

Comparison of Machine Learning Algorithms for Polycystic Ovary Syndrome (PCOS) Classification

Sabila Sulistyani¹, Sumanto², Rizal Amegia Saputra³

Program Studi Informatika Fakultas Teknik dan Informatika Universitas Bina Sarana Informatika

Corresponding Author e-mail: sabilasty@gmail.com, sumanto@bsi.ac.id, rizal.rga@bsi.ac.id

Article History:

Received: 25-09-2026

Revised: 30-09-2026

Accepted: 07-10-2026

Keywords: Classification, K-Nearest Neighbor, Machine Learning, Polycystic Ovary Syndrome, Support Vector Machine

Abstract : Polycystic Ovary Syndrome (PCOS) is a hormonal disorder that can be challenging to detect due to its diverse clinical manifestations. This study aims to compare the performance of K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Logistic Regression for PCOS classification under the same experimental conditions. A quantitative approach was applied to a secondary PCOS Diagnosis dataset containing 1,000 patient records and five predictor variables: Age, BMI, Menstrual Irregularity, Testosterone Level, and Antral Follicle Count. Data were processed using Altair AI Studio through preprocessing, Min-Max normalization, 80:20 data splitting, modeling, and evaluation. KNN used $k = 5$, SVM used an RBF kernel, and Logistic Regression was applied for binary classification. Performance was evaluated using accuracy, precision, recall, and F1-score. KNN achieved the highest accuracy (96.00%), recall (90.91%), and F1-score (90.91%), while SVM achieved the highest precision (91.89%). Logistic Regression obtained the lowest overall performance. These results indicate that KNN demonstrated the strongest overall performance for PCOS classification under the experimental conditions used. Further validation using larger and more diverse clinical datasets is recommended.

How to Cite: Sabila Sulistyani, Sumanto, Rizal Amegia Saputra. (2026). The title. Ambidextrous Journal of Innovation Efficiency and Technology in Organizations. 5(3). Pp 544-556
<https://doi.org/10.61536/ambidextrous.v5i03.879>

 <https://doi.org/10.61536/ambidextrous.v5i03.897>

This is an open-access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) License.



Introduction

Polycystic Ovary Syndrome (PCOS) is a hormonal disorder commonly found among women of reproductive age and is one of the reproductive health issues encountered globally. The global prevalence of PCOS is estimated at 8–13%, depending on the diagnostic criteria used (Lin et al., 2025). This condition is characterized by hyperandrogenism, ovulatory dysfunction, and polycystic ovaries and is associated with various metabolic and cardiovascular complications, mental health disorders, and endometrial cancer (Lim et al., 2023). In Indonesia, attention to PCOS has also developed alongside the increasing number of cases among women of reproductive age. At Dr. Kariadi General Hospital Semarang, the number of cases increased from 66 cases in 2020 to 104 cases in 2022 (Salsabila et al., 2024). Meanwhile, the prevalence of PCOS among adolescent girls is estimated at 6–10% and is

expected to continue increasing along with lifestyle changes, such as high-calorie food consumption, reduced physical activity, and increased psychological stress (Murthi et al., 2025).

The high incidence of PCOS still presents challenges in the diagnosis and early detection process. Approximately 75% of PCOS cases are reported to remain medically undiagnosed because of the complexity of their symptoms and limitations in available diagnostic methods (Alfiryal et al., 2025). Conventional diagnosis, which includes medical interviews, hormone level examinations, and ultrasonography, is also not equally accessible across all healthcare facilities. This condition can lead to delayed diagnosis, potentially increasing the risk of complications and reducing patients' quality of life. Therefore, an approach that can support a more effective PCOS detection process is needed. One approach that can be used is machine learning, which can automatically analyze clinical data to identify patterns and generate supporting information for decision-making (Prasetya & Edward, 2025).

The next issue concerns differences in the characteristics and approaches used in machine learning-based PCOS classification studies. KNN relies on the proximity between data points, SVM uses an optimal hyperplane to separate classes, while Logistic Regression uses a probabilistic approach for binary classification. These differences in characteristics may cause each algorithm to produce different classification capabilities when applied to the same data. In previous research, Pitaloka and Kusnawi (2023) reported a KNN accuracy of 93%, while Permana et al. (2025) obtained an SVM accuracy of 90.74%. Taha et al. (2025) also reported that Logistic Regression achieved an accuracy of 91.7%.

Differences in previous research results cannot yet be used to directly compare the capabilities of the three algorithms because each study applied different experimental scenarios in terms of preprocessing, feature selection, and model optimization. Related studies also show that PCOS classification research uses KNN, SVM, Logistic Regression, and ensemble learning approaches with diverse dataset characteristics and processing techniques. These differences produce varying evaluation values, so the performance of each algorithm cannot yet be directly compared. Therefore, testing using the same dataset, preprocessing stages, data splitting ratio, and evaluation methods is needed so that the performance characteristics of each algorithm can be compared more objectively.

