Data and Metadata. 2025; 4:1075
doi: 10.56294/dm20251075

ORIGINAL

Predicting Surabaya's Rainfall: A Comparative Study of Naïve Bayes, K-Nearest Neighbor, and Random Forest

Predicción de las precipitaciones en Surabaya: un estudio comparativo de Naïve Bayes, K-vecino más cercano y bosque aleatorio

Arip Ramadan[1], Muhammad Axel Syahputra[2], Dwi Rantini[2,3] ✉, Ratih Ardiati Ningrum[2,3], Muhammad Noor Fakhruzzaman[2,3], Aziz Fajar[2,3], Maryamah[2,3], Muhammad Mahdy Yandra[2], Najma Attaqiya Alya[2], Mochammad Fahd Ali Hillaby[2], Alhassan Sesay[4]

[1]Information System Study Program, School of Industrial and System Engineering, Telkom University Surabaya Campus. Surabaya, 60231, Indonesia.
[2]Data Science Technology Study Program, Faculty of Advanced Technology and Multidiscipline, Universitas Airlangga. Surabaya, 60115, Indonesia.
[3]Research Group of Data-Driven Decision Support System, Faculty of Advanced Technology and Multidiscipline, Universitas Airlangga. Surabaya, 60115, Indonesia.
[4]Faculty of Transformative Education, the United Methodist University. Sierra Leone.

Cite as: Rantini D, Hillaby MFA, Alya NA, Yandra MM, Maryamah, Fajar A, et al. Predicting Surabaya's Rainfall: A Comparative Study of Naïve Bayes, K-Nearest Neighbor, and Random Forest. Data and Metadata. 2025; 4:1075. https://doi.org/10.56294/dm20251075

Submitted: 02-10-2024 **Revised:** 08-02-2025 **Accepted:** 16-06-2025 **Published:** 17-06-2025

Editor: Dr. Adrián Alejandro Vitón Castillo

Corresponding Author: Dwi Rantini ✉

ABSTRACT

Introduction: accurate rainfall prediction plays a critical role in climate change adaptation, particularly in mitigating the risks of extreme droughts and floods. Reliable forecasts support sustainable water resource and agricultural management, contributing to reduced socio-economic vulnerability. This study aims to analyze rainfall conditions in Surabaya City and evaluate the performance of three classification methods to determine the most effective model for rainfall classification.
Method: this is a descriptive observational study using secondary data from the Meteorology, Climatology, and Geophysics Agency Maritime Station in Surabaya, covering the period from January 2019 to December 2023. The dataset consists of 1822 daily weather observations, including rainfall, sunshine duration, temperature, wind speed, and humidity. After preprocessing, the rainfall variable was categorized into multiple classes. Three classification methods—Naïve Bayes, K-Nearest Neighbor, and Random Forest—were applied. Model performance was evaluated using accuracy, precision, recall, AUC-ROC, and loss function values.
Results: all models achieved high accuracy, exceeding 0,93. Although Naïve Bayes showed slightly lower accuracy than the other two methods, it had the highest AUC-ROC and the lowest loss function value, indicating better class discrimination and generalization.
Conclusions: the Naïve Bayes classifier is the most effective method for rainfall classification in Surabaya City. Among the predictor variables, sunshine duration is identified as the most influential factor in rainfall classification, followed by humidity, temperature, and wind speed.

Keywords: Rainfall; Classification; Naïve Bayes Classifier; K-Nearest Neighbor; Random Forest; Climate Prediction.

RESUMEN

Introducción: la predicción precisa de las precipitaciones desempeña un papel fundamental en la adaptación al cambio climático, en particular en la mitigación de los riesgos de sequías e inundaciones extremas. Los pronósticos fiables respaldan la gestión sostenible de los recursos hídricos y la agricultura, contribuyendo así

a la reducción de la vulnerabilidad socioeconómica. Este estudio tiene como objetivo analizar las condiciones de las precipitaciones en la ciudad de Surabaya y evaluar el rendimiento de tres métodos de clasificación para determinar el modelo más eficaz.
Método: este es un estudio observacional descriptivo que utiliza datos secundarios de la Estación Marítima de la Agencia de Meteorología, Climatología y Geofísica en Surabaya, que abarca el período comprendido entre enero de 2019 y diciembre de 2023. El conjunto de datos consta de 1822 observaciones meteorológicas diarias, que incluyen precipitaciones, duración de la insolación, temperatura, velocidad del viento y humedad. Tras el preprocesamiento, la variable de precipitación se clasificó en múltiples clases. Se aplicaron tres métodos de clasificación: Naïve Bayes, K-vecino más cercano y Bosque aleatorio. El rendimiento del modelo se evaluó mediante la exactitud, la precisión, la recuperación, el AUC-ROC y los valores de la función de pérdida.
Resultados: todos los modelos alcanzaron una alta precisión, superior a 0,93. Si bien el método Naïve Bayes mostró una precisión ligeramente inferior a la de los otros dos métodos, obtuvo el AUC-ROC más alto y el valor de función de pérdida más bajo, lo que indica una mejor discriminación de clases y generalización.
Conclusiones: el clasificador Naïve Bayes es el método más eficaz para la clasificación de la precipitación en la ciudad de Surabaya. Entre las variables predictoras, la duración de la insolación se identifica como el factor más influyente en la clasificación de la precipitación, seguida de la humedad, la temperatura y la velocidad del viento.

Palabras clave: Precipitación; Clasificación; Clasificador Bayesiano Ingenuo; K-Vecino Más Próximo; Bosque Aleatorio; Predicción Climática.

INTRODUCTION

Weather conditions describe the short-term state of the atmosphere in a specific location, involving factors such as temperature, wind speed, humidity, sunshine duration, and rainfall.[1] Among these, rainfall is a key meteorological variable that significantly influences changes in weather patterns.[2] It is defined as the volume of precipitation that falls on a flat surface over a specified period and is measured in millimeters (mm).[3] Globally, changing rainfall patterns have disrupted agriculture, infrastructure, and disaster response. According to the World Meteorological Organization, extreme rainfall events have increased by over 30 % in the past two decades due to climate change. Indonesia, a tropical country with diverse climatic zones, is particularly vulnerable to irregular and intense rainfall.

As one of Indonesia's largest metropolitan areas, Surabaya is prone to unpredictable weather variations, especially during the rainy season. Meteorology, Climatology, and Geophysics Agency reported that from 2019 to 2023, Surabaya experienced an annual average rainfall of 1800–2200 mm, with several extreme events triggering urban floods and infrastructure damage. For instance, in early 2023, daily rainfall reached 150 mm in a single day—well above the daily average of 60–80 mm. These events disrupt public mobility, affect economic activities, and highlight the urgency of accurate rainfall prediction models to support early warning systems and disaster risk reduction strategies. Advancements in computing technology have enabled the integration of machine learning methods into weather forecasting, especially in classifying and predicting rainfall occurrences.[4] Among the widely used methods are Naïve Bayes, K-Nearest Neighbor (KNN), and Random Forest classifiers.

Several previous studies have compared classification methods for weather, including the Naïve Bayes, Random Forest, and Decision Tree methods. Mistry et al.[5] used daily weather observation data from various Australian weather for their research. Other studies have discussed the classification of rainfall using the K-Nearest Neighbor method, as conducted by Huang et al.[6]. Pandey et al.[7] compared the Logistic Regression and Random Forest methods for rainfall prediction. Research by Shaji et al.[8], utilized meteorological data from India to evaluate and compare the performance of various machine learning classification algorithms for weather forecasting tasks. Then, research by Chen et al.[9] used the KNN method for rainfall detection. These studies affirm the potential of machine learning in enhancing weather prediction accuracy.

However, comparative studies focused on Surabaya's unique geographical and meteorological conditions remain scarce. Each classifier offers specific advantages and drawbacks. Naïve Bayes offers computational efficiency and is well-suited for handling categorical data, though it operates under the assumption of conditional independence among its features. K-Nearest Neighbor is intuitive and flexible but can be sensitive to noise and non-informative features, especially in high-dimensional data. Random Forest, a robust ensemble method, is known for its high accuracy and ability to handle large, complex datasets through aggregation of decision trees.

Therefore, the objective of this research is to compare the performance of Naïve Bayes, K-Nearest Neighbor, and Random Forest classifiers in rainfall classification using historical weather data from Surabaya between 2019 and 2023. The study also aims to identify rainfall trends and evaluate the models' predictive capabilities using accuracy, precision, recall, and AUC-ROC metrics. The findings are expected to support the development of adaptive systems for weather monitoring, disaster preparedness, and urban planning in data-driven ways.

https://doi.org/10.56294/dm20251075

METHOD

Data Set

This research is an observational and descriptive study that utilizes secondary data obtained from the Meteorology, Climatology, and Geophysics Agency. The dataset spans a five-year period, from January 2019 to December 2023, and focuses on weather observations in Surabaya, Indonesia. The study was started from June 2023 and completed in August 2023, and all data analysis and modeling were carried out using Python programming language.

The dataset includes the following variables: rainfall (response variable) and four predictor variables—temperature, humidity, sunshine duration, and wind speed. The response variable (rainfall) was preprocessed and categorized into discrete classes based on predefined rainfall thresholds consistent with Meteorology, Climatology, and Geophysics Agency classification guidelines. These categories serve as the target for classification in this study. Detailed variable descriptions are presented in table 1.

The methodology involved data cleaning, normalization, and categorization of the response variable, followed by the application of three machine learning models: Naïve Bayes, K-Nearest Neighbor, and Random Forest. Model performance was assessed using a train-test split approach (80:20), and evaluation metrics included accuracy, precision, recall, and AUC-ROC. The data processing and analysis steps were fully scripted to ensure reproducibility and transparency.

