

COMPARISON OF DECISION TREE AND NAIVE BAYES METHODS FOR RAINFALL CLASSIFICATION USING A WEATHER DATASET WITH A WEB-BASED APPLICATION

Yuda Samudra¹, Amin Hidayat², Nanang³

^{1,2,3} Department of Informatics Engineering, Faculty of Computer Science, Pamulang University, South Tangerang 15310, Indonesia

¹dosen02623@unpam.ac.id

²dosen02615@unpam.ac.id

³dosen02599@unpam.ac.id

Abstract

Rainfall prediction is an important component of weather analysis as it provides valuable information to support decision-making in sectors such as agriculture, transportation, and environmental management. Although various studies have compared machine learning algorithms for rainfall classification, many of them lack detailed discussion on dataset characteristics and practical system implementation. Therefore, this study aims to evaluate and compare the performance of Decision Tree and Naive Bayes algorithms for rainfall classification while considering dataset characteristics and implementing the model in a web-based application. The dataset used in this study consists of 2,500 records with meteorological parameters including temperature, humidity, wind speed, cloud cover, and atmospheric pressure. The data underwent preprocessing, including data cleaning and label encoding, where rainfall was represented as 1 and no rainfall as 0. The dataset was divided into training and testing sets, and both algorithms were applied to build classification models. Model performance was evaluated using confusion matrix, accuracy, and ROC curve analysis. The results show that the Decision Tree algorithm achieved an accuracy of 1.00 (100%), while the Naive Bayes algorithm achieved 0.972 (97.2%). Although Decision Tree shows superior performance, the perfect accuracy may indicate potential overfitting, and therefore the results should be interpreted carefully. Furthermore, the developed models were successfully implemented into a web-based application that enables users to perform rainfall prediction interactively. This study demonstrates that Decision Tree provides better performance for rainfall classification in the given dataset, while also highlighting the importance of considering dataset characteristics and evaluation methods. The integration of machine learning models into a web-based system provides a practical contribution for real-world weather prediction applications.

Keywords: Rainfall Classification, Decision Tree, Naive Bayes, Machine Learning, Weather Dataset, Web-Based Application

I. INTRODUCTION

Rainfall is one of the key elements of weather that significantly affects various aspects of human life, including agriculture, transportation systems, water resource management, and disaster prevention. Reliable information regarding rainfall conditions is essential for individuals, organizations, and government agencies to support effective planning and decision-making. However, predicting or identifying rainfall conditions using weather parameters is not always straightforward because weather data usually consist of multiple variables that interact in complex ways. For this reason, appropriate analytical methods are needed to process

weather data in order to generate meaningful information related to rainfall conditions [1], [2].

With the rapid development of information technology, data mining techniques have become widely used for analyzing large volumes of data and discovering hidden patterns. In the context of weather analysis, data mining approaches can be utilized to classify rainfall conditions based on several meteorological parameters such as temperature, humidity, wind speed, cloud cover, and atmospheric pressure. These parameters are commonly used indicators for determining whether rainfall is likely to occur. By applying classification algorithms to weather datasets, it becomes possible to develop systems that can recognize rainfall patterns more effectively and efficiently [3].

Various machine learning and data mining methods have been applied to solve classification problems. Among the commonly used techniques are Decision Tree and Naive Bayes, which are popular due to their simplicity and efficiency in handling classification tasks. Decision Tree is a classification approach that represents decision rules in the form of a tree structure, making the resulting model easier to interpret. Previous research has shown that Decision Tree algorithms perform well when applied to structured datasets in classification problems [4]. Meanwhile, Naive Bayes is a probabilistic classification technique derived from Bayes' theorem that assumes independence between variables. This method is recognized for its computational efficiency and its ability to deliver reliable results in many classification applications, including weather-related data analysis.

Previous studies have also conducted comparisons between different classification algorithms to determine which method performs better for particular datasets. The findings of these studies suggest that the effectiveness of a classification algorithm can vary depending on the characteristics of the dataset and the features used in the analysis [5].

However, most of these studies primarily focus on model performance comparison without providing sufficient explanation regarding dataset characteristics, handling of data imbalance, or integration of the developed models into practical systems such as web-based applications. In addition, limited studies discuss how the characteristics of weather datasets influence the performance differences between classification algorithms.

