Students' Dropout Prediction

Aditya Akolkar
Drexel University
aaa588@drexel.edu

Chethas Anil Reddy
Drexel University
car445@drexel.edu

Manish Chandrashekar
Drexel University
mc4484@drexel.edu

Pranjalee Kshirsagar
Drexel University
pk658@drexel.edu

Vishal Hagargundgi
Drexel University
vh347@drexel.edu

March 18, 2024

Abstract

In the realm of education, particularly in the context of online learning environments, the challenge of predicting and mitigating student dropout rates looms large. This document encapsulates a meticulously crafted machine learning project dedicated to addressing this pertinent issue. Utilizing Logistic Regression, Naive Bayes, and Decision Tree algorithms, we selected these three models for ensemble alongside a thorough examination of demographic, academic, and behavioral data. Our project aims to untangle the complex interplay of factors influencing student attrition.

Armed with a meticulously curated dataset, the project embarks on the development of predictive models, each algorithm meticulously tuned to extract optimal performance.

The evaluation phase of the project serves as the litmus test, where the efficacy of each model is rigorously scrutinized. Metrics such as accuracy, precision, recall, and F1-score emerge as beacons guiding the way forward, providing invaluable insights into the strengths and weaknesses of each algorithm. It is in this crucible of evaluation that the true potential of these predictive models is unveiled, offering a glimmer of hope in the endeavor to stem the tide of student attrition.

1 Introduction

The issue of student dropout is a matter of significant concern globally, drawing continuous attention from the scientific community. While academic success is crucial for students, dropout rates persist at significant levels across nearly all countries, influenced by various factors that hinder students from completing their studies. Notably, a considerable portion of dropouts occurs within the initial weeks of enrollment [1], highlighting the urgency of addressing this issue. The repercussions of high dropout rates extend beyond individuals to impact institutions and society at large, signaling a need for educational quality enhancement [1]. Despite the multitude of factors contributing to dropout, researchers strive to forecast dropout rates, enabling the timely implementation of preventive measures and strategies. The analysis of dropout rates dates back several decades to seminal models developed by early researchers [2]. Since then, numerous models and techniques have emerged, garnering attention in deciphering the underlying factors driving student dropout [3].

Online learning emerges as an optimal solution for individuals juggling work commitments while pursuing education. With its flexibility free from temporal and spatial constraints, online learning facilitates knowledge acquisition and degree attainment, accommodating busy schedules seamlessly. The emergence of the COVID-19 pandemic further accelerated the shift from traditional face-to-face learning to online platforms across universities worldwide, aiming to mitigate virus transmission risks [4]. However, the autonomy and time management required in online learning environments also pose challenges, with dropout rates being a significant concern [5]. Universities offering online programs face the challenge of proactively supporting students at risk of dropping out, necessitating intervention from lecturers or administrators.

In the current era, Artificial Intelligence (AI) techniques are increasingly integrated into education, revolutionizing various aspects of higher learning. Many universities are embracing AI technologies, leveraging diverse Machine Learning methods to forecast enrollment trends, predict academic performance, and identify at-risk students who may dropout. This adoption of AI fosters quality enhancement and facilitates the implementation of proactive strategies to improve student retention [6]. Our research builds upon the foundation laid by previous studies, focusing on dropout prediction in online learning environments. We aim to implement logistic regression, naive Bayes, and decision tree algorithms using the UC Irvine machine learning dataset to refine and enhance the accuracy and interpretability of dropout prediction mod-

els. Additionally, we intend to explore socio-economic factors, study habits, and other variables that may contribute to dropout rates, providing a comprehensive understanding of this complex phenomenon. Ultimately, our study seeks to contribute to the field of educational data mining and student retention in online education by exploring novel approaches to predicting dropout rates and developing strategies to improve student outcomes and retention in online learning platforms.

2 Related Work

Predicting student dropout rates is of paramount importance for educational institutions and online platforms. Early identification of students at risk of dropping out allows for timely interventions and support mechanisms to be put in place, ultimately improving student retention rates and learning outcomes. The IEEE paper titled "Machine Learning Algorithms for Early Predicting Dropout Student Online Learning" by Meta Amalya Dewi, Felix Indra Kumiadi, Dina Fitria Murad, Sucianna Ghadati Rabiha and Awards Romli presents a comprehensive exploration of various machine learning algorithms applied to the task of predicting dropout rates in online learning environments.