Since the research uses publicly available secondary data with no personal or sensitive information, no ethical clearance was required. However, data integrity and proper attribution to the Meteorology, Climatology, and Geophysics Agency source were maintained throughout the study. All data were securely stored and processed on institutional computing resources. This methodological framework ensures that the study can be replicated in different regions or datasets with similar variables and structure.

Table 1. Research Variable

Number	Variable	Variable Name	Unit	Explanation
1	Y	Rainfall	Millimeter 0: < 5 mm/day 1: 5 - < 20 mm/day 2: 20 - < 50 mm/day 3: 50 - ≤100 mm/day 4: >100 mm/day	0: very light 1: light 2: moderate 3: heavy 4: very heavy
2	X_1	Temperature	Celsius	Average temperature per day
3	X_2	Humidity	Percent	Average humidity per day
4	X_3	Sunshine duration	Hour	Duration of sunshine
5	X_4	Wind speed	Meters per second	Average wind speed per second

Naïve Bayes

Naïve Bayes classifier is based on a probability theorem known as Bayes' theorem.[10] This algorithm has proven to be effective in predictive modeling and is widely used to classify high-dimensional training datasets.[11] Naïve Bayes is a probabilistic classification method that leverages statistical probability theorems. Its fundamental principle asserts that the presence of a given feature is conditionally independent of other features, thus earning it the 'Naïve' label. The name "Bayes" in Naïve Bayes comes from Bayes' theorem or the law that provides a method for conditional probability. That is, the probability of an event depends on prior knowledge of some related events. Naïve Bayes uses this approach to generate classification models. Bayes' theorem has a general form in equation (1):

$$P(\theta|Y) = \frac{P(Y|\theta)P(\theta)}{P(Y)} \qquad (1)$$

Where:

Y is the observation data, θ is the parameter of Y, P(θ|Y) is the posterior distribution of θ, P(θ) is prior distribution of θ, P(Y|θ) is the likelihood function of Y given θ, and P(Y) is the marginal probability of Y.

K-Nearest Neighbor

As an approach, K-Nearest Neighbor (KNN) applies the principles of supervised learning algorithms.[12] KNN classifies new data points by assigning them to the category that holds the majority among their closest neighbors.[13] The main purpose of this algorithm is to use attributes and training samples to classify new objects.[14]

The steps for solving problems using the KNN method are as follows:

1. The parameter k must be established, representing the count of nearest neighbors.
2. Calculate the distance of each data sample with the data being tested.
3. Sort the data based on the distance from the smallest to the largest.
4. Observe the number of decisions that are the most for the k data taken.
5. If there are two or more classes $\acute{\omega}_i$ that are the closest neighbors of the test data x, then a balanced condition (conflict) occurs and a conflict resolution strategy is used.
6. For each class involved in the conflict, determine the distance between x and class $\acute{\omega}_i$ (representing the class) based on the E nearest neighbors found in class $\acute{\omega}_i$.
7. If the m-th training pattern of class $\acute{\omega}_i$ is involved in the conflict, then the distance between x and class $\acute{\omega}_i$ is calculated using equation (2).

$$d_i = \frac{1}{E}\sum_{j=1}^{n} |(X_j - Y_j)| \quad (2)$$

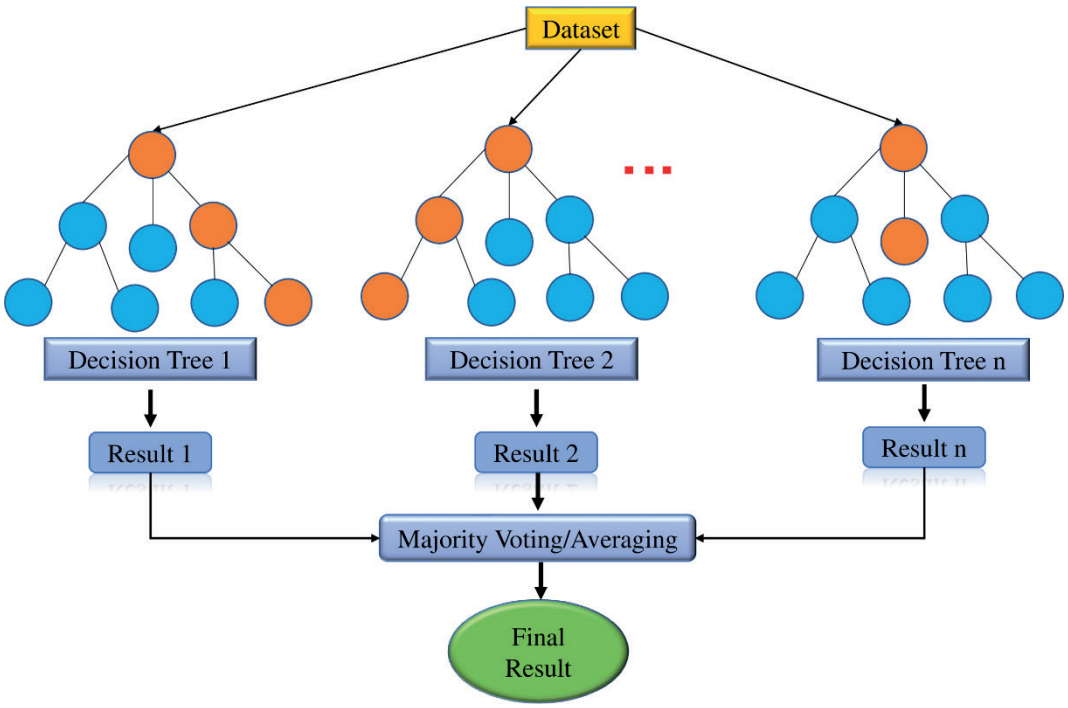
Where:

d_i is the distance between vectors X and Y, E is the number of dimensions or length of vector, n is the index that runs through each dimension of vector, X_j and Y_j is the j-th component of vectors X and Y respectively.

These steps detail the process of using the KNN method to predict the class or category of a data being tested based on the classes of its nearest neighbors.

Random Forest

Random Forest is a classification system consisting of a collection of structured classification trees.[15] In the Random Forest architecture, each constituent decision tree is characterized as an independent and identically distributed random vector. These individual trees then contribute to the final classification by voting for the mode of predicted classes based on the input features. The Random Forest algorithm has several configurable parameters, including the number of decision trees to be created, the criteria used to evaluate the split at each node, and the maximum depth limit of the decision tree.[16] The following is a visualization of Random Forest.



Source: Pham et al.[17]

Figure 1. Random Forest Model

a) Number of Decision Trees: the number of decision trees is a critical parameter in the Random Forest algorithm. This parameter determines the total trees in the Random Forest method, which play a role in the decision-making process and classification prediction. A larger number of trees in Random Forest can affect the prediction process and accuracy. Improvement in model performance can be influenced by the optimality of the number of decision trees.

b) Data Separation Criteria: the data separation criterion is a significant value in evaluating the

Random Forest method. This value indicates the extent to which the model fits or does not fit (in regression) or the extent to which a node can be considered pure (in classification). This criterion plays a role in measuring the quality of the node model fit at the regression stage or the purity of the node at the classification stage. Large values indicate poor model performance.

c) Max Depth: a parameter in the Random Forest algorithm that sets the maximum depth of each tree in the decision forest. Each level of complexity in the decision tree is limited by the maximum tree depth. Increasing the depth of the tree will increase the level of calculation complexity, but also increase the computational cost (execution time). This parameter plays an important role in controlling model complexity and managing computational efficiency.

In this algorithmic framework, a 'forest' of prediction trees (decision trees) is constructed, where each tree's structure is determined by a random vector independently sampled across all trees. The final predictions from the Random Forest are achieved through a consensus mechanism: majority selection for classification and averaging for regression, based on the individual tree outputs. For a Random Forest consisting of N trees, the formula can be expressed in equation (3).

$$\mathrm{l(y)} = argmax_c\left(\sum_{n=1}^{N} I_{h_n(y)=c}\right) \quad (3)$$

Where:

I is the indicator function and h_n is the n-th tree of Random Forest.

Model Selection

The evaluation model used in this research is a confusion matrix, which provides an accuracy value of the algorithm validation against the dataset used. Confusion matrix is a common visualization tool used in supervised learning, where each column of the matrix reflects an example of a predicted class, and each row represents the actual occurrence in that class.[18] An explanation of the confusion matrix is given in table 2.

Table 2. Confusion Matrix

Confusion Matrix		Prediction	
		Negative	Positive
Actual	Negative	TN (True Negative)	FN (False Negative)
	Positive	FP (False Positive)	TP (True Positive)

Source: Fahmy Amin M[18]

Actual condition is the classification of previously predicted rainfall status. Prediction, on the other hand, is the result of the classification of status variables generated by the program.

Some of the requirements that have been defined for the classification matrix include:

Accuracy is defined as the proportion of accurate forecasts out of the total forecasts produced. The accuracy formula can be expressed in equation (4):

$$Accuracy = \frac{TN+TP}{TN+FN+FP+TP} \quad (4)$$

Recall indicates the percentage of all relevant items that were correctly retrieved. The formula for calculating Recall can be expressed in equation (5):

$$Recall = \frac{TP}{FN+TP} \quad (5)$$

Precision is the proportion of correct positive case predictions. The formula for calculating this can be expressed in equation (6):

$$Precision = \frac{TP}{FP+TP} \quad (6)$$

Area Under the Curve and Receiver Operating Characteristic (AUC-ROC) are two-dimensional tools used to

evaluate classification performance with two class decisions.[19] Each object is associated with one of the two elements in the pair of sets, namely positive or negative. In the ROC curve, the true positive rate is plotted on the Y-axis, while the false positive rate is plotted on the X-axis. AUC values can be divided into several groups:[20]

a) Value >0,90 – 1,00: Excellent Classification.
b) Value >0,80 – 0,90: Good Classification.
c) Value >0,70 – 0,80: Fair Classification.
d) Value >0,60 – 0,70: Poor Classification.
e) Value 0,50 – 0,60: Failure.