Based on these problems, this study aimed to compare the performance of K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Logistic Regression algorithms in Polycystic Ovary Syndrome (PCOS) classification using consistent testing conditions. All algorithms were applied to the same dataset using identical preprocessing stages and evaluation methods so that the comparison results could be obtained consistently. The urgency of this study lies in the need to obtain information regarding the capabilities of each algorithm as a support for early PCOS detection based on clinical data. The novelty of this study lies in comparing KNN, SVM, and Logistic Regression using the same dataset, the same preprocessing stages, and consistent evaluation methods, allowing differences in algorithm performance to be analyzed under consistent testing conditions.

Method

This study used a quantitative approach with a machine learning method to develop a Polycystic Ovary Syndrome (PCOS) classification model based on patient clinical data. The data used were secondary data from the PCOS Diagnosis dataset obtained through Kaggle. The machine learning method was used to identify patterns in the data and generate classifications based on the available clinical variables (Taye, 2023; Prasetya & Edward, 2025).

Data processing was performed using Altair AI Studio through preprocessing, data cleaning, data transformation, data splitting, modeling, and evaluation stages. The preprocessing stage included handling missing values using Replace Missing Values with the attribute mean and normalization using the Min-Max method on a range of 0 to 1. These stages were conducted to improve data readiness before the classification process (Cofre-Martel et al., 2021; Ortiz et al., 2024). The data were then divided into 80% test data and 20% testing data to ensure that the model was tested using data not used during training (Joseph, 2022; Sivakumar et al., 2024).

The research data consisted of 1,000 patient records with six attributes: Age, BMI, Menstrual Irregularity, Testosterone Level, Antral Follicle Count, and PCOS Diagnosis. Five attributes were used as predictor variables, while PCOS Diagnosis was used as the target variable, with a category of 0 for patients not diagnosed with PCOS and 1 for patients diagnosed with PCOS. All data were used in the study, so the algorithms were compared using the same dataset and data characteristics.

The research procedure began with dataset collection, preprocessing, data splitting, modeling, and evaluation. The three algorithms compared were K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Logistic Regression. KNN used Euclidean Distance with $k = 5$, SVM used the Radial Basis Function (RBF), while Logistic Regression was used for binary classification (Schober & Vetter, 2021; Du et al., 2024). All algorithms used the same dataset, preprocessing process, and 80:20 data splitting to maintain consistency in the comparison and reduce the potential for data leakage (Ichwani et al., 2026).

Model evaluation was conducted using Performance (Binomial Classification) in Altair AI Studio by examining the confusion matrix, accuracy, precision, recall, and F1-score. The evaluation results were used to compare the performance of the three algorithms in classifying PCOS on the testing data. Using multiple evaluation metrics allowed model performance to be assessed more comprehensively, rather than based solely on accuracy.

Results and Discussion

Research Results

The research results for Polycystic Ovary Syndrome (PCOS) classification are presented based on the research stages conducted using Altair AI Studio. As an initial stage, this study used a PCOS dataset containing patient clinical data as the data source for the classification process. The dataset is shown in Figure 1.

	A	B	C	D	E	F
1	Age	BMI	Menstrual_Irregularity	Testosterone_Level(ng/dL)	Antral_Follicle_Count	PCOS_Diagnosis
2	24	34,7	1	25,2	20	0
3	37	26,4	0	57,1	25	0
4	32	23,6	0	92,7	28	0
5	28	28,8	0	63,1	26	0
6	25	22,1	1	59,8	8	0
7	38	19,3	0	28,4	6	0
8	24	20,2	1	72,5	29	0
9	43	20,2	1	85,8	17	0
10	36	20,6	0	50,4	5	0
11	40	20,4	0	82	21	0
12	28	28,9	0	97,2	17	0
13	28	21,1	1	36,3	8	0
14	41	23,9	0	61,9	11	0
15	38	33,2	1	43	21	1
16	21	26,1	1	83,4	13	1
17	25	29,3	0	66,2	8	0
18	41	20,9	0	70,8	23	0
19	20	21,3	0	83,8	5	0
20	39	18,7	0	51,7	14	0
21	38	20,9	0	93,2	8	0
22	19	22,7	0	62,6	13	0
23	41	21	1	32,6	17	0
24	29	19,5	1	75,7	18	0
25	23	20,1	1	83,5	14	0
26	19	25,8	0	45,3	10	0
27	45	21,5	1	88,6	20	0
28	38	24,2	1	92,5	10	0

Source: (Research Results, 2026)

Figure 1. Raw Data

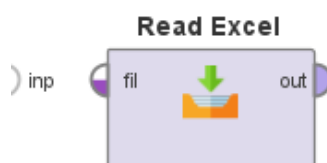
Data Preprocessing Results

The data preprocessing stage was conducted to prepare the dataset before it was used for algorithm modeling and testing. In this study, data preprocessing included dataset import, missing values checking, and data transformation using Altair AI Studio. The results of each preprocessing stage are described as follows.

1. Dataset Import

At this stage, the dataset was first imported into Altair AI Studio using the Read Excel operator. Using this operator, Excel-formatted dataset files can be read and imported into Altair AI Studio,

making them ready for processing in subsequent research stages. The configuration of the Read Excel operator used is shown in Figure 2.