The study conducted by the aurhors serves as a significant precursor to our research endeavor. By reviewing and analyzing the methodologies and findings presented in the IEEE paper, we gained valuable insights into the challenges and opportunities associated with dropout prediction in online education. The paper elucidates the complexities of the dropout prediction task and highlights the potential of machine learning algorithms in addressing this issue. Moreover, it provides a benchmark for evaluating the performance of different predictive models and serves as a reference point for comparing the efficacy of our proposed approach.

The research presented in the IEEE paper employs a diverse set of machine learning algorithms, including decision trees, logistic regression, support vector machines, and neural networks, to develop predictive models for dropout prediction. Through rigorous experimentation and evaluation, the authors assess the performance of these algorithms in terms of accuracy, precision, recall, and F1 score. Additionally, they investigate the impact of various features and data preprocessing techniques on the predictive capabilities of the models.

While the IEEE paper lays a solid foundation for dropout prediction in online learning, our research extends and builds upon its findings in several key aspects. We focus specifically on the implementation of logistic regression, naive bayes, and decision tree algorithms, for dropout prediction using the UC Irvine machine learning dataset. By delving deeper into the intricacies of the said approaches and leveraging the insights gained from the IEEE paper, we aim to enhance the accuracy and interpretability of dropout prediction models in online education settings.

Another study, conducted by [7], employed Gradient Boosting, Random Forest, Support Vector Machine, and Ensemble models to predict student dropout across various stages of university studies. These stages included pre-enrollment and the end of each semester during the first two years of study. In addition to academic data, the models integrated students' personal information, family background, environmental factors, and their interaction with online learning tools. The research reported an accuracy of 82.91% in detecting dropouts by the conclusion of the first semester, which increased to 91.5% by the end of the fourth semester.

Similarly, [8] achieved a dropout prediction accuracy ranging from 77% to 93% using data spanning four academic years. Their classification models, including Logistic Regression, Decision Tree, Random Forest, Naive Bayes, and SVM, utilized features such as course views, assignment scores, tests, examinations, and projects. These models demonstrated the capability to predict dropout probability with satisfactory precision as early as the conclusion of a single semester.

Furthermore, J. Niyogisubizo et al. presented a study in [9] employing ensemble machine learning techniques for predicting student dropout. Their dataset primarily comprised access logs, examination results, assignment grades, project submissions, and test scores. By employing Random Forest, Extreme Gradient Boosting, Gradient Boosting, and Feed-forward Neural Networks, their models achieved training accuracies ranging from 86.67% to 96.67% and testing accuracies from 76.67% to 92.18%. Such predictive capabilities offer valuable insights for universities seeking to identify and mitigate dropout risks effectively.

In another study, Cechinel et al [33], set out to identify students at risk of dropout by exploring optimal combinations of datasets and classification algorithms. They tested various classification algorithms, including Naive Bayes, Random Forest, AdaBoost, Multilayer Perceptron, K-Nearest Neighbor, and Decision Tree, using student data sourced from the Moodle platform. Across 13 datasets featuring diverse data features, AdaBoost emerged as particularly effective.

Similarly, Lorenz et al[34], applied logistic regression and decision trees to anticipate student dropout occurrences at the Karlsruhe Institute of Technology (KIT).The study proposed a methodical approach easily adaptable to other institutions. Findings revealed a slight superiority of decision trees over logistic regression in prediction accuracy. Notably, both methods achieved remarkably high accuracies of up to 95ing robust classification performance exceeding 83the first semester.

In conclusion, these papers offer essential context and groundwork for our research endeavor, shaping our investigation into machine learning algorithms for predicting dropout rates in online learning settings. Drawing upon the methodologies and insights laid out in these papers,

we aspire to introduce innovative approaches and advancements to the realm of educational data mining and student retention in online education.

3 Methodology

3.1 Dataset Description:

The dataset, sourced from the UCI Machine Learning Repository, is known for its high-quality datasets. It offers a comprehensive perspective on various attributes concerning undergraduate students, encompassing demographics, socio-economic factors, and academic performance[31]. Presented in a tabular format, with rows representing individual students and columns detailing their attributes, it is well-suited for scrutinizing factors influencing student dropout and academic achievement across different fields of study. The dataset includes diverse attributes such as marital status, application mode, course, attendance schedule, previous qualifications, nationality, parental qualifications and occupations, special needs, financial status, gender, scholarship status, age at enrollment, and international status. Moreover, it incorporates additional metrics concerning curricular units in the first semester, along with external economic indicators like unemployment rate, inflation rate, and GDP. In approaching the task of predicting student dropout in online learning, a systematic methodology involves initial data preprocessing steps to handle missing values, eliminate irrelevant features, and encode categorical variables into numerical representations. Subsequently, the dataset is partitioned into training and testing sets for model evaluation, followed by the utilization of Ensemble techniques to enhance accuracy and resilience by amalgamating predictions from multiple models. In Figure 1, the dataset exclusively encompasses essential columns alongside their respective value types, offering a concise representation of the key attributes