Based on these limitations, this study aims to analyze and compare the performance of the Decision Tree and Naive Bayes algorithms for rainfall classification using a weather dataset while considering dataset characteristics and model evaluation aspects. Furthermore, this research integrates the classification models into a web-based application to provide a practical tool for rainfall prediction. The main contribution of this study lies in the combination of comparative analysis and system implementation, as well as providing a clearer evaluation of model performance in the context of rainfall classification. Through this research, it is expected that the study will determine which algorithm offers better performance for rainfall classification and provide a practical system that can assist users in analyzing weather data more efficiently [6], [7].

II. RESEARCH METHOD

This study employs an experimental method with a data mining approach to perform rainfall classification based on a weather dataset. The main objective of this research is to compare the performance of two classification algorithms, namely Decision Tree and Naive Bayes [8], in determining rainfall conditions using several meteorological parameters.

The dataset used in this study consists of 2,500 weather records, including attributes such as temperature, humidity, wind speed, cloud cover, and atmospheric pressure. The dataset was obtained from Kaggle, a publicly available data science platform. Specifically, the dataset was accessed from the "SVM Regression" dataset provided by Imran Mik23 on Kaggle

(<https://www.kaggle.com/datasets/imranmik23/svm-regression>). The data were utilized as secondary data for machine learning analysis.

This dataset was selected because it provides relevant meteorological parameters commonly used in rainfall classification studies. After data collection, the dataset underwent preprocessing stages, including data cleaning, label encoding, and normalization to ensure data quality and suitability for classification modeling [9].

The stages of the research conducted in this study are illustrated in Figure 1, which presents the overall workflow of the research process.

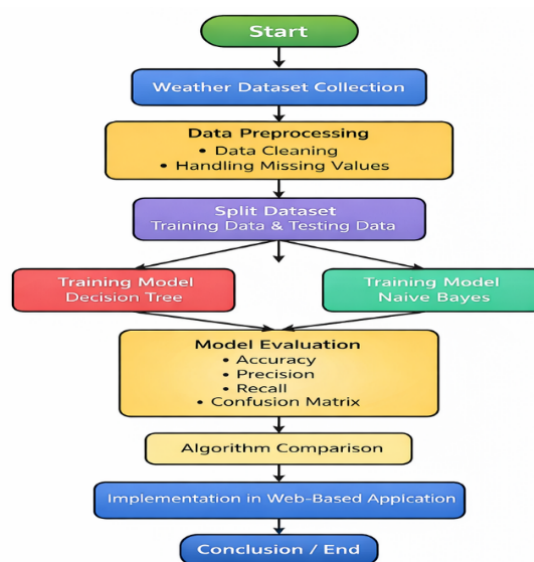


Figure 1. Research Method Flowchart

The research process begins with the collection of the weather dataset, followed by the data preprocessing stage. In this stage, data cleaning is performed to ensure data quality, including handling missing values, removing inconsistent or incomplete records, and transforming data into an appropriate format for the classification process. Furthermore, label encoding is applied to convert categorical target variables into numerical form. If necessary, data normalization is also conducted to ensure that all features are on a comparable scale. Additionally, the class distribution of the dataset is analyzed to identify potential data imbalance issues that may affect model performance.

After preprocessing, the dataset is divided into training and testing sets using a specified ratio (e.g., 80% training data and 20% testing data). To improve the reliability of the evaluation and reduce the risk of overfitting, cross-validation techniques are also applied during the model training process. The training data is used to build the classification models, while the testing data is used to evaluate the performance of the trained models [10], [11].

Subsequently, the classification process is carried out using the Decision Tree and Naive Bayes algorithms. Both algorithms are applied to classify rainfall conditions based on the available weather attributes. After the models are developed, performance evaluation is conducted using several metrics, including accuracy, precision, recall, F1-score, and

confusion matrix. In addition, ROC curve analysis is used to measure the model's ability to distinguish between classes. These evaluation metrics provide a more comprehensive assessment of model performance [12], [13].

The results obtained from the classification and evaluation processes are then analyzed to determine which algorithm demonstrates better performance in rainfall classification. The analysis also considers the relationship between dataset characteristics and model performance. Finally, the developed classification models are implemented into a web-based application to facilitate users in performing rainfall prediction and observing the comparison results between Decision Tree and Naive Bayes methods in a practical and interactive manner [14], [15].

III. RESULTS AND DISCUSSION

In this chapter, the results obtained from processing the weather dataset using the Decision Tree and Naive Bayes classification methods are presented and discussed. The research results include several stages such as data preprocessing, dataset distribution analysis, model performance evaluation using confusion matrix and accuracy metrics, and comparison of the performance of the two algorithms used in this study. In addition, the trained model is implemented in a web-based application to facilitate users in predicting rainfall based on the weather parameters provided. Through the testing and analysis conducted, this study aims to determine which method provides the best performance for rainfall classification using the weather dataset applied in this research.

