Comparison of K-Nearest Neighbors (KNN) and Decision Tree with Binary Particle Swarm Optimization (BPSO) in Predicting Employee Performance

Isti Amelia Isnaeni, Sandra Indriani, Muhammad Rizaq Nuriz Zaman, Andi Nugroho

Abstract — Human resource management has a significant role in influencing organizational performance. As a company that seeks to increase objectivity and decision making that relies on data, this research focuses on exploring the application of Machine Learning algorithms as a transformative approach to employee performance evaluation at PT XYZ. When the current process still uses conventional methods such as manual scoring systems and subjective managerial assessments, bias and lack of transparency may occur. This research discusses technical aspects in implementing Machine Learning algorithms using K-Nearest Neighbors (KNN) and Decision Trees . Before carrying out the classification process, the optimization stage is carried out using the Binary Particle Swarm Optimization (BPSO) algorithm to determine the optimal hyperparameter values. The results of the classification process are then evaluated using the Confusion Matrix . The dataset used in this research has 5 classes so it requires a Multi-Class classification (MCC) approach. This research describes the process of determining the final results of evaluation metrics using the MCC Confusion Matrix with 5 classes. The final results show that the highest F1-Score was obtained from the KNN Algorithm at 84.36% and the Decision Tree Algorithm at 79.8%. Thus, this research contributes to detailing the effectiveness of applying Machine Learning algorithms for evaluating employee performance in the organizational context of PT XYZ.

Keywords — Decision Tree, K-Nearest Neighbors (KNN), Binary Particle Swarm Optimization, Classification, Performance Review, Machine Learning, Multi-Class Classification.

I. INTRODUCTION

In its influence on the performance of an organization, it has been proven that human resource management plays a significant role [1]. In the performance evaluation of an organization, including at PT XYZ in the property sector, employee performance assessments are conducted every semester to identify career goals, areas for improvement, and to support decisions on promotions and salary increases. Although the evaluations are done manually without system support, this conventional method presents challenges in integrating qualitative, quantitative, and subjective data for an accurate representation of performance.

Manual scoring and subjective managerial assessments are prone to bias and subjectivity, impacting the fairness and integrity of evaluations. Unfair assessments can also diminish employee motivation, satisfaction, and productivity. Therefore, this research aims to employ a supervised learning approach, specifically the K-Nearest Neighbors (KNN) and Decision Tree algorithms, to predict the outcomes of employee performance reviews at PT XYZ. By analyzing various data sources, these Machine Learning algorithms are expected to identify factors contributing to high performance or areas in need of improvement. The hope is that the results of this study will provide personalized feedback and training recommendations to employees based on their performance data.RELATED TO WORK

A. Machine Learning

Machine Learning is a part of Artificial Intelligence (AI) that automates the process of creating statistical and analytical models that enable systems to learn from information, recognize patterns, and make predictions with human involvement [2]. Apart from that, in Machine Learning, the process of adapting and improving performance over time through experience is also an important and inseparable aspect.

Machine Learning includes several approaches such as Supervised Learning, Unsupervised Learning, and Reinforcement Learning. Approaches in Supervised Learning and Unsupervised Learning are used to explore hidden information and relationships between data, which will ultimately help future decision makers to take appropriate action [3]. The difference between these two main classes lies in the presence of labels in the subset of training data used [4]. In Reinforcement Learning, an agent is trained to interact with the environment and learn through experimentation, by optimizing its actions based on positive or negative feedback. Mach i ne Learning can be useful for various industries. In the finance industry sector, Machine Learning applications can solve problems such as portfolio analysis [5].

B. Decision Trees

Decision trees are a machine learning method that is very popular and often used [6]. Decision Trees are often used because of their simple nature, easy to understand approach, and good efficiency [7]. Decision trees break down data into smaller subsets based on relevant attributes. At each node in the tree, attributes are selected based on the attribute selection method that produces the most effective separation in classification.

The advantages of Decision Trees include their ability to be suitable for solving regression and classification problems, reliable interpretation capabilities, ability to handle categorical and quantitative values, can fill in missing values in attributes with the most likely values, and high performance due to the efficiency of the tree traversal algorithm [8].

C. K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) represents a straightforward yet highly effective supervised learning classification approach, acknowledged as one of the top ten classical Machine Learning algorithms in the 21st century [9]. This algorithm, tailored for pattern classification, operates based on the nearest neighbor rule [10] and is categorized as a non-parametric method due to its lack of assumptions about the underlying data distribution [11]. A notable advantage of KNN lies in its non-parametric nature, eliminating the need for a training process. Specifically, it can classify queries without necessitating prior knowledge of the statistical properties of the training examples [12].

A key aspect of KNN involves determining the distance or similarity between training and test data. Various methods, such as Minkowsky, Manhattan, Chebychev, Euclidean, Cosine similarity, Kendall's Rank Correlation, and Hamming Distance, have been employed to measure distance or similarity in KNN [13]. The KNN classification algorithm opts for Hamming Distance, which is defined as follows:

$$d_H(x, y) \ = \ w_H(x - y) \tag{1}$$

With:

d_H	: Hamming distance
w_H	: Hamming weight
$d_H(x, y)$	: the number of places of two vectors
x	: integer 0
y	: integer 1

D. MATLAB

Matlab, derived from "Matrix Laboratory," stands as a widely utilized numerical computing environment and programming language within the fields of science, engineering, and computing, developed by MathWorks. Serving as software for image processing and fragmentation, Matlab incorporates a variety of pertinent algorithms and functions [14]. Notably, it features an integrated development environment (IDE) enabling users to create intuitive graphical user interfaces (GUI) [15]. Moreover, it offers interactive simulations and a valuable toolkit for swift development of predictions using Machine Learning algorithms [16]. In the realms of data analysis and Machine Learning, datasets can be characterized by numerous attributes, not all of which contribute relevant or crucial information. Therefore, the execution of Feature Selection becomes crucial, enabling the reduction of dataset dimensions while preserving vital information, thereby enhancing computational efficiency and mitigating overfitting issues. Feature Selection not only eliminates irrelevant features but also eliminates redundant ones, resulting in a more effective feature set for Machine Learning [17].

