*Journal of Autonomous Intelligence (2023) Volume 6 Issue 1*
doi:10.32629/jai.v6i1.583

ORIGINAL RESEARCH ARTICLE

# Designing new student performance prediction model using ensemble machine learning

**Rajan Saluja[1*], Munishwar Rai[1], Rashmi Saluja[2]**

[1] *MMICTBM, MMDU, Mullana, Ambala Cantt, Haryana 133203, India. E-mail: rajansaluja.mca@piet.co.in*
[2] *Department of Computer Applications, PIET, Samalkha, Haryana 132102, India.*

## ABSTRACT

Academic success for students in any educational institute is the primary requirement for all stakeholders, i.e., students, teachers, parents, administrators and management, industry, and the environment. Regular feedback from all stakeholders helps higher education institutions (HEIs) rise professionally and academically, yet they must use emerging technologies that can help institutions to grow at a faster pace. Early prediction of students' success using trending artificial intelligence technologies like machine learning, early finding of at-risk students, and predicting a suitable branch or course can help both management and students improve their academics. In our work, we have proposed a new student performance prediction model in which we have used ensemble machine learning with stacking of four multi-class classifiers, decision tree, k-nearest neighbor, Naïve Bayes, and One vs. Rest support vector machine classifiers. The proposed model predicts the final grade of a student at the earliest possible time and the suitable stream for a new student. A student dataset of over a thousand students from five different branches of an engineering institute has been taken to test the results. The proposed model compares the four-machine learning (ML) techniques being used and predicts the final grade with an accuracy of 93%.

***Keywords:*** Ensemble Machine Learning; Decision Tree; K-Nearest Neighbor; Naïve Bayes; One vs. Rest Support Vector Machine

**ARTICLE INFO**

Received: 2 April 2023
Accepted: 23 April 2023
Available online: 24 May 2023



## 1. Introduction

A large amount of data is produced in educational institutes every year. In any education system, a student spends a lot of hours in the classroom, offline or online, learning from teachers, preparing assignments at home, appearing in the examination, and gets a grade. Students of different learning abilities get a grade according to their academic records, like test scores, assignment marks, theory scores, and practical examination scores. Teachers and administrators maintain every student's complete record, i.e., from admission time to completion of the course, as it is the requirement for certified agencies, affiliation boards, universities, and accreditation bodies. This large amount of data can be used to maintain and raise the standards of that institute. Adopting ensemble machine learning techniques with the data, early predictions like how many students will get excellent grades, how many will drop out, how many students will be placed in which company, how many students have the wisdom to be entrepreneurs, which subject result will be outstanding, and which branch or course is suitable for a student can be made. Predicting grades or making early predictions of at-risk students or dropouts is advantageous for both the students and the institute. Ensemble machine learning techniques are being applied in variety of complex problems[1], like fraud detection, disease prediction, and remote sensing. In

the same way, these techniques can be applied in the selection of a particular course for a student, predicting the final grade of a student, or classifying students into different classes like A, B, and C at the earliest which will help students improve their academic success. As students with low predicted grades can be given more classes and academic facilities for a subject, and high-grade students can be provided with add-on courses to enhance their skills, a lot of high-cost learning management systems (LMS) have been proposed in COVID times. The real problem with all available LMS is accuracy and timing when it is predicted. If prediction is done at the midpoint or end of the study, then it is of no use. We have designed a model that can be used to predict students' grades after the first semester. Several predictive techniques, like support vector machine (SVM)[2,3], decision tree (DT)[4–7], k-nearest neighbors (KNN)[8,9] and Gaussian Naïve Bayes (GNB)[8,10,11], and data mining[4] are being utilized to solve the concerned problem.

Choosing the best prediction model[12] for a particular kind of problem among a wide range of predictive models is a difficult task, the ensemble ML technique makes the task easy by producing the best of all models. Our work focusses on the design of a new prediction model that uses the ensemble ML technique stacking. Data for our work has been collected from an engineering institute, PIET, Samalkha, Delhi NCR, India. This data is perfectly suitable for ensemble machine learning technique implementation. Our paper is structured in following manner: Section 2 describes the all-research questions; in Section 3, literature review has been done; methodology of the research work has been described in Section 4; results and analysis are mentioned in Section 5 and conclusions are made in Section 6.

## 2. Aims of the research and research questions

Our study aims to design a new model for student academic performance prediction that can predict the final degree grade of a student with the help of the Python Jupiter Notebook. Our research work has been partitioned into two parts: in the first part, our aim was to collect the dataset, clean it and find the pre-processing techniques, and feature selection that would be best to predict the students' final grades; in the second part, we designed a new ensemble machine learning model, with the help of which, academic performance could be predicted at the earliest with improved accuracy. The following research questions have been covered in our work:

1) What are the various multi-class classification models that are being used to find out students' grades?

2) Which classification technique is best for predicting students' grades?

3) What is the adverse effect of an imbalanced dataset, and what is the impact of SMOTE?

4) Does stacking improve evaluation parameters like accuracy, recall, precision, and F1-Score?

5) What is the impact of selecting the meta model in stacking?

## 3. Literature survey

Predicting students' grades at the earliest is a desired factor to improve students' success. Educational institutions must develop new emerging solutions that can help predict success at the earliest and with great accuracy in a shorter time. Students from different backgrounds, learning abilities, social and economic status, or boards appear in the process of taking admission in a particular course, and every student's aim is to get success. Our proposed model uses the stacking[13] SVM One vs. Rest[14] algorithm with decision tree classifier[15], Gaussian Naïve Bayes[16], and KNN classifier[17] on the Python Jupiter Notebook platform. In addition, our work compares the accuracy and different parameters of these four machine learning techniques before and after SMOTE (synthetic minority oversampling technique). As our problem is a multi-class classification problem, that means classification tasks that have multiple values in their target variable like an image classification problem, a handwriting classification problem, or a student's grade classification problem. The number of class labels in our work is fixed at 3, i.e., Class C for below-average students, Class B for average students and Class A for good students. The algorithms that have been selected for the same problem are:

- k-nearest neighbors
- support vector machine
- Gaussian Naïve Bayes
- decision tree
- ensemble machine learning.

**KNN** is the simplest classification technique. The performance of this technique is independent of the schema of the dataset. Whenever a new entry is to be made in data, with the help of Euclidean distance formula, its neighbors are calculated. The majority class that has its k nearest neighbors is selected for a new entry. KNN[18] has been used for a student's grade prediction in recent work with 83% accuracy to determine at-risk students at the earliest. Bujang *et al*., in their study[2], used the KNN algorithm for predicting a student's grades for the two subjects with an accuracy of more than 90%. KNN has been used in an ensemble machine learning algorithm[19] for predicting a student's dropout. Rohilla *et al*.[20] explored the use of the KNN algorithm for prediction of diseases in plants. KNN has been preferred in prediction as it requires a shorter training period. The problem associated with KNN is that, in cases of high dimensionality, the results are not so accurate[21].

