Optimizing Dropout Prediction in University Using Oversampling Techniques for Imbalanced Datasets

I Ketut Resika Arthana*, I Made Dendi Maysanjaya, Gede Aditra Pradnyana, and Gede Rasben Dantes

Faculty of Engineering and Vocational, Universitas Pendidikan Ganesha, Singaraja, Bali, Indonesia
Email: resika@undiksha.ac.id (I.K.R.A.); dendi.maysanjaya@undiksha.ac.id (I.M.D.M.); gede.aditra@undiksha.ac.id (G.A.P.); rasben.dantes@undiksha.ac.id (G.R.D.)
*Corresponding author

Manuscript received December 28, 2023; revised February 7, 2024; accepted March 18, 2024; published August 13, 2024

***Abstract*—The phenomenon of student dropout is a significant concern within universities. Institutions must accurately predict the likelihood of student dropout to address this issue effectively. The prediction of student dropout aids universities in identifying early signs of student challenges. Moreover, it enables institutions to implement proactive measures to mitigate dropout rates. This paper presents a novel approach for selecting a classification algorithm to predict student dropout to aid universities in identifying early signs of student dropout. Moreover, it enables institutions to implement proactive measures to mitigate dropout rates. Each university possesses its academic dataset attributes, which can be leveraged for predicting potential dropout cases of student dropout. Our methodology begins with attribute selection, dataset preprocessing, and comparative evaluation of classification algorithms based on priority performance metrics. The research case study is conducted at Universitas Pendidikan Ganesha (Undiksha). The model selection was based on comparing classification algorithm performance, including Naïve Bayesian, Decision Tree (DT), and K-Nearest Neighbors (KNN). The dataset for this research was collected from the Information Academic System of Undiksha, encompassing students who graduated or dropped out between 2013 and 2023. It should be noted that the dataset exhibits class imbalance. Hence, this research utilized the Synthetic Minority Over Sampling Technique (SMOTE) algorithm to address the imbalance in low-sized datasets. The original and oversampled datasets were subjected to each classification algorithm. We chose Recall as the primary evaluation metric to prioritize ensuring that actual dropouts are not incorrectly predicted as graduates. This research demonstrates that the KNN classification algorithm, applied to the oversampled dataset, achieves the highest Recall value of 93.5%, Precision of 94.1%, F1-Score of 93.5%, and AUC value of 97.9%.**

***Keywords*—drop out, oversampling, K-Nearest Neighbors (KNN), decision tree, Naïve Bayesian**

I. Introduction

The phenomenon of student dropout in universities is a concern for top management. Dropout refers to students who discontinue their education without completing their degrees. Students drop out of universities for various reasons, including financial constraints, personal issues, academic difficulties, and lack of support from peers and professors [1, 2]. University dropouts face significant disadvantages in the labor market, which can significantly impact their future opportunities. High dropout rates can damage the university's reputation, leading to implications for its funding and enrollment rates.

Universities must implement significant measures to address the issue of dropout rates in higher education [3, 4]. One approach to addressing this problem involves using machine learning algorithms to predict students who are likely to drop out [5, 6]. The algorithms analyze data from student records such as Grade Point Average (GPA) [7], social economy status, attendance, course grades [8], family background [9], and course completion percentages to identify patterns and indicators of students at risk of academic struggle or potential dropout. Universities can employ machine learning techniques, including classification algorithms, to develop models for predicting student dropout.

Previously, numerous studies have employed machine learning algorithms to forecast student performance and the likelihood of dropping out. A comparison between the classification algorithms KNN and Decision Tree has been conducted by Tariq *et al.* [10]. The result of the experimental evaluation of this research revealed that the accuracy of Decision Tree (DT) is 70%, while that of K-Nearest Neighbors (KNN) was 75%. A limitation of this research is the failure to account for the unbalanced dataset between students dropping out and graduating. According to Gupta *et al.* [11], unbalanced data might lead to biased outcomes and poor machine learning performance.

Another study has been done by Yukselturk [12] using several models to predict dropout students in an online education program. Several models, such as KNN, Decision Tree, Naïve Bayesian, and Neural Network, were compared to determine the best predictive performance. The study found that KNN exhibited the highest sensitivity at 87%, followed by 79.7% for DT, 76.8% for NN, and 73.9% for NB. The dataset comprises 120 graduate students (63.49%) and 69 (36.51%) who dropped out of the program. The imbalanced classes in the dataset have not been addressed as a challenge in this study. Such conditions can lead to poor predictive results, especially in the minority class (dropped-out students) [13].

Various data mining models, including random forest, and logistic regression, Naïve Bayesian Algorithm, JRip, Interpretable Classification Rule Mining (ICRM) [14], Association rules mining, ANN based algorithm, Fuzzy Inference System [15], Logistic Regression, Classification and Regression Trees (CART), C4.5, J48, (BayesNet), SimpleLogistic, Extreme Learning Machine[16] have been utilized to deal with the issue of student dropout. Dropout prediction can also be identified during the learning process in Massive Open Online Courses (MOOCs) [17, 18]. In the Massive Open Online Course (MOOC), the method used to predict whether a student will drop out employs machine learning techniques. This approach analyzes various factors, including the student's engagement in activities like viewing lecture videos and contributing to forum discussions during

the course [19]. Additionally, it considers past data from MOOCs to enhance its predictions [20]. Additionally, data mining techniques can be integrated with qualitative data analysis to pinpoint potential cases of student dropout [21].

Many studies have overlooked the issue of data imbalance, despite the predictive capabilities of these models in predicting student dropout. It is imperative to address this concern to enhance the predicted accuracy of machine learning models [22].