E. Binary Particle Swarm Optimization (BPSO)

Kennedy and Eberhart introduced BPSO to enable the operation of the PSO algorithm in a binary search space [18]. The BPSO Algorithm for Attribute Selection addresses the attribute selection problem, which involves searching for Boolean assignments to propositional variables to maximize the number of simultaneously satisfied clauses [19]. This approach is grounded in [20]:

$$V_i^{k+1} = \omega \ \times \ V_i^k + c_1 r_1 \left(P_{\text{best i}} - X_i^k\right) + c_2 r_2 \left(G_{\text{best i}} - X_i^k\right) \tag{2}$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \tag{3}$$

With :

V_i^{k+1}	= New particle position
V_i^k	= Actual particle position
X_i^k	= Old Particle Point
ω	= Inertial weight
c_1 and c_2	= Cognitive and social parameters
r_1 and r_2	= Random value between 0 and 1
$G_{\text{best i}}$	= The best global particle position of the leader swarm
$P_{\text{best i}}$	= The local best position of the iparticle

F. Cunfusion Matrix

Confusion Matrix serves as a performance assessment tool for structured classification problems in machine learning, especially when the output consists of 2 or more classes [21]. Widely used in classification scenarios, the Confusion Matrix proves valuable for evaluating the efficiency of model performance. True-positives and true-negatives within the matrix denote instances correctly classified by the model, reflecting actual quantities [22].

II. METHODOLOGY

The data utilized in this study constitutes primary data acquired directly from the researcher's employing company. Primary data, being raw and firsthand, is deemed reliable, considered authentic and objective in research. In line with this explanation, the researcher employed data stemming from the outcomes of a performance review conducted at PT XYZ towards the close of 2022. The dataset encompasses 191 rows and 8 columns. In this research, a classification algorithm will be employed in the ongoing project to aid in decision-making based on the analysis of Employee Performance conducted, along with the identification of key skill sets.

The ensuing flow diagram illustrates the research process:

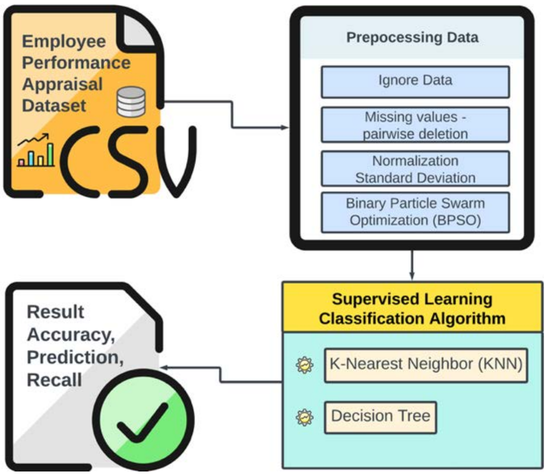

Employee Performance Appraisal Dataset
CSV
Prepocessing Data
Ignore Data
Missing values - pairwise deletion
Normalization Standard Deviation
Binary Particle Swarm Optimizition (BPSO)
Supervised Learning Classification Algorithm
K-Nearest Neighbor (KNN)
Decision Tree
Result Accuracy, Prediction, Recall

Figure 1 . Research Flow Chart

In this study, primary data from PT XYZ, titled the "Employee Performance Appraisal Dataset" in Microsoft Excel format, was employed by the researchers. The data underwent a preprocessing phase to enhance its quality and usability, ensuring a more effective mining process. This preprocessing involved techniques like data omission, handling Missing Values, and normalizing the data to standardize attribute scales. The feature selection phase utilized the Binary Particle Swarm Optimization (BPSO) algorithm, aiming to choose pertinent features before classification to optimize model performance and mitigate overfitting. Classification was executed using the K-Nearest Neighbors (KNN) algorithm and Decision Tree to identify the optimal model for predicting performance review results. A comparative analysis of these algorithms yielded new insights into the model's performance under diverse conditions, including addressing uneven class distributions. The metric evaluation stage, employing the Confusion Matrix, furnished crucial information such as accuracy, precision, recall, and F1-Score, facilitating the assessment of model performance during the classification phase.

A. Data Cleaning

1. Describing Dataset

The dataset used is in Microsoft Excel format which contains information from the results of employee performance reviews at PT XYZ in 2022.

Table 1 . Dataset Attribute Information

No	Atribut	Number	Information
1	NIK	1	Nomor Induk Karyawan
2	Employee Group	2	Status Perjanjian Kerja Karyawan
3	Level Jabatan	3	Level atau Posisi Hierarki Karyawan di Perusahaan
4	Department	4	Status Subdivisi Karyawan
5	Direktorat	5	Status Divisi Karyawan
6	Learning Agility	6	Komponen Penilaian Performance Review berdasarkan Kemampuan suatu Karyawan untuk belajar dan beradaptasi dengan cepat
7	Kompetensi	7	Komponen Penilaian Performance Review berdasarkan sikap, pengetahuan, skill dan keterampilan sesuai dengan standar perusahaan
8	Talent 2021	8	Hasil Performance Review 2021 yang menjadi komponen Penilaian Performance Review 2022
9	Talent 2022	9	Hasil Performance Review 2022

2. Missing Value

The initial phase in preprocessing involves addressing Missing Values. This step proves beneficial in mitigating bias, preserving data integrity, and enhancing model performance. The approach employed to manage Missing Values is Deletion, achieved by removing rows or samples containing features with null values. This method is applied when the volume of deleted data is minimal and does not substantially influence the overall data representation. Furthermore, given that the data utilized is primary, employing alternate methods like filling empty values by computing the average or median could impact the data's interpretation and integrity.