A. Research Dataset

The dataset used in this study is a weather dataset consisting of 2,500 records with several meteorological parameters used to determine rainfall conditions. The dataset includes attributes such as temperature, humidity, wind speed, cloud cover, and atmospheric pressure, while the rain attribute serves as the target variable indicating whether rainfall occurs (rain) or does not occur (no rain).

In addition to the input features, the dataset characteristics, particularly the class distribution between rain and no rain instances, are analyzed to understand their potential impact on model performance. Understanding these characteristics is important, as imbalanced data distribution may influence the learning process of classification algorithms.

These attributes are used as input variables in the classification process using the Decision Tree and Naive Bayes algorithms. Before being applied to the classification models, the dataset undergoes a preprocessing stage to ensure data quality, consistency, and suitability for machine learning analysis.

B. Data Preprocessing

In the preprocessing stage, several data preparation techniques were applied to ensure that the dataset could be effectively used in the classification process. Data cleaning was conducted to identify and handle potential issues such as missing values, inconsistent entries, and noise in the dataset. Based on the dataset inspection, the data were found to be

complete and did not contain significant missing values, allowing the analysis to proceed without major data imputation. Furthermore, label encoding was applied to transform the categorical target variable into numerical form, where rainfall is represented as 1 and no rainfall as 0. This transformation is necessary because machine learning algorithms such as Decision Tree and Naive Bayes require numerical input for model training.

In addition, feature scaling or normalization can be applied to ensure that all input variables are on a comparable scale, which may improve model performance. The dataset distribution was also analyzed to identify the presence of class imbalance, as imbalanced data can bias the model toward the majority class and affect classification performance. Therefore, proper preprocessing plays a crucial role in improving the reliability and effectiveness of the classification models.

C. Rainfall Data Distribution

After the preprocessing stage was completed, the next step was to analyze the distribution of rainfall data in the dataset. This analysis was conducted to determine the proportion between the rain and no rain classes, as class distribution is an important factor that can influence the performance of classification models. Understanding the distribution of the target variable provides insight into dataset characteristics and helps identify the presence of class imbalance. The distribution of rainfall data in the dataset is presented in Figure 2.

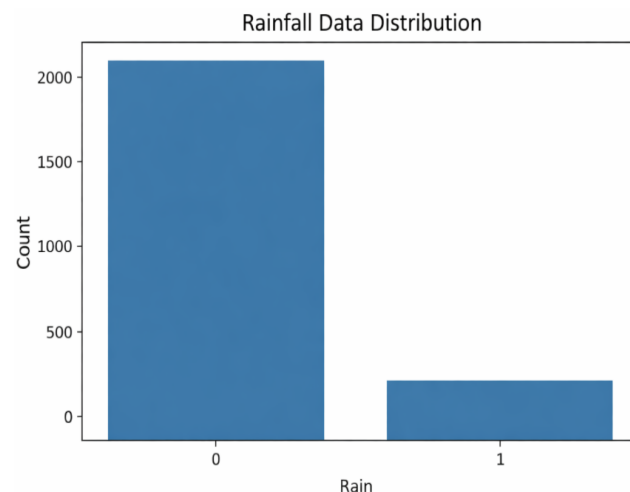


Figure 2. Distribution of Rainfall Data

Based on Figure 2, it can be observed that the number of "no rain" (0) instances is significantly higher than the number of "rain" (1) instances. This indicates that the dataset exhibits a class imbalance, where the majority of the data belongs to the no rain category.

Class imbalance can affect the performance of classification algorithms, as models tend to be biased toward the majority class. This condition may lead to high accuracy values while the model performs poorly in correctly identifying the minority class (rain). In particular, probabilistic models such as Naive Bayes are more sensitive to imbalanced data because they rely on class probability distributions.

On the other hand, Decision Tree algorithms are generally more robust in handling imbalanced datasets, as they can

capture decision boundaries based on feature splits without relying on strict probabilistic assumptions. This difference in handling class imbalance may partially explain the performance gap observed between the two algorithms in this study.

Therefore, analyzing data distribution is essential not only to understand dataset characteristics but also to interpret model performance results more critically and avoid misleading conclusions based solely on accuracy metrics.

D. Evaluation of the Decision Tree Model

In this stage, the classification model built using the Decision Tree algorithm was evaluated to assess its performance in classifying rainfall and non-rainfall data. Model evaluation was conducted using a confusion matrix, which compares the predicted results with the actual class labels. The confusion matrix provides detailed information regarding correctly classified instances as well as misclassification errors. The evaluation results of the Decision Tree model are presented in Figure 3.