The **GNB** ML classification technique works on Bayes' theorem. It is a classification technique with strong independence assumptions. Here, independence refers to the idea that the presence of one value of an attribute does not influence the presence of another. Let $x_i$ be a feature vector, and $y$ be the target class label, and $P(y)$ be the relative frequency of $y$ in the training dataset. Then $P(x_i \mid y)$ is calculated in GNB as in equation 1:

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma^2{}_y}} e^{-\frac{(x_i-\mu_y)^2}{2\sigma^2{}_y}} \quad (1)$$

Bujang *et al*.[2] used GNB to predict the grade of students with an accuracy of more than 90%. Naseer *et al*.[22] predicted the complexity level of coding among the teams who were working on the same software project using the GNB algorithm. Wang *et al*.[23] proposed a new technique called multinomial Naive Bayes tree (MNBT) for both types of data, continuous and discrete. The problem associated with GNB is that it is highly scalable to the number of estimator data points and sometimes it has a zero-frequency problem[21].

The **DT classifier** is a well-structured technique for both types of classification problems. This technique generates a series of observations from the dataset in relation to its various features. A new observation is planted, just like in a binary tree and the data present on that node is further divided into another partition of data values that have different characteristics. Classes are framed on the leaves in which the dataset is split. Hussain and Khan[24] have used DT for the classification of a batch of students into two different grade classes. Similarly, in the study of Hutagaol and Suharjito[19], DT has been used to predict early dropouts in an ensemble ML technique. A little effort is required for the pre-processing of data in the case of DT implementation as normalization and scaling are not needed, even missing values also do not affect the output much, but training the model takes more time in comparison with other available techniques as the complexity level is high for DT[21].

The **support vector machine** technique is generally used for the both types of classification and regression problems. SVM technique for classification can be used for binary and multi-class classification problems. Two variants of SVM for multi-class classification are One vs. One (OvO) and One vs. Rest (OvR). OvR is an empirical method that performs by dividing the multiclass problem into multiple binary problems. SVM binary classifier is used to train it to solve every single binary classification problem, and predictions are made using the model that is the most accurate. One vs. Rest SVM works efficiently when there is a significant margin between classes, which gives better performance in less memory and time, so mostly used in classification problems. There may be a problem using this algorithm when there are millions of pieces of data and a very large number of classes say more than 10 or 20, as this technique creates one binary model for each[21]. SVM has been used to propose a prediction model in the study of Bujang *et al*.[2] and that of Hussain and Khan[24], for predicting the grade in an imbalanced dataset. Students' grades were predicted with SVM in the study of Sorour *et al*.[25] with an accuracy of 86%, and the results were compared with artificial neural networks (ANN). Park and Yoo[26] used the SVM technique to predict an early dropout student in an online course. Singh[27] experimented with a tactic to predict enthusiasm among students towards higher education using SVM and ANN. Burman and Som[28] have predicted the three class students' grades with 90% accuracy with the help of the SVM technique. SVM has been used by Altabrawee *et al*.[29] for prediction of stu-

dents' performance.

**Ensemble Machine Learning** is a technique where multiple ML models are combined to solve a complex intelligence problem. Individually, each model may perform well on some data but not on other data. When multiple models are combined, their weaknesses cancel each other out, and strength comes as the output of combined model. Various types of ensemble machine learning are bagging, stacking and boosting. Some of the applications of ensemble learning are remote sensing, emotion recognition in speech, detection of diseases, detection of fraud, etc. Stacking[30] is one of the most preferred ensemble techniques[31] that can be used to build a new model with the help of existing models to improve the overall model's performance. It has been used in many complicated problems[32–35] as stacking enables us firstly to train multiple individual models to solve the same problem, and then based on their performance, it creates a new model with improved performance. The architecture of the stacking model[1] includes two types of models, level 0 models, or base models (two or more than two), and a meta-model at level 1, like in **Figure 1**. The stacking ensemble method includes training the primary level models with the original dataset, prediction data from level 0 models, training the level 1 model with prediction data collected by level 0 models, and final prediction.

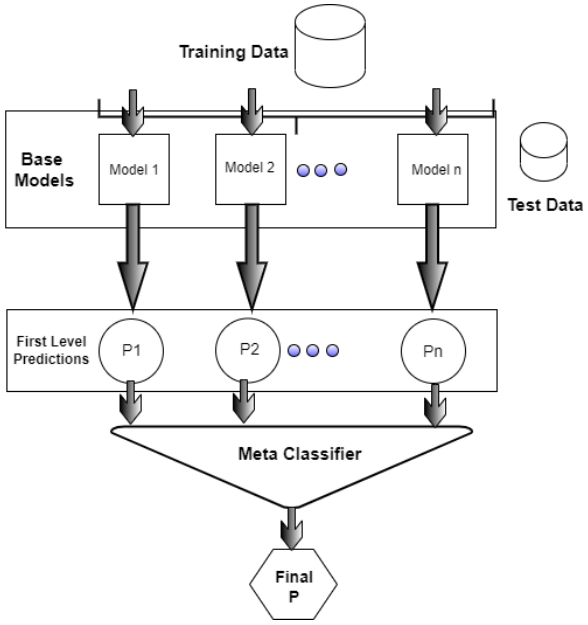


**Figure 1.** Stacking architecture used in the proposed model.

# 4. Methodology

The methodology followed in our work has been described as shown in **Figure 2**. Work has been composed of the following parts:

1) Data collection, cleaning, preprocessing, and data mining

2) Proposed model design training

3) Testing the model.

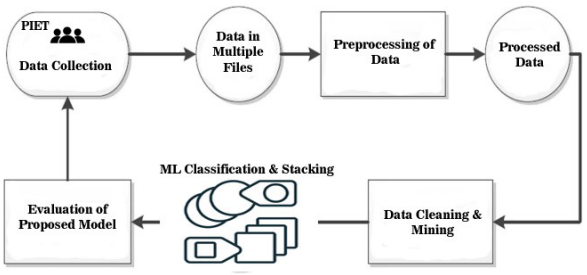


**Figure 2.** Methodology.

## 4.1 Data collection, cleaning, preprocessing, and data mining

Data was collected from the administrative department of an engineering institute, PIET, Samalkha, Haryana, India. Data was collected in the form of several separate Excel files, like batchwise admission data and result data. In admission data, basic information parameters like name, admission number, father's name, address, date of birth, gender, contact number, percentage or CGPA in high school, senior secondary school, admission test data, contact number of student or guardian, and day boarding or hostel, etc., were mentioned. In the result data file, students' 1$^{st}$ to 8$^{th}$ semester marks were mentioned in case of engineering courses and in the case of degree course 1$^{st}$ to 6$^{th}$ semester marks were there. Marks were converted into percentages as different engineering branches have different numbers of subjects with different weightages. The data was cleaned with Excel features and formulas to make the dataset ready for processing. Data was combined from multiple files into a single Excel sheet by combining two sheets of admission and result for a particular batch, as shown in **Figure 3**.

