

Improving k-nearest neighbor performance using permutation feature importance to predict student success in study

Gd. Aditya Jana Satvika, I. N. Sukajaya, I Gede Aris Gunadi

Department of Computer Science, Universitas Pendidikan Ganesha, Singaraja, Indonesia

Article Info

Article history:

Received Dec 12, 2023

Revised Apr 4, 2024

Accepted May 7, 2024

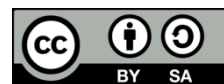
Keywords:

Classification
Feature importance
Feature selection
On-time graduation
Supervised learning

ABSTRACT

The timely graduation of students is a critical indicator of academic quality assessment. Therefore, universities should use effective predictive systems to identify earlier potential lateness of graduation. This study aimed to improve the K-nearest neighbor (K-NN) algorithm's ability to predict student on-time graduation. It evaluated K-NN algorithm performance with and without the permutation feature importance (PFI) technique, using a dataset of 460 student graduation records from 2014 to 2017. The training data was oversampled, adjusting the ratio of minority class samples from 13% to 100% of the majority class samples. The result shows that integrating PFI into the K-NN model improved K-NN performance by 10 iterations of the PFI process, N-shuffle varying from 10 to 100 for each iteration, and a minority class sample ratio of 25%. The accuracy score improved from 90.22% to 92.39%, precision from 50.00% to 62.50%, F1-score from 52.63% to 58.82%, while recall remained consistent at 55.56%. The PFI analysis showed that achievement index for the 1st semester or IPS 1 had the least impact on the model. The study suggested using a comprehensive approach to determine the n-shuffle of PFI based on the number of test data for a more accurate feature contribution pattern.

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



Corresponding Author:

I. N. Sukajaya

Department of Computer Science, Universitas Pendidikan Ganesha
Singaraja, Buleleng, Bali, Indonesia

Email: nyoman.sukajaya@undiksha.ac.id

1. INTRODUCTION

In Indonesia, the quality of higher education is determined by the accreditation grade awarded by the National Accreditation Board for Higher Education (*Badan Akreditasi Nasional-Perguruan Tinggi* or BAN-PT). Timely graduation of students is an indicator of the quality of a university or program used as a benchmark in the accreditation process [1]. Regulation number 5 of 2019 from BAN-PT outlines the accreditation instrument for bachelor's programs. According to this regulation, a program will receive a score of 4 for the on-time graduation indicator if the percentage of on-time student graduations (PTW) is 50% or higher. If PTW is less than 50%, the score is calculated using the formula $1+(6 \times \text{PTW})$ [2]. It is crucial to identify students who are potentially late in completing their studies, enabling prompt evaluation and academic support to students. Based on the observation of 460 students' graduation data of the education of informatics engineering (*pendidikan teknik informatika* or *PTI*) program, Faculty of Engineering and Vocational Studies, Universitas Pendidikan Ganesha (Undiksha), Singaraja, Bali in 2014-2017. It was found that only 50 students completed their education on time, while 410 other students experienced late completion of their studies. These findings indicate not only individual challenges in achieving optimal academic performance but also risks to program quality standards. Therefore, a predictive method is needed to

identify students' ability to complete their education in a timely manner as early as possible. This will enable stakeholders and students to evaluate and plan strategically, reducing the significant number of late graduations.

Data classification is a commonly used method to predict the potential for late graduation of students. Several machine learning algorithms have been implemented for this purpose, including Naïve Bayes, support vector machine (SVM), C4.5, artificial neural network, and K-nearest neighbor (K-NN) [3]–[10]. However, it was not previously examined to what extent each feature contributed to the performance of the utilized model and what factors could influence each feature's contribution. K-NN has disadvantages such as vulnerability to high data dimensions and irrelevant features [11]. The presence or absence of irrelevant features strongly influences the accuracy of the K-NN algorithm [12]. Additionally, the high dimensionality of data directly impacts algorithm and model accuracy [13]. Therefore, to filter out irrelevant features in the K-NN model, a comprehensive approach is necessary.

Feature selection is necessary to address high data dimensionality [14], [15] and eliminate irrelevant or redundant features in a given case [16]. It is also important to improve learning performance, prevent overfitting, and reduce computational costs [17]. The objective of this study is to optimize the performance of the K-NN algorithm by selecting features using the permutation feature importance (PFI) method. PFI can be used in various machine learning models, including K-NN, and has a high computational speed without requiring repeated model training. The PFI process accounts for feature interactions [18], allowing for the identification of relevant and effective feature combinations in predicting timely student graduation. This step aims to reduce irrelevant dimensions of the data and optimize the K-NN algorithm with PFI for graduation prediction. This study also investigates the impact of data balancing on the model's performance.

2. METHOD

This study focuses on optimizing the performance of K-NN for predicting the timely graduation of students in the Undiksha PTI program based on previous students graduation history data. The PFI method was used to optimize the performance of K-NN by removing features that have no significant contribution to the K-NN model. Figure 1 illustrates the general design of the study.