Source: (Research Results, 2026)

Figure 2. Read Excel Operator

After the dataset was successfully imported into Altair AI Studio, the attributes were configured through the Edit Parameter List: Data Set Meta Data Information menu. This stage aimed to ensure that each attribute was correctly recognized according to its role in the classification process. The results of the attribute settings can be seen in Figure 3.

column index	attribute meta data information	role
0	Age	integer attribute
1	BMI	real attribute
2	Menstrual_Irreg	integer attribute
3	Testosterone_L	real attribute
4	Antral_Follicle_	integer attribute
5	PCOS_Diagno	binominal label

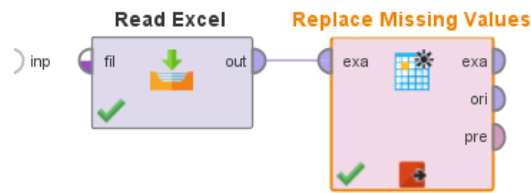
Source: (Research Results, 2026)

Figure 3. Data Attribute Information

Figure 3 shows the attribute information used in this study, along with the configuration of their roles in Altair AI Studio. The PCOS_Diagnosis attribute was set as the label, which functions as the classification target to be predicted by the model. Meanwhile, the other attributes were used as input variables in the model learning process. This configuration was necessary so that the classification process could proceed according to the research objectives.

2. Data Cleaning Results

After the dataset import process was completed, a data quality check was conducted to ensure that there were no missing values in the dataset used. This check used the Replace Missing Values operator connected to the Read Excel operator through the output (out) port. Through this process, all attributes in the dataset could be examined to identify the presence of missing values. Figure 4 shows the implementation of missing values checks using Altair AI Studio.



Source: (Research Results, 2026)

Figure 4. Replace Missing Values Operator

Through this process, Altair AI Studio checks each attribute in the dataset to identify any missing values. The results of this check are shown in Figure 5.

Result History						
ExampleSet (Replace Missing Values)						
	Name	Type	Missing	Statistics	Filter (6 / 6 attributes)	
✓	PCOS_Diagnosis	Binominal	0	Negative 0 Positive 1	Values 0 (801), 1 (199)	
✓	Age	Integer	0	Min 18 Max 45	Average 31.771	
✓	BMI	Real	0	Min 18.100 Max 35	Average 26.387	
✓	Menstrual_Irregularity	Integer	0	Min 0 Max 1	Average 0.530	
✓	Testosterone_Level(ng/dL)	Real	0	Min 20 Max 99.800	Average 60.160	
✓	Antral_Follicle_Count	Integer	0	Min 5 Max 29	Average 17.469	

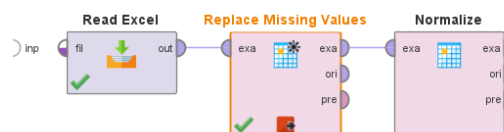
Source: (Research Results, 2026)

Figure 5. Missing Values Check Results

The results of the check showed that none of the attributes in the dataset contained empty or missing values. This condition indicates that the dataset used in this study was complete from the outset, so an imputation step was unnecessary and the dataset could proceed directly to the data transformation stage.

3. Data Transformation Results

After the data were confirmed to have no missing values, the next stage was data transformation using the Normalize operator. In this study, transformation was performed using Min-Max Normalization to equalize the range of numerical attribute values so that differences in data scale would not affect the classification process. The implementation of the data transformation process in Altair AI Studio is shown in Figure 6.



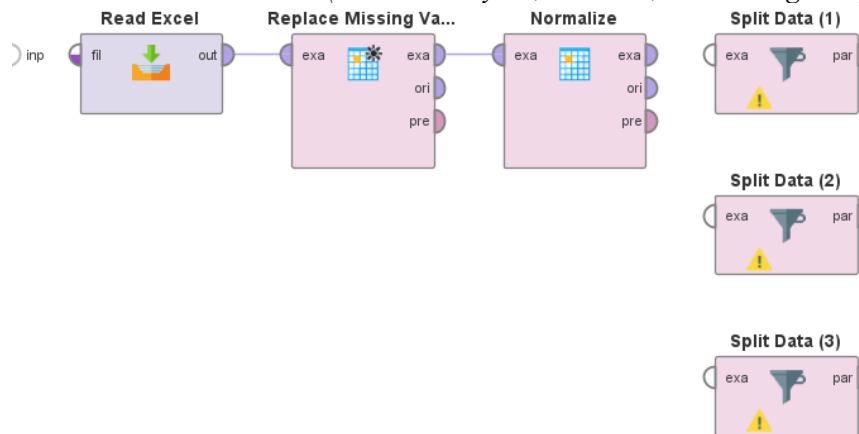
Source: (Research Results, 2026)

Figure 6. Normalize Operator

Once the data cleaning stage is complete, the data is forwarded to the Normalize operator via the output port (exa), as shown in Figure 6. This stage transformed all numerical attribute values to a scale of 0 to 1, according to the predetermined normalization method.