Column Name	Description
Marital status	The marital status of the student. (Categorical)
Application mode	The method of application used by the student. (Categorical)
Application order	The order in which the student applied. (Numerical)
Course	The course taken by the student. (Categorical)
Daytime/evening attendance	Whether the student attends classes during the day or in the evening. (Categorical)
Previous qualification	The qualification obtained by the student before enrolling in higher education. (Categorical)
Nationality	The nationality of the student. (Categorical)
Mother's qualification	The qualification of the student's mother. (Categorical)
Father's qualification	The qualification of the student's father. (Categorical)
Mother's occupation	The occupation of the student's mother. (Categorical)
Father's occupation	The occupation of the student's father. (Categorical)
Displaced	Whether the student is a displaced person. (Categorical)
Educational special needs	Whether the student has any special educational needs. (Categorical)
Debtor	Whether the student is a debtor. (Categorical)
Tuition fees up to date	Whether the student's tuition fees are up to date. (Categorical)
Gender	The gender of the student. (Categorical)
Scholarship holder	Whether the student is a scholarship holder. (Categorical)
Age at enrollment	The age of the student at the time of enrollment. (Numerical)
International	Whether the student is an international student. (Categorical)
Curricular units 1st sem (credited)	The number of curricular units credited by the student in the first semester. (Numerical)
Curricular units 1st sem (enrolled)	The number of curricular units enrolled by the student in the first semester. (Numerical)
Curricular units 1st sem (evaluations)	The number of curricular units evaluated by the student in the first semester. (Numerical)
Curricular units 1st sem (approved)	The number of curricular units approved by the student in the first semester. (Numerical)

Figure 1. Features in Student dropout dataset.

The Figure 2 below illustrates the distribution of students based on their graduation status, with the label "0" representing graduates and "1" denoting dropouts. Notably, the dropout count surpasses more than half of the graduate count, underscoring a significant disparity. This discrepancy underscores the importance of our research, as it offers valuable insights for organizations seeking to identify students at risk of dropout and provide timely interventions to support them.

Figure 2. Count of Target values.

The Figure 3 represents the distribution of graduates (Target = 0) and dropouts (Target = 1) categorized by gender. Notably, it reveals a nearly equal number of male and female students who dropped out. However, a notable disparity emerges in the graduation figures, with the count of female graduates exceeding more than twice the count of male graduates.

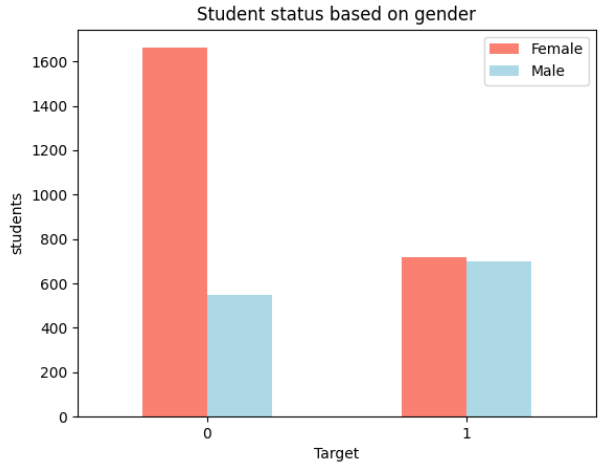

Figure 3. Count of Target values based on gender.

In our study, we employed various data analysis techniques to explore the relationships within our dataset and gain insights into the factors influencing student outcomes. One such technique that proved instrumental in our analysis was the use of correlation heatmaps. A correlation heatmap(Figure 4) is a graphical representation of the correlation matrix of a dataset, where each cell in

the heatmap represents the correlation coefficient between two variables. These coefficients quantify the strength and direction of the linear relationship between variables, ranging from -1 to 1. Positive correlations in our dataset are depicted in dark blue, while negative correlations are represented by light yellow. The intensity of the color reflects the strength of the correlation, with darker shades indicating stronger correlations.