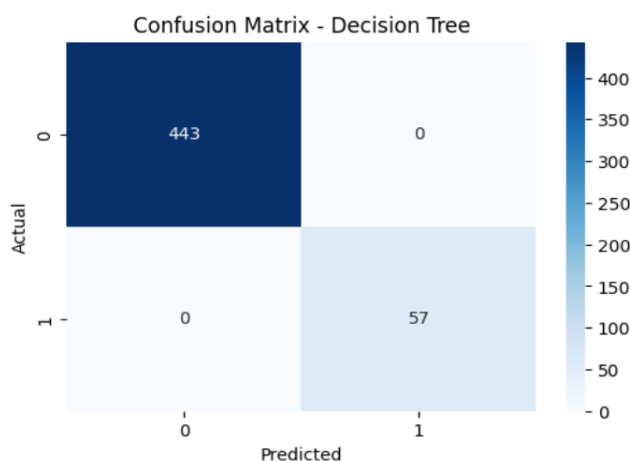


Figure 3. Confusion Matrix of the Decision Tree Model

Based on Figure 3, the Decision Tree model correctly classified 443 instances of no rain (True Negative) and 57 instances of rain (True Positive), with no False Positive or False Negative cases. This result indicates that the model achieved an accuracy of 100%.

Although this result suggests that the Decision Tree model performs exceptionally well, such perfect accuracy is uncommon in real-world classification problems. This condition may indicate potential overfitting, where the model learns the training data too closely and may not generalize well to unseen data. Overfitting is more likely to occur when the dataset is relatively limited or when the model complexity is high, as is often the case with Decision Tree algorithms.

Furthermore, the presence of class imbalance in the dataset may also contribute to this result, as the model may become biased toward dominant patterns in the data. Therefore, relying solely on accuracy as an evaluation metric may lead to misleading conclusions. Additional evaluation metrics such as precision, recall, and F1-score, as well as validation techniques like cross-validation, are necessary to provide a more reliable assessment of the model's performance.

Thus, while the Decision Tree model shows excellent performance on the given dataset, further validation is required

to ensure its robustness and generalization capability in real-world applications.

E. Naive Bayes Model Evaluation

In addition to the Decision Tree model, this study also evaluates the performance of the Naive Bayes algorithm in classifying rainfall data. Model evaluation is conducted using a confusion matrix, which compares the predicted results with the actual class labels to assess classification performance. The confusion matrix provides information regarding correctly classified instances as well as misclassification errors. The evaluation results of the Naive Bayes model are presented in Figure 4.

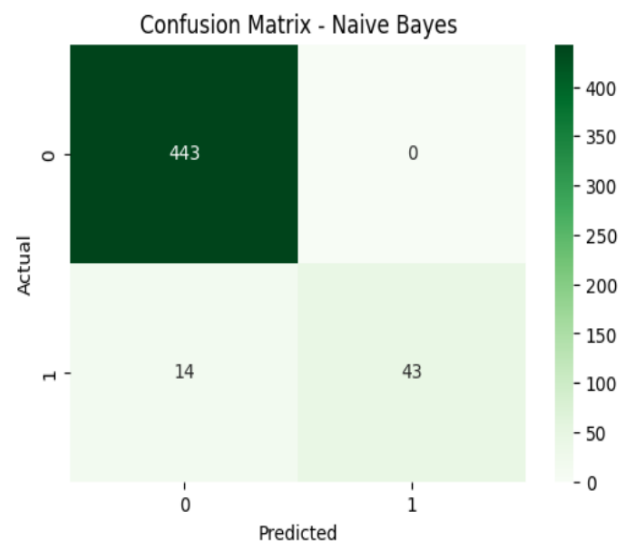


Figure 4. Confusion Matrix of the Naive Bayes Model

Based on Figure 4, the Naive Bayes model correctly classified 443 instances of no rain (True Negative) and 43 instances of rain (True Positive). However, the model produced 14 misclassifications, where actual rain instances were predicted as no rain (False Negative). There are no False Positive cases, indicating that the model tends to predict the majority class more confidently. The overall accuracy achieved by the Naive Bayes model is 0.972 (97.2%), which indicates strong classification performance.