Data Splitting Results

After preprocessing, the dataset was divided into two parts—test data and testing data—using the Split Data operator as the basis for model training and evaluation of classification results. The implementation flow of the Split Data operator is shown in Figure 7.



Source: (Research Results, 2026)

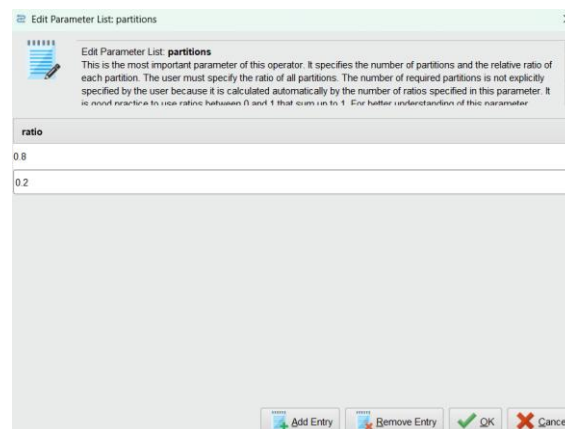
Figure 7. Split Data Operator

The Split Data Operator is used to divide the dataset into training and test data, which are used for the model training and evaluation stages, respectively. The split ratio applied is 80:20, with 80% as test data and the remaining 20% as test data. The results of the dataset split are presented in Table 1.

Table 1. Data Splitting

No	Data Type	Total Data	Data Percentage
1	Train Data	800	80%
2	Test Data	200	20%
3	Total Data	1000	

The data splitting ratio was configured using the parameters of the Split Data operator. The parameter settings used in this study are shown in Figure 8.

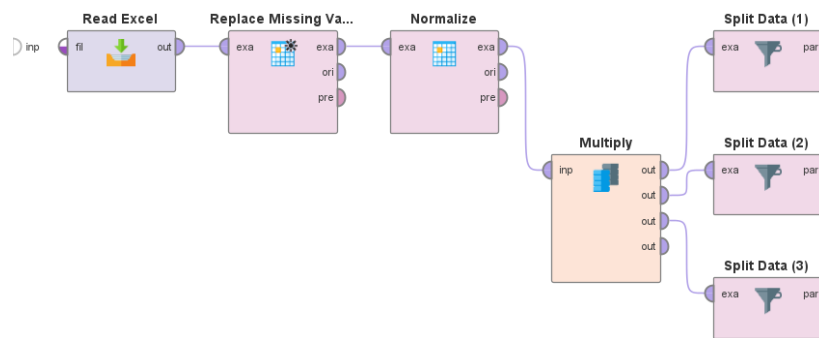


Source: (Research Results, 2026)

Figure 8. Parameter List

Algorithm Modeling Results

At the modeling stage, the Multiply operator was used to support the comparison of several classification algorithms applied in this study. This operator allowed the preprocessed data to be distributed to several processing paths simultaneously, so that each algorithm received the same dataset for training and testing. Thus, the performance results of each algorithm could be compared more objectively. The implementation of the Multiply operator in this study is shown in Figure 9.



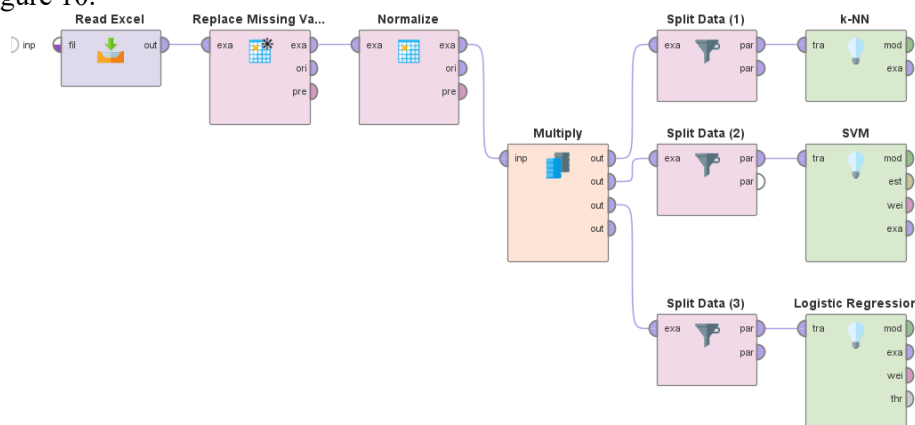
Source: (Research Results, 2026)

Figure 9. Multiply Operator

Next, the output ports of the Multiply operator were connected to three Split Data operators, each designated for the K-Nearest Neighbor, Support Vector Machine, and Logistic Regression algorithms. This configuration was intended to ensure that each algorithm used the same data during training and testing so that the resulting performance could be compared objectively.