B. Data Transformation

1. Data Encoding

In the process of data transformation, the initial task is to convert categorical data into numerical format. The chosen method for this transformation is Label Encoding. This technique assigns a unique integer number to each distinct category value. Once each category value is assigned, the values within the dataset are substituted with the corresponding numeric labels. The transformed variable retains the original variable's structure, maintaining the same number of data points in the same sequence. The only alteration is that the data format has transitioned into numeric form.

The results of the Data Encoding process also eliminate features with a cardinality that exceeds 10, such as NIK. Because features with this cardinality will make the optimization and classification process time consuming. In addition, the classification model will consider NIK to be a numerical value which will reduce the model performance.

Table 2 . Dataset after the Data Encoding process

No.	NIK	Employee Group	Level Jabatan	Department	Direktorat	Learning Agility	Kompetensi	Talent 2021	Talent 2022
1	10155	0	0	0	0	0	0	3	4
2	10014	0	0	5	1	0	0	3	3
3	10151	0	0	6	0	0	0	3	4
4	10111	0	0	7	2	0	0	3	4
5	10138	0	0	2	2	0	0	4	4
…	…	…	…	…	…	…	…	…	…
163	PK055	1	4	1	0	1	1	2	2
164	PK109	1	4	1	0	0	0	2	3
165	PK137	1	4	1	0	0	0	2	3
166	PK087	1	4	6	0	1	1	3	3
167	PK025	1	4	6	0	0	1	3	3

2. Data Normalization (Data Normalization)

In contrast to Data Enoding, data normalization aims to change the scale of numerical features into a scale that has a consistent range. Data normalization ensures that each different scale feature contributes equally.

The data normalization method used is Min-Max . This method will perform a linear transformation on the original data to a range from 0.0 to 0.1.

Min-Max Formula:

$$V' = \frac{V - \min_A}{\max_A - \min_A}\left(new_{\max_A} - new\_min_A\right) + new\_min_A \quad (4)$$

Table 3. Dataset after Data Normalization

No.	Employee Group	Level Jabatan	Direktorat	Department	Learning Agility	Kompetensi	Talent 2021	Talent 2022
1	0	0	0	0	0	0	0.2	0.2666667
2	0	0	0.066667	0.33333333	0	0	0.2	0.2
3	0	0	0	0.4	0	0	0.2	0.2666667
4	0	0	0.133333	0.46666667	0	0	0.2	0.2666667
5	0	0	0.133333	0.13333333	0	0	0.2666667	0.2666667
…	…	…	…	…	…	…	…	…
163	0.066666667	0.266666667	0	0.06666667	0.066666667	0.06666667	0.1333333	0.1333333
164	0.066666667	0.266666667	0	0.06666667	0	0	0.1333333	0.2
165	0.066666667	0.266666667	0	0.06666667	0	0	0.1333333	0.2
166	0.066666667	0.266666667	0	0.4	0.066666667	0.06666667	0.2	0.2
167	0.066666667	0.266666667	0	0.4	0	0.06666667	0.2	0.2

C. Optimization with Binary Particle Swarm Optimization (BPSO)

Binary Particle Swarm Optimization (BPSO) is a modified result of the Particle Swarm Optimization (PSO) algorithm whose aim is to find an optimal feature subset that maximizes (or minimizes) a certain objective function at the feature selection stage. BPSO algorithm is considered effective in carrying out feature selection because it uses binary number operations and evaluates each particle based on an objective function that measures the quality of each solution.

The optimization stage using the BPSO algorithm was carried out using Matlab R_2022a. This algorithm continues to iterate until a stopping criterion is met, such as a maximum number of iterations or a satisfactory solution is reached. The following are the final results of selecting features with the best accuracy:

Table 4. After optimization with BPSO

No	Department	Kompetensi	Talent 2021
1	0	0	0,2
2	0,333333333	0	0,2
3	0,4	0	0,2
4	0,466666667	0	0,2
5	0,133333333	0	0,266666667
6	0,066666667	0	0,2
…	…	…	…
162	0,066666667	0,066666667	0,2
163	0,066666667	0,066666667	0,133333333
164	0,066666667	0	0,133333333
165	0,066666667	0	0,133333333
166	0,4	0,066666667	0,2
167	0,4	0,066666667	0,2

D. Model Training

After preprocessing the data, the next step is the classification process using the K-Nearest Neighbors (KNN) algorithm and Decision Tree .

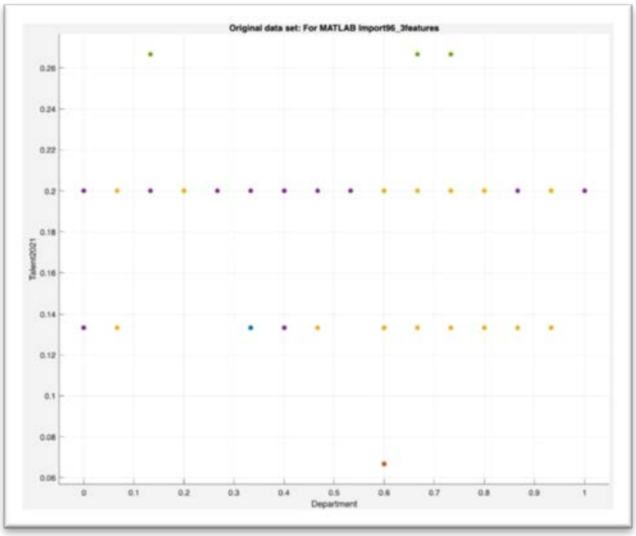


Figure 2 . Scatter Plots

Figure 2. is a scatter plot of the performance review dataset used in this research based on attributes x (department) and y (talent 2021). From the picture in Above you can see that the scatter plot has parallel points and has 5 classes, namely green, purple, orange , blue, red.

