Proposed Approach for Overcoming the Impact of Unbalanced Distribution in Predicting Students' Performance

Gabrijela Dimić [1], Ljiljana Pecić [1]

[1] *Academy of Technical and Art Applied Studies, School of Electrical and Computer Engineering, Vojvode Stepe 283, 11010 Belgrade, Serbia*

Abstract **– The paper presents a method for mitigating the impact of an unbalanced distribution of multidimensional class features on grade prediction accuracy. For the purposes of the case study, an educational data set named APOD was created by integrating data from heterogeneous sources. The input features and the multidimensional class feature were defined. The effectiveness of adopting the Synthetic Minority Over-Sampling Technique (SMOTE) to handle data imbalance issues was explored using various classification methods. To determine which algorithm performed best in terms of minority class distribution, three experiments were carried out. The SMOTE approach with automatic minority class detection and a 100% sampling factor demonstrated a considerable improvement in model performance for four out of five classifiers that were tested. The primary objective of the study described in this paper is to address the problem of predicting students' final grades in situations where a small dataset causes data imbalance. Small datasets provide insufficient representation of instances within specific classes, resulting in unreliable models with poor performance in predicting student success. The proposed approach for implementing SMOTE is based on an algorithm for identifying minority classes, with a predetermined minimum number of samples per class. This approach enables the development of precise models for predicting students' final test results, even with small educational datasets. The contribution of the proposed research lies in achieving greater accuracy in predicting students' final grades, regardless of dataset size and the presence of minority classes.**

Keywords **– Classification, SMOTE, unbalananced distribution, machine learning, educational data mining.**

DOI: 10.18421/TEM134-20
https://doi.org/10.18421/TEM134-20

Corresponding author: Gabrijela Dimić,
Academy of Technical and Art Applied Studies, School of Electrical and Computer Engineering, Vojvode Stepe 283, 11010 Belgrade, Serbia.
Email: gdimic@gs.viser.edu.rs

Received: 26 May 2024.
Revised: 23 September 2024.
Accepted: 08 November 2024.
Published: 27 November 2024.

1. Introduction

The employment of modern e-systems in educational processes has led to the generation of large-scale and diversified datasets. Although the collected raw data represents a rich resource, it is not inherently valuable. Educational systems recognize the potential of analyzing available data to enhance their efficiency. Many studies focussing on the application of machine learning techniques and educational data mining have been carried out in an effort to find information and knowledge essential for enhancing the educational process. Applying data mining techniques to examine information taken from educational settings is known as educational data mining, or EDM [1]. One of the core issues in the field of educational data mining is classification. Minaei-Bidgoli *et al.* [2] conducted a comparative evaluation of six classifiers to predict student performance using data from the LON-CAPA web educational system. The authors demonstrated that combining classifiers significantly improves the predictive model and reduces errors for each implemented classifier. Bresfelean et al. [3] suggested applying classification and clustering techniques to educational data in order to identify trends of student success or failure. In the paper [4], authors compared different techniques for student classification based on data obtained from the Moodle course.

In the paper [5], the authors conducted a comparative analysis of classifier performance to identify the most suitable classifiers in the case of educational datasets with multidimensional class features. They proposed an ensemble model that uses five different classification algorithms developed using the majority voting concept. The suggested classifier model's precision and accuracy for predicting students' grades in a mixed learning environment significantly improved, according to the results. Fernandes *et al.* in [6] provided a predictive analysis of academic achievement among students enrolled in Federal State of Brazil public schools in the academic years 2015 and 2016. In order to forecast academic results at the end of the school year, classification models based on Gradient Boosting Machine (GBM) were created for each of the two datasets that were classified. Francis and Babu's [7] innovative prediction model for assessing student performance included clustering and classification methods. They demonstrated that the proposed hybrid algorithm, which combines clustering and classification approaches, achieves better accuracy in predicting student success. Jalota and Agrawal [8] analyzed the application of various classification methods to predict levels of student success. A novel model for exam grade prediction was proposed by Yağcı [9] based on a comparative examination of classification models developed with the five different algorithms.

The accuracy of predictive classification models depends on various factors: feature vector dimensionality, data type, dataset size, data anomalies, and class imbalance. A frequent and demanding problem in the classification of educational data is class imbalance. It significantly affects the performance of classification models, especially in datasets with a small number of instances. In cases of class imbalance, machine learning methods tend to generate models biased toward the majority class. Evaluation of implemented classifiers on unbalanced datasets indicates lower classification accuracy for unseen minority class labels.

This paper describes an approach to overcoming the impact of unbalanced distribution of multidimensional class features on the precision of the classification model. A case study was conducted on an educational dataset created by integrating heterogeneous sources. The dataset creation and descriptive analysis were performed in the Python programming environment. Based on the descriptive analysis, multiple minority classes of multidimensional class features were identified.

The SMOTE and the Min-Max data normalisation method are the foundations of the suggested strategy, which addresses class imbalance.

By using five classification algorithms, the SMOTE over-sampling method was tested with varying percentages of synthetic classes. Using the Weka environment, the Naïve Bayes, kNN, Decision Table, J48, and Random Forest algorithms were used to assess the performance of the created classification models.