Algorithm Implementation

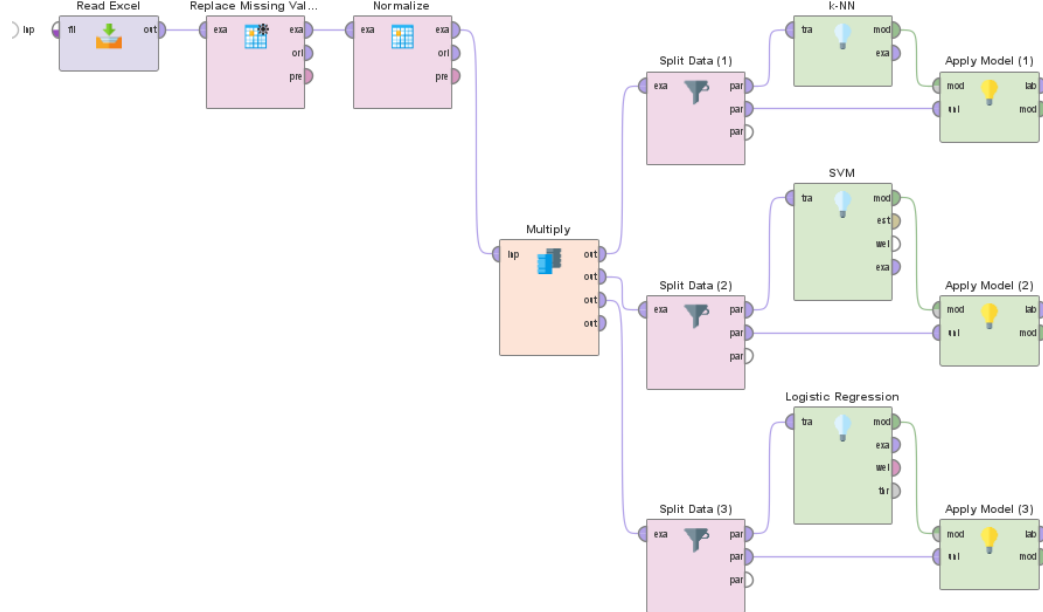
The classification algorithm implementation phase is carried out after the data splitting process using the Multiply and Split Data operators is complete. In this phase, partition 1, which contains the test data, is connected to the training port (tra) on each algorithm operator used. This test data represents 80% of the total dataset used in this study. The model training implementation for each algorithm is shown in Figure 10.



Source: (Research Results, 2026)

Figure 10. Multiply Process

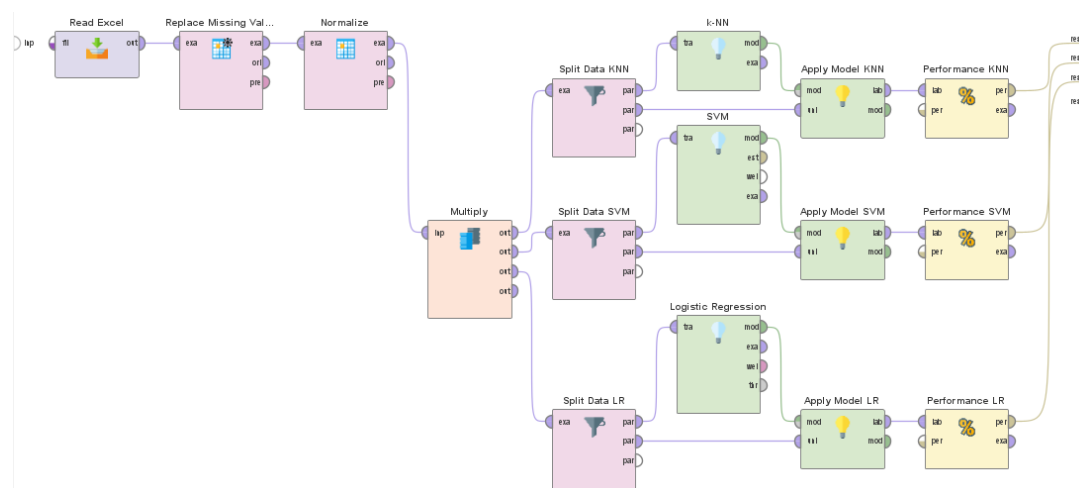
Next, model testing is performed using the Apply Model operator, which applies the trained model to the test data. At this stage, the model (mod) port of each algorithm is connected to the mod port of the Apply Model operator, while partition 2—which contains 20% of the total dataset as test data—is connected to the unlabeled (unl) port. This stage is used to generate predictions for the test data based on the previously developed model. The implementation of the Apply Model operator is shown in Figure 11.



Source: (Research Results, 2026)

Figure 11. Apply Model Process

To measure how well each algorithm performed, model evaluation was conducted using the Performance (Binomial Classification) operator. This operator compares the labels predicted by the model with the actual labels in the testing data, allowing the performance of each algorithm to be measured objectively. The evaluation results consisted of accuracy, precision, recall, and F1-score values, which were used as the basis for comparing the performance of KNN, SVM, and Logistic Regression. The implementation of the model evaluation process is shown in Figure 12.



Source: (Research Results, 2026)

Figure 12. Performance Process

Testing Results

1. Testing of the K-Nearest Neighbor (KNN) Algorithm

In testing the K-Nearest Neighbor (KNN) algorithm, the accuracy was 96.00%, the precision was 90.91%, the recall was 90.91%, and the F1-score was 90.91%. These figures indicate that the KNN algorithm is relatively effective at recognizing PCOS data patterns in the tested dataset, with a relatively low classification error rate compared to the number of correct predictions.