E. Evaluation and Tuning

Based on tests carried out with the KNN algorithm using K 1-10. Using Presets : Fine KNN, Medium KNN, Weighted KNN. Using Distance Metrics: Hamming . Using Distance Weight: Equal, Inverse, Squared Inverse . Using Standardize data: FALSE .

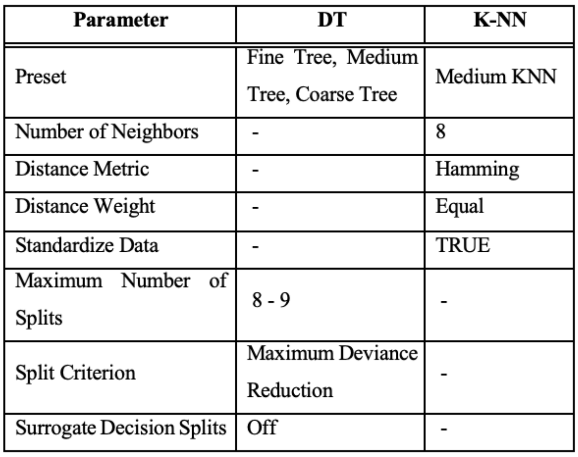
Based on tests carried out with the Decision Tree algorithm using Maximum Number of Splits 1-10. Using Presets: Fine Tree, Medium Tree, Coarse Tree . Using Split Criterion: Gini's Diversity index, Twoing Rule, Maximum Deviance reduction . Using Surrogate Decision Splits: Off .

Table 5. Algorithm Parameter Settings

Parameter	DT	K-NN
Preset	Fine Tree, Medium Tree, Coarse Tree.	Fine KNN, Medium KNN, Weighted KNN
Number of Neighbors	-	1-10
Distance Metric	-	Hamming
Distance Weight	-	Squared Inverse
Standardize Data	-	FALSE/TRUE
Maximum Number of Splits	1-100	-
Split Criterion	Gini's Diversity Index, Twoing Rule , Maximum Deviance Reduction	-
Surrogate Decision Splits	Off	-

The results of the best settings for the KNN and decision tree algorithms are in **Table 6.**

Table 6. Results of the best settings for K-Nearest Neighbors and Decision Tree .



Parameter	DT	K-NN
Preset	Fine Tree, Medium Tree, Coarse Tree	Medium KNN
Number of Neighbors	-	8
Distance Metric	-	Hamming
Distance Weight	-	Equal
Standardize Data	-	TRUE
Maximum Number of Splits	8 - 9	-
Split Criterion	Maximum Deviance Reduction	-
Surrogate Decision Splits	Off	-

The next step is to find out in more detail about the Confusion Matrix calculations from the Model Parameters with the best values in the K-Nearest Neighbors (KNN) and Decision Tree Algorithms .

The dataset used in this research has 5 parameter scores (5 classes) to determine the final score for each employee in the 2022 Performance Review at PT XYZ, which is as follows:

Table 7. 2022 Performance Review Parameter Score at PT XYZ

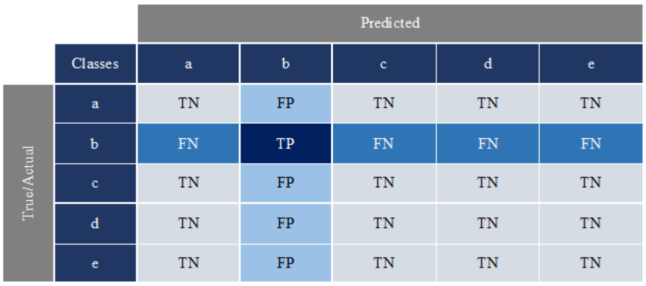
Skor	Indeks Skor	*Feature Selection*	*identifier*
Road Runner	4	0.2666667	e
Promotable	3	0.2	d
Solid Contributor	2	0.133333333	c
Sleeping Tiger	1	0.066666667	b
Unfit	0	0	a

In Machine Learning, the classification involving labels with more than two classes is determined Multi-Class Classification (MCC). In this type of classification, each model predicts the potential occurrence of multiple classes for each sample. In contrast to Binary Classification, the formulas and methodologies utilized for calculating Accuracy, Precision, Recall, and F1-Score exhibit significant differences in their approach. Within MCC, numerous approaches are applicable based on specific conditions or the significance attributed to each class and event within those classes.

One such approach is the Weighted-Average method, where Performance Metrics are individually computed for each class. Subsequently, a weight is assigned to each class based on its frequency. Essentially, this approach operates akin to Macro-Average. However, instead of dividing Precision and Recall based on the number of classes, it offers each class a fair representation proportional to its frequency in the dataset. This proves beneficial when dealing with an imbalanced dataset, yet desiring equal importance for each class.