Various data balancing methods have been developed to enhance prediction accuracy in the minority class. Typically, the minority class represents the interest group, and an imbalanced minority class significantly affects prediction outcomes [22]. According to a study by Mduma [23], some balancing techniques were applied to Uwezo and India datasets to predict student dropout better. The SMOTE with the Edited Nearest Neighbors (SMOTE ENN) algorithm has shown the best results for prediction compared to other algorithms, such as Random Under Sampling, Random Over Sampling, SMOTE with Tomek, Synthetic Minority Over Sampling, and also Synthetic Minority Over Sampling Technique (SMOTE) with Edited Nearest Neighbors (ENN) [24, 25].

The characteristic dataset of dropout students in Undiksha was similar to most datasets in other universities, with features such as low sample size and an imbalanced dataset between dropout students and graduated students classes. To address this problem, an oversampling algorithm will be applied to the low-sized data to generate more optimal prediction results. This research utilizes the Synthetic Minority Over Sampling Technique (SMOTE) algorithm to generate sampling data in low-sized classes. The algorithm helps generate new sampling data by combining examples from targeted classes [23]. Later, several classification models will be applied to the datasets, with or without oversampling, to find the best performance results.

The selection of performance evaluation indicators in classification should be aligned with the problem's priority. In the context of predicting Potential Dropout Students, we establish the following priority statement: "It is preferable to predict a student as a potential dropout when they are not, rather than failing to predict a dropout when they are." Therefore, minimizing false negatives becomes paramount. Consequently, we prioritize recall as a performance evaluation metric, as higher recall scores correlate with lower false negative rates. However, other indicators such as Precision, Accuracy, F1-Score, and Area under the ROC Curve (AUC ROC) remain under consideration.

This study proposes a novel method for predicting university dropout rates using oversampling techniques and machine learning algorithms. Several machine learning methods are evaluated to identify the optimal prediction model, including Decision Trees, K-Nearest Neighbours (KNN), and Naïve Bayes. To select the optimal algorithm, we initially collected the dataset from the academic information system at Undiksha. Subsequently, we duplicated this dataset into two versions: the first comprising the original dataset with imbalanced data and the second involving the dataset with the SMOTE algorithm applied to oversample the low-sized data. Each dataset underwent training using a classification algorithm: Decision Trees, K-Nearest Neighbors (KNN), and Naïve Bayes. Following this, we assessed performance metrics, including Recall, Accuracy, Precision, F1-Score, and AUC ROC.

Every university has a dataset that can predict student dropout rates. The primary contribution of this paper lies in elucidating the process of attaining the optimal model for dropout prediction, coupled with introducing a novel approach to preprocessing data and delineating priority evaluation metrics.

II. Methodology

This study employs the steps of the Cross Industry Standard Process for Data Mining (CRISP-DM). CRISP-DM is the methodology that widely accepted standard process model for implementing data mining initiatives across many industries [26]. As depicted in Fig. 1, the CRISP DM methodology consists of (1) Business Understanding, (2) Data Understanding, (3) Data Preparation, (4) Modelling, (5) Evaluation, and (6) Deployment.

Fig. 1. Process model for dropout prediction with CRISP-DM.

A. Business Understanding

A university dropout student is a student who discontinues their academic program before completing it. Dropout rates can be influenced by various factors such as financial constraints, academic challenges, personal issues, lack of motivation, and insufficient support.

Addressing the issue of student dropout is urgent due to its adverse effects on both individuals and society. From an individual perspective, dropping out can result in wasted time and resources and negatively impact future career prospects. On a broader societal scale, it contributes to a decline in the overall educational attainment of the population, thereby impeding economic development and limiting social mobility.

Anticipating potential dropout cases is crucial for institutions to identify students at risk of leaving and provide targeted interventions to prevent it. Predictive analytics can analyze academic achievement, attendance, and engagement data to identify patterns indicative of a student's likelihood of dropping out. This study empowers universities to provide proactive support and resources to help students overcome obstacles, increasing their likelihood of completing their studies. University data can be leveraged to identify potential dropouts. Undiksha can utilize its existing academic and financial data to forecast students who may be at risk of dropping out.

B. Data Understanding

The dataset was gathered from the Information Academic System at Universitas Pendidikan Ganesha (Undiksha), comprising records of 17,904 students from 2013 to 2023. It contains attributes such as student ID, name, program of study, GPA, school of origin, type of student tuition, admission type for new students, and graduation status. Explanation of data attributes is described in Table 1.

Table 1. Data attribute description

Attribute	Description
NIM	Student Id
Name	Name of students
Entry Year	Entry year of students
Program Study	Program Study of Students
CGPA	Cumulative Grade Point Average
School Origin	**Name of previous high school**
Student Tuition Type	In Undiksha, the student tuition type consists of KIPK/Bidikmisi, Student Tuition Type 1, and Student Tuition Type 7.
New Student Admission Pathway	In Undiksha, students can register to Undiksha via pathway SNMPTN (based on performance in senior high school), SBMPTN (based on score test held by the Ministry of Education and Culture), and Local Test (Test by Undiksha)
Status	Status of Students, Graduated or Dropped Out.

C. Data Preparation

Based on the distribution dataset between Graduated and Dropped Out Students (Fig. 2), there is a data imbalance between classes. The dataset has 14,983 (84%) records of graduated students and 2,921 (16%) records of dropout students. The class data imbalance in classification can pose challenges in making accurate predictions. This problem stems from bias in the classification model, which tends to prioritize predicting the majority class over the minority class. To mitigate this issue, we employ oversampling methods to balance the sample sizes in the minority class before classification.