Based on the confusion matrix, the model correctly classified 152 non-PCOS data points and 40 PCOS data points. In addition, 4 PCOS data points were predicted as non-PCOS (false negatives),

and 4 non-PCOS data points were predicted as PCOS (false positives). The number of correct predictions was higher than the number of classification errors, resulting in high evaluation values for the KNN algorithm. The confusion matrix resulting from testing the K-Nearest Neighbor algorithm can be seen in Figure 13.

accuracy: 96.00%

	true 0	true 1	class precision
pred. 0	152	4	97.44%
pred. 1	4	40	90.91%
class recall	97.44%	90.91%	

Source: (Research Results, 2026)

Figure 13. Testing Results Algorithm K-Nearest Neighbor (KNN)

2. Testing of the Support Vector Machine (SVM) Algorithm

In testing the Support Vector Machine (SVM) algorithm, the accuracy was 93.50%, the precision was 91.89%, the recall was 77.27%, and the F1-score was 83.95%. These results indicate that SVM is still capable of classifying PCOS data with good performance, although its evaluation results remain below those of KNN.

Based on the confusion matrix results, this model successfully classified 153 non-PCOS data points and 34 PCOS data points correctly. In addition, 10 PCOS data points were predicted as non-PCOS (false negatives), and 3 non-PCOS data points were predicted as PCOS (false positives). The higher number of false negatives compared to KNN indicates that some PCOS cases were still not successfully identified by the model. The confusion matrix generated from testing the Support Vector Machine algorithm is shown in Figure 14.

accuracy: 93.50%

	true 0	true 1	class precision
pred. 0	153	10	93.87%
pred. 1	3	34	91.89%
class recall	98.08%	77.27%	

Source: (Research Results, 2026)

Figure 14. Testing Results Algorithm Support Vector Machine (SVM)

3. Testing of the Logistic Regression Algorithm

In testing the Logistic Regression algorithm, the accuracy was 89.50%, precision was 78.05%, recall was 72.73%, and the F1-score was 75.29%. These results indicate that Logistic Regression is capable of classifying PCOS data but performs less well than KNN and SVM. The model's predictions on the test data were then analyzed using a confusion matrix to determine the number of correct classifications and classification errors. The confusion matrix resulting from the Logistic Regression algorithm test is shown in Figure IV.15.

Based on the confusion matrix results, the model successfully classified 147 non-PCOS data points and 32 PCOS data points correctly. However, 12 PCOS data points were still predicted as non-PCOS (false negatives), and 9 non-PCOS data points were predicted as PCOS (false positives). The number of errors indicates that the model still has limitations in accurately distinguishing between the PCOS and non-PCOS classes. The confusion matrix from the Logistic Regression algorithm test is shown in Figure 15.

accuracy: 89.50%

	true 0	true 1	class precision
pred. 0	147	12	92.45%
pred. 1	9	32	78.05%
class recall	94.23%	72.73%	

Source: (Research Results, 2026)

Figure 15. Testing Results Algorithm Logistic Regression

Based on the confusion matrix, Logistic Regression successfully classified 147 non-PCOS data points and 32 PCOS data points correctly. However, 12 PCOS data points were still predicted as non-PCOS (false negatives), and 9 non-PCOS data points were predicted as PCOS (false positives). The number of errors indicates that the model still has limitations in accurately distinguishing between the PCOS and non-PCOS classes.

The 12 false negatives affected the model's ability to recognize PCOS cases, as reflected in the recall value of 72.73%. This means that some data that actually belonged to the PCOS class were not successfully recognized by the model. Meanwhile, the presence of 9 false positives also affected the precision value, which reached only 78.05%. Thus, the classification errors produced by Logistic Regression resulted in lower accuracy, precision, recall, and F1-score values than those of the other algorithms.

One possible reason for the lower performance of Logistic Regression is that the relationship between the attributes used and the PCOS class in the research dataset may not be fully captured by a linear approach. Logistic Regression performs classification based on the relationship between predictor variables and class probabilities, so its performance can be influenced by the patterns of relationships among the attributes in the data. The attributes used in this study include Age, BMI, Menstrual Irregularity, Testosterone Level, and Antral Follicle Count. Meanwhile, KNN performs classification based on data proximity, and SVM uses the Radial Basis Function (RBF) kernel; thus, the two algorithms employ different approaches to recognizing data patterns. These differences in methodological characteristics may be one factor contributing to Logistic Regression's lower performance on this research dataset.

Compared to the other algorithms, Logistic Regression produced a higher number of classification errors. KNN produced 4 false negatives and 4 false positives, while SVM produced 10 false negatives and 3 false positives. Logistic Regression produced 12 false negatives and 9 false positives. This indicates that KNN was more capable of recognizing patterns in the research dataset, while SVM also had better classification performance than Logistic Regression. Thus, the lower performance of Logistic Regression was reflected not only in its accuracy value but also in its precision, recall, F1-score, and number of classification errors.