The results of the conducted experiments show an improvement in classification performances by applying the SMOTE method to all minority classes of multidimensional class features. The performance of the data classification models in the educational system was shown to be significantly improved by the over-sampling method and normalisation techniques.

2. Related Works

The selection of a method to address the problem of data imbalance can be challenging. There are numerous different oversampling methods available to tackle this issue. An overview of studies addressing strategies for resolving the imbalance issue in educational data will be given in this part. Over-sampling, under-sampling, and SMOTE techniques were tested by Thammasiri *et al*. [10] in an effort to address the imbalance issue in educational data. Four classification models were created. The best classification result, with an accuracy of 90.24%, was obtained by combining the support vector machine (SVM) classification method with the SMOTE over-sampling technique, according to performance study. In order to predict undergraduate study success, Mueen *et al.* [11] compared the effectiveness of three classification methods. The SMOTE over-sampling method was implemented to address the data imbalance problem. The combined use of One-Sided Selection and SMOTE techniques to balance the distribution of educational data sets was explained by Pristyanto *et al.* [12]. The results showed improved performance of all listed classifiers when the SMOTE method was used to address the imbalance problem in educational data. To identify the best feature vector from the educational data set, Dimić *et al.* suggested a methodology in [13] that applies four feature selection techniques and the SMOTE method. The outcomes have shown that, for the feature vector chosen using the Correlation-based Feature Selection approach, the Naïve Bayes model significantly improved in accuracy and decreased in false negative rate (FNR).

Ghorbani and Ghousi [14] conducted a comparative analysis of different techniques for addressing the problem of imbalanced data when predicting student performance. Two data sets were used for the research.

Eight classifiers were used to examine how well the applicable resampling techniques addressed the issue of imbalanced data. The Random Forest classifier and the SVM-SMOTE resampling method were shown to produce the greatest results, according to the findings. In order to maintain balance and minimise the detrimental effects of noise, Tariq *et al*. [15] recommended a methodology based on the CTGAN model, the NCC classifier, and the SMOTE Iterative-Partitioning Filter algorithm. This allowed for an increase in the size of the educational data set without compromising balance. On a small educational data set, Rattan *et al.* [16] performed a comparative analysis of the performance of the classification algorithms k-Nearest Neighbour, Naïve Bayes, CHAID decision tree, and Random Forest with and without SMOTE. Semi-supervised over-sampling techniques were suggested by Jahin *et al.* [17] as a solution to the imbalance in binary class labels issue, with the goal of enhancing student grade classification performance. Flores *et al.* investigated eight prediction models' accuracy in predicting potential dropout rates among university students by balancing class labels using the SMOTE technique in [18]. Upon comparing the performance of the developed classification models, the Random Forest predictive model exhibited the best levels of accuracy (96.8%) and robustness (96.78%). In [19], Bujang *et al.* analyzed published studies from 2015 to 2021 with the most commonly implemented methods for addressing the problem of imbalance in educational data sets. The efficiency of these methods in addressing unbalanced classification was demonstrated. The results of the study showed that the SMOTE over-sampling method is the most widely used strategy for resolving the imbalance issue that is affecting the accuracy of student grade prediction at the data level. The use of various sample approaches to solve the issue of class imbalance with varying degrees of imbalance was examined by Wongvorachan *et al.* [20]. They used a data set from the Longitudinal Study of high school students from 2009. The effectiveness of the applied methods was tested using the Random Forest classifier. The findings demonstrated a considerable improvement in the efficiency of educational data classification for slightly unbalanced data and composite sampling for very imbalanced data.

3. Background

This section explains the theoretical background for the methodologies and algorithms used in the preprocessing and classification phases of the research detailed in this publication.

3.1. Classification

One way to consider the concept of classification is as a supervised learning method [21]. Creating a classification model involves training instances with known label values. A classifier is a function M that, given an instance x, predicts a class label y. It is defined by the following equation:

$$\hat{y} = M(x) \tag{1}$$

where x =$(x_1, x_{2,\dots,} x_d)^{\ T} \in R_d$ depicts an instance in a space with dimensions of d, and $\hat{y} \in \{c_1, c_2, \dots, c_k\}$ is a feature that depicts the prediction class's values.

A test set containing unknown class label values is used to evaluate the classifier's performance. After establishing reliability, the model is deployed for classifying instances with unknown classes.

3.2. SMOTE

Unreliable classifier models with inadequate performance are produced by a data set where the number of instances in one class is substantially lower than the number of instances in another class [22], [23]. The classifier is trained on majority classes, resulting in low sensitivity to minority classes during the testing phase. Using methods for establishing data balance at the phase of preprocessing provides the foundation for resolving this issue. SMOTE is the most common and efficient oversampling method in many application domains [24]. By identifying the nearest neighbors of the same class, synthetic instances of the minority class are created. Because of its versatility and ease of use, the SMOTE approach is frequently employed in cases with data imbalance [25].

