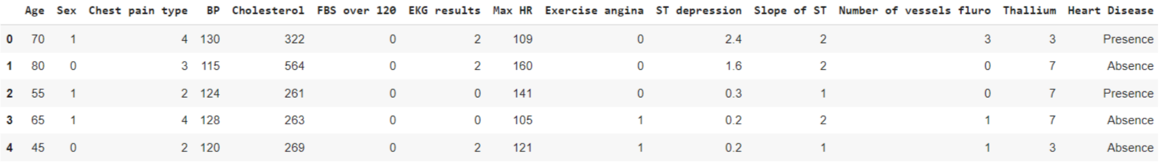


Figure 1. Research Process Framework

Define Goal: is to clearly determine what problem will be solved, in this case it is to predict whether someone has or is indicated for heart disease or not from the collected medical data and information.

Data Preparation: is a stage starting from data collection, data cleaning, determining data features and data normalization. In this research, only primary data sources were used which came from www.kaggle.com, from the research results of Prakhar Kapoor. With link: https://www.kaggle.com/datasets/kapoorprakhar/cardio-health-risk-assessment-dataset. The Cardio Health Risk Assessment dataset includes detailed medical and demographic information of patients, such as age, cholesterol levels, blood pressure, and lifestyle factors. This dataset is designed to develop and test machine learning models to predict heart disease risk. This provides a valuable resource for researchers and healthcare professionals aiming to improve diagnostic accuracy and patient outcomes in cardiovascular health. For more details, see Figure 2 as follows:



	Age	Sex	Chest pain type	BP	Cholesterol	FBS over 120	EKG results	Max HR	Exercise angina	ST depression	Slope of ST	Number of vessels fluro	Thallium	Heart Disease
0	70	1	4	130	322	0	2	109	0	2.4	2	3	3	Presence
1	80	0	3	115	564	0	2	160	0	1.6	2	0	7	Absence
2	55	1	2	124	261	0	0	141	0	0.3	1	0	7	Presence
3	65	1	4	128	263	0	0	105	1	0.2	2	1	7	Absence
4	45	0	2	120	269	0	2	121	1	0.2	1	1	3	Absence

Figure 2. Research's Dataset

The following is an explanation of the features in the data shown in Figure 2: Age; is the patient's age, Sex; is the patient's gender, Chest pain type; is a type of pain in the chest, BP; are blood pressure or blood pressure, Cholesterol; is the cholesterol level, FBS; is fasting blood sugar or blood sugar levels, ECG results; is the result of the electrocardiographic test, Max HR; is the maximum value of heartbeat rate or average heart rate, Exercise Angina; is pain in the chest that occurs due to exercise, ST depression; is the level of depression, Slope of ST; is the rate of increase in heart rate, Number of Vessels; is the number of blood vessels stained with fluoroscopy, and Thallium; is the result of a stress test.

Model Design: is the stage of designing a model according to an algorithm. The core of machine learning is at the model design stage. In model design, the researcher determines the algorithm to be used, in this research it is the Gaussian Naive Bayes algorithm.

Model Training: is a training or machine learning stage from previously established models and algorithms.

Model Evaluation: is the stage where the results of the training model have been obtained and the output analyzed.

Conclusion: is the stage of concluding from the results of the training model and evaluation model, at this stage the desired results are obtained according to the objectives or goals of the research.

2.2. Gaussian Naive Bayes Algorithm

Gaussian Naive Bayes is a classification algorithm based on Bayes' theorem, which assumes that variables are independent of each other and have a continuous normal or Gaussian distribution [10]. The key concept of Gaussian Naive Bayes can be formulated as follows [11]:

Bayes' Theorem Equation:

$$P(C|X) = \frac{P(X|C).P(C)}{P(X)} \tag{1}$$

Where:
P(C|X) is the posterior probability of class C based on feature X
P(X|C) is the probability of feature X given class C
P(C) is the prior probability of class C
P(X) is the marginal probability of class X

Naïve assumption: *The "naive" part of the classifier refers to the assumption that all features are independent of each other based on their class. This simplifies the calculation of the probability: P(X|C)*