Data was combined for four batches of engineering and 5 batches of BCA, city, and location as rural or urban fields were extracted from the address field. Similarly, the zodiac sign field was prepared from date of birth data. The zodiac sign was included in the final dataset as a dependency on the final grade was found over this independent attribute. High school and senior secondary level CGPA

| Sr. N | Roll No | Batch | Branch | Zodiac | Genc | Cat | City | LOC | Phone | Hostel | 10th% | 12th% | PQT | 1st% | 2nd% | 3rd% | 4th% | 5th% | 6th% | 7th% | 8th% | Final% | Avg% | Grade |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2308259 | 2009 | ECE | pisces | M | GEN | PANIPAT | U | M | N | 72.6 | 67.6 | 62 | 60 | 64.8 | 0 | 0 | 0 | 0 | 64.2 | 71 | 0 | 32.44 | 0 |
| 2 | 2807001 | 2007 | CSE | sagit | F | GEN | DELHI | U | B | N | 82 | 80 | 66 | 57 | 63.1 | 63.6 | 66.8 | 68.7 | 71 | 72.5 | 69.3 | 67.8 | 66.49 | 1 |
| 3 | 2807002 | 2007 | CSE | libra | F | GEN | DELHI | U | B | N | 77 | 62 | 60 | 61 | 61.6 | 59.5 | 59.4 | 62.4 | 73 | 72.4 | 71.5 | 66.6 | 65.14 | 1 |
| 4 | 2807003 | 2007 | CSE | capri | M | GEN | SONEPAT | U | B | N | 63 | 69 | 64 | 57 | 63.4 | 61.4 | 62.5 | 69 | 68 | 72.1 | 71 | 67.1 | 65.62 | 1 |
| 5 | 2807004 | 2007 | CSE | virgo | M | GEN | PANIPAT | U | M | N | 68 | 71 | 64 | 60 | 67.9 | 59.4 | 65.4 | 69.5 | 71 | 70.6 | 69.1 | 67.6 | 66.62 | 1 |
| 6 | 2807005 | 2007 | CSE | aqua | M | GEN | JIND | U | B | N | 83 | 71 | 60 | 55 | 61.4 | 59.2 | 61.5 | 61.7 | 69 | 66.3 | 66.6 | 63.6 | 62.56 | 1 |
| 7 | 2807006 | 2007 | CSE | sagit | M | GEN | FARIDABA | U | B | Y | 70 | 49 | 62 | 56 | 60.9 | 57 | 65 | 65.6 | 70 | 69.6 | 68.3 | 65.4 | 64.1 | 1 |
| 8 | 2807007 | 2007 | CSE | cancer | F | GEN | PANIPAT | U | B | N | 67 | 77 | 65 | 61 | 67.1 | 62.5 | 65.7 | 68.6 | 74 | 73.8 | 70.2 | 69 | 67.92 | 1 |
| 9 | 2807008 | 2007 | CSE | capri | M | GEN | UP | R | M2 | Y | 58 | 53 | 36 | 57 | 58.4 | 52.2 | 0 | 57.9 | 60 | 57.5 | 57.5 | 0 | 50.04 | 0 |
| 10 | 2807009 | 2007 | CSE | tauras | M | GEN | SONEPAT | U | B | N | 59 | 51 | 58 | 50 | 57.1 | 52.4 | 61 | 63.4 | 66 | 67.8 | 69.1 | 62.8 | 60.93 | 1 |
| 11 | 2807010 | 2007 | CSE | libra | M | GEN | PANIPAT | R | M | N | 61 | 58 | 54 | 49 | 54.5 | 50.8 | 56.8 | 57.3 | 57 | 58.4 | 58.8 | 56.1 | 55.33 | 1 |
| 12 | 2807011 | 2007 | CSE | pisces | F | GEN | KARNAL | U | B | N | 62 | 62 | 66 | 66 | 70.6 | 67.4 | 67 | 67 | 69 | 70.8 | 72.1 | 69.1 | 68.76 | 1 |
| 13 | 2807012 | 2007 | CSE | gemini | M | GEN | PANIPAT | U | B | N | 67 | 68 | 69 | 65 | 70.7 | 67.7 | 71.1 | 71.4 | 74 | 75.2 | 79.4 | 73 | 71.86 | 2 |
| 14 | 2807013 | 2007 | CSE | gemini | M | GEN | KARNAL | R | L | N | 57 | 50 | 67 | 58 | 63.9 | 63 | 68.2 | 71.1 | 76 | 75.7 | 79.1 | 71.4 | 69.34 | 1 |
| 15 | 2807014 | 2007 | CSE | cancer | M | GEN | SONEPAT | U | B | N | 71 | 67 | 60 | 60 | 57.9 | 54.4 | 63.3 | 63.7 | 62 | 68.8 | 71.7 | 64.1 | 62.71 | 1 |
| 16 | 2807015 | 2007 | CSE | scorpio | F | GEN | SONEPAT | U | B | N | 57 | 59 | 59 | 61 | 54.3 | 59.5 | 61.7 | 56.8 | 67 | 66.3 | 65.5 | 62.4 | 61.52 | 1 |
| 17 | 2807016 | 2007 | CSE | scorpio | F | GEN | DELHI | R | B | N | 68 | 56 | 74 | 69 | 73.1 | 69.9 | 76.4 | 76.5 | 79 | 74 | 77.2 | 75.1 | 74.44 | 2 |
| 18 | 2807017 | 2007 | CSE | sagit | F | GEN | SONEPAT | U | B | N | 79 | 71 | 68 | 63 | 68.4 | 67.8 | 68.7 | 68.9 | 73 | 73.7 | 76.7 | 71 | 69.95 | 1 |