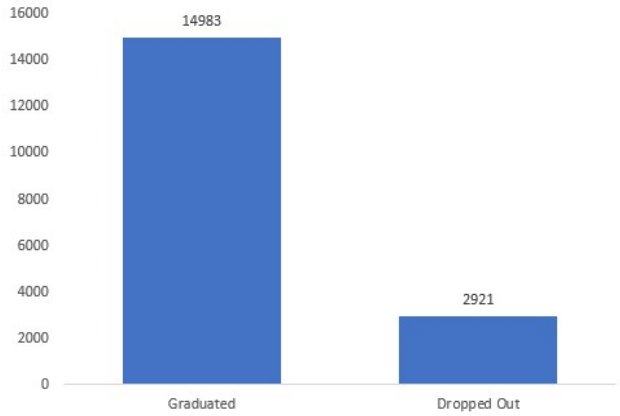


Fig. 2. Distribution dataset between graduated and dropped out students.

The Synthetic Minority Over-sampling Technique (SMOTE) is a sampling technique used to address class imbalance issues in classification tasks. The SMOTE algorithm generates artificial samples for the underrepresented class to achieve a balanced dataset. This study will employ the SMOTE method to create artificial samples for the minority class. The Synthetic Minority Over-Sampling Technique (SMOTE) algorithm follows these steps:

1) Select a minority class sample that needs to be oversampled.
2) Determine the k-Nearest Neighbors for the chosen minority sample.
3) Choose one of the K-Nearest Neighbors and generate a synthetic sample by taking a linear combination of the selected minority sample and the chosen neighbor.
4) Repeat the previous step until the desired number of synthetic samples has been generated.
5) The oversampling level can be modified by specifying the preferred ratio of minority class instances to majority class instances in the resulting dataset.

After applying the SMOTE algorithm to the dataset, we have 14,983 instances of Graduated Students and 14,983 instances of Drop Out Students (Fig. 3).

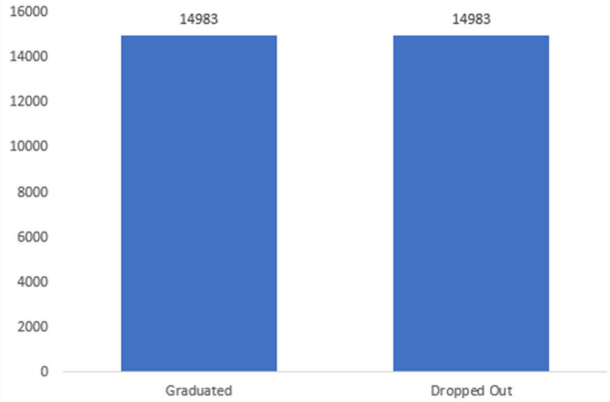


Fig. 3. Balanced dataset after apply smote.

D. Modelling

This study evaluates performance model classification to predict student drop out using metrics such as Accuracy, Precision, Recall, F1-Score, and Area under the ROC Curve (AUC ROC).

The primary priority performance metric is recall since we focus on reducing false negative numbers. The classification models used in this research are Naïve Bayes, Decision Tree, and K-Nearest Neighbors (KNN).

1) Classification algorithm

a) Naïve Bayesian

Naive Bayes is a probabilistic method used in machine learning for classification. Naive Bayes assumes that the attributes are conditionally independent given the class, meaning that the presence or lack of a feature does not influence the presence or absence of other features. This assumption is vital but only sometimes feasible in real-world situations. It simplifies the calculation of probabilities and allows for efficient training and prediction [27].

The Naive Bayes classifier is a mathematical method that uses probabilistic calculations to determine the most suitable classification for a specific piece of data in a given problem area. This classifier can function as a versatile toolkit suitable for a wide range of classification tasks across various domains [28].

Naive Bayes is a simple yet powerful probabilistic machine learning method for categorization tasks. The Bayes theorem assumes that the features are conditionally independent given the class. The Bayes theorem assumes that the features are

conditionally independent given the class. The algorithm calculates the posterior probability of a class by considering the prior probability of the class and the likelihood of the features supplied to the class. The probability can be expressed using either the Gaussian or multinomial distribution, and the prior probability can be derived from the training data or assumed to follow a uniform distribution.

b) Decision tree

The decision tree is a commonly employed and easily comprehensible machine-learning method for classification and regression tasks. The system creates a hierarchical model illustrating decisions and their potential outcomes. Due to its simplicity, interpretability, and effectiveness, decision tree algorithms are often used in various fields, such as finance, medicine, and engineering.

The construction of a decision tree in machine learning is based on information entropy. Entropy is a measure of the randomness or uncertainty in a dataset. The decision tree algorithm selects the feature that most effectively divides the data into similar groups by maximizing either the information gain or the gain ratio. Information gain represents the decrease in entropy from splitting the data based on a specific property.

In conclusion, a decision tree is an algorithm that constructs a tree-like model of decisions and their consequences based on information entropy. The algorithm iteratively selects the attribute that best splits the data into homogeneous subsets based on the information gain or the gain ratio. Decision trees are frequently utilized in diverse fields because of their simplicity, interpretability, and effectiveness in classification and regression tasks.

c) K-Nearest Neighbour (KNN)

K-Nearest Neighbour (KNN) is a fundamental non-parametric classification method applicable to classification and regression tasks. KNN assigns a class to a data point by considering the classes of its nearest neighbours in the feature space. Before training the model, it is essential to determine the number of neighbours (k) as a hyperparameter in KNN.

In KNN, to classify a new data point, the algorithm calculates the distance between the data point and all other data points in the training set. KNN offers several advantages, including its simplicity of understanding and implementation and its lack of need for a training process. However, it does have limitations, such as sensitivity to the selection of the distance measure and the value of k, as well as underperforming on high-dimensional data. In practice, KNN is frequently employed as a baseline algorithm for comparison with more complex models.

K-Nearest Neighbors (KNN) is a straightforward and intuitive machine-learning technique suitable for classification and regression problems. The fundamental concept involves categorizing a data point according to the class or value of its closest neighbours in the feature space. The approach calculates the distance between a data point and all other data points in the training set. Subsequently, it identifies the k-nearest neighbours to the new data point for classification. The effectiveness of KNN is influenced by the selection of the distance measure and the value of k, which makes it a common choice as a baseline approach in machine learning studies.