4. Algorithm Performance Comparison

After testing the three algorithms, the next stage was to compare the performance of each algorithm based on accuracy, precision, recall, and F1-score. The comparison aimed to identify the algorithm that provided the best performance in classifying PCOS. The performance comparison results for the three algorithms are presented in Table 2.

Table 2. Comparison of Algorithm Testing Results

Algorithm	Accuracy	Precision	Recall	F1-Score
K-Nearest Neighbor (KNN)	96.00%	90.91%	90.91%	90.91%
Support Vector Machine (SVM)	93.50%	91.89%	77.27%	83.95%
Logistic Regression	89.50%	78.05%	72.73%	75.29%

Source: (Research Results, 2026)

Based on Table 2, the K-Nearest Neighbor (KNN) algorithm showed the highest performance among the algorithms, obtaining an accuracy 96,00%, precision 90,91%, recall 90,91%, and F1-score 90,91%. These results show that KNN had a good ability to classify both PCOS and non-PCOS data in a balanced manner. The higher performance of KNN may be associated with the algorithm's characteristic of determining the data class based on its proximity to other test data. In this study, KNN used a k value of 5 and weighted voting, so data points with shorter distances had a greater influence in determining the class. In addition, the data underwent Min-Max Normalization so that all numerical attributes were within a uniform range and distances between data points could be calculated more optimally.

Meanwhile, the Support Vector Machine (SVM) achieved an accuracy of 93.50%, a precision of 91.89%, a recall of 77.27%, and an F1-score of 83.95%. The SVM's precision value of 91.89% was

slightly higher than that of KNN, indicating that the positive PCOS classification generated by SVM were relatively accurate. However, the lower recall value indicates that some PCOS cases were still not successfully identified by the model. Based on the confusion matrix, SVM produced 10 false negatives, which is higher than KNN, which produced 4 false negatives.

This performance difference may be related to the characteristic of SVM that forms a hyperplane to separate data classes. This study used the Radial Basis Function (RBF) kernel, which allows SVM to handle nonlinear relationship patterns. Nevertheless, this configuration did not produce class separation as effective as KNN on this research dataset. This is reflected in the higher number of false negatives, resulting in a lower recall value for SVM than for KNN. The SVM parameters used in this study included the RBF kernel with a gamma value of 0.1 and several other parameters.

Meanwhile, Logistic Regression achieved the lowest performance, with an accuracy of 89.50%, a precision of 78.05%, a recall of 72.73%, and an F1-score of 75.29%. This performance was also supported by the confusion matrix results, which showed 12 false negatives and 9 false positives. The number of errors was higher than that of KNN and SVM, resulting in a lower ability of Logistic Regression to distinguish between the PCOS and non-PCOS classes. As explained in the previous discussion, one possible cause is that the relationship between the attributes in the dataset and the PCOS class may not be fully represented by the linear approach of Logistic Regression.

Thus, the differences in performance among the three algorithms in this study may be associated with differences in the mechanisms each algorithm uses to recognize data patterns. KNN determines the class based on data proximity, allowing it to produce good results on the research dataset after normalization. SVM uses hyperplane formation with the RBF kernel and therefore can handle more complex patterns, but in this study it still produced more false negatives than KNN. Meanwhile, Logistic Regression uses a probabilistic approach based on the relationship between predictor variables and classes, resulting in the lowest performance on the research dataset. Under the same preprocessing, dataset, and data splitting conditions, these differences indicate that the characteristics of classification methods can affect the model's ability to recognize PCOS data patterns.

Conclusion

Based on the research results, a comparison of the K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Logistic Regression algorithms for classifying Polycystic Ovary Syndrome (PCOS) revealed differences in performance when the three algorithms were applied to the same dataset, preprocessing, and data splitting. KNN achieved an accuracy of 96.00%, precision of 90.91%, recall of 90.91%, and an F1-score of 90.91%. SVM achieved an accuracy of 93.50%, precision of 91.89%, recall of 77.27%, and an F1-score of 83.95%, while Logistic Regression achieved an accuracy of 89.50%, precision of 78.05%, recall of 72.73%, and an F1-score of 75.29%. These results indicate that the characteristics of classification mechanisms affect the model's ability to distinguish between PCOS and non-PCOS patterns in the research dataset. In practice, the research results can provide supporting information for algorithm selection in the development of PCOS classification systems based on clinical data, particularly when a model with a relatively low classification error rate is required.

This study has limitations because it uses only one secondary dataset consisting of 1,000 records and six attributes, with five attributes serving as predictors; therefore, its performance results cannot yet be generalized to a broader population or other clinical datasets. This study also used only one preprocessing scenario, an 80:20 data splitting, and specific algorithm parameter configurations, so it does not yet represent the full range of possible model performance. Future research is recommended to use a more diverse dataset from a broader range of clinical sources, implement cross-validation, hyperparameter tuning, and feature selection, and compare more algorithms to obtain a more comprehensive picture of performance. Further development could also consider integrating the model into clinical decision support systems while ensuring validation and evaluation on real-world data before practical deployment.