The following equation describes how to produce a synthetic sample by linearly combining two samples from the minority class:

$$X_{sint\_new} = X_i + \left(X_j - X_i\right) * \alpha \tag{2}$$

X_{sint_new} is a new synthetic instance, X_i original instance, X_j one of the nearest neighbors selected based on Euclidean distance, α random value between 0 and 1.

3.3. Normalization

One of the most widely used techniques for normalising data during the data preprocessing phase is the Min-Max method [26]. The procedure is based on scaling values to a specific range. For each feature in the data set, the minimum value is transformed into zero, the maximum value is transformed into one. Decimal numbers between 0 and 1 are used for representing other values. The procedure of Min-Max normalisation is carried out using the equation 3:

$$X_{norm} = \frac{X - \min(X)}{\max(X) - \min(X)} \quad (3)$$

X is the original data value, *min(X)* is the lowest value in the data set for the specific feature, and *max(X)* is the largest value in the data set for the same feature. *X_norm* is the normalised value.

3.4. Näive-Bayes (NB)

The "naive" nature of the NB classification algorithm, derives from the assumption of conditional independence of the class [27], [28]. This indicates that feature values for a particular class label have an independent effect on other feature values. Small data sets can benefit from its simplicity and flexible probabilistic model. A continuous model can be trained by estimating the data density rather than the distribution of values. This method is predicated on the general form of normal distribution of data, which is uncommon in real-world educational environments. The primary goal is to use the Bayes theorem of posterior probability [29] to maximise the $P(C_i|X)$ value, where i is the class index. This may be computed using (4):

$$P(C_i|X) = P(X|\ C_i) \cdot P(C_i) \quad (4)$$

Calculating the likelihood that an instance X belongs to class Ci and choosing the class with the highest posterior probability comprises the classification process.

3.5. k-Nearest Neighbor (kNN)

One way to describe kNN is as a straightforward classification method. [30]. It helps to solve issues related to categorisation and regression. This method "memorises" the training dataset rather than gaining knowledge from it. kNN classifies new input data according to how similar it is to previously taught data, so classifying data into meaningful clusters or subsets.

The class with which the supplied data shares nearest neighbours is allocated to it. The Manhattan, Euclidean, and Minkowski functions are used to calculate the distance for the new data point to join the cluster. The Euclidean distance metric is made use of in this study.

3.6. Decision Table (DT)

DT machine learning algorithm [31] is a simple and interpretable method for classification based on decision tables. These tables represent rules that connect input features with classes. First, a subset of the relevant input features for classification is chosen from the available features using the DT technique. This is usually done through feature selection processes to reduce complexity and improve model accuracy. Based on the selected features, the DT algorithm generates rules in the form of "if-then" statements, linking specific feature values to particular classes. The generated rules are organized into a decision table, where each combination of feature values appearing in the training set is assigned a corresponding class. The algorithm searches the decision table for the relevant rule whenever a new data instance needs to be classified. The corresponding class is assigned if the feature values of the new instance match any rules. Nearest match and majority class procedures are employed if there is not a directly matching rule.

3.7. J48

For the purpose of creating decision trees, Ross Quinlan's C4.5 method is implemented by the J48 algorithm [32]. It is implemented in the Weka open-source software [33]. The generation of the decision tree starts with the root node, representing the most significant feature selected based on the Gain ratio measure. Subsets of the dataset are created according to the chosen feature's values. The process is repeated recursively for each subset, creating new nodes and branches of the tree. Gain ratio is used to select features for creating tree nodes. For all child nodes, the information gain measure is calculated and normalized by the number of feature values, reducing bias towards features with many values. After the tree is created, pruning is applied to remove branches that do not significantly contribute to the model's accuracy.

3.8. Random Forest (RF)

In order to produce better predictions than a single decision tree could produce, RF [34] combines numerous decision trees.

A randomly chosen collection of characteristics and a randomly chosen portion of data are used to construct each tree. This indicates that throughout training, each tree only sees and utilises a part of the available data and features. Once a group of trees is assembled, every tree in the collection can supply its forecast for fresh input data. Whereas in regression each tree gives its numerical forecast, in classification each tree casts a vote for a class. Usually, voting or averaging all of the tree projections determines the final prediction. The quantity of trees in the forest, the amount of features each tree uses, and the tree's depth are some of the important RF parameters. The Gini Index criteria, which measures a feature's impurity in relation to classes, is used by RF to pick features. The tree is not pruned; instead, it is constructed using a variety of features to reach its maximum depth. Studies have indicated that feature selection criteria have less of an impact on classifier performance than pruning methods selection.

3.9. Evaluation Metrics

In order to ensure the precision and accuracy of the classification of new, unseen data, the model's performance should be evaluated.

Accuracy is a measure of accurately classified instances and indicates the likelihood that an instance will be successfully classified. The number of classified instances where the estimated class label values match the actual values is known as true positive (TP). The amount of instances in which the estimated class label values deviate from the actual values is known as false positives (FP). False negatives (FN) are instances of incorrectly classified negative examples, whereas true negatives (TN) show the amount of occurrences that are actually negatively labelled.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (5)$$

Precision and *Recall* are two other significantly important measures for evaluating models. The percentage of correctly classified positive instances among all positively classified instances is known as *Precision*. *Recall* is a metric that shows the proportion of accurately classified instances that are positive out of all truly positive instances.