In scenarios involving an imbalanced dataset, particularly in studies predicting employee performance outcomes, it becomes crucial to accord equal weight to each parameter or class. Hence, the most suitable approach to employ is the Weighted-Average. The subsequent section outlines the implementation of the calculation for each Performance Metric using the Weighted-Average approach:

Table 8. Confusion Matrix Illustration 5 Classes



		Predicted				
	Classes	a	b	c	d	e
True/Actual	a	TN	FP	TN	TN	TN
	b	FN	TP	FN	FN	FN
	c	TN	FP	TN	TN	TN
	d	TN	FP	TN	TN	TN
	e	TN	FP	TN	TN	TN

a. the Accuracy, Precision, Recall and F1-Score performance metrics in each class:

$$\text{accuracy}_c = \frac{(\text{TP}_c + \text{TN}_c)}{(\text{TP}_c + \text{FP}_c + \text{TN}_c + \text{FN}_c)} \quad (5)$$

$$\text{precision}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c} \quad (6)$$

$$\text{Recall}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FN}_c} \quad (7)$$

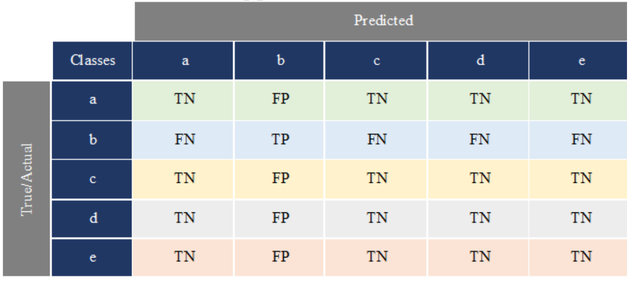
$$\text{F1} - \text{Score}_c = \frac{2 \text{ x precision}_c \text{ x recall}_c}{\text{precision}_c + \text{recall}_c} \quad (8)$$

Where TP_c, TN_c, FP_c, dan $\text{FN}_c$are the predicted numbers for TP, TN, FP, and FN from classification in class C.

b. Calculate each Support amount

Support represents the number of samples or how many numbers actually appear in each class in the dataset. As illustrated in the Confusion Matrix table Below, the amount of support for each class is calculated from the horizontal observation results. For example, class a is calculated based on the number of supports that have been highlighted in orange.

Table 9 . Class Confusion Matrix 5x5 illustration of the number of supports for each class



		Predicted				
	Classes	a	b	c	d	e
True/Actual	a	TN	FP	TN	TN	TN
	b	FN	TP	FN	FN	FN
	c	TN	FP	TN	TN	TN
	d	TN	FP	TN	TN	TN
	e	TN	FP	TN	TN	TN

c. Calculating Performance Metrics globally using the Weighted-Average approach

$$\text{weighted accuracy} = \frac{\sum_{c=0}^{q-1} \text{accuracy}_c S_c}{\sum_{c=0}^{q-1} S_c} \qquad (9)$$

$$\text{weighted precision} = \frac{\sum_{c=0}^{q-1} \text{precision}_c S_c}{\sum_{c=0}^{q-1} S_c} \qquad (10)$$

$$\text{weighted recall} = \frac{\sum_{c=0}^{q-1} \text{recall}_c S_c}{\sum_{c=0}^{q-1} S_c} \qquad (\ 11)$$

$$\text{weighted F1} - \text{Score} = \frac{\sum_{c=0}^{q-1} \text{F1}-\text{Score}_c S_c}{\sum_{c=0}^{q-1} S_c} \text{ [23]} \qquad (12)$$

Where q is the number of classes and S_c is the number of supports from each class.

Classification Report on The best Parameter Model with the K-Nearest Neighbors (KNN) Algorithm.

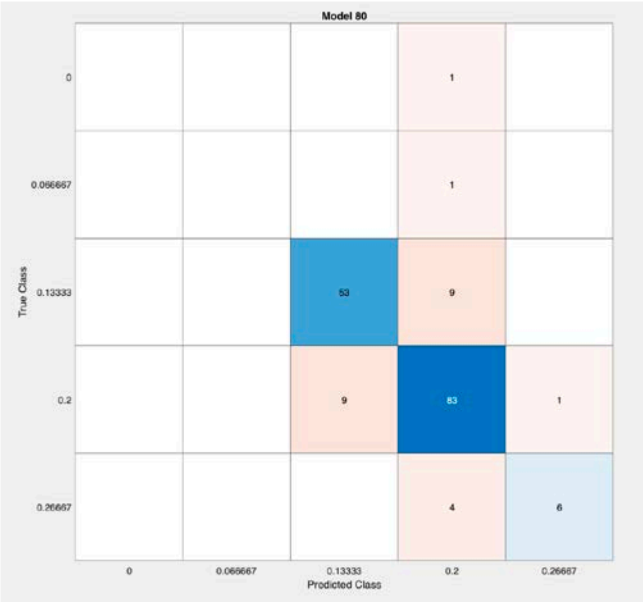
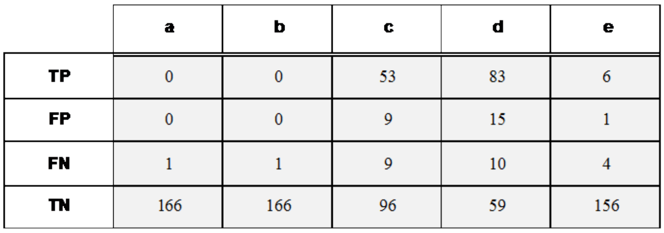


Figure 3 . Confusion Matrix Observation Results of the Best Parameters of the KNN Algorithm

1. Determine and calculate TP, FP, FN, and TN from each class

5x5 Confusion Matrix table illustration in above, then the TP, FP, FN and TN values for each class are obtained as follows:

Table 10 . TP, FP, FN, and TN results in each of the best parameter classes with the KNN algorithm