**Figure 3.** Excel representation of data after preprocessing.

and marks were converted into percentages to form uniformity. One more reason for the conversion of marks into percentages was that there were different types of students at different board and field levels and total marks were different for different state boards and fields. The dataset was loaded with the help of the Panda library in Python. An observation was done and it was found that there are a total of 25 fields, including 8 categorical fields (branch, zodiac, gender, category, city, location, phone, and hostel), 5 integer type fields (Sr. No, Roll No, Batch, PQT, Grade) and remaining float type data fields and dataset looks like in the **Figure 4**.

| Attribute | dtype | Attribute | dtype |
|---|---|---|---|
| Roll No | int64 | PQT | int64 |
| Batch | int64 | Ist% | float64 |
| Branch | object | 2nd% | float64 |
| Zodiac | object | 3rd% | float64 |
| Gender | object | 4th% | float64 |
| Cat | object | 5th% | float64 |
| City | object | 6th% | float64 |
| LOC | object | 7th% | float64 |
| Phone | object | 8th% | float64 |
| Hostel | object | Final% | float64 |
| 10th% | float64 | Avg% | float64 |
| 12th% | float64 | Grade | int64 |

**Figure 4.** Dataset attributes.

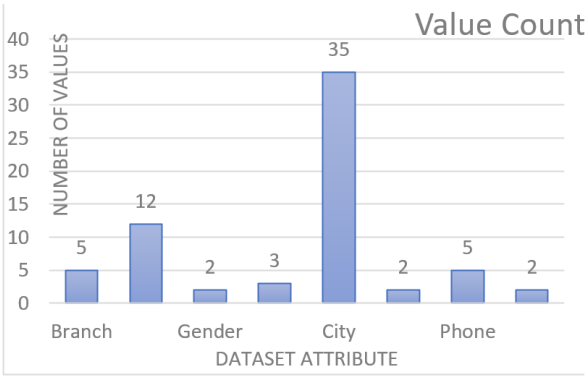


**Figure 5.** Categorical attributes value count.

Using sns.barplot, a bar graph was plotted between the category value and its count as shown in the **Figure 5**. It can be observed that the total number of branches that have been taken in this dataset is 5, categories have been provided with three values, totaling 35 kinds of cities from which students have come to the institute either from rural or urban locations. We had to finalize which attributes to exclude or include in the final dataset according to the impact of the attribute on the final grades of the student. With the help of heatmap, it can be observed which variables are to be included for the prediction model, as only positive corelated values can be selected, as shown in **Figure 6**.

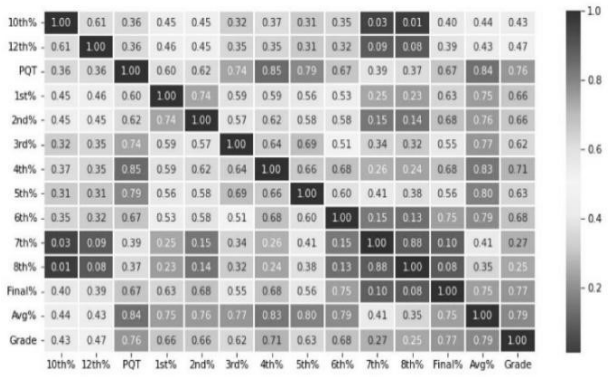


**Figure 6.** Attributes correlation heatmap.

As we had to do the prediction at the earliest, there was no need to consider the 2nd semester to 8th semester percentages. Similarly, negative corelated attributes were removed from the dataset. The next step was to convert all categorical features into category-specific integer values. Zodiac sign features were encoded into 12 different categories from 0 to 11, hostel into 0 and 1, 35 cities from 0 to 34, rural and urban locations into 0 and 1, similarly, contact phone categories into 0 to 4 using Labelencoder(). After ignoring some null values in attributes and filling some with mean values, the final dataset that was used for training our proposed model is shown in **Figure 7**.

```
newdata.head()
```

| | Zodiac | Cat | City | LOC | Phone | Hostel | 10thP | 12thP | PQT | 1stP |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 7 | 1 | 26 | 1 | 2 | 0 | 72.6 | 67.6 | 62 | 59.62 |
| **1** | 8 | 1 | 3 | 1 | 0 | 0 | 82.0 | 80.0 | 66 | 57.43 |
| **2** | 6 | 1 | 3 | 1 | 0 | 0 | 77.0 | 62.0 | 60 | 61.24 |
| **3** | 3 | 1 | 34 | 1 | 0 | 0 | 63.0 | 69.0 | 64 | 57.33 |
| **4** | 11 | 1 | 27 | 1 | 2 | 0 | 68.0 | 71.0 | 64 | 60.48 |

**Figure 7.** PIET students data after label encoding.

## 4.2 Proposed model design and training

Our objective was to design and propose a student success prediction model. Success is being defined in the terms of final grades or percentage. There may be a lot of grade variations, or percentage variations: some students may get excellent grades, most of the students may get average grades and some students may get below average grades; so the dataset will have multiclass dependent attributes and may be imbalanced for any educational institution. As shown in **Figure 8**, variation between two grade classes can be seen easily.

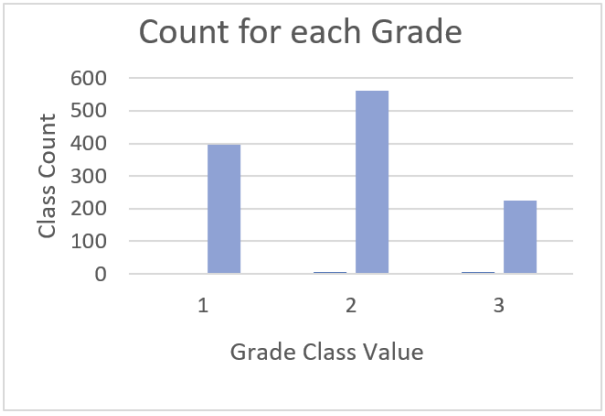


**Figure 8.** Count for grade class.

SMOTE has been used in our proposed model (**Figure 9**) so that imbalanced multiclass data can be synchronized, which will be effective for better prediction. After oversampling, the number for three grade classes becomes even for all classes, as shown in **Figure 10**. We have used four multi-class classification ML techniques DT, KNN, NB, and SVM OvR. Our model has the following components:

- Training the model with four ML techniques (DT, KNN, NB and OvR) and testing the model to determine first level prediction accuracy;
- Using SMOTE for handling the imbalanced data, if any;
- Training the model again with the new train-test dataset and second level prediction accuracy;
- Stacking all classification ML models and designing a new ensemble ML technique with OvR as meta and other models as base models;
- Training the final model with SMOTEd train data and finding final level prediction accuracy with collective accuracy test data.