Research Flow

Here is how the research we conducted works and the flow diagram can be seen in figure 2.

Collecting data

Data collection in this research was obtained through assistance from the Meteorology, Climatology, and Geophysics Agency Maritime of Surabaya City. The data obtained were 1,822 daily from 2019 to 2023. The data has several parameters such as rainfall, duration of sunlight, wind speed, temperature, and humidity.

Pre-processing Data

The pre-processing stage begins by inputting raw data, the next thing to do is data cleaning such as checking missing values with actions taken evenly on the missing data. The data that has been cleaned is then labeled for the "Rainfall" variable to become the "Status" variable with values 0, 1, 2, 3, and 4 according to the rainfall range value. Then, exploratory data analysis is carried out as a data analysis approach by displaying data visualization to see the data distribution and see if there is data that needs to be deleted. The table 3 is the data distribution of each class. The next step is to divide the data into training data and testing data with a ratio of 80:20.

Table 3. Distribution the Number of Rainfall Data in Each Class

Class 0	Class 1	Class 2	Class 3	Class 4
1,743	45	26	7	0

Hyperparameter Tuning

The training process of the three models was carried out with "GridSearchCV" to determine the best parameters. Table 4 shows each variable to be tuned.

Table 4. Parameters to be Used for Tuning Each Model

Classification Methods	The Best Parameters	Explanation
Naïve Bayes	var_smoothing: '1e-9', '1e-8', '1e-7', '1e-6'	var_smoothing: add a small value to the feature variance to make the model more stable. This is useful to prevent zero values in probability calculations that can cause errors or unwanted results.
K-Nearest Neighbor	Metric: 'euclidean', 'manhattan', 'minkowski' n_neighbor: 1, 2, 3, 4, 5, 6 weights: 'uniform', 'distance'	Metric: calculates the distance between data points n_neighbor: determines the number of nearest neighbors considered for classification or regression. weights: determines how weights are assigned to neighbors in making predictions.
Random Forest	max_depth: None, 10, 20, 30 max_features: 'auto', 'sqrt', 'log2' min_samples_leaf: 1, 2, 4 min_samples_split': 2, 5, 10 n_estimators: 10, 50, 100, 200	max_depth: specifies the maximum depth of trees in the forest. max_features: specifies the maximum number of features to consider for splitting at each node. min_samples_leaf: specifies the minimum number of samples that must be present in the last leaf (terminal node). min_samples_split: specifies the minimum number of samples required to split an internal node. n_estimators: specifies the number of trees in the forest.

https://doi.org/10.56294/dm20251075

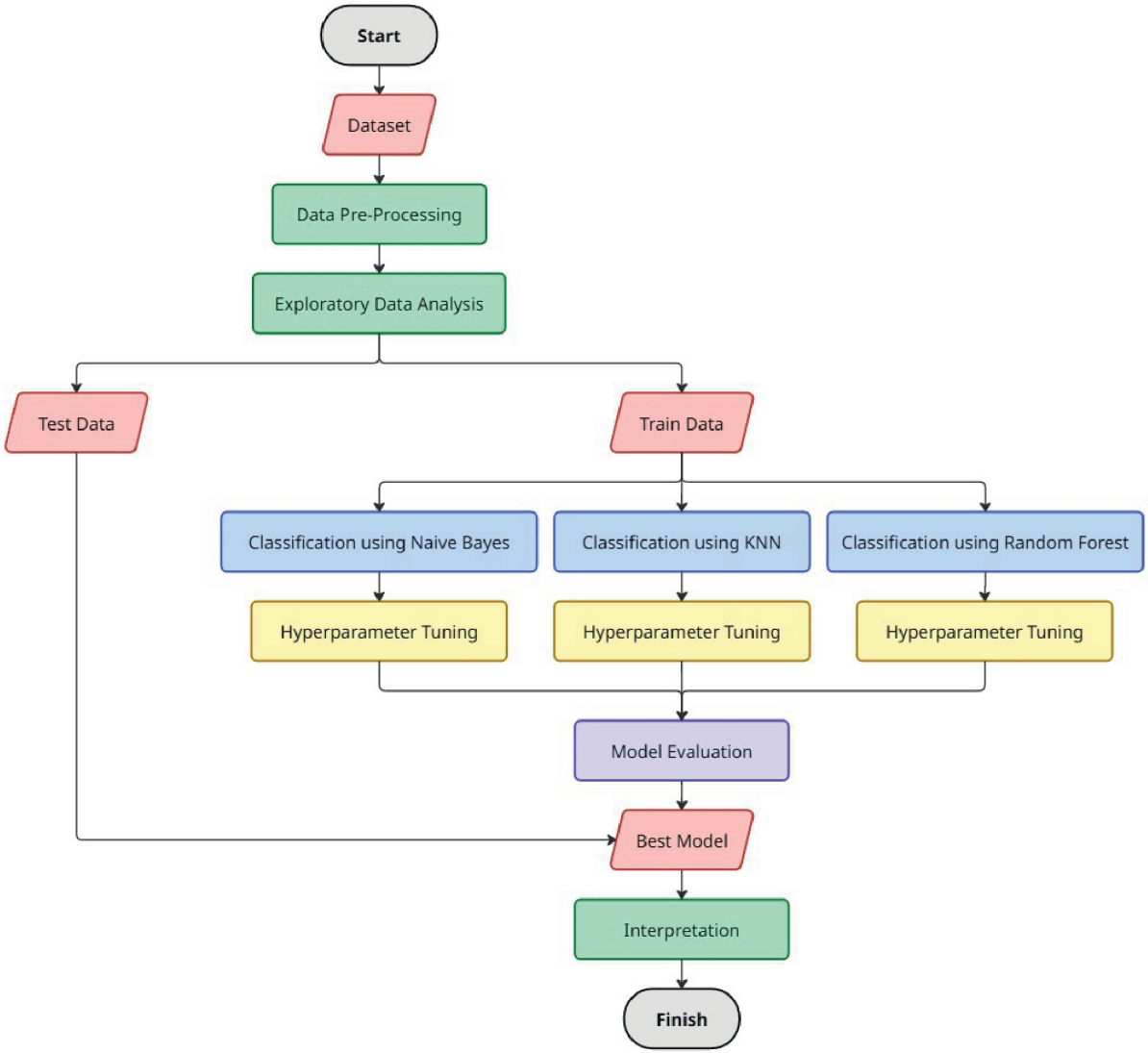


Figure 2. Research Flow Chart

Rainfall Classification Model

The best parameter results obtained from the hyperparameter tuning process will be used for training the rainfall classification model using the three methods, namely Naïve Bayes, K-Nearest Neighbor, and Random Forest. The results obtained are in the form of a confusion matrix model evaluation that will display metrics including: Accuracy, Precision, Recall, and AUC-ROC. Then evaluate the feature importance using Random Forest to see which variables have the most influence on rainfall in the Surabaya city. Furthermore, the loss function is carried out using categorical cross-entropy, because this function is very commonly used in multi-class classification problems.[21] Categorical cross-entropy computes the dissimilarity between the ground truth probability distribution and the model's output probability distribution. By using categorical cross-entropy, the model will learn to minimize the difference between predictions and actual values, thereby increasing classification accuracy. From the results of the model comparison, the best model was obtained by looking at its accuracy value.

RESULTS

Preprocessing Data

Data preprocessing to prepare data for easy analysis, the collected data is preprocessed with the aim of cleaning, organizing, and changing the data into a format that is easier to process in further analysis. Table 5 shows the data after preprocessing, due to limitations, 5 examples were taken to be displayed.

Table 5. Data Display After Preprocessing

Rainfall	Duration Sunshine	Temperature	Wind Speed	Humidity
0	1,3	27,4	4	80
0	0	28,2	5	75
1,72	5,9	25,4	3	94
1,72	0	25,4	5	92
1,72	4,5	27,6	2	87

Exploratory Data Analysis

The next step is exploratory data analysis as a data analysis approach by displaying data visualization. Figure 3 shows the number of days with various levels of rainfall from 2019 to 2023. The rainfall category is classified into four statuses: very light (Status 0, represented by the blue line), light (Status 1, represented by the orange line), moderate (Status 2, represented by the green line), and heavy (Status 3, represented by the red line). The main observation shows that the number of days with very light rainfall is very high compared to other categories, consistently around 350 days per year. This indicates that almost every day of the year experiences rain with very light intensity.

Meanwhile, the number of days with light rainfall increased slightly from 2019 to 2020 and remained relatively stable until 2023, with the number of days ranging from 20 to 25 days per year. The number of days with moderate rainfall is very low and there is almost no significant variation over the five-year period, ranging from 0 to 10 days per year. The number of days with heavy rainfall is also very low, similar to moderate rainfall, with the number of days ranging from 0 to 5 days per year, with no significant increase over this period.