Gaussian Distribution Equation:

$$P(x_i|C) = \frac{1}{\sqrt{2\pi\sigma^2 C}} exp\left(-\frac{(x_i-\mu C)^2}{2\sigma^2 C}\right) \tag{2}$$

Where:
x_i is the value of the feature
μC is the mean of the features in class C
σC is the standard deviation of the features in class C
The process of the algorithm is as follow:

1. Training phase:
   a. Estimating the mean and standard deviation of each feature for each class of training data
   b. Calculates the initial probability of each class based on their frequency in the training data.
2. Prediction phase:
   a. For each sample to be classified, calculate the probability of each feature using the Gaussian probability density function.
   b. Multiply all these possibilities together for each class.
   c. Multiply by the initial probability of each class.
   d. The class with the highest final probability value is selected as the prediction.

The advantages of this algorithm are:

1. Simple: easy to implement and understand.
2. Efficient: requires little training data for parameter estimation.
3. Scalability: can handle larger datasets.

The disadvantages are:

1. Independent assumptions: assumptions that have strong independence, may not actually hold in real-world scenarios, have the potential to decrease the accuracy of the classifier.
2. Gaussian assumption: assumes that features are normally distributed, which may not always be the case.

2.3. Confusion Matrix

A confusion matrix, which shows a classification model's accuracy, is a tool for assessing machine learning performance. The number of true positives, true negatives, false positives, and false negatives is shown in the confusion matrix [12]. A confusion matrix has matrix dimensions of 4 x 4 or more, based on how many classes need to be predicted, look at figure 3 which is an example of a 4 x 4 confusion matrix.

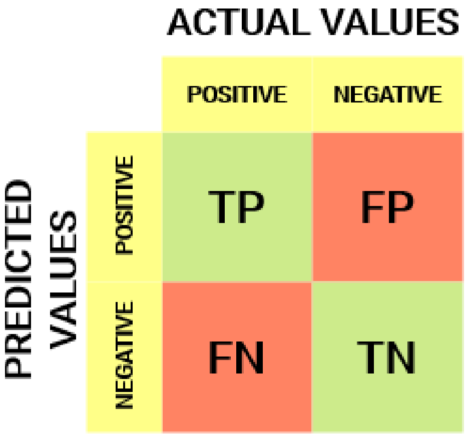


Figure 3. Confusion Matrix

The important terms of the confusion matrix are as follows:

True Positive (TP): *The expected and actual values coincide, or the actual class and the expected class match. The model predicts a positive value, and the true value is positive.*

True Negative (TN): *Either the predicted class and the true class match, or the projected value and the true value match. The model predicts a negative value, while the true value is negative.*

False Positive (FP): *The value that was expected is off. The model predicts a positive number, while the actual value is negative.*

False Negative (FN): *It is not the projected value. The model predicts a negative value, while the actual value is positive.*

Accuracy: *describes the accuracy of the model in predicting.*

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \tag{3}$$

Precision: *describes the accuracy between the actual value and the predicted value.*

$$Precission = \frac{TP}{TP+FP} \tag{4}$$

Recall: *is the ratio of all those predicted correctly*

$$Recall = \frac{TP}{TP+FN} \tag{5}$$

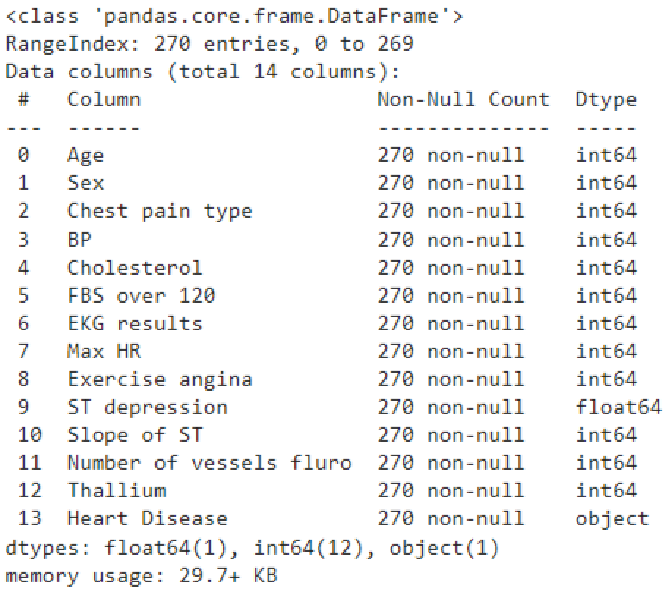
F1-score or F measure: *describes the comparison of weighted average precision and recall* [13].