The SMOTEd dataset has same values for all three grade classes, so the dataset is balanced now. The dataset was divided into independent and dependent attributes, with X having all independent features and y having target grade values, with the help of train and test split data (X_train, X_test, y_train, and y_test) with 80:20 ratio of training and testing data samples. DT, KNN, GNB, and SVM OvR were used for the first-round prediction of our proposed model, as these four models predicted student performance with more than 80% accuracy for our student dataset. The processed dataset was finally split into 80:20 ratio after comparing the

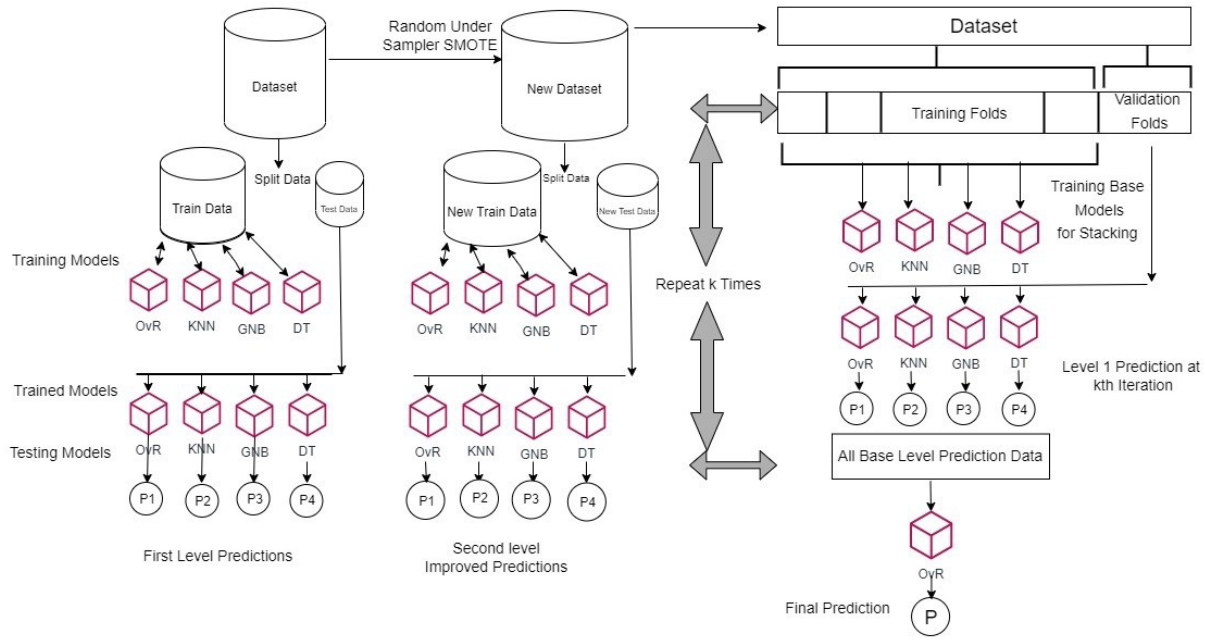


**Figure 9.** The proposed student performance prediction model.

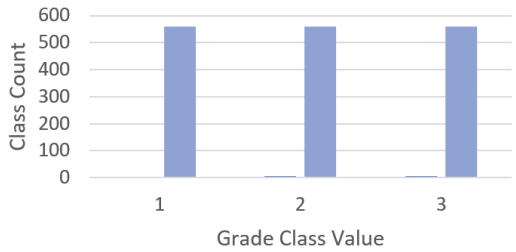


**Figure 10.** Count for three grade classes students after SMOTE.

performance of all models with different splits like 50:50, 70:30 and 90:10 train test data. All necessary libraries and components like decision tree classifier, One vs. Rest classifier, k-neighbors classifier, and the GNB were imported, and all models were trained with training data.

## 4.3 Proposed model testing

Testing was done with the help of both normal and SMOTE test data and results were validated with one batch of results. The model's performance was tested in three steps. First level predictions were calculated on an average basis over multiple runs with a normal dataset for four multi-class classification models, then second level predictions were calculated with SMOTE data and then final prediction was made with the help of stacking all models together. The different parameters for which analysis done were accuracy, recall, precision, and F1-Score, as shown in **Table 1**.

All four models behaved in the same pattern with different sampling criteria for evaluation and predicted students' grades with accuracy greater than 85%. The analysis for first-level predictions for all multiclass classification models has been described in **Table 2**. Results are being analyzed after multiple runs of testing, and all parameters were calculated by the average of three class labels 0, 1, and 2. It was observed that DT and KNN predicted

**Table 1.** Evaluation parameters and detail

| Sr. No. | Parameters | Detail |
|---|---|---|
| 1. | Accuracy | Accuracy is the basic metric that is used to evaluate any ML model's performance. Accuracy is the number of correct predictions made over all the predictions made.<br>(TP + TN)/(TP + TN + FP + FN) |
| 2. | Recall | Recall means how many of the positive cases the model correctly predicted, over all the positive cases in the dataset, also called as sensitivity.<br>TP/(TP + FN) |
| 3. | Precision | Precision means how many correct positive predictions there are.<br>TP/(TP + FP) |
| 4. | F1-Score | F1-Score is a parameter that is calculated by combining both precision and recall. It is the harmonic mean of the two parameters.<br>2 × (precision × recall)/(precision + recall) |

the performance with an accuracy level of 88%, KNN with 87%, and OvR with 85%. Recall, precision, and F1-Score values for all models were nice and stable in every sampling criterion.

**Table 2.** Analysis of prediction accuracy without SMOTE

| Model | Accuracy | Recall | Precision | F1-Score |
|---|---|---|---|---|
| DT | 0.88 | 0.88 | 0.88 | 0.88 |
| OvR | 0.85 | 0.87 | 0.85 | 0.84 |
| KNN | 0.87 | 0.87 | 0.87 | 0.87 |
| GNB | 0.88 | 0.88 | 0.88 | 0.88 |

The count of all three grade classes was different, and the dataset was imbalanced. Imbalanced data is hard to handle as it increases skewness[36]. SMOTE[37] is the widely used technique to handle the imbalanced data. SMOTE creates new minority class data and copies existing data. This duplicate data is generated by randomly selecting one or more of the k-nearest neighbors for each data point in the minority class. We have applied various SMOTE techniques to our model, and the dataset was reconstructed as shown in **Figure 10**. Training was applied again to four models KNN, DT, SVM, and GNB with new SMOTE training data.

SMOTE functionalities available in Python under the Imbalanced-learn Library[38] were implemented, and testing was done for four models. Random Under Sampler and SMOTE methods were observed as the better choice as both techniques increased the accuracy percentage significantly as shown in **Table 3**.