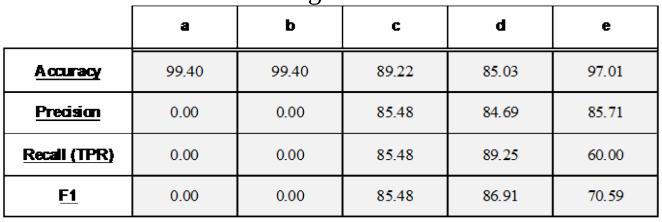


	a	b	c	d	e
TP	0	0	53	83	6
FP	0	0	9	15	1
FN	1	1	9	10	4
TN	166	166	96	59	156

2. the Accuracy, Precision, Recall and F1-Score Performance Metrics of each class

Based on the formula already defined in above, the Accuracy, Precision, Recall and F1-Score performance metrics for each class are obtained as follows:

Table 11. Accuracy, Precision, Recall and F1-Score performance metrics for each class of best parameters with the KNN algorithm



	a	b	c	d	e
Accuracy	99.40	99.40	89.22	85.03	97.01
Precision	0.00	0.00	85.48	84.69	85.71
Recall (TPR)	0.00	0.00	85.48	89.25	60.00
F1	0.00	0.00	85.48	86.91	70.59

3. Determine the amount of support for each class

The following is a calculation of total support based on the number of numbers that actually appear in each class:

- Class Support a

$$S_a = 0 + 0 + 0 + 1 + 0 = 1$$

- Class Support b

$$S_b = 0 + 0 + 0 + 1 + 0 = 1$$

- Class Support c

$$S_c = 0 + 0 + 53 + 9 + 0 = 62$$

- Class Support d

$$S_d = 0 + 0 + 9 + 83 + 1 = 93$$

- Class Support e

$$S_e = 0 + 0 + 0 + 4 + 6 = 10$$

4. Determine Accuracy, Precision, Recall, and F1-Score using the Weighted Average approach

The following are the respective calculations of Accuracy, Precision, Recall, and F1-Score for each class based on the formula that has been defined in on:

$$\text{weighted accuracy} = \frac{\sum_{c=0}^{q-1} \text{accuracy}_c S_c}{\sum_{c=0}^{q-1} S_c}$$

$$= \frac{(99.4\ \times 1)+(99.4 \times 1)+(89.22 \times 62)+(85.03 \times 93)+(97.01 \times 10)}{1+1+62+93+10}$$

$$= \frac{14608.38}{167}$$

$$= \mathbf{87.48}$$

$$\text{weighted precision} = \frac{\sum_{c=0}^{q-1} \text{precision}_c S_c}{\sum_{c=0}^{q-1} S_c}$$

$$= \frac{(0\times1)+(0\times1)+(85.48\times62)+(84.69\times93)+(85.71\times10)}{1+1+62+93+10}$$

$$= \frac{14033.67}{167}$$

$$= \mathbf{84.03}$$

$$\text{weighted recall} = \frac{\sum_{c=0}^{q-1} \text{recall}_c S_c}{\sum_{c=0}^{q-1} S_c}$$

$$= \frac{(0\times1)+(0\times1)+(85.48\times62)+(89.25\times93)+(60\times10)}{1+1+62+93+10}$$

$$= \frac{14200}{167}$$

$$= \mathbf{85.03}$$

$$\text{weighted F1} - \text{Score} = \frac{\sum_{c=0}^{q-1} \text{F1}-\text{Score}_c S_c}{\sum_{c=0}^{q-1} S_c}$$

$$= \frac{(0\times1)+(0\times1)+(85.48\times62)+(86.91\times93)+(70.59\times10)}{1+1+62+93+10}$$

$$= \frac{14088.6}{167}$$

$$= \mathbf{84.36}$$

Classification Report on The best Parameter Model with the Decision Tree Algorithm.

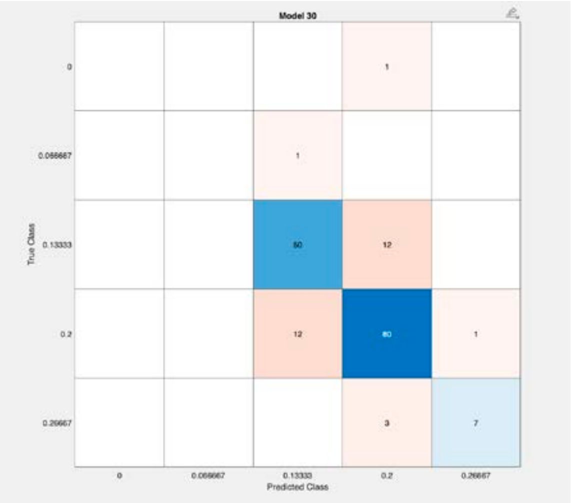


Figure 4 . Confusion Matrix Observation Results of the Best Parameters of the Decision Tree Algorithm

1. Determine and calculate TP, FP, FN, and TN from each class

5x5 Confusion Matrix table illustration in above, then the TP, FP, FN and TN values for each class are obtained as follows:

Table 12 . TP, FP, FN, and TN results for each of the best parameter classes using the Decision Tree algorithm

	a	b	c	d	e
TP	0	0	50	80	7
FP	0	0	13	16	1
FN	1	1	12	13	3
TN	166	166	92	58	156

2. the Accuracy, Precision, Recall and F1-Score Performance Metrics of each class

Based on the formula already defined in above, the Accuracy, Precision, Recall and F1-Score performance metrics for each class are obtained as follows:

Table 13 . Accuracy, Precision, Recall and F1-Score performance metrics for each class of best parameters with the Decision Tree algorithm

	a	b	c	d	e
Accuracy	99.40	99.40	85.03	82.63	97.60
Precision	0.00	0.00	79.37	83.33	87.50
Recall (TPR)	0.00	0.00	80.65	86.02	70.00
F1	0.00	0.00	80.00	84.66	77.78

3. Determine the amount of support for each class

The following is a calculation of total support based on the number of numbers that actually appear in each class:

- Class Support a

$$S_a = 0 + 0 + 0 + 1 + 0 = 1$$

- Class Support b

$$S_b = 0 + 0 + 0 + 1 + 0 = 1$$

- Class Support c

$$S_c = 0 + 0 + 53 + 9 + 0 = 62$$

- Class Support d

$$S_d = 0 + 0 + 9 + 83 + 1 = 93$$

- Class Support e

$$S_e = 0 + 0 + 0 + 4 + 6 = 10$$

4. Determine Accuracy, Precision, Recall, and F1-Score using the Weighted Average approach

The following are the respective calculations of Accuracy, Precision, Recall, and F1-Score for each class based on the formula that has been defined in on:

$$\text{weighted accuracy} = \frac{\sum_{c=0}^{q-1} \text{accuracy}_c S_c}{\sum_{c=0}^{q-1} S_c}$$

$$= \frac{(99.4\ \times1)+(99.4\times1)+(85.03\times62)+(82.63\times93)+(97.6\times10)}{1+1+62+93+10}$$

$$= \frac{14131.7}{167}$$

$$= \mathbf{84.62}$$

$$\text{weighted precision} = \frac{\sum_{c=0}^{q-1} \text{precision}_c S_c}{\sum_{c=0}^{q-1} S_c}$$

$$= \frac{(0\times1)+(0\times1)+(79.37\times62)+(83.33\times93)+(87.5\times10)}{1+1+62+93+10}$$

$$= \frac{13545.63}{167}$$

$$= \mathbf{81.11}$$

$$\text{weighted recall} = \frac{\sum_{c=0}^{q-1} \text{recall}_c S_c}{\sum_{c=0}^{q-1} S_c}$$

$$= \frac{(0\times1)+(0\times1)+(80.65\times62)+(86.02\times93)+(70\times10)}{1+1+62+93+10}$$

$$= \frac{13700}{167}$$

$$= \mathbf{82.04}$$

$$\text{weighted F1} - \text{Score} = \frac{\sum_{c=0}^{q-1} \text{F1}-\text{Score}_c S_c}{\sum_{c=0}^{q-1} S_c}$$

$$= \frac{(0\times1)+(0\times1)+(80\times62)+(84.66\times93)+(77.78\times10)}{1+1+62+93+10}$$

$$= \frac{13610.79}{167}$$

$$= \mathbf{81.50}$$

III. RESULT

The results of the comparison of the KNN and Decision Tree algorithms show that the Accuracy, Precision, Recall and F1-Score values are higher for KNN.

Table 14 . Comparison of KNN and Decision Tree Algorithms with BPSO

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1-Score**
KNN	87.48%	84.03%	85.03%	84.36%
Decision Trees	84.62%	81.11%	82.04%	81.50%

IV. SUMMARY

In conclusion, this study has conducted a thorough exploration of Machine Learning algorithms, specifically concentrating on the comparison between Decision Tree and K-Nearest Neighbors (KNN) within the context of employee performance evaluation at PT XYZ. Our findings exhibit promising outcomes, demonstrating the efficacy of both algorithms in classification tasks, achieving an F1-Score of 79.8% for Decision Tree and 84.36% for K-NN. Moreover, this research entails an optimization phase utilizing Binary Particle Swarm Optimization (BPSO) to ascertain optimal hyperparameters, thereby enhancing the precision of the classification model. The integration of BPSO not only enhances algorithmic performance but also illustrates the adaptability of Swarm Intelligence techniques in augmenting the efficiency of Machine Learning processes.

This investigation navigates the intricacies of real-world scenarios through the utilization of a Confusion Matrix with Multi-Class Classification. The meticulous evaluation provides a nuanced comprehension of the algorithm's performance across diverse classes, offering insights into strengths and potential areas for improvement. The study contributes to the broader discourse on algorithm selection for performance evaluation, offering valuable insights for both practitioners and researchers.

Future research endeavors should delve deeper into hyperparameter refinement and incorporate strategies for handling Imbalanced Data to enhance the comprehensiveness of performance evaluation studies and attain more optimal accuracy outcomes. Overall, this research underscores the significance of informed algorithm selection, optimization techniques, and a clear understanding of Confusion Matrix approaches that consider class distribution for achieving high-quality results in Machine Learning applications.

Suggestions for future research encompass broadening the comparison of Machine Learning algorithms, addressing data imbalances, and conducting detailed investigations into hyperparameter tuning. Considering a wider array of algorithms, including linear models like Support Vector Machines (SVM) or Neural Networks, could provide comprehensive insights into algorithmic performance. Confronting imbalanced datasets, exploring specific techniques such as oversampling, undersampling, or leveraging SMOTE can enhance the performance of classification models. Additionally, detailed research into advanced hyperparameter tuning strategies, such as Bayesian optimization or genetic algorithms, can efficiently yield optimal hyperparameter configurations compared to traditional methods.

REFERENCES

[1] TFA Aziz, S. Sulistiyono, H. Harsiti, A. Setyawan, A. Suhendar, and TA Munandar, "Group decision support system for employee performance evaluation using combined simple additive weighting and Borda," in IOP Conference Series: Materials Science and Engineering , Institute of Physics Publishing, May 2020. doi: 10.1088/1757-899X/830/3/032014

[2] D. Theng and M. Theng, “Machine Learning Algorithms for Predictive Analytics: A Review and New Perspectives Cloud Computing, Cloud Computing Federation, Virtual Machine Management View project Machine Learning Data Science View project Machine Learning Algorithms for Predictive Analytics: A Review and New Perspectives”, doi: 10.37896/HTL26.06/1159.

[3] KM Iraqi, IEEE Computer Society, University of Karachi. Department of Computer Science, and Institute of Electrical and Electronics Engineers, ICISCT'20 : 2nd International Conference on Information Science and Communication Technology : 8th-9th February 2020.