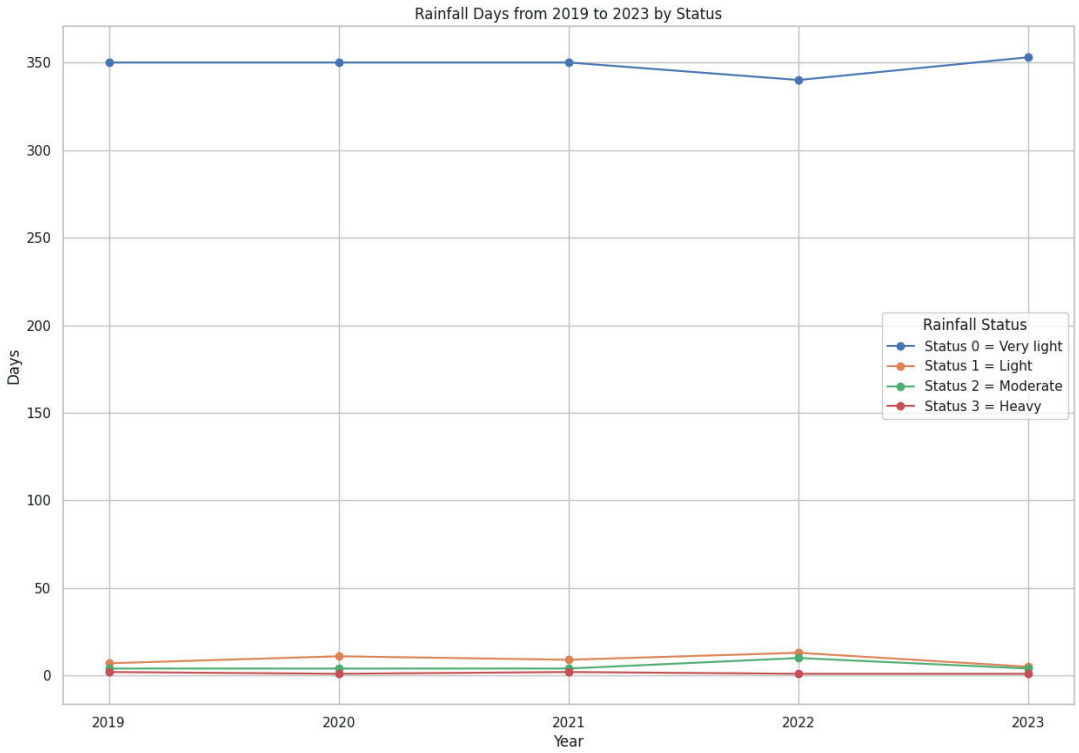


Figure 3. Rainfall Distribution 2019 - 2023

From this visualization, it can be concluded that the analyzed area experiences very light rainfall most of the year with very few days with light, moderate, and heavy rainfall. Table 5 is the data distribution of each class. From this visualization, it can be concluded that the analyzed area experiences very light rainfall almost throughout the year with very few days having light, moderate, and heavy rainfall.

Based on figure 4 shows the distribution of several weather variables based on data from 2019 to 2023. The distribution of sunshine duration shows quite low variability with a consistent median throughout the five years. In 2019, there were several outliers showing sunshine duration values that were much lower than in other years. The years 2020 to 2023 showed a more uniform distribution without any striking outliers. Wind speed has a more varied distribution compared to sunshine duration. The box plot shows some outliers in all years, especially

https://doi.org/10.56294/dm20251075

in 2019 and 2020, indicating the occurrence of winds with much higher speeds. The years 2021 to 2023 have a more concentrated distribution with few outliers. The humidity distribution shows very high values with a consistent median approaching 100 % for each year. There are no striking outliers, indicating that humidity in this region tends to be consistently high and stable throughout the year without significant fluctuations. The temperature distribution shows low variability with a consistent median throughout the five years. The box plot shows some outliers, especially in 2019 and 2020, indicating lower temperature values. The years 2021 to 2023 have a more uniform distribution with no striking outliers. The results of this visualization show that humidity and sunshine duration tend to be stable from year to year with little variation. Wind speed and temperature show more variation and outliers, indicating the presence of extreme weather conditions in certain years.

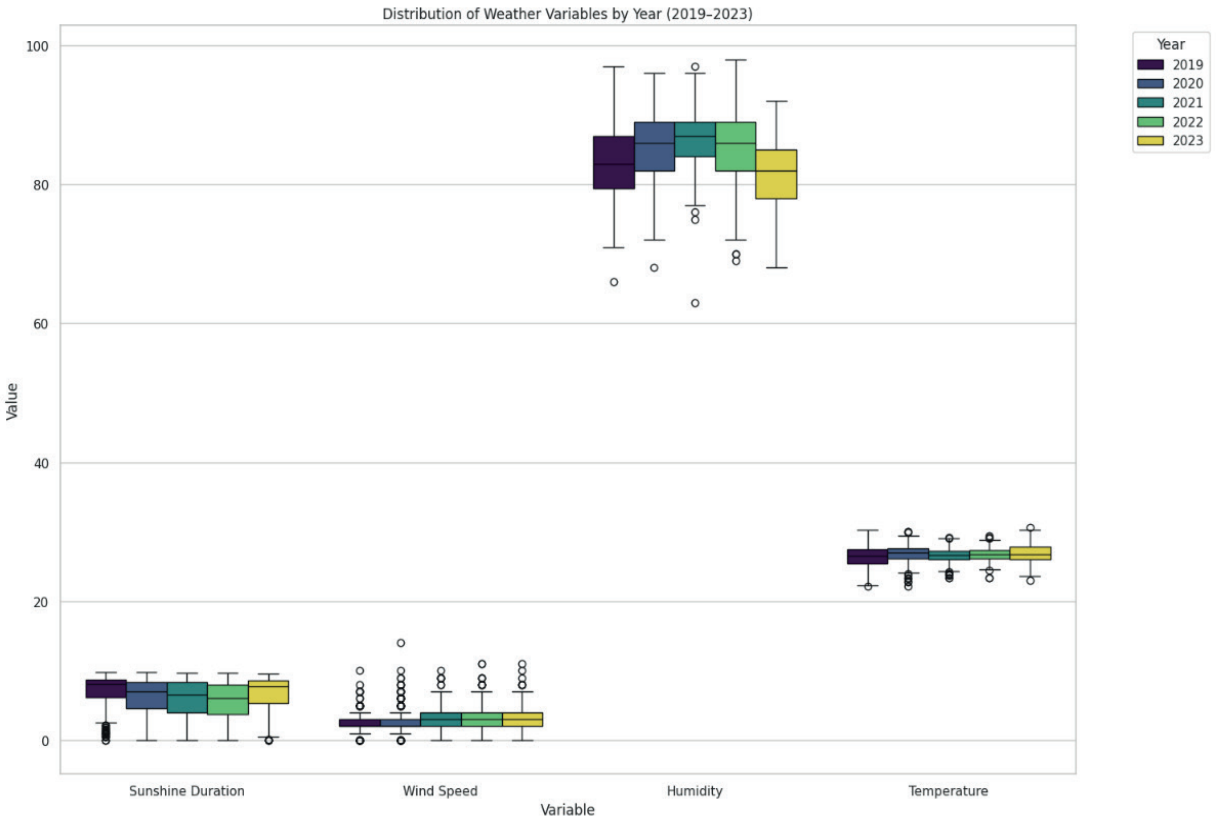


Figure 4. Distribution of Weather Variables 2019 – 2023

Splitting Data

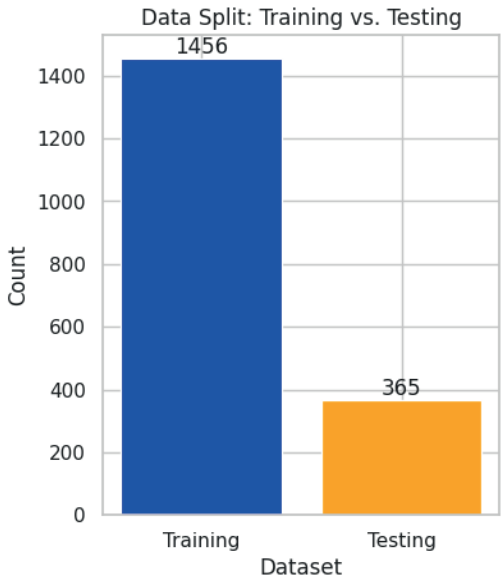


Figure 5. Distribution of Training Data and Testing Data

https://doi.org/10.56294/dm20251075

In the context of classification, data is generally divided into two main parts, namely training data and testing data. Determining the proportion between training data and testing data does not have a standard rule, so it can be adjusted to the needs of the research. Therefore, in this research, figure 4 shows the partition between training data and testing data used.

Based on figure 5, the total number of data used in this research is 1822 data. Of this amount, 80 % is allocated for training data, which is 1456 data, and 365 data is used as testing data. The proportion of 80 % for this training data is determined by the researcher by referring to previous studies.

Hyperparameter Tuning

The following are the results of hyperparameter tuning for three models: Naïve Bayes, K-Nearest Neighbor (KNN), and Random Forest shown in table 6.

Table 6. Results of Hyperparameter Tuning for Each Model

Classification Method	The Best Parameter	The Best Accuracy
Naïve Bayes	var_smoothing: 1e-09	0,9539
K-Nearest Neighbor	Metric: Euclidean, n_neighbor: 6, weights: uniform	0,9594
Random Forest	max_depth': none, 'max_features': auto, 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 10	0,9594

Based on table 6 For the Naïve Bayes method, the optimized parameter is var_smoothing, which is used to add a small variance to the variance of each feature to avoid division by zero. After fitting with GridSearchCV for several candidates, the best parameter found was var_smoothing is 1×10^{-9}. The Best Accuracy achieved with this parameter was 0,9539, indicating that the model with this parameter performed well.

The optimized parameters for K-Nearest Neighbor were metric, n_neighbor, and weights. The metric parameter determines the distance used, n_neighbor determines the number of neighbors considered, and weights determines the weight given to neighbors. After fitting with GridSearchCV, the best parameters found were metric = 'euclidean', n_neighbor = 6, and weights = 'uniform'. The best accuracy achieved with this parameter was 0,9594.