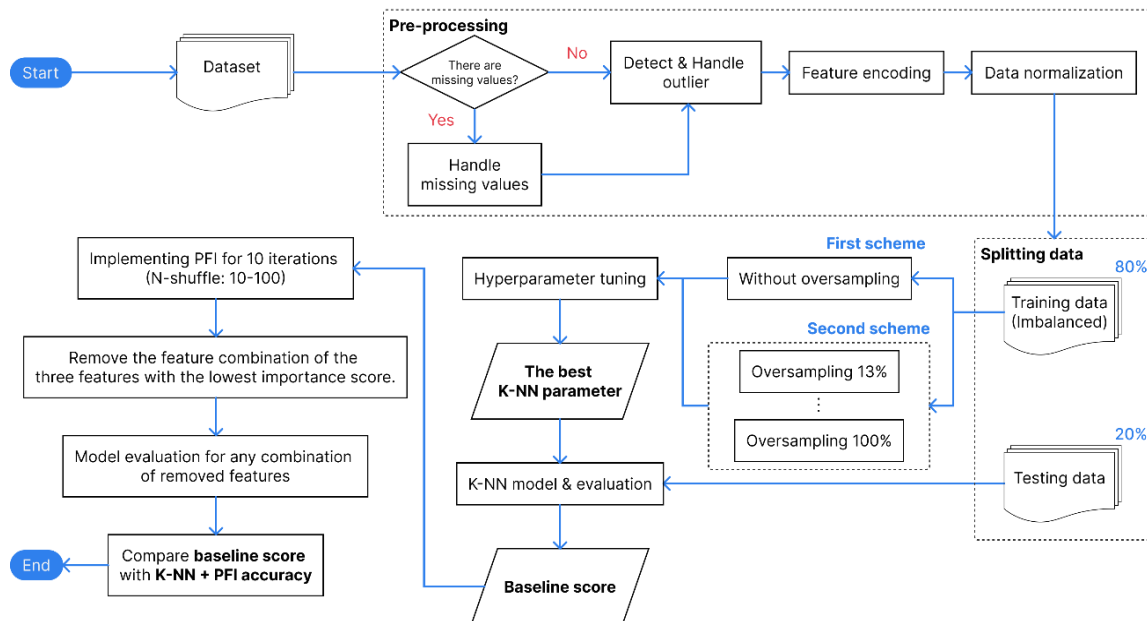


Figure 1. Proposed method

2.1. Dataset

The dataset of this study comprises student data from the Undiksha PTI program spanning the cohort period from 2014 to 2017. These data were collected from the technical implementation unit for technology, information and communication (UPA-TIK) of Undiksha. The obtained data was observed and integrated into a single .csv file. There are two types of variables used in this study, namely independent

variables (predictors) and dependent variable (determined by other variables). Table 1 gives a descriptive overview of the features involved.

Table 1. Feature information of Undiksha PTI student graduation dataset

Features	Values	Scales	Description
Gender	Male, female	Nominal	Student gender
Parents' income	Numeric	Ratio	Parents monthly income
UKT cost	Numeric	Ratio	Tuition fees, determined by parents' economic
Number of repeated courses (1 st -4 th semester)	Numeric	Ratio	Courses are repeated in the first four semester
IP semester (IPS 1-4)	0-4	Ratio	Achievement index for the 1 st -4 th semester
Graduation status	On-time, late	Nominal	Dependent attribute or data class

2.2. Data preprocessing

The student graduation dataset were prone to several problems that could adversely impact the performance of the K-NN model. The identified issues in the student graduation dataset included missing values, outlier data, and data encoding problems. Therefore, the preprocessing stage was crucial to improve the data quality by addressing these issues [19].

2.2.1. Dealing with missing values

One of the challenges in machine learning study is dealing with incomplete datasets containing missing values, which can lead to misleading analysis and inaccurate decisions [20]. The presence of missing values can compromise the quality of the data and can be the most influential source of weakness in machine learning. In this study, the imputation method was used as a strategy to deal with missing data. Each data feature used implements a different imputation method. For the gender feature, the mode imputation method was used because it belongs to categorical data. The parents' income and UKT cost features used the median imputation method and the IPS 1-4 features were imputed with a value of 0 to replace the null value.

2.2.2. Dealing with outlier data

Outlier data are typically defined as anomalous data characterized by a set of observations with extreme values. The term 'extreme value' refers to a value that is significantly different from the majority of values within its respective group, as if it resulted from a different mechanism. In order to identify outlier data in this study, a threshold was set to classify data as outliers by transforming the study data into a standard score (z-score). For large samples with more than 80 observations, the assessment guideline states that the z-score threshold is set at 3 [21]. Therefore, any observation with a z-score ≥ 3 is classified as an outlier. The mathematical formula for calculating the z-score, as in (1).

$$Z = \frac{x - \mu}{\sigma} \quad (1)$$

In (1), the z-score (Z) represents the standard score of an observed value (x) in relation to the population mean (μ) and standard deviation (σ).