2) Modelling process

As shown in Fig. 4, The initial step starts with selecting the dataset from the Information Academic System at Undiksha. We created two versions of the dataset: the first version without oversampling and the second with oversampling. We use the SMOTE algorithm to oversample low-sized datasets. Each dataset is then applied to each classification model. Specifically, we employ Naïve Bayes, Decision Tree, and KNN as classification models.

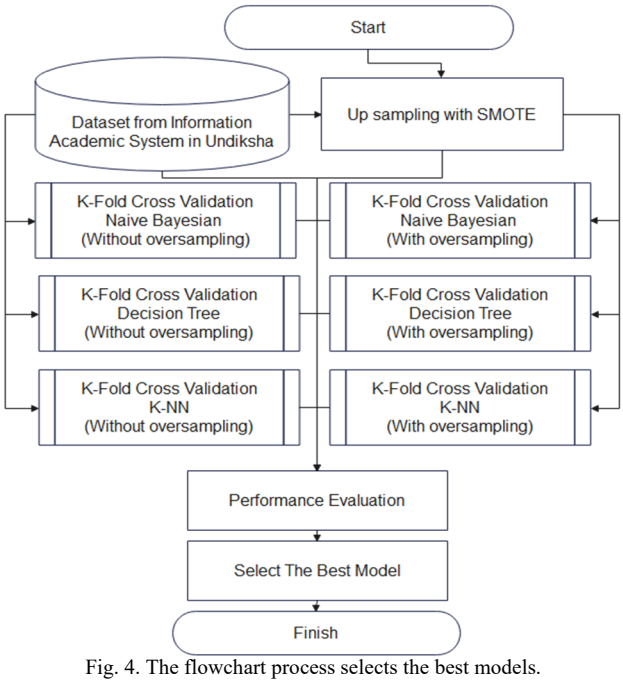


Fig. 4. The flowchart process selects the best models.

This study employs K-Fold Cross-Validation to evaluate the model's performance. We chose the value of K = 10 because 10-fold cross-validation is commonly used in applied machine learning. In 10-fold cross-validation, the dataset is divided into K subsets, with K-1 subsets used for training and the remaining subset for evaluation in each iteration. In 10-fold Cross-Validation, the dataset is divided into 10 roughly equal-sized parts (folds). The model is trained with 9 folds, and the remaining fold is used for testing. This process is repeated 10 times with a different fold reserved for testing. The evaluation of the results obtained through 10-fold cross-validation is discussed in the Evaluation section.

This study employed the Python programming language and its Scikit-learn (SKlearn) library for data preprocessing, classification, and evaluation. Data underwent initial cleaning and preprocessing in the preprocessing stage using various SKlearn modules, including feature selection, one-hot encoding, and oversampling. Subsequently, several classification models, including KNN, decision trees, and Naïve Bayes, were trained and evaluated on the preprocessed data using cross-validation techniques. Finally, the performance of the models was assessed using standard evaluation metrics, such as accuracy, precision, recall, F1-Score, and AUC-ROC. Python and its associated libraries facilitated efficient and effective implementation of this study's data preprocessing, classification, and evaluation pipelines.

Fig. 5 shows the code in Python to classify data with the KNN Algorithm. Then, we select evaluation metrics and

assess the algorithm with cross-validation. We employ 10-fold cross-validation. The evaluation metrics, namely accuracy, precision, recall, F1-Score, and AUC, are displayed based on the cross-validation results. We apply the same method in Naïve Bayesian and Decision Tree. The code for the oversampling dataset shown in Fig. 6.

```
def c_KNN(X, y):
  knn = KNeighborsClassifier(n_neighbors=5)
  scoring = ['accuracy', 'precision_macro', 'recall_macro', 'f1_macro', 'roc_auc']
  cv_results = cross_validate(knn, X, y, cv=10, scoring=scoring, error_score='raise')
  print('Accuracy:', cv_results['test_accuracy'].mean())
  print('Precision:', cv_results['test_precision_macro'].mean())
  print('Recall:', cv_results['test_recall_macro'].mean())
  print('F1 Score:', cv_results['test_f1_macro'].mean())
  print('AUC:', cv_results['test_roc_auc'].mean())
```

Fig. 5. Code in python for the classification with KNN, use cross-validation, and show evaluation performance.

```
from imblearn.over_sampling import SMOTE
smote = SMOTE(sampling_strategy='minority')
X.columns = X.columns.astype(str)
X_resampled, y_resampled = smote.fit_resample(X, y)
```

Fig 6. Code python for SMOTE oversampling.

E. Evaluation

This study utilizes five parameters to estimate the performance of classification. The examined attributes encompass accuracy, Precision, recall, F1-Score, and AUC ROC. Accuracy is a metric that quantifies the degree to which projected outcomes align with the actual results, particularly in scenarios where the data is evenly distributed over multiple classes. However, due to the disproportionate distribution of data in our dataset, the assessment of classifier performance will rely more on precision and recall metrics [29, 30]. The F1-Score is calculated by taking the harmonic mean of Precision and Recall, offering a balanced evaluation of both measurements. The Area Under the Curve of the Receiver Operating Characteristic (AUC ROC) is a commonly utilized performance measure in binary classification tasks. The assessment metric evaluates the model's ability to distinguish between positive and negative events by computing the Area Under the Receiver Operating Characteristic (ROC) curve. The ROC curve illustrates how the True Positive Rate (TPR) and False Positive Rate (FPR) change at various threshold levels. Eqs. (1)–(5) sequentially calculate accuracy, precision, recall, F1-Score, and AUC ROC.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

$$AUC\ ROC = \int_0^1 TPR\left(FPR^{-1}(x)\right) dx \quad (5)$$

F. Deployment

The best-performing classification model identified through the comparison was deployed in the Academic Information System and used real-world data to predict potential dropout students. Subsequently, an Application Programming Interface (API) was developed, accessible from the Executive Information System. This feature enables top management to identify students at risk of dropping out.