Despite this high accuracy, the presence of False Negative predictions suggests that the model has limitations in correctly identifying rainfall events. This limitation can be attributed to the fundamental assumption of the Naive Bayes algorithm, which assumes independence among input features. In real-world weather data, variables such as humidity, cloud cover, and temperature are often correlated, which violates this assumption and may reduce the model's effectiveness.

Additionally, the class imbalance observed in the dataset may further influence the model's performance, as Naive Bayes relies on probability distributions that can be biased toward the majority class. This explains why the model produces no False Positive cases but fails to correctly classify some rain instances. Therefore, although the Naive Bayes algorithm demonstrates relatively high accuracy, its performance is slightly lower compared to the Decision Tree model. This difference highlights the importance of considering both dataset characteristics and algorithm assumptions when interpreting

classification results. Furthermore, additional evaluation metrics such as precision, recall, and F1-score are necessary to provide a more comprehensive assessment of the model's performance, particularly in handling the minority class.

F. ROC Curve Decision Tree

In addition to using a confusion matrix, the performance of the Decision Tree model was further evaluated using the Receiver Operating Characteristic (ROC) curve. The ROC curve is used to measure the ability of a classification model to distinguish between positive and negative classes by illustrating the relationship between the True Positive Rate (TPR) and the False Positive Rate (FPR) across different threshold values. The effectiveness of the model is commonly quantified using the Area Under the Curve (AUC) value. The ROC curve of the Decision Tree model is presented in Figure 5.

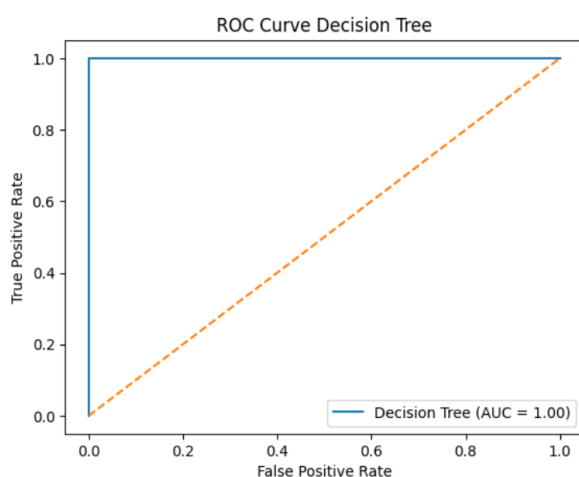


Figure 5. ROC Curve of the Decision Tree Model

Based on Figure 5, the ROC curve of the Decision Tree model shows an AUC value of 1.00, indicating an excellent ability to distinguish between rainfall and non-rainfall classes. The curve is located very close to the top-left corner of the graph, which reflects a high True Positive Rate and a low False Positive Rate.

However, similar to the accuracy results obtained from the confusion matrix, the perfect AUC value should be interpreted with caution. In practical machine learning applications, an AUC value of 1.00 is rarely achieved and may indicate that the model is overfitting the dataset. This means that the model may have learned the training data too specifically and may not generalize well to new, unseen data.

Additionally, the presence of class imbalance in the dataset may also influence the ROC performance, as models trained on imbalanced data can appear to perform very well when evaluated using certain metrics. Therefore, relying solely on ROC and AUC values may not provide a complete evaluation of model performance.

To ensure a more reliable assessment, it is important to complement ROC analysis with other evaluation metrics such as precision, recall, and F1-score, as well as validation techniques like cross-validation. These additional evaluations help provide a more comprehensive understanding of the model's robustness and generalization capability.

G. Model Accuracy Comparison

After evaluating each classification model using the confusion matrix and ROC curve, the next step is to compare the performance of the Decision Tree and Naive Bayes algorithms. This comparison aims to identify which algorithm performs better in classifying rainfall based on the given weather dataset. Although accuracy is used as a primary metric to measure overall performance, relying solely on accuracy may not provide a complete evaluation, especially in the presence of class imbalance. The comparison of model accuracy between the Decision Tree and Naive Bayes algorithms is presented in Figure 6.