2.2.3. Feature encoding

Before applying the K-NN method, it is necessary to transform the categorical features in the data set into numerical data so that they can be used in the K-NN method calculation. The choice of feature encoding method depends on the condition and type of categorical data. In this study, the one-hot encoding method was used to process nominal data types, where each category of data creates a new feature [22]. For example, the gender feature was converted into two separate features, is_Male and is_Female, each with the values 0 and 1 (true/false). Meanwhile, the label encoding method was used for the graduation status feature, which acts as a dependent attribute, because it is only a data class. In this context, the class 'on-time' is represented by the value 1, while the class 'late' is represented by the value 0.

2.2.4. Data normalization

The normalization process aims to standardize the values of the data, ensuring a range between 0 and 1. Data normalization is essential in this study due to the significant variation in feature ranges, making the data more comparable. In addition, data normalization does not significantly increase the memory and processing power requirements [23]. The study used the min-max normalization approach, which normalizes the range of values to 0 and 1. The calculation for the min-max normalization is indicated in (2) [24].

$$x' = \frac{x_i - \min(x)}{\max(x) - \min(x)} \quad (2)$$

In (2), x' represents the result of normalization obtained by normalizing a specific value (x_i) based on the minimum ($\min(x)$) and maximum ($\max(x)$) values of the attribute.

2.3. Splitting data

Data splitting is an important step in machine learning for reliable model evaluation and model generalization. This process involved segmenting the dataset into different subsets for training and testing. The data splitting ratio used in this study was 80:20 for 459 datasets, in which 80% (367 data) of the datasets were used as training data, and the remaining 20% (92 data) were used as testing data.

2.4. Handle imbalanced data

The study dataset had an imbalance issue, resulting in the machine learning model performing better on the majority data than on the minority data. To solve this issue, the K-means SMOTE method had been implemented to create synthetic data in the training set to balance the minority and majority class data. K-means SMOTE consists of three steps: clustering, filtering, and oversampling [25]. Experiments were conducted to explore how the performance of the model was affected by a gradual increase in the minority class sample size. Sample ratios ranging from 13% to 100% of the majority class sample size were used.

2.5. Hyperparameter tuning for K-NN

The K-NN algorithm is non-parametric, meaning that its parameters must be determined before used. To find the best combination of values, hyperparameters should be adjusted through experimentation. The GridSearchCV library was used to perform hyperparameter tuning and test the number of K-neighbors, weighting, and distance measurement metrics in K-NN. Each parameter combination underwent 10 tests using K-fold cross-validation, and the average value was calculated. The K-NN parameters used in the hyperparameter tuning procedure are presented in Table 2.

Table 2. Parameters tested in hyperparameter tuning

Parameters	Values
N Neighbors	1,50,2 (N ranges from 1 to 50, only for the odd numbers)
Weight	Uniform, distance
Metrics	Minkowski, Euclidean, Manhattan

2.6. Process of PFI method

PFI evaluates each feature's contribution to changes in model performance. If randomizing a feature's value doesn't alter the model performance, the feature is deemed insignificant. Conversely, if there's a significant change, the feature is considered impactful. PFI for random forest was introduced in 2001 [26], and based on this concept, a model-agnostic version, known as a dependency model, was later proposed [27]. The following outlines the process of the PFI algorithm.

- 1) Use the K-NN method to predict test data and determine the accuracy or baseline score.
- 2) Select column/feature X_n , and randomize its value.
- 3) Perform prediction on new test data (X_n features that have been randomized).
- 4) Obtain the new accuracy (permuted accuracy).
- 5) Calculate the importance score, $\text{importance score}(F) = \text{Baseline score} - \text{permuted accuracy}$.
- 6) Repeat steps 2-6 for new columns/features and further iterations.

2.7. Testing scheme and model evaluation

To see the effectiveness of the PFI implementation on the K-NN model, two comprehensive test schemes were conducted. First, K-NN was trained with training data without oversampling. Second, K-NN was trained with training data that has undergone oversampling to balance the classes. The oversampling was done by varying the ratio of minority class samples (13% to 100% of the majority class). Each scheme includes hyperparameter tuning to obtain optimal K-NN parameters. The evaluation was conducted and the accuracy score was used as a baseline to calculate the importance score in PFI. The PFI process was performed in 10 iterations with different N-shuffle values (10 to 100, multiples of 10). It was done to see the general pattern of feature importance for the K-NN model. The average importance score of 10 iterations was calculated for each feature, and then the features were sorted in descending order based on the average importance score. In the implementation of PFI for the K-NN model, information was obtained in the form of

a ranking of the contribution of each feature to the K-NN model. To optimize the performance of the K-NN model, tests were conducted by removing some features based on the lowest importance score. Experimentally, the three features with the lowest contribution were selected and all possible combinations or subsets of features were tested. The best results of the evaluation by removing the combination of three features were compared with the K-NN model without PFI. In addition, the effect of the oversampling implementation on the PFI results was also observed in this study, providing a comprehensive overview of the model performance improvement.