**Table 3.** Analysis of prediction accuracy with SMOTE

| Models testing after SMOTE | Accuracy | Recall | Precision | F1-Score |
|---|---|---|---|---|
| DT | 0.90 | 0.90 | 0.90 | 0.90 |
| OvR | 0.87 | 0.87 | 0.87 | 0.87 |
| KNN | 0.90 | 0.90 | 0.90 | 0.90 |
| GNB | 0.88 | 0.88 | 0.88 | 0.88 |

It was observed that after SMOTE, each model performed better than without SMOTE. KNN and DT predicted the grade with an accuracy level of 90%, and the model that was used as meta model in our design, SVM OvR, predicted the student performance with accuracy, precision, recall, and an F1-Score of 87%. The third component of our proposed model increases the overall prediction accuracy with the help of ensemble machine learning technique. SVM One vs. Rest was selected as the meta-model, and all four ML classification models were kept as base models for our design. After various testing environment parameters, the prediction accuracy has significantly increased in our model design.

## 5. Results and discussion

The model has been tested for the same dataset with all types of oversampling techniques[39–41] like SMOTE, SVMSMOTE, Borderline SMOTE, Random Under Sampler, and ADASYN. The analysis has been mentioned in **Table 4**. We observed the highest prediction accuracy of all three classes is up to 93%, when OvR has been chosen as a meta model and KNN, NB, SVM, and DT are taken as base models for first level of stacking and a 10-fold repeated stratified k value. The oversampling technique in the best performance case has been observed as Random Under Sampler. Other SMOTE techniques also performed better in our proposed model, as model predicted students' performance with more than 90% accuracy. The model exhibited the highest performance rates in precision, recall, and F1-Score.

All other three combinations were also tested, where KNN, DT, or GNB was selected as meta models. From the graphs shown in **Figure 11**, it can be easily interpreted that in the case of any SMOTE method (Random Under Sampler, Boarder Line SMOTE, SVMSMOTE, SMOTE or ADASYN), our proposed model M1 has performed best of all other ensemble models M2, M3, and M4. The M2 model (DT as meta) predicted students' grades with maximum of 91% accuracy with SVMSMOTE. The model M3 (KNN as meta) performed with an accuracy of 91% with SVMSMOTE and the model M4 (GNB as meta) predicted the grade with maximum of 91% accuracy with Random Under Sampler SMOTE. In case of M2 and M4 models where accuracy and other parameters for performance of models were good, but in both models, the training period was very lengthy.

## 6. Conclusions

In this paper, we have designed and proposed a model for predicting students' performance in an educational institution using ensembled machine learning technique, i.e., stacking of OvR as a meta and DT, KNN, GNB, and SVM as base models, and

**Table 4.** Results analysis of the proposed student performance prediction model

| Ensemble model | Meta model | Base models | Stacking SMOTE | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|---|---|---|
| Proposed model M1 | SVM One vs. Rest | DT, OvR, KNN, GNB | Random Under Sampler | **0.93** | **0.93** | **0.93** | **0.93** |
| | | | Boarder Line Smote | 0.91 | 0.92 | 0.91 | 0.91 |
| | | | SVMSMOTE | 0.91 | 0.91 | 0.91 | 0.91 |
| | | | ADASYN | 0.90 | 0.90 | 0.90 | 0.90 |
| | | | SMOTE | 0.92 | 0.92 | 0.92 | 0.92 |
| M2 | Decision tree | DT, OvR, KNN, GNB | Random Under Sampler | 0.89 | 0.89 | 0.89 | 0.88 |
| | | | Boarder Line Smote | 0.89 | 0.88 | 0.88 | 0.88 |
| | | | SVMSMOTE | 0.91 | 0.91 | 0.91 | 0.90 |
| | | | ADASYN | 0.89 | 0.89 | 0.89 | 0.88 |
| | | | SMOTE | 0.90 | 0.90 | 0.89 | 0.89 |
| M3 | K-nearest neighbors | DT, OvR, KNN, GNB | Random Under Sampler | 0.90 | 0.90 | 0.90 | 0.90 |
| | | | Boarder Line Smote | 0.90 | 0.90 | 0.90 | 0.89 |
| | | | SVMSMOTE | 0.91 | 0.91 | 0.90 | 0.91 |
| | | | ADASYN | 0.89 | 0.89 | 0.89 | 0.89 |
| | | | SMOTE | 0.91 | 0.91 | 0.91 | 0.91 |
| M4 | Gaussian Naïve Bayes | DT, OvR, KNN, GNB | Random Under Sampler | 0.91 | 0.91 | 0.91 | 0.91 |
| | | | Boarder Line Smote | 0.89 | 0.89 | 0.89 | 0.89 |
| | | | SVMSMOTE | 0.90 | 0.90 | 0.90 | 0.90 |
| | | | ADASYN | 0.88 | 0.88 | 0.88 | 0.88 |
| | | | SMOTE | 0.91 | 0.92 | 0.91 | 0.91 |