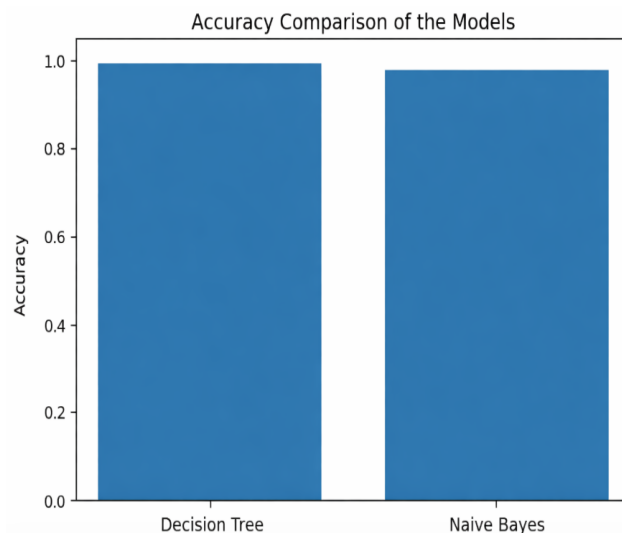


Figure 6. Accuracy Comparison of the Classification Models

Based on Figure 6, the Decision Tree model achieves an accuracy of 1.00 (100%), while the Naive Bayes model achieves an accuracy of 0.972 (97.2%). At first glance, this result indicates that the Decision Tree algorithm outperforms the Naive Bayes algorithm. However, this result should be interpreted carefully.

The perfect accuracy achieved by the Decision Tree model may indicate potential overfitting, where the model learns the dataset too specifically and may not generalize well to unseen data. In contrast, the Naive Bayes model, although slightly lower in accuracy, produces several misclassifications, which may reflect a more realistic performance in handling unseen data.

The difference in performance can also be explained by the characteristics of the algorithms. Decision Tree is capable of capturing complex and non-linear relationships between features without requiring assumptions about feature independence. On the other hand, Naive Bayes assumes independence among features, which may not hold in weather datasets where variables such as humidity, temperature, and cloud cover are often correlated.

In addition, the presence of class imbalance in the dataset may influence the evaluation results, as models tend to favor the majority class. Therefore, it is important to consider additional evaluation metrics such as precision, recall, and F1-score, as well as validation techniques like cross-validation, to

obtain a more comprehensive comparison of model performance.

Overall, while the Decision Tree model shows higher accuracy in this study, the results should be interpreted critically by considering dataset characteristics, algorithm assumptions, and potential overfitting issues. The results of this study are consistent with previous research that has demonstrated the effectiveness of Decision Tree in classification tasks involving structured datasets. For instance, [4] reported that Decision Tree performs well in weather-related classification problems due to its ability to capture complex relationships among features.

Similarly, [7] found that Decision Tree outperformed Naive Bayes in rainfall classification, particularly when the dataset contains correlated variables such as humidity and temperature. This finding is in line with the results of this study, where Decision Tree achieved higher accuracy compared to Naive Bayes.

However, unlike previous studies that primarily focus on algorithm comparison, this study also considers dataset characteristics such as class imbalance and implements the model into a web-based application. Therefore, this research not only provides comparative analysis but also contributes to practical implementation in rainfall prediction systems.

H. Additional Performance Evaluation Metrics

To provide a more comprehensive evaluation beyond accuracy, additional performance metrics including precision, recall, and F1-score are presented in Table 1.

Tabel 1. Performance Evaluation Metrics of Decision Tree and Naive Bayes Models

Model	Accuracy	Precision	Recall	F1-Score
Decision Tree	1.00	1.00	1.00	1.00
Naive Bayes	0.972	1.00	0.754	0.86

Table 1 presents the evaluation results of the Decision Tree and Naive Bayes models using additional performance metrics, including precision, recall, and F1-score. These metrics provide a more comprehensive evaluation beyond accuracy, particularly in handling class imbalance.

The Decision Tree model achieves perfect scores across all metrics, which indicates excellent classification performance but may also suggest potential overfitting. On the other hand, the Naive Bayes model achieves high precision but lower recall, indicating that the model fails to correctly identify some rainfall instances (false negatives). This result shows that Naive Bayes is less effective in detecting the minority class compared to Decision Tree.

I. Web-Based System Implementation

In the final stage of this study, the classification models developed using the Decision Tree and Naive Bayes algorithms were implemented into a web-based application. This implementation aims to provide a practical tool that enables users to predict rainfall conditions based on input weather parameters. The application was developed using the Python programming language with the Flask framework as the backend, combined with a user-friendly web interface that allows users to input weather data through a browser.

The interface of the web-based rainfall prediction system is presented in Figure 7 and Figure 8. Users are required to input several weather parameters, including temperature, humidity, wind speed, cloud cover, and atmospheric pressure. After entering the required values, users can click the “Predict” button to generate the prediction result, indicating whether the condition is classified as Rain or No Rain.