In this study, the performance of the K-NN model was evaluated using a confusion matrix, including the accuracy score by (3), the recall by (4), the precision by (5), and the F1-score by (6). This matrix provided a summary of the prediction by comparing the actual value and the prediction result [28], consisting of TP and TN for correct prediction and FP and FN for incorrect prediction. The confusion matrix table in Table 3 provided an assessment of the effectiveness of the K-NN model in predicting the on-time graduation of students at Undiksha PTI program.

Table 3. Confusion matrix

		Predicted value	
		0	1
Actual value	0	True negative (TN)	False positive (FP)
	1	False negative (FN)	True positive (TP)

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

$$\text{Recall score} = \frac{TP}{TP+FN} \tag{4}$$

$$\text{Precision score} = \frac{TP}{TP+FP} \tag{5}$$

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \times 100\% \tag{6}$$

3. RESULTS AND DISCUSSION

The study conducted experiments in two schemes. First, K-NN was trained using training data without oversampling. Second, K-NN was trained using training data oversampled to balance the minority and majority classes. The performance of K-NN in each approach was evaluated, and the accuracy score was used as the baseline score to calculate the importance of each feature in PFI.

3.1. K-NN implementation without oversampling technique

Before applying the PFI method to identify features that did not contribute significantly to the K-NN model, an evaluation of the model was performed using the confusion matrix. The evaluation aimed to obtain an accuracy, precision, recall, and F1-score. Accuracy score was used as the baseline score in the PFI calculation process. In evaluating the efficacy of K-NN in predicting the timely graduation of students in Undiksha PTI program, the prediction results were summarized in the confusion matrix table shown in Table 4. In addition, comprehensive results for the calculated performance of the model are also presented in Table 5.

Table 4. Confusion matrix of K-NN implementation

		Predicted value	
		0 (late)	1 (on-time)
Actual value	0 (late)	81	2
	1 (on-time)	7	2

Table 5. Performance score of K-NN implementation

Measurement	Percentage
Accuracy score	90.22%
Recall score	22.22%
Precision score	50.00%
F1-score	30.77%

The confusion matrix of the K-NN model showed its performance. It accurately predicted class 0 (late) 81 instances, but misclassified it as class 1 (on-time) 2 instances. Conversely, it correctly predicted class 1 (on-time) in 2 instances, but misclassified it as class 0 (late) in 7 instances. The accuracy score was 90.22%, indicating overall correctness. The recall score was 22.22%, representing the proportion of correctly predicted class 1 (on-time) instances. The precision score was 50.00%, indicating the accuracy of predicted class 1 (on-time) instances. The F1-score was 30.77%, combining precision and recall. These results guided the application of PFI, using accuracy as a baseline to determine feature importance scores.

3.2. K-NN and PFI implementation without oversampling technique

In this study, the PFI method was used to assess the importance of different features in predicting on-time graduation of Undiksha PTI students. The PFI process was performed iteratively for 10 iterations with different number of shuffles (10, 20, 30, 40, 50, 60, 70, 80, 90, and 100) in each iteration. Each permuted feature was predicted and evaluated using the confusion matrix to obtain an accuracy score (permuted score) as a result of randomization. Furthermore, the original accuracy (baseline score) was compared with the permuted accuracy to obtain the importance score of the corresponding feature. The importance score was calculated as the average over all iterations. The results are shown in Figure 2, where the features are sorted in descending order of their average importance score, with the most contributing features listed first. This iterative process was intended to provide a more comprehensive understanding of the results.