$$F - measure = \frac{2*Recall*Precission}{Recall+Precission} \tag{6}$$

3. Result and Discussion

3.1. Dataset Preparation

The implementation of this research is using the Python language with an IDE using https://colab.research.google.com. This stage begins with importing the libraries needed in this program, including: *numpy, pandas, matplotlib, tensorflow* and *sklearn*. Then load the data from Google Drive, then explore the data using the *info*() function which can be seen in Figure 4, analyze the data distribution which can be seen in Figure 5, clean the data from null values, and separate numeric values and categorical values. Categorical values must be converted to numeric. After that, the dataset which has been turned into numeric values is all normalized with *StandardScaler* from the *Sklearn* library. The dataset is then divided into a training set and a testing set.



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 270 entries, 0 to 269
Data columns (total 14 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   Age                      270 non-null    int64  
 1   Sex                      270 non-null    int64  
 2   Chest pain type          270 non-null    int64  
 3   BP                       270 non-null    int64  
 4   Cholesterol              270 non-null    int64  
 5   FBS over 120             270 non-null    int64  
 6   EKG results              270 non-null    int64  
 7   Max HR                   270 non-null    int64  
 8   Exercise angina          270 non-null    int64  
 9   ST depression            270 non-null    float64
 10  Slope of ST              270 non-null    int64  
 11  Number of vessels fluro  270 non-null    int64  
 12  Thallium                 270 non-null    int64  
 13  Heart Disease            270 non-null    object 
dtypes: float64(1), int64(12), object(1)
memory usage: 29.7+ KB
```

Figure 4. Data Information

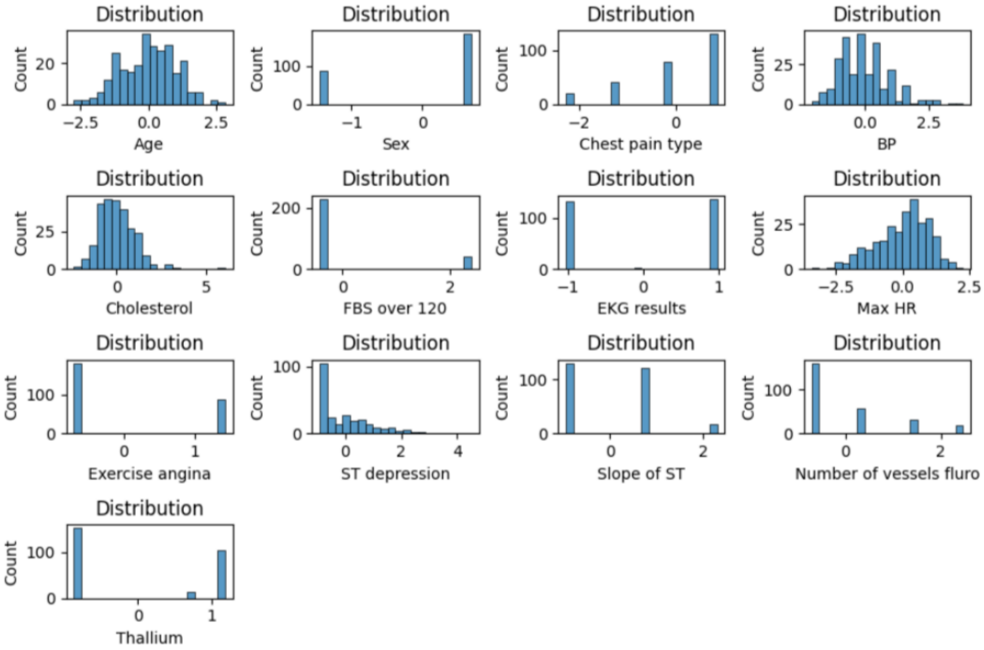


Figure 5. Data distribution from each feature

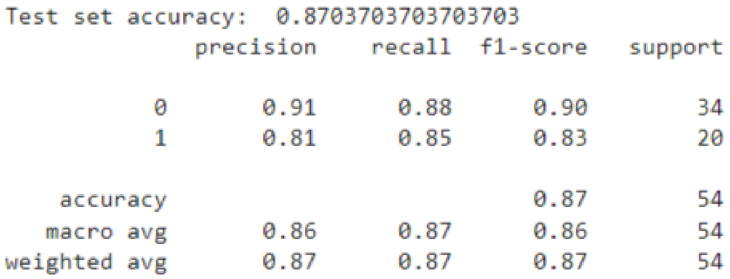
3.2. Model Design and Training

At this stage the algorithm that will be tested is Gaussian Naive Bayes, configured as the following source code.