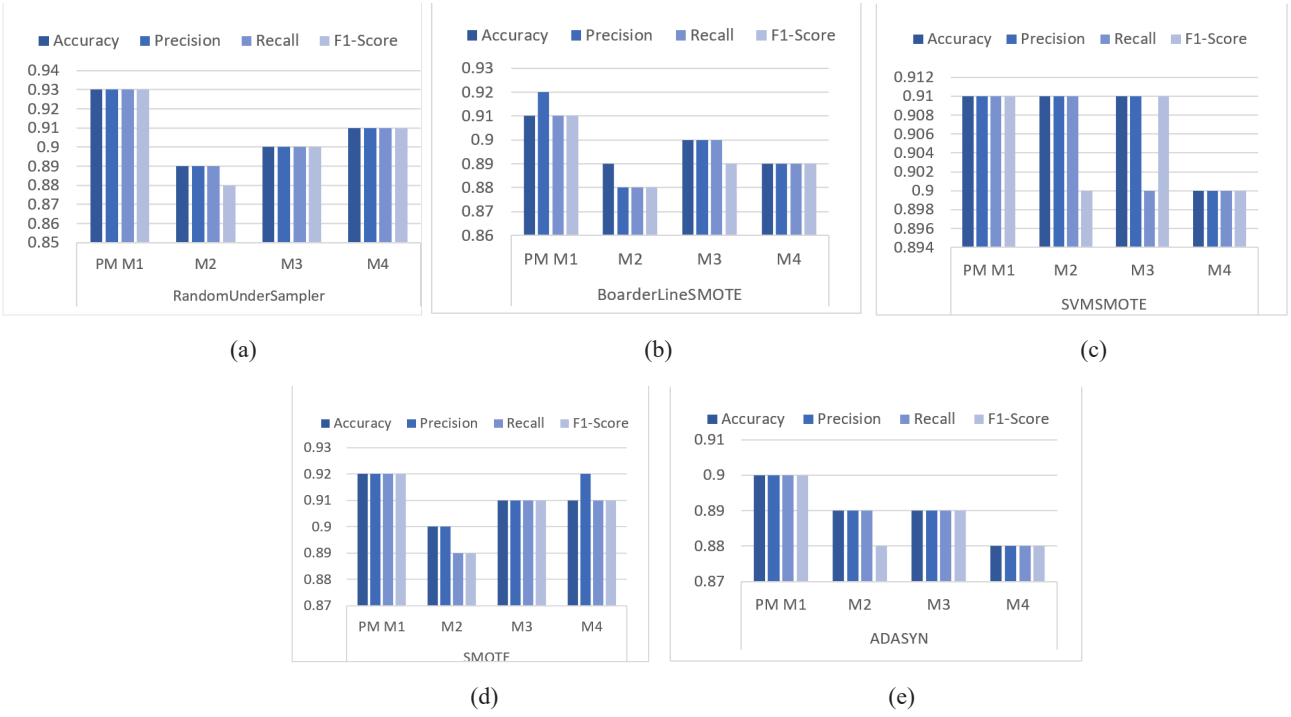


**Figure 11.** The performance comparison of the proposed model M1 with other ensemble models M2, M3, and M4. (a) Performance comparison in Random Under Sampler; (b) Boarder Line SMOTE; (c) SVMSMOTE; (d) SMOTE; (e) ADASYN.

the oversampling technique Random Under sampler SMOTE, that can predict students' performance at the earliest possible time so that necessary actions can be taken. Our proposed model predicts students' final grades at the end of the first semester with an accuracy of 93%, similarly, precision, recall and F1-Score up to 93%. In other variants of SMOTE techniques, our proposed model has shown the best performance out of the other existing models, as described in literature where authors have predicted the three classes students' grades with a maximum of 90% accuracy. The prediction accuracy in a research work was 98%, but that was for only one course and in our research, 5 courses have been included in the dataset so that prediction can be done in an environment where students of different academic caliber study. Our aim in designing this model is to provide the best prediction so that stu-

dents of any branch should succeed in their course. We successfully tested our model on PIET College data and presented the importance of stacking and SMOTE methods that can efficiently give better results to improve students' grade prediction. We elaborated that with the use of our proposed model, prediction accuracy improves from 80% to 90%. The proposed model can be implemented for other multiclass classification problems like disease prediction, image classification, fraud detection, and filtering emails. In our future work, we will use this model to predict the course of study for students at institute. We will further investigate the use of more appropriate ensemble machine learning models that can predict the output with greater accuracy. We conclude that our proposed model gives more accurate results and can help any educational institute improve their students' academic performance.

## Acknowledgements

I would like to express my sincere gratitude to the administrative staff of Panipat Institute of Engineering & Technology, Samalkha, Haryana, India, for their continuous support.

## Conflict of interest

The authors declare no conflict of interest.

## References

1. Jiang W, Chen Z, Xiang Y, *et al*. SSEM: A novel self-adaptive stacking ensemble model for classification. IEEE Access 2019; 7: 120337–120349. doi: 10.1109/ACCESS.2019.2933262.2.
2. Bujang SDA, Selamat A, Ibrahim R, *et al*. Multiclass prediction model for student grade prediction using machine learning. IEEE Access 2021; 9: 95608–95621. doi: 10.1109/ACCESS.2021.3093563.
3. Pang Y, Judd N, O'Brien J, Ben-Avie M. Predicting students' graduation outcomes through support vector machines. In: 2017 Frontiers in Education Conference (FIE); 2017 Oct 18–21; Indianapolis, IN, USA. New York: IEEE; 2017. doi: 10.1109/FIE.2017.8190666.
4. Ünal F. Data mining for student performance prediction in education. In: Birant D (editor). Data mining—Methods, applications, and systems. London: IntechOpen; 2021. doi: 10.5772/intechopen.91449.
5. Palacios CA, Reyes-Suárez JA, Bearzotti LA, *et al*. Knowledge discovery for higher education student retention based on data mining: Machine learning algorithms and case study in Chile. Entropy 2021; 23(4): 485. doi: 10.3390/e23040485.
6. Kaunang FJ, Rotikan R. Students' academic performance prediction using data mining. In: 2018 Third International Conference on Informatics and Computing (ICIC); 2018 Oct 17–18; Palembang, Indonesia. New York: IEEE; 2019. doi: 10.1109/IAC.2018.8780547.
7. Ruiz S, Urretavizcaya M, Rodríguez C, Fernández-Castro I. Predicting students' outcomes from emotional response in the classroom and attendance. Interactive Learning Environments 2020; 28(1): 107–129. doi: 10.1080/10494820.2018.1528282.
8. Cervera DEM, Parra OJS, Prado MAA. Forecasting model with machine learning in higher education ICFES exams. International Journal of Electrical and Computer Engineering 2021; 11(6): 5402–5410. doi: 10.11591/ijece.v11i6.pp5402-5410.
9. Sethi K, Jaiswal V, Ansari MD. Machine learning based support system for students to select stream (subject). Recent Advances in Computer Science and Communications 2020; 13(3): 336–344. doi: 10.2174/2213275912666181128120527.
10. Marbouti F, Diefes-Dux HA, Madhavan K. Models for early prediction of at-risk students in a course using standards-based grading. Computers & Education 2016; 103: 1–15. doi: 10.1016/j.compedu.2016.09.005.
11. Pushpa SK, Manjunath TN, Mrunal TV, *et al*. Class result prediction using machine learning. In: 2017 International Conference on Smart Technology for Smart Nation (SmartTechCon); 2017 Aug 17–19; Bengaluru, India. New York: IEEE; 2018. doi: 10.1109/SmartTechCon.2017.8358559.
12. Tuggener L, Amirian M, Rombach K, *et al*. Automated machine learning in practice: State of the art and recent results. In: 2019 6th Swiss Conference on Data Science (SDS); 2019 Jun 14; Bern, Switzerland. New York: IEEE; 2019. doi: 10.1109/SDS.2019.00-11.
13. Pavlyshenko B. Using stacking approaches for machine learning models. In: 2018 IEEE 2nd International Conference on Data Stream Mining and Processing (DSMP); 2018 Aug 21–25; Lviv, Ukraine. New York: IEEE; 2018. doi: 10.1109/DSMP.2018.8478522.
14. Xu J. An extended one-versus-rest support vector machine for multi-label classification. Neurocomputing 2011; 74(17): 3114–3124. doi: 10.1016/j.neucom.2011.04.024.
15. Trabelsi A, Elouedi Z, Lefevre E. Decision tree classifiers for evidential attribute values and class