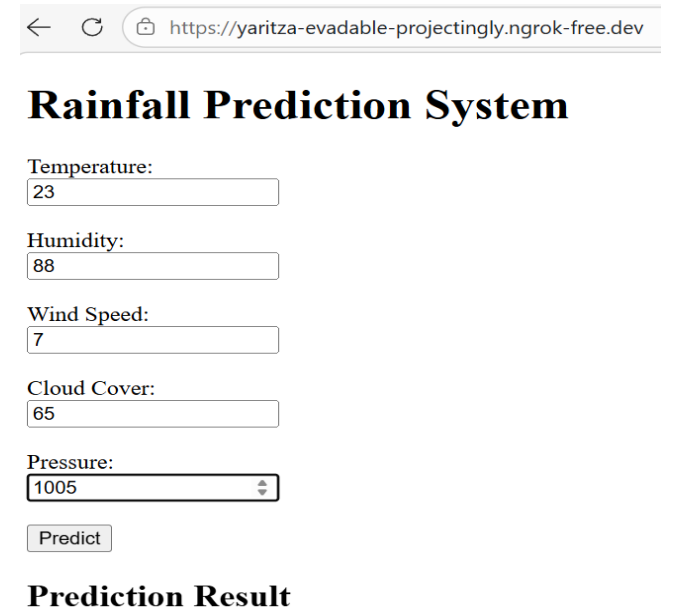


Figure 7. Web-Based Rainfall Prediction System Interface Showing Rain Prediction Result

Based on the testing scenario shown in Figure 7, when the input values are temperature = 23, humidity = 88, wind speed = 7, cloud cover = 65, and pressure = 1005, the system predicts Rain. This result reflects that higher humidity and cloud cover tend to increase the likelihood of rainfall, which is consistent with general meteorological patterns.

← ↻ 🌐 <https://yaritza-evadable-projectingly.ngrok-free.dev>

Rainfall Prediction System

Temperature:

Humidity:

Wind Speed:

Cloud Cover:

Pressure:

Prediction Result

No Rain

Figure 8. Web-Based Rainfall Prediction System Interface
Showing No Rain Prediction Result

In contrast, as shown in Figure 8, when the input values are temperature = 30, humidity = 45, wind speed = 4, cloud cover = 10, and pressure = 1005, the system predicts No Rain. This indicates that lower humidity and cloud cover are associated with a lower probability of rainfall.

The implementation of the classification models into a web-based system demonstrates the practical applicability of machine learning techniques in supporting weather prediction tasks. However, the reliability of the system is highly dependent on the performance and generalization ability of the underlying models. Considering that the Decision Tree model achieved perfect accuracy, which may indicate overfitting, further validation using larger and more diverse datasets is necessary to ensure the robustness of the system in real-world scenarios.

Therefore, while the developed system provides a useful and interactive tool for rainfall prediction, future improvements are required, including enhancing model generalization, incorporating additional evaluation metrics, and testing the system with real-world data to improve its reliability and applicability.

IV. CONCLUSION

Based on the findings of this study, both Decision Tree and Naive Bayes algorithms can be effectively applied to classify rainfall conditions using weather parameters such as temperature, humidity, wind speed, cloud cover, and atmospheric pressure. The experimental results show that the Decision Tree model achieves higher accuracy compared to the Naive Bayes model, with values of 1.00 (100%) and 0.972 (97.2%), respectively. This indicates that the Decision Tree algorithm is more capable of capturing the relationships among weather attributes in the dataset used in this study. However, the perfect accuracy obtained by the Decision Tree model

should be interpreted with caution, as it may indicate potential overfitting and limited generalization capability. Therefore, the evaluation results should not rely solely on accuracy but also consider other performance metrics and validation techniques to ensure model reliability. The implementation of the classification models into a web-based application demonstrates the practical applicability of machine learning techniques in supporting rainfall prediction. The developed system allows users to input weather parameters and obtain prediction results in a fast and interactive manner, which can be useful for decision-making in areas such as agriculture, transportation, and environmental monitoring. Despite these contributions, this study has several limitations, including the use of a single dataset and limited exploration of other classification algorithms. Therefore, future research is recommended to use larger and more diverse datasets, apply additional machine learning methods such as Random Forest or Support Vector Machine, and incorporate more comprehensive evaluation metrics. Furthermore, improving the robustness and scalability of the web-based system is also necessary to enhance its applicability in real-world scenarios.