In the Random Forest method, the optimized parameters were max_depth, max_features, min_samples_leaf, min_samples_split, and n_estimators. The max_depth parameter specifies the maximum depth of the tree, max_features specifies the number of features considered to split each node, min_samples_leaf specifies the minimum number of samples required to become a leaf, min_samples_split specifies the minimum number of samples required to split a node, and n_estimators specifies the number of trees in the forest. After fitting with GridSearchCV, the best parameters found were 'max_depth': none, 'max_features': auto, 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 10. The best accuracy achieved with these parameters was 0,9594.

Naïve Bayes Method

The next step after dividing the training and testing data and get the most optimal hyperparameter tuning is to analyze the classification method using testing data. The first classification method applied in this research is the Naïve Bayes. The following presents the results of the discussion based on the analysis using the Naïve Bayes method. Confusion matrix of Naïve Bayes shown in table 7.

Table 7. Confusion Matrix of Naïve Bayes Method

Actual	Prediction			
	0	1	2	3
0	342	0	4	0
1	8	0	0	0
2	9	0	0	0
3	2	0	0	0

Table 7 shows four different classes of rainfall, labeled as Actual 0, Actual 1, Actual 2, and Actual 3. These classes reflect different categories of rainfall. In this confusion matrix analysis, it was found that for Class 0, out of 346 samples that are actually Class 0, the model successfully classified 342 samples correctly (true

https://doi.org/10.56294/dm20251075

positives). However, there are 4 samples that are misclassified as Class 2. No Class 0 samples are misclassified as Class 1 or Class 3. This shows that the model has a very high level of accuracy in classifying Class 0, but there are still errors in distinguishing between Class 0 and Class 2. For Class 1, out of 8 samples that are actually Class 1, none are correctly classified as Class 1 (true positives = 0). All of these samples are misclassified as Class 0. This shows that the model is unable to distinguish Class 1 from Class 0, indicating a major problem in recognizing Class 1. For Class 2, out of 9 samples that are actually Class 2, none are correctly classified as Class 2 (true positives = 0). All these samples are also wrongly classified as Class 0. This is similar to Class 1, indicating that the model is unable to distinguish Class 2 from Class 0. For Class 3, out of the 2 samples that are actually Class 3, none of them are correctly classified as Class 3 (true positives = 0). All these samples are wrongly classified as Class 0. This again indicates that the model is unable to distinguish Class 3 from Class 0.

$$Accuracy = \frac{342 + 0 + 0 + 0}{365} = 0.9370$$

Obtained from the calculation of accuracy, which is 0,9370. This value indicates that the Naïve Bayes method although this accuracy gives a general idea of the model's performance, it is also important to look at other evaluation metrics, especially in cases where there is data imbalance between classes.

$$Precision_0 = \frac{342}{(342 + 19)} = 0.947$$
$$Precision_1 = \frac{0}{(0 + 0)} = nan$$
$$Precision_2 = \frac{0}{(0 + 4)} = 0$$
$$Precision_3 = \frac{0}{(0 + 0)} = nan$$

The precision for class 0 reaches 0,947 or about 94,7 %. This means that of all predictions indicating the sample as Class 0, about 94,7 % are actually Class 0. The high precision value for Class 0 indicates that the model is very effective in predicting this class with few false positives. The precision for class 1 is undefined or nan (not a number). This usually occurs because there are no positive predictions made by the model for Class 1. In other words, the model never identified a sample as Class 1, so there is no basis for calculating precision. This indicates a significant weakness in the model's ability to detect Class 1. The precision for class 2 is 0. This indicates that there are no correct model predictions for Class 2. All predictions made for Class 2 are false positives. A precision value of 0 indicates that the model completely failed to identify samples that were truly from Class 2. The precision for Class 3 is nan (not a number). Just like Class 1, this occurs because there are no positive predictions made for Class 3. The model never identified a sample as Class 3, so precision cannot be calculated. This again shows that the model is unable to detect samples from Class 3.

$$Recall_0 = \frac{342}{(342 + 4)} = 0.988$$

$$Recall_1 = \frac{0}{(0 + 8)} = 0$$

$$Recall_2 = \frac{0}{(0 + 9)} = 0$$

$$Recall_3 = \frac{0}{(0 + 2)} = 0$$

The recall for Class 0 is 0,988 or around 98,8 %. This means that the model is almost always correct in identifying samples that are actually from Class 0. In other words, of all samples that are truly Class 0, around 98,8 % are correctly identified by the model. The high recall value indicates a very good performance in detecting Class 0. The recall for class 1 is 0, which means that the model failed to identify a single sample that actually came from Class 1. The model completely failed to detect this class, indicating that no Class 1 samples were correctly identified by the model.

The recall for class 2 is at 0. This indicates that the model failed to identify a single sample that actually

came from Class 2. Just like Class 1, the model completely failed to detect this class. The recall for class 3 is 0, which means that the model failed to identify a single sample that actually came from Class 3. The model also completely failed to detect samples from Class 3.

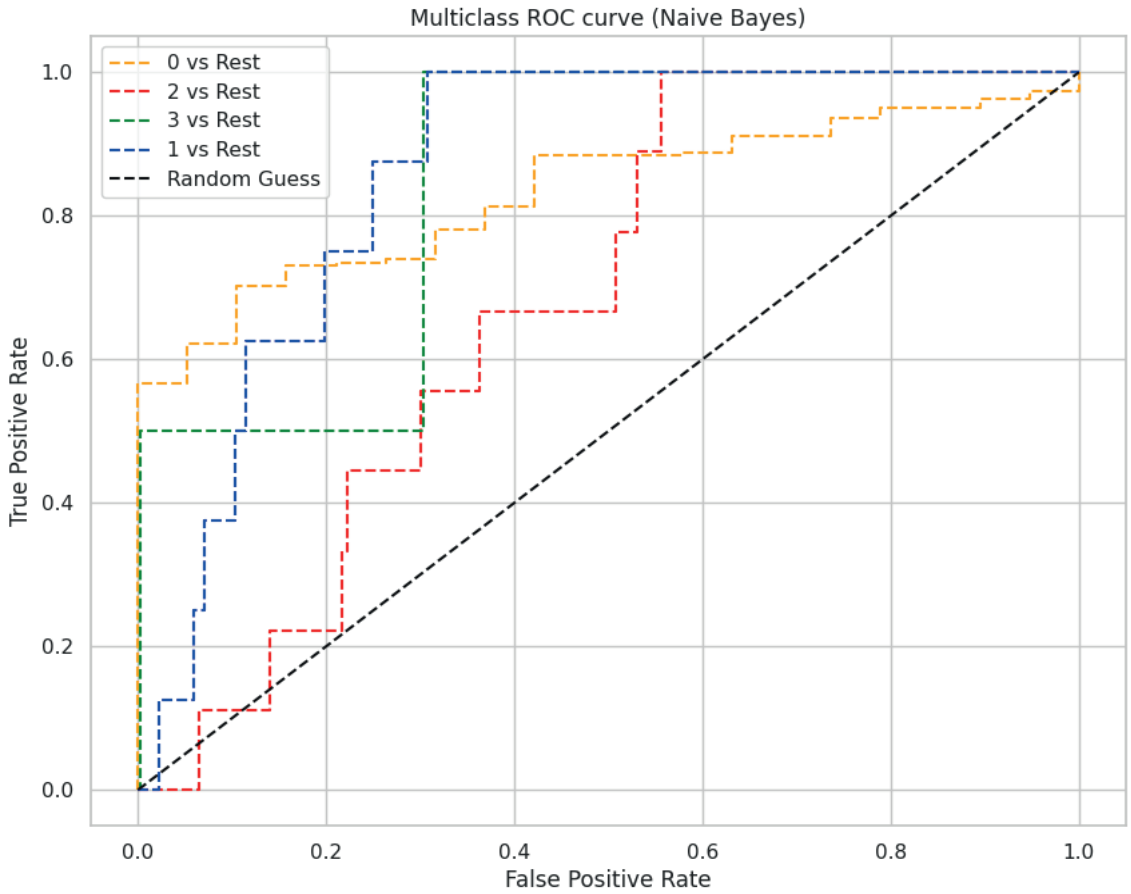


Figure 6. AUC – ROC Curve Naïve Bayes

Overall, from table 7, the Naive Bayes model has a very good performance in distinguishing Class 0 and Class 1 from other classes. Based on figure 6, the ROC curve approaching the upper left corner of the graph. However, the model shows lower performance for Class 2 and Class 3, with the ROC curve being further away from the upper left corner and closer to the random guess line. Overall, with an ROC-AUC value of 0,804, the model shows quite good performance for multiclass classification tasks, although there is still room for improvement, especially in detecting Class 2 and Class 3.

K-Nearest Neighbor Method

The next classification method applied in this research is K-Nearest Neighbor (KNN). The following presents the results of the discussion based on the analysis using the method with a value of k = 6. The confusion matrix of KNN is shown in table 8.

Table 8. Confusion Matrix of K-Nearest Neighbor Method (k=6)

Actual	Prediction			
	0	**1**	**2**	**3**
0	346	0	0	0
1	8	0	0	0
2	9	0	0	0
3	2	0	0	0

https://doi.org/10.56294/dm20251075

Table 8 above shows four different classes of rainfall, labeled as Actual 0, Actual 1, Actual 2, and Actual 3. For Class 0, the model successfully identified 346 samples out of a total of 365 samples that were truly included in Class 0. This is indicated by the True Positive (TP) value of 346 and False Positive (FP) of 0, indicating that there were no incorrect predictions for this class. However, there are 19 samples from other classes (8 from Class 1, 9 from Class 2, and 2 from Class 3) that are misclassified as Class 0, indicating a number of False Negatives (FN) for these classes. In contrast, the model performance for Class 1, Class 2, and Class 3 is very poor.