III. Results and Discussion

The performance evaluation from classification models is shown in Table 2. The average evaluation metric value is high overall, except for Naïve Bayesian, which exhibits a significantly lower value than other algorithms.

Table 2. Summary performance evaluation

Algorithm	Acc	Prec.	Recall	F1	AUC
Without oversampling					
Decision Tree	0.903	0.847	0.893	0.853	0.893
Naïve Bayesian	0.834	0.417	0.5	0.455	0.593
KNN	**0.963**	**0.970**	**0.896**	0.925	0.931
With oversampling					
Decision Tree	0.917	0.927	0.917	0.916	0.917
Naïve Bayesian	0.536	0.537	0.536	0.490	0.594
KNN	0.935	0.941	**0.935**	**0.935**	**0.979**

A. Comparison Classification Algorithm in a Dataset without Oversampling

This research utilized two datasets to compare prediction performance. Performance evaluation of the original dataset (imbalanced and without oversampling) indicates that the KNN algorithm outperforms other algorithms, exhibiting higher evaluation results across all metrics. The evaluation results suggest that the academic dataset of Undiksha is compatible with the KNN algorithm, highlighting the ability of the KNN model to handle large and noisy datasets effectively [31]. In the KNN classification model without oversampling, it is notable that recall has the lowest value (0.896%) compared to other metrics, including accuracy (96.3%), precision (97.9%), F1-Score (92.5%), and AUC (93.1%).

Following these results, the Decision Tree algorithm also demonstrates high evaluation results and closely approximates the performance of the KNN classification model (with an accuracy of 90.3%). In contrast, the Naïve Bayesian classification algorithm exhibits the lowest evaluation results across all metrics compared to other classification algorithms. This suggests that the probability-based model is unsuitable for classifying dropout or non-dropout cases in Undiksha's dataset. These results may be attributed to the characteristic of Naïve Bayes itself, as it assumes that each attribute is independent given the class [32]. Naïve Bayes ignores the relations between attributes and focuses on the relationship between attribute and class variable only.

B. Comparison Classification Algorithm in the Dataset with Oversampling

In the second version of datasets with the SMOTE

oversampling technique applied, all Decision Tree model evaluation metrics improved compared to the results without oversampling. Similarly, the Naïve Bayes model exhibited an increase in all evaluation metrics, except for accuracy, which notably decreased from 0.834 to 0.536. On the other hand, the KNN algorithm experienced a decrease in accuracy (from 0.963 to 0.935) and precision (from 0.970 to 0.941) while witnessing an increase in recall (from 0.896 to 0.935), F1-Score (from 0.925 to 0.935), and AUC (from 0.931 to 0.979) metrics.

Overall, the performance evaluation scores of all algorithms demonstrated improvement. The recall values for all three algorithms increased when compared to the recall values of the data without oversampling, indicating the efficacy of the oversampling technique in reducing false negative values. The KNN model maintains the highest scores across all evaluation metrics, followed by the Decision Tree and Naïve Bayes with the lowest scores.

This study found that using SMOTE improved recall by reducing false negatives, consistent with previous research [33], that the impact of SMOTE can vary across different algorithms and datasets. While SMOTE may improve classification accuracy, it can also introduce errors. The positive effects of SMOTE are particularly noticeable in algorithms like KNN and Decision Trees. However, despite a marginal improvement in recall, Naive Bayesian has shown a significant decline in Precision, Accuracy, and F1-Score.

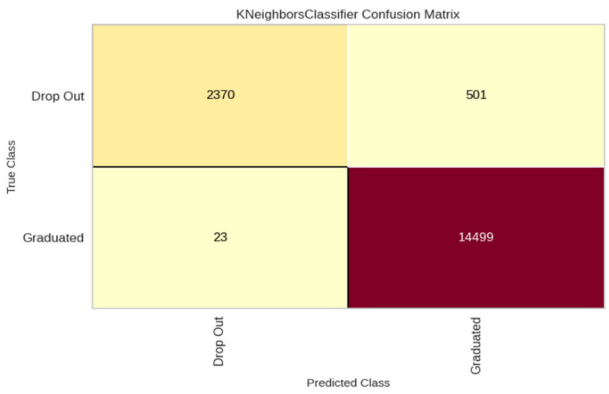
C. KNN Oversampling vs KNN Non-oversampling

The KNN model exhibits the highest performance in both datasets compared to other models. However, as depicted in Table 3, the accuracy and Precision values were higher in the classification without oversampling. Conversely, Recall, F1, and AUC metrics demonstrated superior performance in classification with the applied SMOTE technique.

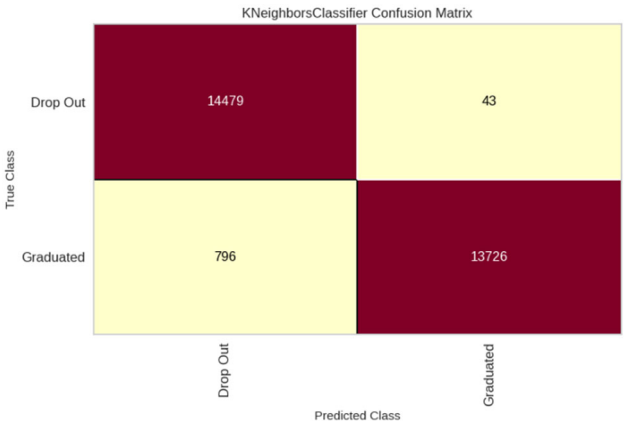
Table 3. KNN algorithm performance evaluation

Algorithm	Acc	Prec.	Recall	F1	AUC
Without oversampling	**0.963**	**0.970**	0.896	0.925	0.931
With oversampling	0.935	0.941	**0.935**	**0.935**	**0.979**

KNN models augmented with the SMOTE oversampling technique exhibit a notable reduction in false negative instances (where the actual status is dropout but predicted as graduated). Prediction using the original dataset reveals 501 false negatives (Fig. 7(a)), which decreases to 43 using the SMOTE technique (Fig. 7(b)).