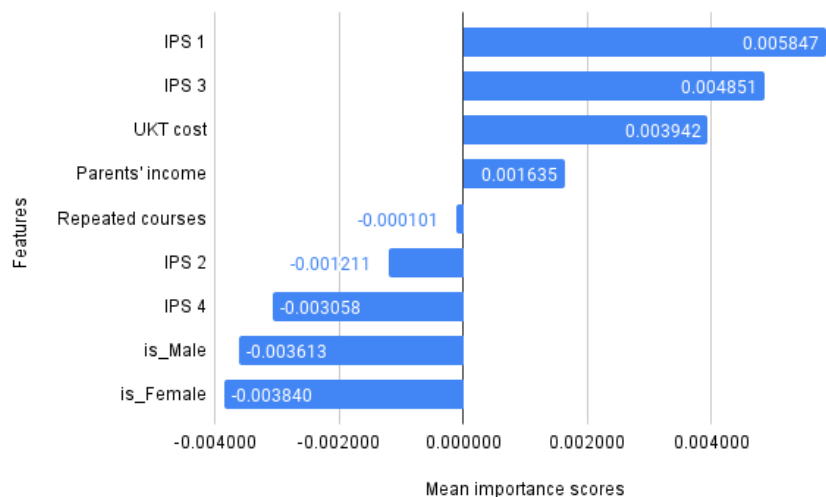


Figure 2. Ranking of feature contributions to the K-NN model

Although the contribution of each feature was known, additional testing was required to determine the number and specific feature with the lowest importance scores that needed to be removed. This was done to optimize the performance of the K-NN model in predicting on-time graduation of the students. In this study, a trial was conducted by selecting three features with the lowest importance scores: is_Female, is_Male, and IPS 4. The test was performed by brute force by systematically removing all possible combinations or subsets of features from the three selected features. This comprehensive method covered all possibilities and allowed to evaluate the performance of the model for each removal combination. Figure 3 showed the test results of the K-NN model without oversampling techniques enhanced with PFI. The results of the seven tests shown in Figure 3(a) indicate that the is_Female, is_Male, and IPS 4 features had no contribution to the K-NN model, and at the same time, by removing the is_Female, is_Male, and IPS 4 features, the performance of the K-NN model could be improved. Figure 3(b) compared the performance of the K-NN model without the PFI process and the K-NN model with the PFI process (removing the is_Female, is_Male, and IPS 4 features).

The first test scheme showed that the use of PFI improved the performance of the K-NN model, resulting in a high accuracy score. However, it was found that the built model was still biased as it struggled to effectively predict minority class data. The unbalanced proportion of training data had falsely inflated the accuracy score, while the confusion matrix and low recall score indicated the model's limitations in predicting minority class data. This was due to the model learning more from the majority class data because

the training data had a highly unbalanced class distribution. In the second testing scheme, an oversampling technique was implemented to address the issue of imbalanced data.

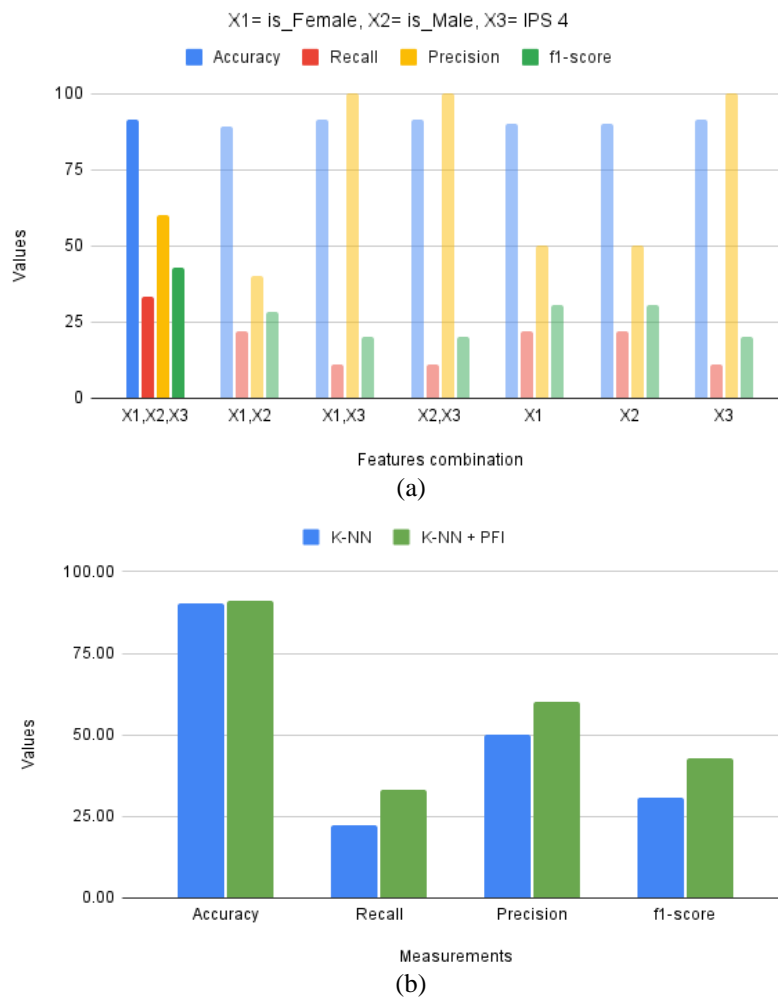


Figure 3. Performance score comparison of (a) K-NN+PFI based on deleted feature combination and (b) K-NN with best K-NN+PFI

3.3. K-NN implementation with oversampling technique

In the second phase of testing, the K-NN model was trained using oversampled training data. In this study, the K-means SMOTE oversampling method was used to create synthetic data for the minority class by employing a ratio of minority class samples ranging from 13% to 100% of the majority class samples. Thus, the K-NN model was tested 88 times with different ratios. The K-NN model was evaluated using a confusion matrix to obtain accuracy, recall, precision, and F1-scores for each ratio. The results of testing the K-NN model and the effects of using training data with different oversampling ratios are shown in Figure 4. The effect of oversampling training data on the accuracy and recall of the model was shown in Figure 4(a). It is evident that the use of oversampling improved the recall scores compared to not using oversampling. This implied that increasing the number of minority class samples in the imbalanced training data improved the model’s ability to predict the minority class data. However, Figure 4(b) showed that increasing the number of minority data also leads to a decrease in the precision scores of the K-NN model.