```
# GaussianNB
from sklearn.naive_bayes import GaussianNB

gaussian_model = GaussianNB()
gaussian_model.fit(X_train, y_train)
y_prediction = gaussian_model.predict(X_test)
print("Test set accuracy: ", accuracy_score(y_test, y_prediction))
print(classification_report(y_test, y_prediction))
```

first we need to import sklearn.naive_bayes library and GaussianNB as the algorithm. And then we fitting the model. The fitting model function will do the training. After that we predict the X_test variable. the output is as we can see in figure 5 as follows:



```
Test set accuracy:  0.8703703703703703
                       precision    recall  f1-score   support

           0       0.91      0.88      0.90        34
           1       0.81      0.85      0.83        20

    accuracy                           0.87        54
   macro avg       0.86      0.87      0.86        54
weighted avg       0.87      0.87      0.87        54
```

figure

Figure 5. Output from training and prediction

The result of the training shows obtained accuracy value of 0.87; precision of class 0=0.91, class 1=0.81; recall of class 0=0.88, class 1=0.85; f1-score is class 0=0.90, class 1=0.83.

3.3. Model Evaluation

To evaluate the model, we use a confusion matrix, which is also facilitated by the Sklearn library. The following is the configuration for the confusion matrix:

```
from sklearn.metrics import confusion_matrix as cm

score = round(accuracy_score(y_test, y_prediction), 3)
cm1 = cm(y_test, y_prediction)
sns.heatmap(cm1, annot=True, fmt=".0f")
plot.xlabel('Predicted Number')
plot.ylabel('Actual Number')
plot.title('Confusion Matrix Accuracy Score: {0}'.format(score), size = 10)
plot.show()
```

Meanwhile, the visual output can be seen in Figure 6. There we can see the values of TP=30, FP=4, TN=17, and FN=3. The calculation is as follows:

Accuracy = (30 + 17) / 30 + 4 + 17 + 3 = 0.87

Precision = 30 / (30 + 4) = 0.88

Recall = 30 / (30 + 3) = 0.90

F1 – score = (2 X 0.90 X 0.88) / (0.90 + 0.88) = 1.584/1.78 = 0.89

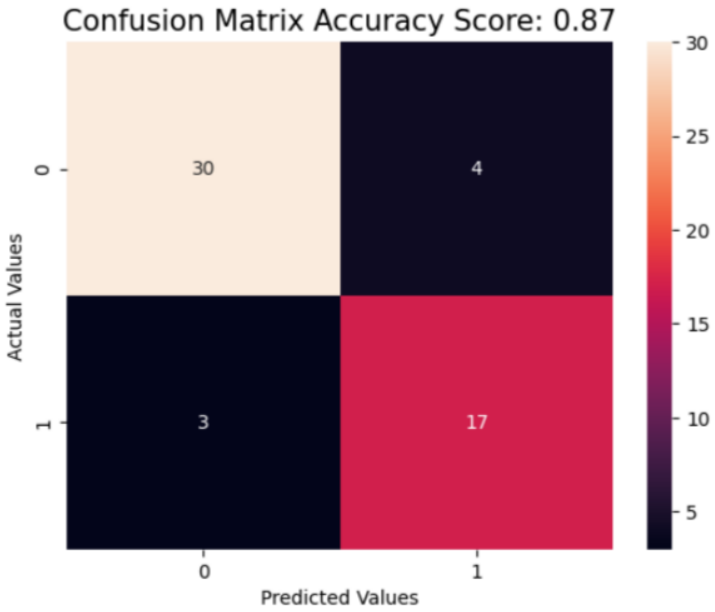


Figure 6. Confusion Matrix

4. Conclusion

From the results of implementing the Gaussian Naive Bayes model with the cardio health risk assessment dataset, it can be concluded that Gaussian Naive Bayes is still higher than the Linear Discriminate Analysis carried out by Isnanto, namely 81.22% [1]. With Gaussian Naive Bayes, accuracy results were obtained at 0.87%, precision at 0.88%, recall at 0.90%, f1-score at 0.89%. However, this algorithm is still below when compared to the Decession Tree of 90% in Dewi's research [2].

5. Acknowledgement