(a)



(b)

Fig. 7. Confusion matrix KNN (a) without oversampling; (b) with oversampling.

KNN model with applied SMOTE oversampling technique shows the best performance evaluation results compared to another model. It also has the highest recall value (main priority performance). This approach successfully reduces false negative numbers from 501 to 43. The chosen model presented in this study can be used to accurately predict students at risk of dropping out and needing early intervention.

In imbalanced datasets, the SMOTE algorithm applied to KNN showed positive performance, similar to the results found in research on the diagnosis of diabetes [34]. The study showed that implementing SMOTE with KNN improved accuracy by 8.25%.

D. Selected Model for Dropout Prediction

The KNN model, augmented with the SMOTE oversampling technique, demonstrates superior performance evaluation results compared to alternative models. Additionally, it achieves the highest recall value, prioritizing performance. This methodology effectively mitigates false negative instances, reducing them from 501 to 43. The selected model showcased in this study holds promise for accurately identifying students at risk of dropout and facilitating timely interventions.

E. Learning Curve to Identify Overfitting

As depicted in Fig. 8, the learning curve showcases the relationship between the training and validation accuracy of a K-Nearest Neighbors (KNN) model utilized to predict student dropout rates.

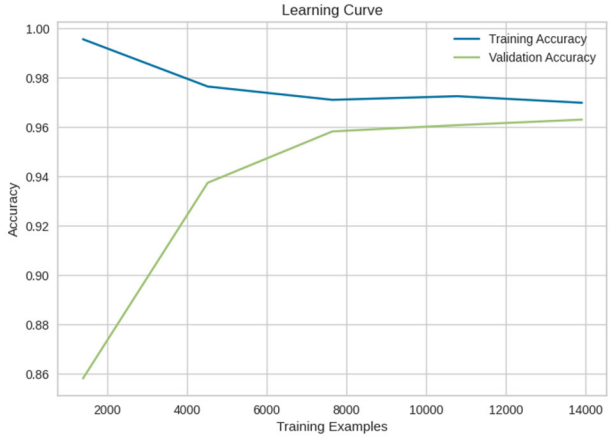


Fig. 8. Learning curve KNN with oversampling.

As the number of training examples increases, the training accuracy slightly decreases while the validation accuracy increases, as depicted by the curve. Initially, with fewer training cases, the training accuracy remains high, while the validation accuracy is notably lower. This disparity suggests potential model overfitting to the training data, implying that the model memorizes the data rather than discerning the underlying patterns.

Increasing the number of training examples results in a slight decrease in training accuracy, suggesting that the model enhances its ability to generalize rather than memorize the training data. Consequently, the validation accuracy shows improvement, signifying that the model's predictions are becoming more accurate on unseen data.

When the validation accuracy remains constant and aligns with the training accuracy as the number of training examples increases, increasing the amount of data enhances the model's ability to generalize. If the validation accuracy stays the same or improves as more data is added, the model demonstrates resilience and does not suffer from overfitting.

In line with the insights provided by Parmezan *et al.* [35], the importance of selecting an appropriate model for forecasting in the context of temporal data can be extended to the domain of predicting student dropouts. Models that exhibit strong generalization capabilities across diverse datasets are paramount for effectively identifying students at risk of dropping out. In predicting dropouts, the model must generalize effectively across various datasets to identify at-risk students accurately. A stable or increasing validation accuracy with additional data suggests that the model will perform consistently in real-world scenarios. The learning curve illustrates that the KNN model for predicting dropout is undergoing fine-tuning and is anticipated to provide precise predictions on new data.

IV. Discussion

The field of predicting student dropout potential has been extensively explored, presenting unique challenges ranging from determining data attributes and model selection to addressing the imbalance between students who drop out and those who do not, as well as the metrics used for selecting the optimal model. The selection of various attributes to predict dropouts has been the subject of numerous studies. However, which features may be effectively utilized alongside various machine learning classifiers for forecasting student dropout remains to be seen [36]. This work's limitation is using existing attributes from our academic information system, such as Entry Year, Program of Study, CGPA, School Origin, Student Tuition Type, and New Student Admission Pathway. The disparity in attributes used among studies predicting dropout potential renders direct comparisons between the models used in this research and those in existing studies unfeasible. This addition enriches the discussion by acknowledging the data's specificity and implications for the study's comparability and generalizability. It also sets a clear direction for future research, emphasizing the importance of attribute selection in improving dropout prediction models. The issue of dataset imbalance has yet to be addressed in prior research. However, it is a common phenomenon in higher education that the data on students who drop out and those who do not will significantly differ, where an imbalanced dataset can impact the performance of classification algorithms and potentially lead to model overfitting [11]. In this study, addressing the imbalanced dataset with the Synthetic Minority Over-sampling Technique (SMOTE) has enhanced model performance. Moreover, it is crucial to determine a priority metric when selecting the best model. Previous studies [6, 7, 15, 36–39] have not explicitly mentioned the priority metric for determining the best model. In this research, the best model is chosen based on the highest Recall value to avoid misclassifying actual dropouts as non-dropouts (False Positive), thus ensuring that at-risk students receive appropriate attention from the university. Rarely do other studies assess whether the chosen model is prone to overfitting. A good-performance model must be evaluated for potential overfitting [35]. This study employs a Learning Curve to ascertain whether the model exhibits overfitting. For Universitas Pendidikan Ganesha, the K-Nearest Neighbors (KNN) algorithm, coupled with SMOTE for handling imbalanced datasets, emerged as the most suitable classification algorithm based on available attributes and data characteristics.