3.4. K-NN and PFI implementation with oversampling technique

The next step was to improved the performance of the K-NN model for each variation of the oversampling ratio used in the training data by implementing the PFI. A comparison of accuracy, recall, precision, and F1-score results is present in Figure 5 before and after PFI was applied. The baseline, marked with a dotted line, displays the actual K-NN score, while the solid line represents the K-NN score optimized

with PFI. After PFI implementation for each oversampling ratio variation, the K-NN performance was improved. The best model performance was achieved when the minority class samples ratio was 25% of the majority class samples. This resulted in an improvement in the accuracy score from 90.22% to 92.39%, precision from 50.00% to 62.50%, and F1-score from 52.63% to 58.82%, while recall remained consistent at 55.56%. In the model with a minority class ratio of 25%, the K-NN model attained the highest accuracy, with recall, precision, and F1-score being more balanced compared to when using other oversampling ratios. In this scenario, the IPS 1 feature was found to have the lowest contribution to the model and was consequently removed to reduce the dataset’s dimensionality. The IPS 1 feature was also identified as the feature that often received the lowest importance score for different ratio settings.

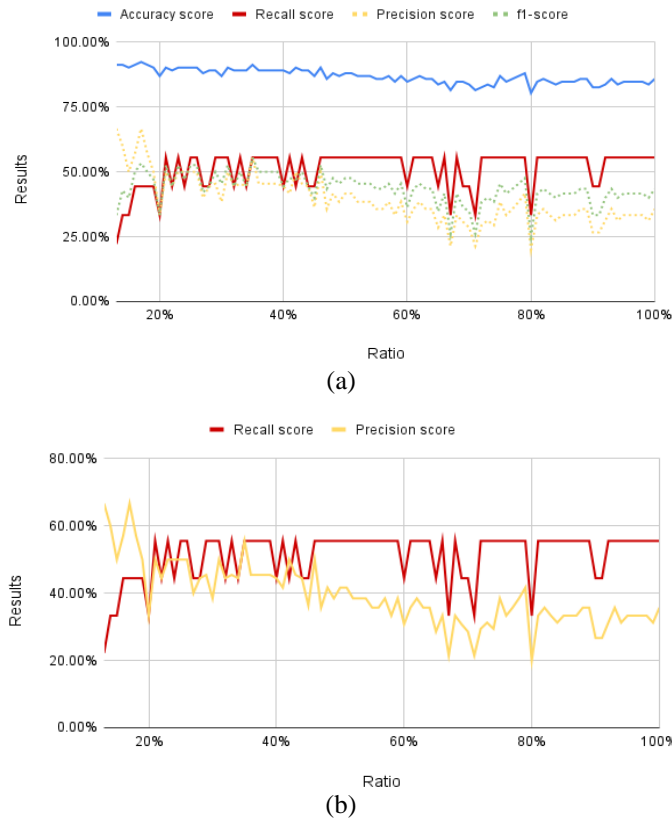


Figure 4. Impact of increasing recall score on (a) accuracy score and (b) precision score