labels. Fuzzy Sets and Systems 2019; 366: 46–62. doi: 10.1016/j.fss.2018.11.006.
16. Rezaeijo SM, Abedi-Firouzjah R, Ghorvei M, Sarnameh S. Screening of COVID-19 based on the extracted radiomics features from chest CT images. Journal of X-Ray Science and Technology 2021; 29(2): .229–243. doi: 10.3233/XST-200831.
17. Churcher A, Ullah R, Ahmad J, *et al*. An experimental analysis of attack classification using machine learning in IoT networks. Sensors (Switzerland) 2021; 21(2): 446. doi: 10.3390/s21020446.
18. Akçapınar G, Altun A, Aşkar P. Using learning analytics to develop early-warning system for at-risk students. International Journal of Educational Technology in Higher Education 2019; 16: 40. doi: 10.1186/s41239-019-0172-z.
19. Hutagaol N, Suharjito. Predictive modelling of student dropout using ensemble classifier method in higher education. Advances in Science, Technology and Engineering Systems 2019; 4(4): 206–211. doi: 10.25046/aj040425.
20. Rohilla N, Rai M. Advance machine learning techniques used for detecting and classification of disease in plants: A review. In: 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N); 2021 Dec 17–18; Greater Noida, India. New York: IEEE; 2021. doi: 10.1109/ICAC3N53548.2021.9725616.
21. Saluja R, Rai M. Analysis of existing ML techniques for students success prediction. In: 2022 Seventh International Conference on Parallel, Distributed and Grid Computing (PDGC); 2022 Nov 25–27; Solan, Himachal Pradesh, India. New York: IEEE; 2022. p. 507–512. doi: 10.1109/PDGC56933.2022.10053236.
22. Naseer M, Zhang W, Zhu W. Prediction of coding intricacy in a software engineering team through machine learning to ensure cooperative learning and sustainable education. Sustainability (Switzerland) 2020; 12(21): 8986. doi: 10.3390/su12218986.
23. Wang S, Jiang L, Li C. Adapting naive Bayes tree for text classification. Knowledge and Information Systems 2015; 44: 77–89. doi: 10.1007/s10115-014-0746-y.
24. Hussain S, Khan MQ. Student-Performulator: Predicting students' academic performance at secondary and intermediate level using machine learning. Annals of Data Science 2021; 10: 637–655. doi: 10.1007/s40745-021-00341-0.
25. Sorour SE, Goda K, Mine T. Evaluation of effectiveness of time-series comments by using machine learning techniques. Journal of Information Processing 2015; 23(6): 784–794. doi: 10.2197/ipsjjip.23.784.
26. Park HS, Yoo SJ. Early dropout prediction in online learning of university using machine learning. International Journal on Informatics Visualization 2021; 5(4): 347–353. doi: 10.30630/JOIV.5.4.732.
27. Singh M, Verma C, Kumar R, Juneja P. Towards enthusiasm prediction of Portuguese school's students towards higher education in realtime. In: 2020 International Conference on Computation, Automation and Knowledge Management (ICCAKM); 2020 Jan 9–10; Dubai, United Arab Emirates. New York: IEEE; 2020. doi: 10.1109/ICCAKM46823.2020.9051459.
28. Burman I, Som S. Predicting students academic performance using Support Vector Machine. In: 2019 Amity International Conference on Artificial Intelligence (AICAI); 2019 Feb 4–6; Dubai, United Arab Emirates. New York: IEEE; 2019. doi: 10.1109/AICAI.2019.8701260.
29. Altabrawee H, Ali OAJ, Ajmi SQ. Predicting students' performance using machine learning techniques. Journal of University of Babylon for Pure and Applied Sciences 2019; 27(1): 194–205.
30. Nti IK, Adekoya AF, Weyori BA. A comprehensive evaluation of ensemble learning for stock-market prediction. Journal of Big Data 2020; 7: 20. doi: 10.1186/s40537-020-00299-5.
31. Wibawa AS, Purwarianti A. Indonesian Named-entity Recognition for 15 classes using ensemble supervised learning. Procedia Computer Science 2016; 81: 221–228. doi: 10.1016/j.procs.2016.04.053.
32. Hu X, Zhang H, Mei H, *et al*. Landslide susceptibility mapping using the stacking ensemble machine learning method in Lushui, Southwest China. Applied Sciences 2020; 10(11): 4016. doi: 10.3390/app10114016.
33. Rahman M, Chen N, Elbeltagi A, *et al*. Application of stacking hybrid machine learning algorithms in delineating multi-type flooding in Bangladesh. Journal of Environmental Managemen 2021; 295: 113086. doi: 10.1016/j.jenvman.2021.113086.
34. Chung J, Teo J. Single classifier vs. ensemble machine learning approaches for mental health prediction. Brain Informatics 2023; 10: 1. doi: 10.1186/s40708-022-00180-6.
35. Smirani LK, Yamani HA, Menzli LJ, Boulahia JA. Using ensemble learning algorithms to predict student failure and enabling customized educational paths. Scientific Programming 2022; 2022: 3805235. doi: 10.1155/2022/3805235.
36. Barella VJ, Garcia LPF, de Souto MCP, *et al*. Assessing the data complexity of imbalanced datasets. Information Sciences 2021; 553: 83–109. doi:

10.1016/j.ins.2020.12.006.
37. Bej S, Davtyan N, Wolfien M, *et al*. LoRAS: An oversampling approach for imbalanced datasets. Machine Learning 2021; 110: 279–301. doi: 10.1007/s10994-020-05913-4.
38. Lemaître G, Nogueira F, Aridas CK. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. Journal of Machine Learning Research 2017; 18: 1–5.
39. Davagdorj K, Lee JS, Pham VH, Ryu KH. A comparative analysis of machine learning methods for class imbalance in a smoking cessation intervention. Applied Sciences 2020; 10(9): 3307. doi: 10.3390/app10093307.
40. Seo JH, Kim YH. Machine-learning approach to optimize smote ratio in class imbalance dataset for intrusion detection. Computational Intelligence and Neuroscience 2018; 2018: 9704672. doi: 10.1155/2018/9704672.
41. Ijaz MF, Alfian G, Syafrudin M, Rhee J. Hybrid Prediction Model for type 2 diabetes and hypertension using DBSCAN-based outlier detection, Synthetic Minority Over Sampling Technique (SMOTE), and random forest. Applied Sciences 2018; 8(8): 1325. doi: 10.3390/app8081325.