For Class 1, none of the samples were correctly classified, with a TP of 0 and an FN of 8. All 8 samples that should have been classified as Class 1 were instead classified as Class 0, resulting in an FP is 8 for Class 0. A similar situation occurred in Class 2 and Class 3, where the TP for both classes was also 0. All 9 samples from Class 2 and 2 samples from Class 3 were incorrectly classified as Class 0, with FNs are 9 and 2, respectively.

$$Accuracy = \frac{346 + 0 + 0 + 0 + 0}{365} = 0.9479$$

Obtained from the accuracy calculation, which is 0,9479. This accuracy shows that the K-Nearest Neighbor (KNN) model with k=6 has a very good performance in classifying rainfall data into five different classes. Although this accuracy is very high, it is also important to look at other evaluation metrics, such as precision and recall, especially in cases where there is data imbalance between classes, to ensure a more comprehensive model performance.

$$Precision_0 = \frac{346}{(346 + 19)} = 0.9479$$

$$Precision_1 = \frac{0}{(0 + 0)} = nan$$

$$Precision_2 = \frac{0}{(0 + 0)} = nan$$

$$Precision_3 = \frac{0}{(0 + 0)} = nan$$

Obtained from the precision calculation for class 0 is 0,947 or around 94,7 %. This means that of all the model predictions that state a sample as Class 0, around 94,7 % of them are actually Class 0. The high precision value indicates that the model has a good ability to identify samples that are truly included in Class 0, with a low number of false positives. In other words, the model rarely incorrectly predicts samples that are not Class 0 as Class 0. The evaluation results for Class 1, Class 2, and Class 3 show precision values that cannot be calculated (nan or not a number). These nan values usually appear because there are no positive predictions made by the model for these classes. In other words, the model never classifies any sample as Class 1, Class 2, or Class 3. This situation indicates a serious problem in the model's ability to detect and classify samples into these classes.

$$Recall_0 = \frac{346}{(346 + 0)} = 1.0$$

$$Recall_1 = \frac{0}{(0 + 8)} = 0$$

$$Recall_2 = \frac{0}{(0 + 9)} = 0$$

$$Recall_3 = \frac{0}{(0 + 2)} = 0$$

Obtained from the calculation of recall for Class 0 is 1,0 or 100 %, which means that this model successfully identified all samples that actually came from Class 0 without missing any. With this perfect recall, it can be

concluded that the model has a very good ability to detect all Class 0 samples. Recall for Class 1, Class 2, and Class 3 is 0. This shows that the model is unable to detect a single sample that actually comes from these classes. All samples that should have been classified as Class 1, Class 2, or Class 3 were instead classified as Class 0. The model's inability to detect samples from Class 1, Class 2, and Class 3 indicates a serious problem in the model's ability to recognize the unique characteristics of samples in those classes. The AUC – ROC curve K-Nearest Neighbor results are shown in figure 7.

Based on figure 7 the K-Nearest Neighbor (KNN) model of 0,668 indicates that the KNN classifier has moderate ability to distinguish between classes. This indicates that the model performs better than random guessing but still has room for improvement in the Surabaya City rainfall dataset. The ROC curve and AUC scores provide valuable insights into the performance of the KNN classifier. Although this classifier performs quite well, there is a clear indication that further improvements are possible to achieve better discrimination between classes.

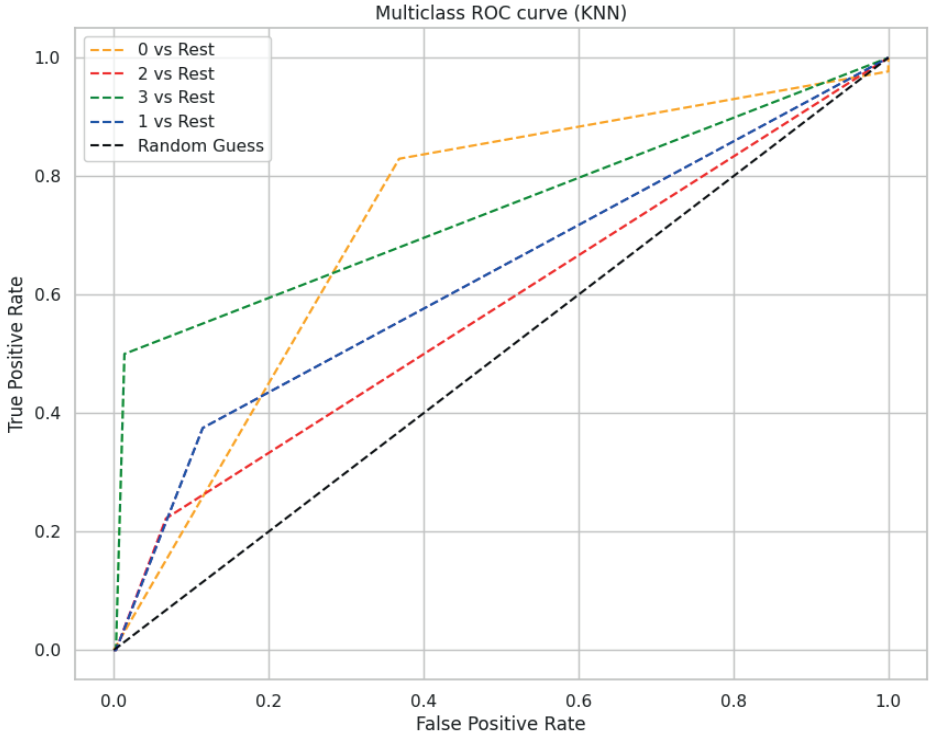


Figure 7. AUC – ROC Curve K-Nearest Neighbor

Random Forest Method

The next classification method used in this research is Random Forest. Result of confusion matrix for Random Forest method is shown in table 9.

Table 9. Confusion Matrix of Random Forest Method

Actual	Prediction			
	0	1	2	3
0	346	0	0	0
1	8	0	0	0
2	9	0	0	0
3	2	0	0	0

The table 9 above shows four different classes of rainfall, labeled as Actual 0, Actual 1, Actual 2, and Actual 3. For Class 0, the model successfully identified 346 samples out of a total of 365 samples that were truly included in Class 0. This is indicated by the True Positive (TP) value of 346 and False Positive (FP) of 0, indicating that there were no incorrect predictions for this class. However, there are 19 samples from other Classes (8 from Class 1, 9 from Class 2, and 2 from Class 3) that are misclassified as Class 0, indicating a number of False Negatives (FN) for these classes. In contrast, the model performance for Class 1, Class 2, and Class 3 is very poor. For Class 1, none of the samples were correctly classified, with a TP of 0 and an FN of 8.

https://doi.org/10.56294/dm20251075

$$Accuracy = \frac{346 + 0 + 0 + 0 + 0}{365} = 0.9479$$

All 8 samples that should have been classified as Class 1 were instead classified as Class 0, resulting in an FP is 8 for Class 0. A similar situation occurred in Class 2 and Class 3, where the TP for both classes was also 0. All 9 samples from Class 2 and 2 samples from Class 3 were incorrectly classified as Class 0, with FNs are 9 and 2, respectively.

The accuracy calculation produces a value of 0,9479. This accuracy shows that the Random Forest method has a very good performance in classifying rainfall data into five different classes. Although this accuracy is very high, it is also important to look at other evaluation metrics, such as precision and recall, especially in cases where there is data imbalance between classes, to ensure a more comprehensive model performance.

$$Precision_0 = \frac{346}{(346 + 19)} = 0.9479$$

$$Precision_1 = \frac{0}{(0 + 0)} = nan$$

$$Precision_2 = \frac{0}{(0 + 0)} = nan$$

$$Precision_3 = \frac{0}{(0 + 0)} = nan$$

Obtained from the calculation of precision for class 0 is 0,947 or around 94,7 %. This means that of all the model predictions that state a sample as Class 0, around 94,7 % of them are actually Class 0. The high precision value indicates that the model has a good ability to identify samples that are truly included in Class 0, with a low number of false positives. In other words, the model rarely incorrectly predicts samples that are not Class 0 as Class 0. The evaluation results for Class 1, Class 2, and Class 3 show precision values that cannot be calculated (nan or not a number). These nan values usually appear because there are no positive predictions made by the model for these classes. In other words, the model never classifies any sample as Class 1, Class 2, or Class 3. This situation indicates a serious problem in the model's ability to detect and classify samples into these classes.

$$Recall_0 = \frac{346}{(346 + 0)} = 1.0$$

$$Recall_1 = \frac{0}{(0 + 8)} = 0$$

$$Recall_2 = \frac{0}{(0 + 9)} = 0$$

$$Recall_3 = \frac{0}{(0 + 2)} = 0$$

Obtained from the calculation of Recall for class 0 is 1,0 or 100 %, which means that this model successfully identified all samples that actually came from Class 0 without missing any. With this perfect recall, it can be concluded that the model has a very good ability to detect all Class 0 samples. Recall for Class 1, Class 2, and Class 3 is 0. This shows that the model is unable to detect a single sample that actually comes from these classes. All samples that should have been classified as Class 1, Class 2, or Class 3 were instead classified as Class 0. The model's inability to detect samples from Class 1, Class 2, and Class 3 indicates a serious problem in the model's ability to recognize the unique characteristics of samples in those classes.