V. Conclusions

In this study, we compared three classification algorithms, namely Naive Bayesian, Decision Tree, and kNN, to predict students at risk of dropping out. We applied the SMOTE algorithm to oversample low-sized datasets to address imbalanced data. We used each dataset for every classification algorithm without and with oversampling. Our primary priority was the recall evaluation performance, chosen to ensure that actual dropouts are not incorrectly predicted as graduates, thus reducing false negatives.

Overall, the KNN Algorithm exhibits the highest performance compared to Decision Tree and Naive Bayesian, both with and without SMOTE oversampling. Notably, KNN with SMOTE oversampling demonstrates superior recall performance compared to KNN without oversampling. Consequently, we conclude that the KNN algorithm with oversampling is the optimal model for predicting dropout students. SMOTE enhances the model's performance by mitigating class imbalance and reducing bias towards the majority classes.

The limitation of this study is the inherent trade-off between increasing recall and managing the rise in false positives, particularly evident in the outcomes of the oversampled dataset. The other limitation is using existing attributes from our academic information system, such as Entry Year, Program of Study, CGPA, School Origin, Student Tuition Type, and New Student Admission Pathway. Hence, the results cannot be compared with other research because of different attributes.

Future research should explore the importance of attribute selection in improving dropout prediction models and additional oversampling techniques and hybrid models to refine the balance between recall, precision, and accuracy further. The impact of integrating more diverse and comprehensive datasets to enrich the training process could provide insights into enhancing model robustness.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

I Ketut Resika Arthana, conducted data collection, experiment, and article writing; I Made Dendi Maysanjaya and Gede Aditra Pradnyana contributed to article writing; and Prof. Gede Rasben Dantes gave reviews and comments. All authors had approved the final version.

REFERENCES

[1] M. A. Aziz et *al.*, "Comparison of K-Medoids algorithm with K-Means on number of student dropped out," in *Proc. 2022 1st International Conference on Smart Technology, Applied Informatics, and Engineering (APICS)*, IEEE, Aug. 2022, pp. 53–58. doi: 10.1109/APICS56469.2022.9918789

[2] M. Revathy, S. Kamalakkannan, and P. Kavitha, "Machine learning based prediction of dropout students from the education university using SMOTE," in *Proc. 2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, IEEE, Jan. 2022, pp. 1750–1758. doi: 10.1109/ICSSIT53264.2022.9716450

[3] M. Utari, B. Warsito, and R. Kusumaningrum, "Implementation of Data mining for drop-out prediction using random forest method," in *Proc. 2020 8th International Conference on Information and Communication Technology (ICoICT)*, IEEE, Jun. 2020, pp. 1–5. doi: 10.1109/ICoICT49345.2020.9166276

[4] M. Sharma and M. Yadav, "Predicting students' drop-out rate using machine learning models: A comparative study," in *Proc. 2022 Third International Conference on Intelligent Computing Instrumentation and Control Technologies (ICICICT)*, IEEE, Aug. 2022, pp. 1166–1171. doi: 10.1109/ICICICT54557.2022.9917841

[5] C. E. Aguirre and J. C. Perez, "Predictive data analysis techniques applied to dropping out of university studies," in *Proc. 2020 XLVI Latin American Computing Conference (CLEI)*, IEEE, Oct. 2020, pp. 512–521. doi: 10.1109/CLEI52000.2020.00066

[6] L. Kemper, G. Vorhoff, and B. U. Wigger, "Predicting student dropout: A machine learning approach," *European Journal of Higher Education*, vol. 10, no. 1, pp. 28–47, 2020. doi: 10.1080/21568235.2020.1718520

[7] L. Aulck, N. Velagapudi, J. Blumenstock, and J. West," Predicting student dropout in higher education," *arXiv Reprint*, arXiv:1606.06364, 2016.

[8] S. Rovira, E. Puertas, and L. Igual, "Data-driven system to predict academic grades and dropout," *PLoS One*, vol. 12, no. 2, 2017. doi: 10.1371/journal.pone.0171207

[9] M. Goga, S. Kuyoro, and N. Goga, "A recommender for improving the student academic performance," *Procedia Soc Behav Sci*, vol. 180, pp. 1481–1488, 2015. doi: 10.1016/j.sbspro.2015.02.296

[10] Tariq, A. Amin, Y. Masood, M. Muzaffar, and J. Iqbal, "Predicting early withdrawal of university students: A comparative study between KNN and decision tree," in *Proc. 2023 4th International Conference on Advancements in Computational Sciences (ICACS)*, IEEE, Feb. 2023, pp. 1–7. doi: 10.1109/ICACS55311.2023.10089706

[11] P. Gupta, A. Varshney, M. R. Khan, R. Ahmed, M. Shuaib, and S. Alam, "Unbalanced credit card fraud detection data: A machine learning-oriented comparative study of balancing techniques," *Procedia Comput Sci*, vol. 218, pp. 2575–2584, 2023. doi: 10.1016/j.procs.2023.01.231

[12] E. Yukselturk, "Predicting dropout student: An application of data mining methods in an online education program," *European Journal of Open, Distance and e-Learning*, vol. 17, no. 1, 118, 2014.