REFERENSI

- [1] G. DE COLOMBIA, "SMART CITIES," *RESEARCHGATE*, pp. 1–51, 2017.
- [2] M. LIYEW AND G. MELESE, "MACHINE LEARNING APPROACHES FOR RAINFALL PREDICTION: A REVIEW," *INTERNATIONAL JOURNAL OF CLIMATOLOGY*, vol. 36, no. 3, pp. 255–272, 2021.
- [3] K. R. B AND G. A, "RAINFALL PREDICTION USING DATA MINING TECHNIQUES: A SURVEY," vol. 83, no. 8, pp. 23–30, 2013.
- [4] Y. FINDAWATI, I. R. I. ASTUTIK, A. S. FITRONI, I. INDRAWATI, AND N. YUNIASIH, "COMPARATIVE ANALYSIS OF NAÏVE BAYES, K-NEAREST NEIGHBOR, AND C4.5 METHODS IN WEATHER FORECASTING," *JOURNAL OF PHYSICS: CONFERENCE SERIES*, vol. 1402, no. 6, pp. 1–6, 2019.
- [5] V. KUMAR, N. KEDAM, O. KISI, S. ALSULAMY, K. M. KHEDHER, AND M. A. SALEM, "A COMPARATIVE STUDY OF MACHINE LEARNING MODELS FOR DAILY AND WEEKLY RAINFALL FORECASTING," *WATER RESOURCES MANAGEMENT*, vol. 39, no. 1, pp. 271–290, 2025.
- [6] A. NOEMAN, D. HANDAYANI, AND A. HISWARA, "DECISION TREE-BASED WEATHER PREDICTION," vol. 10, no. 225, pp. 67–78, 2026.
- [7] M. W. MARTADIANSYAH, A. GHUFRON, R. A. HIDAYAH, D. SALZABILA, AND L. AMANDA, "COMPARISON OF C4.5 ALGORITHM AND NAÏVE BAYES FOR RAINFALL CLASSIFICATION," *JURNAL KOMPUTER DAN ANTARMUKA*, vol. 3, no. 1, pp. 8–17, 2025.
- [8] M. A. SULARSO AND J. JUPRON, "THE EFFECT OF DATA IMBALANCE ON THE PERFORMANCE OF THE NAÏVE BAYES ALGORITHM IN STROKE CLASSIFICATION," *INFOTECH JOURNAL OF INFORMATION TECHNOLOGY*, vol. 6, no. 2, pp. 212–218, 2025.

- [9] I. ENGINEERING ET AL., "FLOOD DISASTER CLASSIFICATION ANALYSIS BASED ON ...," VOL. 5, NO. 2, PP. 109–117, 2021.
- [10] N. A. PRAKOSO INDARYONO, "COMPARATIVE ANALYSIS OF RANDOM FOREST AND NAÏVE BAYES ALGORITHMS FOR RAINFALL CLASSIFICATION BASED ON CLIMATE IN INDONESIA," *JIPi (JOURNAL OF INFORMATICS RESEARCH AND LEARNING)*, VOL. 9, NO. 1, PP. 158–167, 2024.
- [11] M. KOLAM AND D. NOTIFIKASI, "JURNAL RESTI," *RESTI*, VOL. 1, NO. 10, PP. 19–25, 2021.
- [12] R. IRMANITA, S. S. PRASETIYOWATI, AND Y. SIBARONI, "CLASSIFICATION OF MALARIA COMPLICATION USING CART (CLASSIFICATION AND REGRESSION TREE) AND NAÏVE BAYES," *JURNAL RESTI*, VOL. 5, NO. 1, PP. 10–16, 2021.
- [13] H. OKTAVIANTO, H. W. SULISTYO, G. WIJAYA, D. IRAWAN, AND G. ABDURRAHMAN, "COMPARATIVE ANALYSIS OF DECISION TREE AND RANDOM FOREST IN HEALTH TEXT DATA CLASSIFICATION," *BINA INSANI ICT JOURNAL*, VOL. 11, NO. 1, P. 56, 2024.
- [14] A. HURA, "APPLICATION OF MACHINE LEARNING FOR AGRICULTURAL PRODUCTIVITY PREDICTION BASED ON WEATHER DATA IN INDONESIA," *JIARTIN JOURNAL OF ARTIFICIAL INTELLIGENCE NUSANTARA*, VOL. 1, NO. 1, PP. 1–8, 2025.
- [15] A. S. AGUNG, A. A. FAUZI, A. A. NUR RISAL, AND F. ADIBA, "IMPLEMENTATION OF DATA MINING TECHNIQUES FOR RAINFALL PREDICTION CLASSIFICATION USING BMKG DATA IN SOUTH SULAWESI," *JURNAL TEKNO INSENTIF*, VOL. 17, NO. 1, PP. 22–23, 2023.