The AUC – ROC Curve K-Nearest Neighbor results are shown in figure 8. Based on figure 8 for the Random Forest method shows an ROC-AUC score of 0,767 indicating that the Random Forest classifier has a good ability to distinguish between classes. This shows that the model performs quite well and is better than random guessing, with room for some further improvement. The ROC curve and AUC score provide valuable insights into the performance of the Random Forest classifier. The model shows a good ability to distinguish between classes in a multiclass classification task.

https://doi.org/10.56294/dm20251075

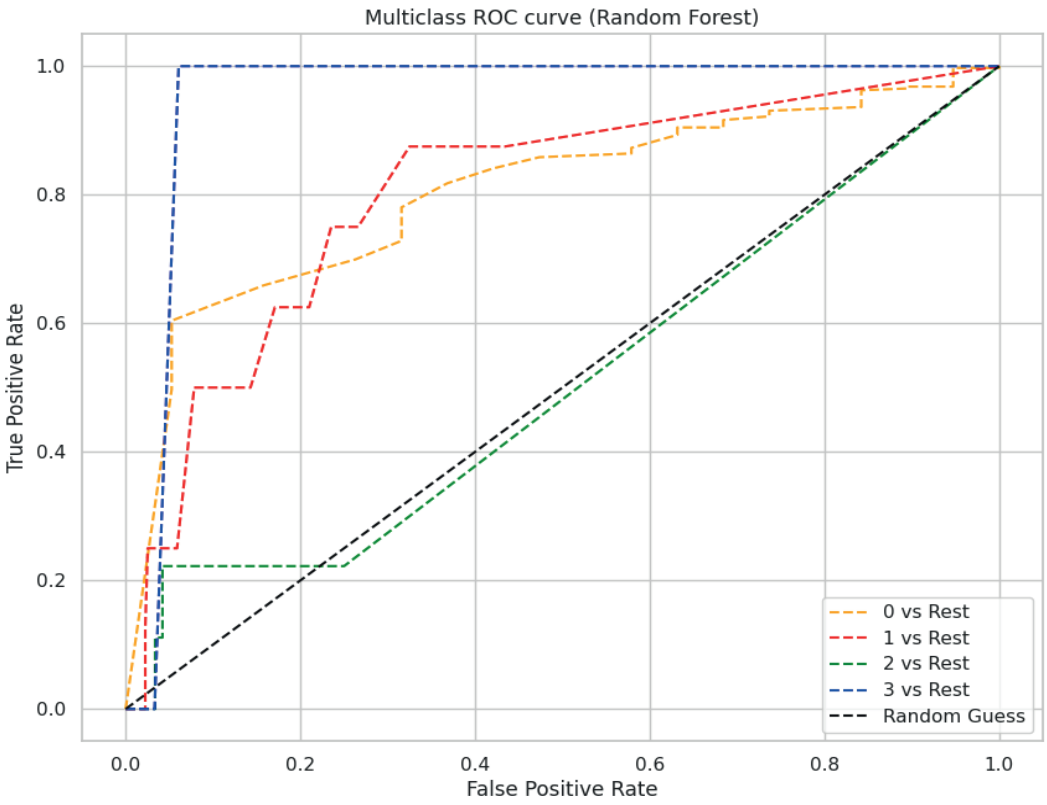


Figure 8. AUC – ROC Curve Random Forest

Loss Function Analysis

In this research, the performance of the three models used was compared based on the loss function value calculated using categorical cross-entropy. The following is an analysis of the results of the comparison of the loss functions of the three models shown in figure 9.

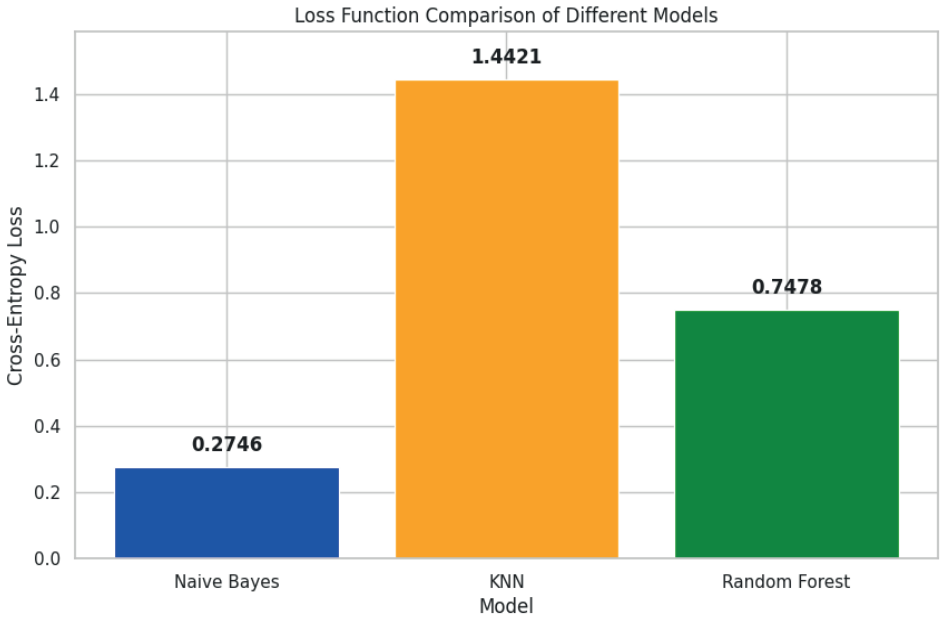


Figure 9. Loss Function Before Tuning

Based on figure 9 for the Naïve Bayes method has a cross-entropy loss value of 0,2746, which is the lowest loss value among the three models. This shows that the prediction of the Naïve Bayes method is closest to the actual probability distribution of the data, so this model is the most accurate in classification compared to KNN and Random Forest. The K-Nearest Neighbor (KNN) model has the highest cross-entropy loss value, which is

https://doi.org/10.56294/dm20251075

1,4421, which shows that this model is less effective in predicting classes compared to other models. The high loss value in KNN can be caused by the k parameter which may not be optimal or the characteristics of the data that are not suitable for the KNN method. Meanwhile, the Random Forest method has a cross-entropy loss value of 0,7478, which shows better performance than KNN but still inferior to Naïve Bayes.

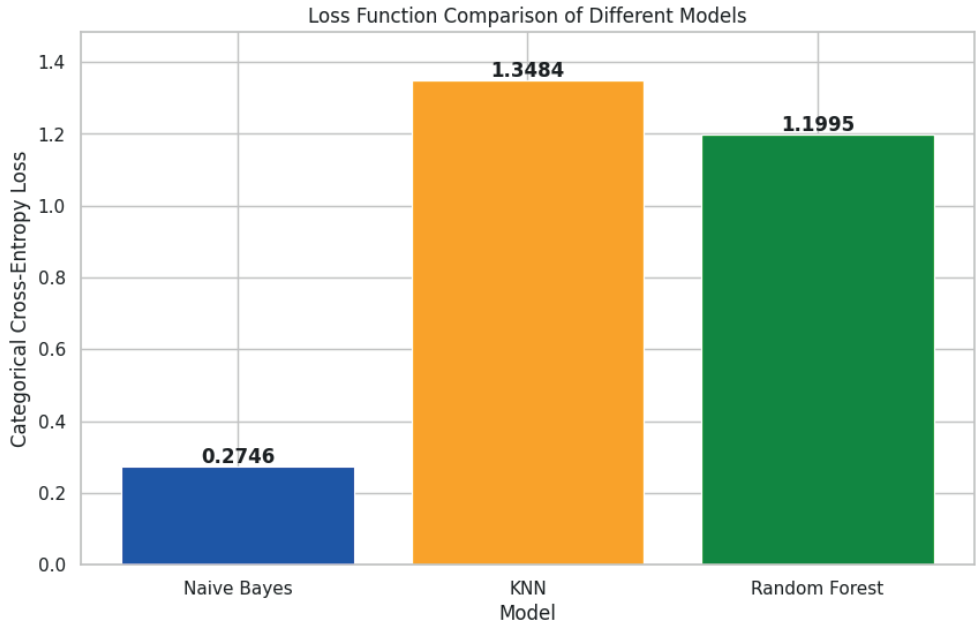


Figure 10. Loss Function After Tuning

The loss function after tuning of the three models is shown in figure 10. Based on figure 10, the categorical cross-entropy loss value for Random Forest is 1,1995, which is higher than the first graph. This may be due to differences in preprocessing or data partition. The loss value for Naive Bayes remains consistently low at 0,2746, confirming that this model is the most reliable in classification based on the dataset used. The K-Nearest Neighbor (KNN) model shows a categorical cross-entropy loss value of 1,3484, which remains high but slightly lower than the first graph. This shows the poor performance consistency of KNN for this dataset.

From the two graphs above, it can be concluded that the Naïve Bayes method is the most effective in classifying data with the lowest categorical cross-entropy loss value. The Random Forest method has quite good performance but is still below Naïve Bayes. The K-Nearest Neighbor (KNN) model shows the worst performance with the highest loss value, indicating that this model is less suitable for the dataset used in this research.

Important Feature Analysis

Feature importance is used to measure how important each feature is to the performance of a classification model. More precisely, feature importance refers to the size of an individual feature's contribution to the performance of a particular classifier. The following are the results of feature importance shown in figure 11. Based on figure 11 shows a visualization of feature importance, which illustrates how important each feature is to the performance of the classification model. Sunshine duration has the highest feature importance score among all features, around 0,38. This shows that sunshine duration is the most influential factor in the rainfall prediction model. This means that changes in sunshine duration make the largest contribution to the model's prediction results. This large influence shows that sunsine duration is a very important variable in determining rainfall in the Surabaya city.

Humidity is the second most important feature with a score of around 0,23. This shows that humidity also has a significant influence on rainfall prediction, although not as large as sunshine duration. Variations in humidity make an important contribution to determining the model's prediction results. Temperature has a feature importance score of around 0,23, making it the third most important feature in the model. Although it has a slightly lower influence than sunshine duration, temperature still plays an important role in rainfall prediction. Temperature variations also contribute significantly to the model prediction results. Wind speed has the lowest feature importance score, around 0,19. This indicates that wind speed is the least influential feature in the rainfall prediction model. However, wind speed still provides a relevant contribution, but not as large as the other three features.