The formulas for *Precision* and *Recall* are given by equations (6) and (7), respectively.

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

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

F1-measure represents the harmonic mean of Precision and Recall.

$$F_1 - measure = \frac{2*(Recall*Precision)}{Recall+Precision} \quad (8)$$

The connection between the True Positive Rate (TPR) and the False Positive Rate (FPR) is depicted by the ROC curve. The area beneath the ROC curve, or AUC (Area Under the Curve), indicates how well the model can distinguish between class values.

$$TPR = \frac{TP}{TP+FN} \quad (9)$$

$$FPR = \frac{FP}{FP+FN} \quad (10)$$

4. Methodology

This section of the paper outlines a novel strategy for improving prediction accuracy in a blended learning environment. To provide a balanced distribution of values in the multidimensional class feature, the proposed framework applies the SMOTE method (Figure 1).

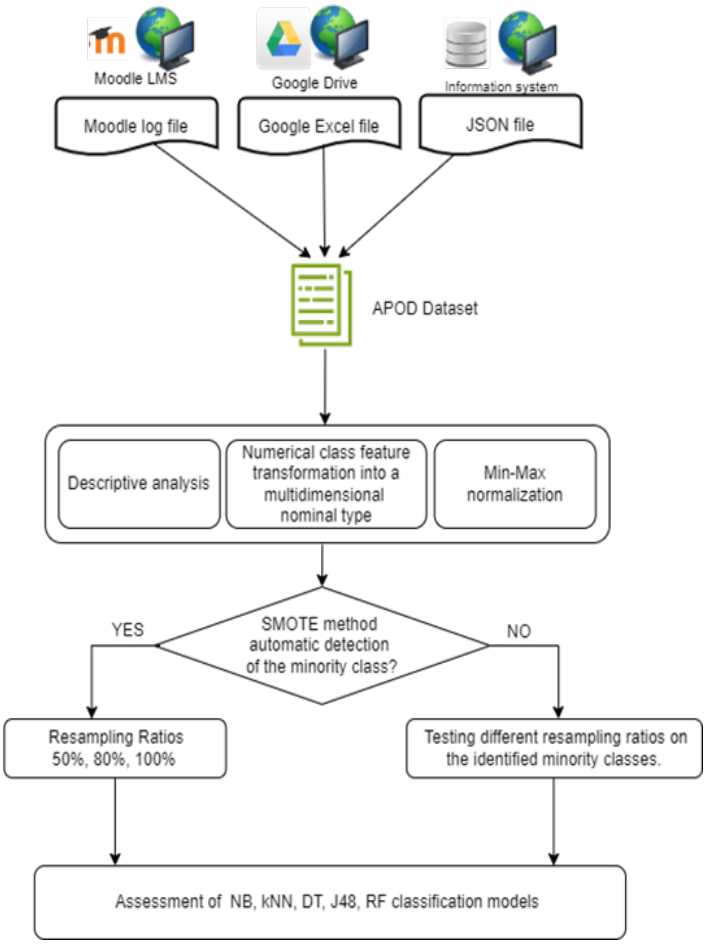

Figure 1. Proposed framework

4.1. Creating an APOD Data Set

For the described case study in this research, data on student activities and grades in the Data Analysis course for the academic year 2022/23 were used. The course was conducted at the School of Electrical and Computer Engineering in Belgrade. The data was extracted from three heterogeneous sources: the Moodle LMS system, Google Drive, and the educational institution's information system. The final dataset, APOD, was formed in a Python environment by integrating Moodle CSV files, Google Excel documents, and JSON reports. The APOD data set contains ten different features for 228 students.

Table 1 displays the overall features of the APOD data set.

Table 1. Description selected features of APOD data set

Name	Description	Min	Max	StDev	Mean
CountF	Forum accesses number	1	60	10.255	8.12
CountI	Course guide accesses number	1	13	3.478	2.802
CountLW	Lab video material accesses number	1	73	12.304	9.676
CountHW	Homework material accesses number	1	117	33.573	19.98
CountLec	Lessons accesses number	1	140	17.807	19.806
AttLec	Points achieved in lectures	0	10	2.215	2.866
LW	Points achieved in lab exercises	0	10	6.65	3.343
HE	Points achieved in homework assignments	0	20	12.756	7.163
ET	Points achieved in exam test	-1	70	23.859	19.868
Grade	Finale grade	3, 5, 6, 7, 8, 9, 10		5.759	1.756

4.2. Preprocessing

During the phase of preprocessing, a descriptive analysis was carried out to examine at the value distribution and figure out how many values were missing for each feature. This procedure was implemented in Python environment using NumPy, SciPy, Pandas, and Matplotlib libraries.

The occurrence of missing values was determined for the features CountF: 62, CountI: 92, CountLW: 21, CountHW: 29, CountLec: 31. Figure 2 displays the distribution of data across input features where the occurrence of missing values was not identified.