References

Alfiryal, N., Sadik, K., Suhaeni, C., & Soleh, A. M. (2025). Deteksi *polycystic ovary syndrome* (PCOS) berbasis *machine learning*: Kombinasi SMOTE, random forest, gradient boosting, dan Bayesian

- optimization. *Indonesian Journal of Applied Statistics*, 8(2), 148–159. <https://jurnal.uns.ac.id/ijas/article/view/109931>
- Cofre-Martel, S., Droguett, E. L., & Modarres, M. (2021). Big machinery data preprocessing methodology for data-driven models in prognostics and health management. *Sensors*, 21(20), 1–29. <https://doi.org/10.3390/s21206841>
- Du, K. L., Jiang, B., Lu, J., Hua, J., & Swamy, M. N. S. (2024). Exploring kernel machines and support vector machines: Principles, techniques, and future directions. *Mathematics*, 12(24), 1–58. <https://doi.org/10.3390/math12243935>
- Ichwani, A., Kesuma, R. I., Setiawan, A., & Wicaksono, I. E. (2026). Preventing data leakage in classification via integrated machine learning pipelines: Preprocessing, feature transformation, and hyperparameter tuning. *Jurnal Teknik Informatika (JUTIF)*, 7(1), 391–410. <https://doi.org/10.52436/1.jutif.2026.7.2.5490>
- Joseph, V. R. (2022). Optimal ratio for data splitting. *Statistical Analysis and Data Mining*, 15(4), 531–538. <https://doi.org/10.1002/sam.11583>
- Lim, J., Li, J., Feng, X., Feng, L., Xia, Y., Xiao, X., Wang, Y., & Xu, Z. (2023). Machine learning classification of polycystic ovary syndrome based on radial pulse wave analysis. *BMC Complementary Medicine and Therapies*, 23(1), 1–15. <https://doi.org/10.1186/s12906-023-04249-5>
- Lin, T., Xie, B., Yang, J., Xu, J., & Chen, F. (2025). Changes in the global burden of polycystic ovary syndrome from 1990 to 2021. *Reproductive Health*, 22(1). <https://doi.org/10.1186/s12978-025-02016-y>
- Murthi, A. K., Wratsangka, R., Sisca, & Kurniawan, Y. (2025). Polycystic ovarian syndrome (PCOS) in adolescent girls in Indonesia: A new burden in women's reproductive health. *Jurnal Biomedika dan Kesehatan*, 8(3), 245–249. <https://doi.org/10.18051/JBiomedKes.2025.v8.245-249>
- Ortiz, B. L., Gupta, V., Kumar, R., Jalin, A., Cao, X., Ziegenbein, C., Singhal, A., Tewari, M., & Choi, S. W. (2024). Data preprocessing techniques for AI and machine learning readiness: Scoping review of wearable sensor data in cancer care. *JMIR mHealth and uHealth*, 12, e59587. <https://doi.org/10.2196/59587>
- Permana, B. A. C., Zulkipli, Wasil, M., & Harianto. (2025). Penerapan kecerdasan buatan untuk menentukan gejala utama *polycystic ovary syndrome* (PCOS) berdasarkan data klinis pasien. *Infotek: Jurnal Informatika dan Teknologi*, 8(2), 543–553. <https://doi.org/10.29408/jit.v8i2.31123>
- Pitaloka, N. T., & Kusnawi. (2023). PCOS disease classification using feature selection RFECV and EDA with KNN algorithm method. *Jurnal Teknik Informatika (JUTIF)*, 4(4), 693–701. <https://doi.org/10.52436/1.jutif.2023.4.4.831>
- Prasetya, F., & Edward. (2025). Diagnosis penyakit menggunakan artificial intelligence (AI): Konsep, bukti ilmiah, dan implikasi klinis. *Jurnal Sains dan Kesehatan*, 6(1), 98–107. <https://doi.org/10.30872/jsk.v6i1.976>
- Salsabila, W., Adyani, K., & Realita, D. F. (2024). Perbandingan pengetahuan dan sikap tentang PCOS pada mahasiswa kesehatan dan non kesehatan. *Journal of Sexual and Reproductive Health*, 4(1), 25–38. <https://doi.org/10.53088/griyawidya.v4i1.1930>
- Schober, P., & Vetter, T. R. (2021). Logistic regression in medical research. *Anesthesia & Analgesia*, 132(2), 365–366. <https://doi.org/10.1213/ANE.0000000000005247>
- Sivakumar, M., Parthasarathy, S., & Padmapriya, T. (2024). Trade-off between training and testing ratio in machine learning for medical image processing. *PeerJ Computer Science*, 10, 1–17. <https://doi.org/10.7717/peerj-cs.2245>

- Taha, R., Zain El Abdin, H., & Musleh, T. (2025). Comparative analysis of supervised machine learning models for PCOS classification using clinical data. *Journal of Engineering Research and Sciences*, 4(6), 16–26. <https://doi.org/10.55708/js0406003>
- Taye, M. M. (2023). Understanding of machine learning with deep learning: Architectures, workflow, applications and future directions. *Computers*, 12(5), 91. <https://doi.org/10.3390/computers12050091>