[13] R. Blagus and L. Lusa, "Joint use of over-and under-sampling techniques and cross-validation for the development and assessment of prediction models," *BMC Bioinformatics*, vol. 16, no. 1, 2015. doi: 10.1186/s12859-015-0784-9

[14] M. Kumar, A. J. Singh, and D. Handa, "Literature survey on educational dropout prediction," *International Journal of Education and Management Engineering*, vol. 7, no. 2, pp. 8–19, 2017. doi: 10.5815/ijeme.2017.02.02

[15] A. Saranya and J. Rajeswari, "Enhanced prediction of student dropouts using fuzzy inference system and logistic regression," *ICTACT Journal on Soft Computing*, vol. 2, 2016. doi: 10.21917/ijsc.2016.0161

[16] J. Chen, "MOOC dropout prediction using a hybrid algorithm based on decision tree and extreme learning machine," *Math Probl Eng*, vol. 2019, 2019. doi: 10.1155/2019/8404653

[17] F. Dalipi, "MOOC dropout prediction using machine learning techniques: Review and research challenges," in *Proc. IEEE Global Engineering Education Conference*, 2018, vol. 2018, pp. 1007–1014. doi: 10.1109/EDUCON.2018.8363340

[18] J. Chen, "A systematic review for MOOC dropout prediction from the perspective of machine learning," *Interactive Learning Environments*, 2022. doi: 10.1080/10494820.2022.2124425

[19] M. Fei and D.-Y. Yeung, "Temporal models for predicting student dropout in massive open online courses," in *Proc. 2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, 2015, pp. 256–263. doi: 10.1109/ICDMW.2015.174

[20] E. B. M. Magalhaes, "Student dropout prediction in MOOC using machine learning algorithms," in *Proc. 2021 Workshop on Communication Networks and Power Systems, WCNPS 2021*, 2021. doi: 10.1109/WCNPS53648.2021.9626227

[21] S. Durso and J. V. A. Cunha, "Determinant factors for undergraduate student's dropout in an accounting studies department of a Brazilian public university," *Educação em Revista*, vol. 34, 2018. doi: 10.1590/0102-4698186332

[22] López, A. Fernández, S. García, V. Palade, and F. Herrera, "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics," *Inf Sci (N Y)*, vol. 250, 2013. doi: 10.1016/j.ins.2013.07.007

[23] N. Mduma, "Data balancing techniques for predicting student dropout using machine learning," *Data (Basel)*, vol. 8, no. 3, 2023. doi: 10.3390/data8030049

[24] G. Douzas and F. Bacao, "Geometric SMOTE: Effective oversampling for imbalanced learning through a geometric extension of SMOTE," *arXiv Reprint*, arXiv:1709.07377, 2017.

[25] X. Yu, M. Zhou, X. Chen, L. Deng, and L. Wang, "Using class imbalance learning for cross-company defect prediction," in *Proc. the International Conference on Software Engineering and Knowledge Engineering*, Knowledge Systems Institute Graduate School, 2017, pp. 117–122. doi: 10.18293/SEKE2017-035

[26] C. Schröer, F. Kruse, and J. M. Gómez, "A systematic literature review on applying CRISP-DM process model," *Procedia Comput Sci*, vol. 181, pp. 526–534, 2021. doi: 10.1016/j.procs.2021.01.199

[27] G. I. Webb, "Naïve bayes," *Encyclopedia of Machine Learning*, Boston, MA: Springer US, 2011, pp. 713–714. doi: 10.1007/978-0-387-30164-8_576

[28] F.-J. Yang, "An implementation of Naive Bayes classifier," in *Proc. 2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, IEEE, Dec. 2018, pp. 301–306. doi: 10.1109/CSCI46756.2018.00065

[29] R. I. Bendjillali, M. Beladgham, K. Merit, and A. Taleb-Ahmed, "Illumination-robust face recognition based on deep convolutional neural networks architectures," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 18, no. 2, 1015, May 2020. doi: 10.11591/ijeecs.v18.i2.pp1015-1027

[30] R. Karim. (Sep. 27, 2023). Illustrated: 10 CNN architectures. *Towards Data Science*. [Online]. Available: https://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d

[31] P. Salim and A. Laksitowening, "Time series prediction on college graduation using KNN algorithm," in *Proc. 2020 8th International Conference on Information and Communication Technology (ICoICT)*, 2020. doi: 10.1109/ICoICT49345.2020.9166238

[32] D. Kanojia and M. Motwani, "Comparison of Naive Basian and K-NN classifier," *International Journal of Computer Applications*, vol. 65, no. 23. pp. 975–8887, 2013.

[33] S. Gupta and A. Gupta, "Dealing with noise problem in machine learning data-sets: A systematic review," *Procedia Comput. Sci.*, vol. 161, pp. 466–474, 2019. doi: 10.1016/j.procs.2019.11.146

[34] A. G. Pertiwi, N. Bachtiar, R. Kusumaningrum, I. Waspada, and A. Wibowo, "Comparison of performance of k-nearest neighbor algorithm using smote and k-nearest neighbor algorithm without smote in diagnosis of diabetes disease in balanced data," *J Phys Conf Ser*, vol. 1524, no. 1, 012048, Apr. 2020, doi: 10.1088/1742-6596/1524/1/012048.

[35] R. S. Parmezan, V. M. A. Souza, and G. E. A. P. A. Batista, "Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model," *Inf Sci (N Y)*, vol. 484, pp. 302–337, May 2019. doi: 10.1016/j.ins.2019.01.076

[36] H. Dasi, "Student dropout prediction using machine learning techniques," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 10, no. 4, pp. 408–414, 2022.

[37] M. Segura, "Machine learning prediction of university student dropout: Does preference play a key role?" *Mathematics*, vol. 10, no. 18, 2022. doi: 10.3390/math10183359

[38] M. Katsuragi, "Dropout prediction by interpretable machine learning model towards preventing student dropout," *Advances in*

Transdisciplinary Engineering, vol. 28, pp. 678–683, 2022. doi: 10.3233/ATDE220700

[39] V. Realinho, J. Machado, L. Baptista, and M. V. Martins, "Predicting student dropout and academic success," *Data (Basel)*, vol. 7, no. 11, 146, Oct. 2022. doi: 10.3390/data7110146