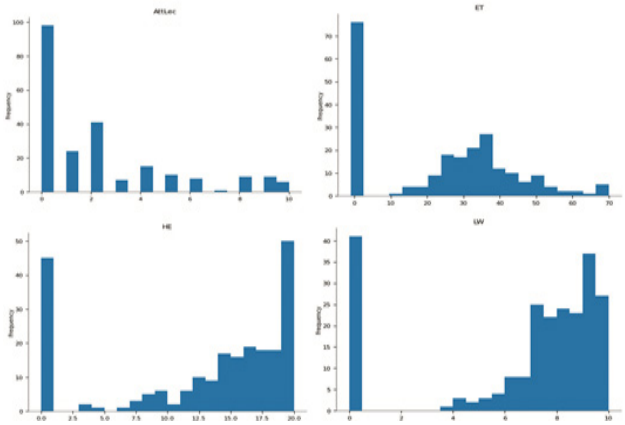

Figure 2. Data distribution for input feature

Figure 3 depicts the distribution of values for the class label Grade.

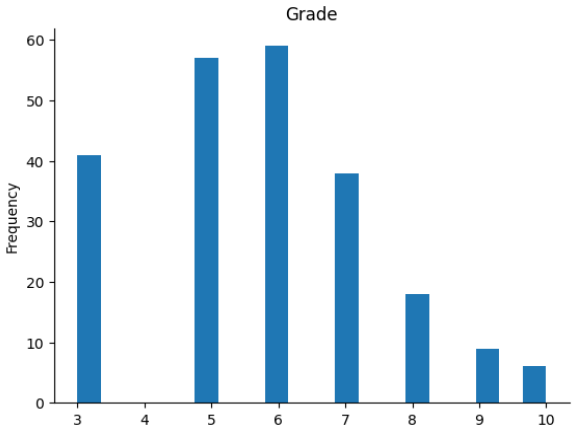

Figure 3. Data distribution for class feature

It is evident from Figure 3's class label histogram that there is an issue with class imbalance in the APOD data set. The presence of three minority classes is noticed. Class 10 (C6) has 6 instances, class 9 (C5) has 9 instances, and class 8 (C4) has 18, which represents a significant difference between these minority and majority classes (classes 3, 5, 6, 7). For this reason, data balancing was performed by applying the SMOTE method.

In order for the values of classes 3, 5, 6, 7, 8, 9, and 10 to transform to the class labels {not take exam, fail, six, seven, eight, nine, and ten}, respectively, the numerical values of the class label Grade were converted into a multidimensional nominal type. Considering that the occurrence of missing values indicates that students did not use certain materials or resources, they were replaced with zero.

After this replacement procedure, different dimensionality and unevenness of the feature value domains were observed in the analyzed dataset. Therefore, the data normalization procedure was carried out using the Min-Max method.

The implementation of the SMOTE method included two cases. The first case relied on the minority class with class label ten being automatically identified. By varying the percentage of synthetic samples to values 50%, 80%, and 100%, the over-sampling process was tested. The second case was based on the detection of minority classes that have less than 50% of samples compared to the majority class. Minority classes with class labels ten, nine, eight were identified. The over-sampling process was tested with different percentages of creating synthetic samples. The implementation of the SMOTE method was performed separately for each identified minority class. Evalution metrics were used to assess the generated classifier models and see how the SMOTE approach affected classification performance. In the study described in the paper, the final grade of students in the Data Analysis course was predicted using the classification algorithms Naïve-Bayes, k-Nearest Neighbour, Decision Table, J48, and Random Forest.

The 10-fold cross-validation approach was used to split the APOD dataset into training and testing sets for each algorithm.

5. Results

In this section, the results of the comparative analysis of classifier models on the APOD dataset under different over-sampling scenarios are presented.

The methodology of the proposed environment was implemented in both Python and Weka environments.

5.1. Comparative Analysis of the Experimental Results

The APOD data set with nominal multidimensional class labels and Min-Max normalization was used. Five classification algorithms were implemented to generate classifier models. The performance evaluation of the generated models was conducted through the analysis of Accuracy, Precision, Recall, F_1-measure, and ROC metrics.

5.1.1. The First Experiment

In the first experiment, the APOD data set with the original number of 228 instances was used, and the performance analysis of classifier models was conducted. Table 2 provides comparison performances of the generated models.

Table 2. Performance of classification algorithms for imbalanced APOD dataset

Algorithms	ACC	Precision	Recall	F1	ROC
NB	0.534	0.537	0.534	0.533	0.587
KNN	0.464	0.470	0.463	0.465	0.419
DT	0.539	0.547	0.539	0.537	0.586
J48	0.574	0.574	0.574	0.574	0.586
RF	**0.578**	**0.580**	**0.577**	**0.578**	**0.597**

5.1.2. The Second Experiment

The second experiment involved applying the SMOTE method with automatic detection of the minority class.

Over-sampling was conducted with percentages of 50%, 80%, and 100% synthetic samples to balance the data. Table 3 displays the generated classifier models' comparative performances.