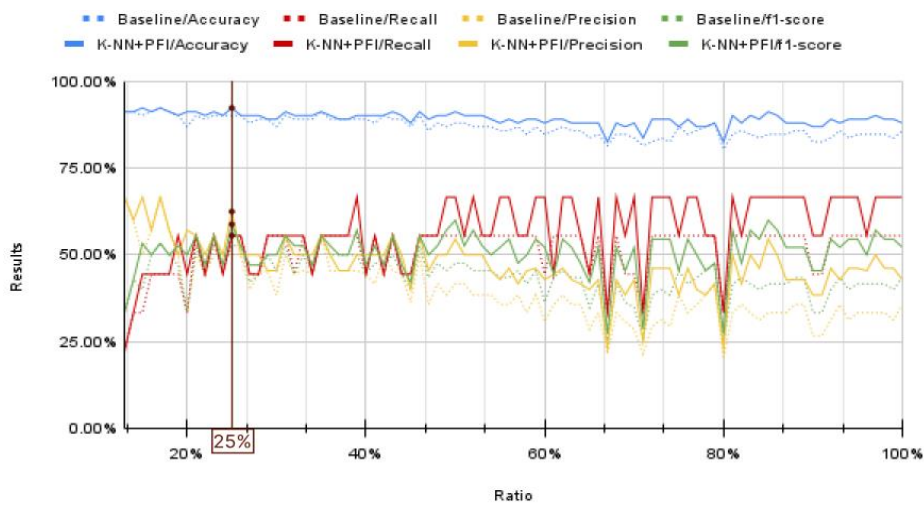


Figure 5. Comparison of baseline and K-NN + PFI measurement results

4. CONCLUSION

Based on the results of the study and discussion above, the implementation of PFI successfully improves the performance of the K-NN model. The best combination between K-NN and PFI was achieved with a ratio of 25%. This resulted in improvements in various evaluation metrics such as accuracy score of 92.39%, precision score of 62.50%, F1-score of 58.82%, and while recall score remained consistent at 55.56%. The IPS 1 feature was identified as having the lowest contribution to the model and was removed to reduce the dimensionality of the dataset. In-depth analysis also showed that PFI could identify features with the lowest contribution, which differ depending on the ratio between the number of samples in the majority and minority classes. The K-NN model in this study showed low recall and precision scores due to the large number of datasets in the “late” class with similar values to those in the “on-time” class. This similarity made accurate predictions difficult for the K-NN model, as it relied on a majority class voting system among nearest neighbors to determine the new data class. The study indicates that achieving a perfect balance between minority and majority class distributions is not always necessary to improve model effectiveness. Further investigation is necessary to understand the impact of different oversampling ratios on model performance for future work. It is also recommended using a comprehensive approach to determine the n-shuffle of PFI based on the number of test data to obtain a more accurate feature contribution pattern.




REFERENCES

- [1] R. Setiawan, A. Aprillia, and N. Magdalena, “Analysis of antecedent factors in academic achievement and student retention,” *Asian Association of Open Universities Journal*, vol. 15, no. 1, pp. 37–47, Mar. 2020, doi: 10.1108/AAOUJ-09-2019-0043.
- [2] Lembaga Akreditasi Mandiri Program Studi Tekniknkan, “Matrix assessment of self-evaluation report and study program performance report (in Indonesian: Matriks penilaian laporan evaluasi diri dan laporan kinerja program studi),” *BAN-PT*, 2021. <https://www.banpt.or.id/peraturan/peraturan-ban-pt/>.
- [3] Hartatik, K. Kusriani, and A. B. Prasetyo, “Prediction of student graduation with naive bayes algorithm,” in *2020 5th International Conference on Informatics and Computing, ICIC 2020*, Nov. 2020, pp. 1–5, doi: 10.1109/ICIC50835.2020.9288625.
- [4] J. G. Perez and E. S. Perez, “Predicting student program completion using Naïve Bayes classification algorithm,” *International Journal of Modern Education and Computer Science*, vol. 13, no. 3, pp. 57–67, Jun. 2021, doi: 10.5815/IJMECS.2021.03.05.
- [5] A. Anggrawan, H. Hairani, and C. Satria, “Improving SVM classification performance on unbalanced student graduation time data using SMOTE,” *International Journal of Information and Education Technology*, vol. 13, no. 2, pp. 289–295, 2023, doi: 10.18178/ijet.2023.13.2.1806.
- [6] D. Y. Putri, R. Andreswari, and M. A. Hasibuan, “Analysis of students graduation target based on academic data record using C4.5 algorithm case study: information systems students of Telkom University,” in *2018 6th International Conference on Cyber and IT Service Management, CITSM 2018*, Aug. 2019, pp. 1–6, doi: 10.1109/CITSM.2018.8674366.
- [7] Laurentinus, O. Rizan, Sarwindah, Hamidah, R. Sulaiman, and P. Fuston, “Data mining using C4.5 algorithm in predicting student graduation,” in *2022 5th International Seminar on Research of Information Technology and Intelligent Systems, ISRITI 2022*, Dec. 2022, pp. 738–743, doi: 10.1109/ISRITI56927.2022.10052793.
- [8] A. M. Olalekan, O. S. Egwuiche, and S. O. Olatunji, “Performance evaluation of machine learning techniques for prediction of graduating students in Tertiary Institution,” in *2020 International Conference in Mathematics, Computer Engineering and Computer Science, ICMCECS 2020*, Mar. 2020, pp. 1–7, doi: 10.1109/ICMCECS47690.2020.240888.
- [9] A. P. Salim, K. A. Laksitowening, and I. Asror, “Time series prediction on college graduation using KNN algorithm,” in *2020 8th International Conference on Information and Communication Technology, ICoICT 2020*, Jun. 2020, pp. 1–4, doi: 10.1109/ICoICT49345.2020.9166238.
- [10] T. Asril, “Prediction of students study period using k-nearest neighbor algorithm,” *International Journal of Emerging Trends in Engineering Research*, vol. 8, no. 6, pp. 2585–2593, Jun. 2020, doi: 10.30534/ijeter/2020/60862020.
- [11] S. C. T. Koumético and H. Toulmi, “Improving KNN model for direct marketing prediction in smart cities,” in *Studies in Computational Intelligence*, vol. 971, 2021, pp. 107–118.
- [12] R. Puspadini, H. Mawengkang, and S. Efendi, “Feature selection on k-nearest neighbor algorithm using similarity measure,” in *MECnIT 2020 - International Conference on Mechanical, Electronics, Computer, and Industrial Technology*, Jun. 2020, pp. 226–231, doi: 10.1109/MECnIT48290.2020.9166612.
- [13] S. Thudumu, P. Branch, J. Jin, and J. (Jack) Singh, “A comprehensive survey of anomaly detection techniques for high dimensional big data,” *Journal of Big Data*, vol. 7, no. 1, p. 42, Dec. 2020, doi: 10.1186/s40537-020-00320-x.
- [14] G. Hu, B. Du, X. Wang, and G. Wei, “An enhanced black widow optimization algorithm for feature selection,” *Knowledge-Based Systems*, vol. 235, p. 107638, Jan. 2022, doi: 10.1016/j.knsys.2021.107638.
- [15] R. A. Khurma, I. Aljarah, A. Sharieh, M. A. Elaziz, R. Damaševičius, and T. Krilavičius, “A review of the modification strategies of the nature inspired algorithms for feature selection problem,” *Mathematics*, vol. 10, no. 3, p. 464, Jan. 2022, doi: 10.3390/math10030464.
- [16] J. Cai, J. Luo, S. Wang, and S. Yang, “Feature selection in machine learning: a new perspective,” *Neurocomputing*, vol. 300, pp. 70–79, Jul. 2018, doi: 10.1016/j.neucom.2017.11.077.
- [17] J. Li *et al.*, “Feature selection: a data perspective,” *ACM Computing Surveys*, vol. 50, no. 6, pp. 1–45, Nov. 2017, doi: 10.1145/3136625.
- [18] M. Christoph, “Interpretable machine learning a guide for making black box models explainable,” in *Book*, 2020, p. 247.
- [19] T. A. Alghamdi and N. Javaid, “A survey of preprocessing methods used for analysis of big data originated from smart grids,” *IEEE Access*, vol. 10, pp. 29149–29171, 2022, doi: 10.1109/ACCESS.2022.3157941.
- [20] M. Alabadla *et al.*, “Systematic review of using machine learning in imputing missing values,” *IEEE Access*, vol. 10, pp. 44483–44502, 2022, doi: 10.1109/ACCESS.2022.3160841.
- [21] J. F. Hair, R. E. Anderson, and R. L. Tatham, “Multi-variate data analysis with readings,” in *Englewood Cliffs, N.J. SE -: Prentice Hall Englewood Cliffs*, 4th ed., 1995.
- [22] K. Potdar, T. S., and C. D., “A comparative study of categorical variable encoding techniques for neural network classifiers,” *International Journal of Computer Applications*, vol. 175, no. 4, pp. 7–9, Oct. 2017, doi: 10.5120/ijca2017915495.