[4] MWB Azlinah, M. Bee, and W. Yap, “Supervised and Unsupervised Learning for Data Science Unsupervised and Semi-Supervised Learning Series Editor: M. Emre Celebi.” [On line]. Available: http://www.springer.com/series/15892

[5] Institute of Electrical and Electronics Engineers and PPG Institute of Technology, Proceedings of the 5th International Conference on Communication and Electronics Systems (ICCES 2020) : 10-12, June 2020.

[6] R.G. Leiva, A.F. Anta, V. Mancuso, and P. Casari, “A novel hyperparameter-free approach to decision tree construction

that avoids overfitting by design," IEEE Access, vol. 7, pp. 99978–99987, 2019, doi: 10.1109/ACCESS.2019.2930235.

[7] J. Biedrzycki and R. Burduk, "Decision tree integration using dynamic regions of competence," Entropy, vol. 22, no. 10, pp. 1–12, Oct. 2020, doi: 10.3390/e22101129.

[8] Institute of Electrical and Electronics Engineers and Manav Rachna International Institute of Research and Studies, Proceedings of the International Conference on Machine Learning, Big Data, Cloud and Parallel Computing : trends, perspectives and prospects : COMITCON-2019 : 14th-16th February, 2019.

[9] J. Hu, H. Peng, J. Wang, and W. Yu, "KNN-P: A KNN classifier optimized by P systems," Theor Comput Sci, vol. 817, pp. 55–65, May 2020, doi: 10.1016/j.tcs.2020.01.001.

[10] T. Adithiyaa, D. Chandramohan, and T. Sathish, "Optimal prediction of process parameters by GWO-KNN in stirring-squeeze casting of AA2219 reinforced metal matrix composites," Mater Today Proc, vol. 21, pp. 1000–1007, 2020, doi: https://doi.org/10.1016/j.matpr.2019.10.051.

[11] S. Ray, "A Quick Review of Machine Learning Algorithms," in 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), 2019, pp. 35–39. doi: 10.1109/COMITCon.2019.8862451.

[12] Z. Pan, Y. Wang, and Y. Pan, "A new locally adaptive K-Nearest Neighbors algorithm based on class discrimination," Knowl Based Syst, vol. 204, Sept. 2020, doi: 10.1016/j.knosys.2020.106185.

[13] M. Ali, LT Jung, AH Abdel-Aty, MY Abubakar, M. Elhoseny, and I. Ali, "Semantic-KNN algorithm: An enhanced version of traditional KNN algorithm," Expert Syst Appl, vol. 151, Aug. 2020, doi: 10.1016/j.eswa.2020.113374.

[14] A. Abdulrahman and S. Varol, "A Review of Image Segmentation Using MATLAB Environment," in 8th International Symposium on Digital Forensics and Security, ISDFS 2020, Institute of Electrical and Electronics Engineers Inc., Jun. 2020. doi: 10.1109/ISDFS49300.2020.9116191.

[15] V. Consonni, G. Baccolo, F. Gosetti, R. Todeschini, and D. Ballabio, "A MATLAB toolbox for multivariate regression coupled with variable selection," Chemometrics and Intelligent Laboratory Systems, vol. 213, Jun. 2021, doi: 10.1016/j.chemolab.2021.104313.

[16] S. Chakrabarti et al., 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC) : 7th-9th January, 2019, University of Nevada, Las Vegas, NV, USA.

[17] X. Hu, Y. Che, X. Lin, and S. Onori, "Battery Health Prediction Using Fusion-Based Feature Selection and Machine Learning," IEEE Transactions on Transportation Electrification, vol. 7, no. 2, pp. 382–398, Jun. 2021, doi: 10.1109/TTE.2020.3017090.

[18] Institute of Electrical and Electronics Engineers, The Tenth International Renewable Energy Congress : 2019 10th International Renewable Energy Congress (IREC) : March 26-28, 2019, Sousse, Tunisia.

[19] DR Nemade and R. Kumar Gupta, "Diabetes Prediction using BPSO and Decision Tree Classifier," 2020.

[20] A. Nugroho, "Comparison of Binary Particle Swarm Optimization And Binary Dragonfly Algorithm for Choosing the Feature Selection."

[21] Dr. 2020, pp. 1–6. doi: 10.1109/CDMA47397.2020.00006.

[22] J. Tanha, Y. Abdi, N. Samadi, N. Razzaghi, and M. Asadpour, "Boosting methods for multi-class imbalanced data classification: an experimental review," J Big Data, vol. 7, no. 1, Dec. 2020, doi: 10.1186/s40537-020-00349-y.

[23] M. Heydarian, T. E. Doyle, and R. Samavi, "MLCM: Multi-Label Confusion Matrix," IEEE Access , vol. 10, pp. 19083–19095, 2022, doi: 10.1109/ACCESS.2022.3151048.

Isti Amelia Isnaeni - Computer Science Department, Mercu Buana University, Jakarta, Indonesia Email: 41819120030@student.mercubuana.ac.id

Sandra Indriani - Computer Science Department, Mercu Buana University, Jakarta, Indonesia Email: 41819120071@student.mercubuana.ac.id

Muhammad Rizaq Nuriz Zaman - Computer Science Department, Mercu Buana University, Jakarta, Indonesia Email: 41819120021@student.mercubuana.ac.id

Andi Nugroho - Phd Student in Computer Sciences – Computer Science Department, Mercu Buana University, Jakarta, Indonesia Email : andi.nugroho@mercubuana.ac.id Scopus Author ID : 57208427717 ORCID : orcidID= https://orcid.org/0000-0002-1713 - 035X.