Table 3. Performance of classification algorithms using SMOTE method with different sampling factor

Algorithms	ACC	Precision	Recall	F1	ROC
sampling factor = 50% (231 instances)					
NB	0.626	0.628	0.626	0.626	0.636
KNN	0.566	0.571	0.566	0.567	0.518
DT	0.648	0.656	0.648	0.648	0.686
J48	**0.678**	0.679	**0.678**	0.678	0.686
RF	0.674	0.675	0.674	0.674	**0.698**
sampling factor =80% (232 instances)					
Algorithms	**ACC**	**Precision**	**Recall**	**F1**	**ROC**
NB	0.637	0.639	0.637	0.637	0.637
KNN	0.566	0.571	0.566	0.567	0.518
DT	0.759	0.767	0.759	0.759	0.797
J48	**0.788**	**0.789**	**0.788**	**0.788**	**0.796**
RF	0.784	0.785	0.784	0.784	0.701
sampling factor =100% (234 instances)					
Algorithms	**ACC**	**Precision**	**Recall**	**F1**	**ROC**
NB	0.723	0.726	0.723	0.722	0.787
KNN	0.771	0.778	0.772	0.772	0.727
DT	0.849	0.855	0.849	0.848	0.890
J48	0.889	0.890	0.889	0.889	0.897
RF	**0.894**	**0.895**	**0.894**	**0.894**	**0.911**

5.1.3 The Third Experiment

In the third experiment, a proposed approach to applying the SMOTE method was implemented. In order to determine the majority class in the multidimensional class feature and the number of instances for each class, an algorithm was devised. The number of samples in the majority class is represented by majority_count, and the minimal threshold for the number of samples per class was set at *min_threshold = 0.5 * majority_count*.

The process of identifying minority classes involved verifying the condition related to the quantity of samples inside a specific class.

Classes with a number of samples less than the minimum threshold, *class_counts[class] < min_threshold*, were identified as minority classes. For each identified minority class, the SMOTE method was implemented with different percentages of synthetic samples to be created. Table 4 displays the generated models' comparison performances.

Table 4. Performance of classification algorithms for APOD dataset using SMOTE method with novel approach

Algorithms	ACC	Precision	Recall	F1	ROC
C4(18 + 80%), C5(9+100%+80%), C6(6+100%+100%+30%) (290 instances)					
NB	0.953	0.953	0.952	0.952	0.993
KNN	0.912	0.913	0.907	0.908	0.945
DT	0.962	0.968	0.962	0.962	0.994
J48	0.978	0.980	0.978	0.978	0.992
RF	**0.979**	**0.981**	**0.979**	0.979	**0.999**
C4(18 + 100%+30%), C5(9+100%+100%+40%), C6(6+100%+100%+80%) (334 instances)					
Algorithms	**ACC**	**Precision**	**Recall**	**F1**	**ROC**
NB	0.962	0.962	0.961	0.961	0.994
KNN	0.928	0.932	0.928	0.929	0.954
DT	0.967	0.971	0.967	0.966	0.996
J48	0.982	0.982	0.982	0.982	0.993
RF	**0.992**	**0.993**	**0.992**	**0.992**	**0.999**

6. Discussion

In the study described in the paper, three experiments were conducted, focusing on improving the accuracy of the classification models on an educational dataset created from heterogeneous sources and defined by a multidimensional class attribute. The analysis was conducted to determine the effect of class imbalance on the developed classifier models' performance. The models' performance metrics are displayed in Table 2 for each of the used algorithms. The ratio of accurate predictions to the total number of evaluated samples is known as accuracy, and it is one of the most widely used evaluation metrics for classifier performance. Despite being simple to comprehend, this statistic ignores a number of crucial aspects that should be taken into account when assessing classifier performance.

RF performed the best out of all classifiers on the original dataset without using the SMOTE method, with an accuracy and other metrics of about 58%. The best class separability among all observed algorithms was indicated by the ROC value. On the other hand, the kNN algorithm had the lowest results among all classifiers, with accuracy below 47% and a low ROC value, indicating poor performance in this case. Näive-Bayes showed modest performance with metric values around 53%, while the ROC value suggested slightly better but still modest class separability. Decision Table exhibited similar performances to Näive-Bayes with slightly higher precision but similar accuracy and ROC value. The J48 algorithm, a decision tree variant, outperformed NB, kNN, and DT with accuracy and other metrics around 57%.

In the first experiment, all generated classifier models had metric results below 60%, indicating that none of them achieved satisfactory and significant performance. RF showed the best performance, while kNN exhibited the weakest. These results highlight the need for better dataset balancing or balancing the number of samples to improve classifier performance. The results of applying the SMOTE method by automatically detecting the minority class are shown in Table 3. Synthetic instance creation was performed with three different values of the sampling factor. All classifiers significantly improved their performance in the case of the SMOTE method with a sampling ratio of 100%. This suggests that data balance improves model learning and increases F1, Accuracy, Precision, and Recall. RF showed the best performance among all algorithms at all levels of balanced data, especially with 100% balanced data, where the ROC value reached 0.911. J48 and DT also showed significant improvements, especially J48, which was very close to RF performance with high ROC values.