- [23] S. G. K. Patro and K. K. Sahu, "Normalization: a preprocessing stage," *Iarjset*, pp. 20–22, Mar. 2015, doi: 10.17148/iarjset.2015.2305.
- [24] I. Izonin, R. Tkachenko, N. Shakhovska, B. Ilchyshyn, and K. K. Singh, "A two-step data normalization approach for improving classification accuracy in the medical diagnosis domain," *Mathematics*, vol. 10, no. 11, p. 1942, Jun. 2022, doi: 10.3390/math10111942.
- [25] G. Douzas, F. Bacao, and F. Last, "Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE," *Information Sciences*, vol. 465, pp. 1–20, Oct. 2018, doi: 10.1016/j.ins.2018.06.056.
- [26] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.
- [27] A. Fisher, C. Rudin, and F. Dominici, "All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously," *Journal of Machine Learning Research*, vol. 20, 2019.
- [28] D. Valero-Carreras, J. Alcaraz, and M. Landete, "Comparing two SVM models through different metrics based on the confusion matrix," *Computers and Operations Research*, vol. 152, p. 106131, Apr. 2023, doi: 10.1016/j.cor.2022.106131.

BIOGRAPHIES OF AUTHORS






Gd. Aditya Jana Satvika    received the S.T. degree in computer engineering from the Telkom University, Indonesia in 2018. Currently, he is a graduate student in the Department of Computer Science, Universitas Pendidikan Ganesha, Indonesia. His research interests include machine learning and intelligent systems. He can be contacted at email: janasatvika8@gmail.com.



I. N. Sukajaya    is a senior lecture in Computer Science Department of Universitas Pendidikan Ganesha. He achieved Dr. degree at Department of Electrical Engineering-Institut Teknologi Sepuluh Nopember (ITS) in 2017 and his Master degree was achieved in 1999 at Department of Informatics Engineering – Institut Teknologi Bandung. Currently; his research topic areas are: data mining, students profiling, and serious game in education. He is also a reviewer in some qualified international journals. He could be contacted through email address: nyoman.sukajaya@undiksha.ac.id.



I Gede Aris Gunadi    received the B.Sc. degree in Physics (specific field: introductory physics) from the Sepuluh Nopember Institute of Technology Surabaya in 2000. He completed the Master program in the Informatics Engineering study program at the Faculty of Information Technology, Institut Teknologi Sepuluh Nopember Surabaya, in 2008, and a doctoral degree in computer science from Gajah Mada University in 2016. He has 12 years of research, teaching, and community service experience in the field of the Physics Education Department, 4 years in the Master Program of Computer Science, and 1 year in the Mathematics Department at Universitas Pendidikan Ganesha. The main topics of his field are data science, simulation, computation, and electronics. He is responsible for teaching courses in the departments of data science, IoT, soft computing, computer simulation, and electronics. He can be contacted at email: igedearisgunadi@undiksha.ac.id.