Based on the results of the feature importance analysis, it can be concluded that sunshine duration is the most influential factor in rainfall prediction in Surabaya City, followed by humidity, temperature, and wind

https://doi.org/10.56294/dm20251075

speed. These results can provide valuable insights for researchers and practitioners in understanding the main factors that influence rainfall and can be used to further optimize the prediction model.

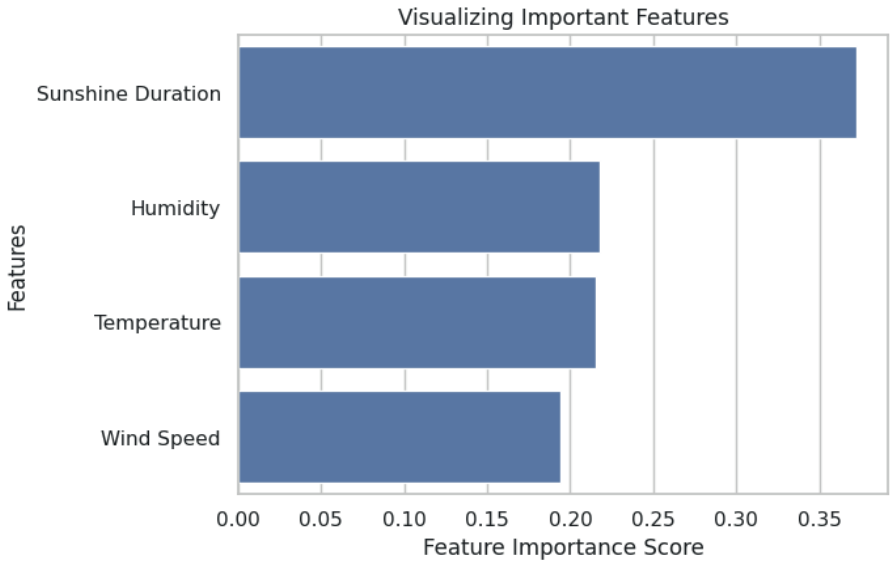


Figure 11. Feature Importance

Model Selection

After performing classification using the Naïve Bayes, K-Nearest Neighbor, and Random Forest, the performance of the three models was obtained based on several evaluation metrics, namely accuracy, precision, recall, and AUC-ROC. Here are the results of the comparison of the three models in table 10.

Table 10. Model Comparison

Clasification Model	Naïve Bayes	K-Nearest Neighbor (k=6)	Random Forest
Accuracy	0,9370	0,9479	0,9479
Precision Class 0	0,9479	0,9470	0,9470
Precision Class 1	nan	nan	nan
Precision Class 2	0	nan	nan
Precision Class 3	nan	nan	nan
Recall Class 0	0,9880	1	1
Recall Class 1	0	0	0
Recall Class 2	0	0	0
Recall Class 3	0	0	0
AUC-ROC	0,8040	0,6680	0,7660

The comparison reveals that while the models exhibit high overall accuracy, their predictive capacity is concentrated on the majority class, indicating a significant class imbalance issue. This undermines their effectiveness in multi-class rainfall classification. The Naïve Bayes model demonstrates superior discriminatory power, as reflected in its performance across evaluation metrics and loss function analysis, suggesting better generalization despite its lower accuracy. The findings imply that conventional classifiers may not sufficiently capture the complexity of rainfall patterns in imbalanced datasets, emphasizing the need for enhanced modeling strategies—such as resampling techniques or cost-sensitive learning—to improve minority class detection and overall model reliability.

BIBLIOGRAPHIC REFERENCES

1. Bluestein HB, Carr FH, Goodman SJ. Atmospheric observations of weather and climate. Atmosphere-Ocean. Taylor & Francis; 2022;60(3 4):149 87.

2. Fowler HJ, Ali H, Allan RP, Ban N, Barbero R, Berg P, et al. Towards advancing scientific knowledge of climate change impacts on short-duration rainfall extremes. Philosophical Transactions of the Royal Society A. The Royal Society Publishing; 2021;379(2195):20190542.

https://doi.org/10.56294/dm20251075

3. Firdiyan N, Muntini MS. The Effect Of Rainfall On The Detection Of Standing Water On Runway. Dans: Journal of Physics: Conference Series. IOP Publishing; 2021. p. 012034.

4. Xu T, Liang F. Machine learning for hydrologic sciences: An introductory overview. Wiley Interdisciplinary Reviews: Water. Wiley Online Library; 2021;8(5):e1533.

5. Mistry MN, Schneider R, Masselot P, Royé D, Armstrong B, Kyselý J, et al. Comparison of weather station and climate reanalysis data for modelling temperature-related mortality. Sci Rep. Nature Publishing Group UK London; 2022;12(1):5178.

6. Huang M, Lin R, Huang S, Xing T. A novel approach for precipitation forecast via improved K-nearest neighbor algorithm. Advanced Engineering Informatics. Elsevier; 2017;33:89 95.

7. Pandey R, Upadhya M, Singh M. Rainfall Prediction Using Logistic Regression and Random Forest Algorithm. Dans: 2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT). IEEE; 2024. p. 663 8.

8. Shaji A, Amritha AR, Rajalakshmi VR. Weather prediction using machine learning algorithms. Dans: 2022 International Conference on Intelligent Controller and Computing for Smart Power (ICICCSP). IEEE; 2022. p. 1 5.

9. Chen PY, Tung CP, Tsao JH, Chen CJ. Assessing future rainfall intensity–duration–frequency characteristics across Taiwan using the k-nearest neighbor method. Water (Basel). MDPI; 2021;13(11):1521.

10. Berrar D. Bayes' theorem and naive Bayes classifier. Elsevier (In Press); 2025;

11. Farid DM, Zhang L, Rahman CM, Hossain MA, Strachan R. Hybrid decision tree and naïve Bayes classifiers for multi-class classification tasks. Expert Syst Appl. Elsevier; 2014;41(4):1937 46.

12. Suyal M, Goyal P. A review on analysis of k-nearest neighbor classification machine learning algorithms based on supervised learning. International Journal of Engineering Trends and Technology. Seventh Sense Research Group; 2022;70(7):43 8.

13. Dhanabal S, Chandramathi S. A review of various k-nearest neighbor query processing techniques. Int J Comput Appl. Citeseer; 2011;31(7):14 22.

14. Deng Z, Zhu X, Cheng D, Zong M, Zhang S. Efficient kNN classification algorithm for big data. Neurocomputing. Elsevier; 2016;195:143 8.

15. Salman HA, Kalakech A, Steiti A. Random forest algorithm overview. Babylonian Journal of Machine Learning. 2024;2024:69 79.

16. Krishnan R, Sivakumar G, Bhattacharya P. Extracting decision trees from trained neural networks. Pattern Recognit. Elsevier; 1999;32(12).

17. Pham TA, Tran VQ. Developing random forest hybridization models for estimating the axial bearing capacity of pile. PLoS One. Public Library of Science San Francisco, CA USA; 2022;17(3):e0265747.

18. Fahmy Amin M. Confusion matrix in three-class classification problems: A step-by-step tutorial. Journal of Engineering Research. Tanta University, Faculty of Engineering; 2023;7(1):0.

19. Espino-Salinas CH, Galván-Tejada CE, Luna-García H, Gamboa-Rosales H, Celaya-Padilla JM, Zanella-Calzada LA, et al. Two-dimensional convolutional neural network for depression episodes detection in real time using motor activity time series of depresjon dataset. Bioengineering. MDPI; 2022;9(9):458.

20. Wardhani SG, Kurniawati A. Implementation of K-Nearest Neighbor Algorithm for Creditworthiness Analysis Using Methods Cross-Industry Standard Process for Data Mining (CRISP-DM). Science (1979). 2025;10(1):152 7.

21. Gowdra N, Sinha R, MacDonell S, Yan W. Maximum Categorical Cross Entropy (MCCE): A noise-robust alternative loss function to mitigate racial bias in Convolutional Neural Networks (CNNs) by reducing overfitting. 2021.

https://doi.org/10.56294/dm20251075

FINANCING

The authors did not receive financing for the development of this research.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest.

AUTHORSHIP CONTRIBUTION

Conceptualization: Arip Ramadan, Dwi Rantini, Muhammad Axel Syahputra.

Data curation: Muhammad Axel Syahputra, Dwi Rantini, Muhammad Noor Fakhruzzaman.

Formal analysis: Arip Ramadan, Dwi Rantini, Ratih Ardiati Ningrum, Muhammad Noor Fakhruzzaman, Aziz Fajar, Muhammad Axel Syahputra, Najma Attaqiya Alya, Muhammad Mahdy Yandra.

Research: Muhammad Axel Syahputra, Najma Attaqiya Alya, Arip Ramadan, Dwi Rantini.

Methodology: Muhammad Axel Syahputra, Dwi Rantini, Ratih Ardiati Ningrum.

Project management: Dwi Rantini, Alhassan Sesay, Maryamah.

Resources: Arip Ramadan, Alhassan Sesay, Maryamah.

Software: Muhammad Axel Syahputra, Najma Attaqiya Alya, Muhammad Mahdy Yandra.

Supervision: Dwi Rantini, Ratih Ardiati Ningrum, Muhammad Noor Fakhruzzaman, Aziz Fajar.

Validation: Dwi Rantini, Ratih Ardiati Ningrum, Muhammad Noor Fakhruzzaman, Aziz Fajar.

Display: Muhammad Axel Syahputra, Najma Attaqiya Alya, Muhammad Mahdy Yandra.

Drafting - original draft: Arip Ramadan, Dwi Rantini.

Writing - proofreading and editing: Arip Ramadan, Dwi Rantini, Muhammad Axel Syahputra, Najma Attaqiya Alya, Muhammad Mahdy Yandra.