The kNN classifier showed the least improvement and generally weaker performance compared to other algorithms, even when SMOTE was applied.

The results of the new approach to applying the SMOTE method, based on the algorithm for detecting minority classes, are shown in Table 4. Minority class detection was based on checking the condition of a minimum threshold for the number of samples per class. Half of the samples in the class that was identified as the majority were used as the minimum threshold for identifying minority classes (min_threshold = 0.5 * majority_count). By adjusting the identified minority classes in relation to the majority class, synthetic data samples were produced. In both cases of the third experiment, all classifiers showed significant performance improvement, with accuracies (ACC) above 90% and very high ROC values (close to or at 0.999 for RF). This indicates that the new approach of the SMOTE method is very effective in balancing the data and improving classifier performance. RF consistently showed the best performance in both cases, with accuracy close to 99% and nearly perfect ROC values. J48 and DT classifiers also exhibited high performance, with J48 approaching RF in results. Although NB and kNN classifiers improved their performance, their results were slightly lower compared to DT, J48, and RF. KNN had the lowest results among all tested classifiers but still showed significant improvements compared to the experiment without the SMOTE method and the second experiment of applying the SMOTE method by automatically detecting the minority class.

7. Conclusion

A large quantity of data is being generated, collected, and stored in all areas, including education, as a result of the digitisation process and the quick growth of information and communication technology. The study and predicting of student progress is one of the major issues facing educational establishments. The process of extracting important and relevant insights from educational data is known as educational data mining. It might, however, run into issues like the effect of unbalanced data on the forecast of student progress. The study described in this paper aims to find an approach to sampling imbalanced educational data sets created from heterogeneous sources with multidimensional class feature. Several classifiers were applied to reach better conclusions. All classifiers were first tested on the original APOD data set. The results show that the classifiers' performance is inadequate for imbalanced data, and that the class attribute's separability was not accomplished to a suitable degree.

By using the SMOTE method with automatic detection of minority classes, a significant improvement in the performance of four classifiers was observed when the sampling ratio was 100%.

However, the kNN classifier showed the least improvement and generally weaker performance compared to other algorithms. Considering that the SMOTE method was applied with automatic detection of minority classes, it was concluded that a different approach should be used in the case of multidimensional class attributes. The proposed approach of using the SMOTE method, based on the algorithm for detecting minority classes, achieves a significant improvement in performance for all implemented classifiers. A balanced creation of synthetic samples of minority classes in relation to the majority class is ensured by setting a minimal threshold for detecting minority classes. In this manner, improved class separability is attained for all implemented classifiers. A larger number of synthetic instances (290 and 334) show how important it is to balance the educational data set in order to achieve meaningful gains in the prediction of students' final grades.

The small number of instances in the original data set limited the selection of appropriate classifiers to handle this issue, which is a shortcoming of the research detailed in the paper. The following paths can be taken for this study's future development. For a more thorough examination, data sets with varying numbers of data gathered across multiple school years should be used. Analysing the effects of various sampling techniques on the accuracy of generated models would involve comparing the performance of classifier ensembles. Furthermore, in the case of data acquired from heterogeneous sources, a significant improvement in the accuracy of predicting student grades would result from additional investigation into the effects of various sampling techniques on the selection of the ideal feature vector.

References:

[1]. Baker, R. S., & Yacef, K. (2009). The state of educational data mining in 2009: A review and future visions. *Journal of educational data mining*, *1*(1), 3-17.

[2]. Minaei-Bidgoli, B., Kashy, D. A., Kortemeyer, G., & Punch, W. F. (2003). Predicting student performance: an application of data mining methods with an educational web-based system. *33rd Annual Frontiers in Education, 1*, T2A-13. IEEE.

[3]. Bresfelean, V. P., Bresfelean, M., Ghisoiu, N., & Comes, C. A. (2008). Determining students' academic failure profile founded on data mining methods. *ITI 2008-30th international conference on information technology interfaces*, 317-322. IEEE.

[4]. Romero, C., Ventura, S., Espejo, P. G., & Hervás, C. (2008). Data mining algorithms to classify students. *Educational data mining 2008*.

[5]. Predić, B., Dimić, G., Rančić, D., Štrbac, P., Maček, N., & Spalević, P. (2018). Improving final grade prediction accuracy in blended learning environment using voting ensembles. *Computer Applications in Engineering Education*, *26*(6), 2294-2306.

[6]. Fernandes, E., Holanda, M., Victorino, M., Borges, V., Carvalho, R., & Van Erven, G. (2019). Educational data mining: Predictive analysis of academic performance of public school students in the capital of Brazil. *Journal of business research*, *94*, 335-343.

[7]. Francis, B. K., & Babu, S. S. (2019). Predicting academic performance of students using a hybrid data mining approach. *Journal of medical systems*, *43*(6), 162.

[8]. Jalota, C., & Agrawal, R. (2019). Analysis of educational data mining using classification. *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, 243–247. IEEE.

[9]. Yağcı, M. (2022). Educational data mining: prediction of students' academic performance using machine learning algorithms. *Smart Learning Environments*, *9*(1), 11.