For further research it will be necessary to compare all existing algorithm and chose the best one to be implemented for such dataset, so we can implement that algorithm in real application to predict heart disease for medical usage.

References

[1] R. R. Isnanto, I. Rashad, and C. Edi Widodo, "Classification of Heart Disease Using Linear Discriminant Analysis Algorithm," *E3S Web Conf.*, vol. 448, pp. 1–11, 2023, doi: 10.1051/e3sconf/202344802053.

[2] L. A. Dewi, "Klasifikasi Machine Learning Untuk Mendeteksi Penyakit Jantung Dengan Algoritma KNN, Decision Tree dan Random Forest," UIN Syarif Hidayatullah, 2023.

[3] J. J. Pangaribuan, H. Tanjaya, and K. Kenichi, "Mendeteksi Penyakit Jantung Menggunakan Machine Learning Dengan Algoritma Logistic Regression," *J. Inf. Syst. Dev.*, vol. 06, no. 02, pp. 1–10, 2021.

[4] A. Ahmadi and N. R. Ganji, "AI-Driven Medical Innovations : Transforming Healthcare through Data Intelligence," *Int. J. BioLife Sci.*, vol. 2, no. 2, pp. 132–142, 2023.

[5] M. Gallego, A. Berman, and M. Crispin, "Artificial intelligence in healthcare: A technological perspective," *Innov. Sustain. Futur. Healthc.*, 2020.

[6] T. A. Mylano, "Mengenal Penyakit Jantung Koroner, Penyebab Kematian Tertinggi di Dunia," *www.siloamhospitals.com*, 2024. https://www.siloamhospitals.com/informasi-siloam/artikel/mengenal-penyakit-jantung-koroner-penyebab-kematian-tertinggi-di-dunia (accessed Jun. 09, 2024).

[7] T. Suryadi, "Kematian Mendadak Kardiovaskuler," *J. Kedokt. Syiah Kuala*, vol. 17, no. 2, pp. 112–118, 2017, doi: 10.24815/jks.v17i2.8990.

[8] M. Abdar, S. R. N. Kalhori, T. Sutikno, I. M. I. Subroto, and G. Arji, "Comparing performance of data mining algorithms in prediction heart diseses," *Int. J. Electr. Comput. Eng.*, vol. 5, no. 6, pp. 1569–1576, 2015, doi: 10.11591/ijece.v5i6.pp1569-1576.

[9] N. M. Ratminingsih, "Penelitian Eksperimental Dalam Pembelajaran Bahasa Kedua," *Prasi*, vol. 6, no. 11, pp. 31–40, 2010.

[10] I. As'ad, "Advancing Healthcare Diagnostics: A Study on Gaussian Naive Bayes Classification of Blood Samples," *Int. J. Artif. Intell. Med. Issues*, vol. 1, no. 2, pp. 115–123, 2023, doi: 10.56705/ijaimi.v1i2.120.

[11] I. Ovyawan Herlistiono and S. Violina, "Naïve Bayes Binary Classification for Film Review," no. 204, 2020.

[12] A. Bhandari, "Confusion Matrix in Machine Learning," *www.analyticsvidhya.com*, 2024. https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/ (accessed Jun. 15, 2024).

[13] M. S. Anggreany, "Confusion Matrix," *www.socs.binus.ac.id*, 2020. https://socs.binus.ac.id/2020/11/01/confusion-matrix/ (accessed Jun. 15, 2024).