[10]. Thammasiri, D., Delen, D., Meesad, P., & Kasap, N. (2014). A critical assessment of imbalanced class distribution problem: The case of predicting freshmen student attrition. *Expert Systems with Applications*, *41*(2), 321-330.

[11]. Mueen, A., Zafar, B., & Manzoor, U. (2016). Modeling and predicting students' academic performance using data mining techniques. *International Journal of Modern Education and Computer Science*, *8*(11), 36-42.

[12]. Pristyanto, Y., Pratama, I., & Nugraha, A. F. (2018). Data level approach for imbalanced class handling on educational data mining multiclass classification. In *2018 International Conference on Information and Communications Technology (ICOIACT,)*310–314. IEEE.

[13]. Dimic, G., Rancic, D., Macek, N., Spalevic, P., & Drasute, V. (2019). Improving the prediction accuracy in blended learning environment using synthetic minority oversampling technique. *Information Discovery and Delivery*, *47*(2), 76-83.

[14]. Ghorbani, R., & Ghousi, R. (2020). Comparing different resampling methods in predicting students' performance using machine learning techniques. *IEEE access*, *8*, 67899-67911.

[15]. Tariq, A., Niaz, Y., & Amin, A. (2021). Systematic Approach for Re-Sampling and Prediction of Low Sample Educational Datasets. *International Journal of Computing and Digital System*.

[16]. Rattan, V., Mittal, R., Singh, J., & Malik, V. (2021, March). Analyzing the application of SMOTE on machine learning classifiers. *2021 International Conference on Emerging Smart Computing and Informatics (ESCI)*, 692-695. IEEE.

[17]. Jahin, D., Emu, I. J., Akter, S., Patwary, M. J., Bhuiyan, M. A. S., & Miraz, M. H. (2021). A novel oversampling technique to solve class imbalance problem: A case study of students' grades evaluation. In *2021 International Conference on Computing, Networking, Telecommunications & Engineering Sciences Applications (CoNTESA)*, 69–75. IEEE.

[18]. Flores, V., Heras, S., & Julian, V. (2022). Comparison of predictive models with balanced classes using the SMOTE method for the forecast of student dropout in higher education. *Electronics*, *11*(3), 457.

[19]. Bujang, S. D. A., Selamat, A., Krejcar, O., Mohamed, F., Cheng, L. K., Chiu, P. C., & Fujita, H. (2022). Imbalanced classification methods for student grade prediction: a systematic literature review. *IEEE Access*, *11*, 1970-1989.

[20]. Wongvorachan, T., He, S., & Bulut, O. (2023). A comparison of undersampling, oversampling, and SMOTE methods for dealing with imbalanced classification in educational data mining. *Information*, *14*(1), 54.

[21]. Han, J., Kamber, M., & Pei, J. (2012). *Data mining concepts and techniques* (3rd ed.). The University of Illinois at Urbana-Champaign, Simon Fraser University.

[22]. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321-357.

[23]. Gu, Q., Cai, Z., Zhu, L., & Huang, B. (2008, December). Data mining on imbalanced data sets. In *2008 International Conference on advanced computer theory and engineering*, 1020-1024. IEEE.

[24]. Elreedy, D., & Atiya, A. F. (2019). A novel distribution analysis for SMOTE oversampling method in handling class imbalance. *Computational Science–ICCS 2019: 19th International Conference, Faro, Portugal, June 12–14, 2019, Proceedings, Part III*, 236–248. Springer International Publishing.

[25]. Fernández, A., Garcia, S., Herrera, F., & Chawla, N. V. (2018). SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *Journal of artificial intelligence research*, *61*, 863-905.

[26]. Patro, S. (2015). Normalization: A preprocessing stage. *arXiv preprint arXiv:1503.06462*.

[27]. John, G. H., & Langley, P. (2013). Estimating continuous distributions in Bayesian classifiers. *arXiv preprint arXiv:1302.4964*.

[28]. Rish, I. (2001, August). An empirical study of the naive Bayes classifier. *IJCAI 2001 workshop on empirical methods in artificial intelligence*, *3*(22), 41-46.

[29]. Walters, C., & Ludwig, D. (1994). Calculation of Bayes posterior probability distributions for key population parameters. *Canadian Journal of Fisheries and Aquatic Sciences*, *51*(3), 713-722.

[30]. Taunk, K., De, S., Verma, S., & Swetapadma, A. (2019). A brief review of nearest neighbor algorithm for learning and classification. *2019 international conference on intelligent computing and control systems (ICCS)*, 1255-1260. IEEE.

[31]. Kohavi, R. (1995, April). The power of decision tables. In *European conference on machine learning*, 174-189. Berlin, Heidelberg: Springer Berlin Heidelberg.

[32]. Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, *1*, 81-106.

[33]. University of Waikato. (n.d.). WEKA. *University of Waikato, New Zealand*. Retreived from: http://www.cs.waikato.ac.nz/ml/weka/ [accessed 10 June 2024].

[34]. Breiman, L. (2001). Random forests. *Machine learning*, *45*, 5-32.