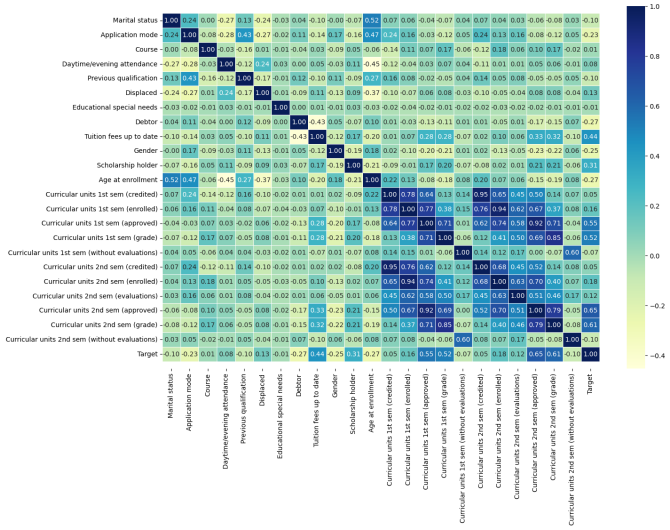

Figure 4. Heatmap Correlations.

By visualizing the correlations between variables in our dataset using a heatmap, we were able to quickly identify patterns and relationships that might not have been immediately apparent from raw data alone. For instance, if we observe a dark blue cell between "Scholarship holder" and "Tuition fees up to date", it suggests that students with scholarships tend to keep their tuition fees up to date. Conversely, a dark green cell between "Age at enrollment" and "Debtor" implies that older students are more likely to be debtors. While correlation coefficients provide insights, it's essential to consider statistical significance. A high correlation coefficient doesn't always imply causation.

3.2 Machine Learning Techniques:

3.2.1 Logistic Regression:

Logistic Regression (LR) [10][11] is a statistical predictive model. For classification, the relationship between the dependent variable and one or more independent variables is measured by estimating probabilities using a logistic function. It calculates a weighted average of a set of variables, which are known as covariates [11] and are provided as an input to the logistic function. The input to the logistic function, z, is shown in equation as below [10]:

$$z = \beta_0 + \sum_{i=1}^{n} \beta_i x_i$$

Where, β_1 to β_n are the coefficients for the feature values, x_1 to x_n. β_0 is a constant. The logistic function is given by equation (3) as below [10], which gives the probability of prediction class:

$$y = \frac{1}{1+e^{-z}}$$

Where z is the logit (logistic) function, y gives the dropout probability of the student; the value of y as 1 and 0, which represents student as a dropout or completer respectively. It is simple, can handle high dimensional and large dataset. However, it cannot handle noisy data; also, it is susceptible to overfitting [11].

3.2.2 Naïve Bayes

The Naïve Bayes algorithm is a classification technique that employs statistical and probabilistic principles. It was introduced by Thomas Bayes [22]. The primary components of the Naïve Bayes Classifier pertain to four fundamental aspects: prior, posterior and conditional probability[23]. The Naive Bayes algorithm offers several advantages, including its ability to perform well with small training data, simplicity in computation and implementation, scalability for handling large datasets, robustness in dealing with incomplete data and tolerance to data noise. [22][24]-[25]

The Naïve Bayes Classifier (NBC) employs a general formula,

$$P(Q|X) = \frac{P(X|Q) \cdot P(Q)}{P(X)}$$

Where:

X - Data with unknown class

Q - The hypothesis X is a specific class

$P(Q|X)$ - Probability of the hypothesis Q given X

$P(Q)$ - Probability of hypothesis Q

$P(X|Q)$ - Probability X in the hypothesis Q

$P(X)$ - Probability X

In order to grasp the Naïve Bayes theorem, it's essential to understand that the classification process relies on gathering multiple pieces of evidence to discern the class of the analyzed sample..Thus, the Bayes theorem is adapted as follows,

$$P(Q|X_1...X_n) = \frac{P(X_1...X_n|Q) \cdot P(Q)}{P(X_1...X_n)}$$

When considering a variable Q as a representation of class and variables $X_1...X_n$ representing the characteristics of the instructions necessary for the classification process, the formula elucidates that the likelihood of certain characteristic samples entering class Q (Posterior) is the product of the probability of the occurrence of class (Prior to sample entry) and the probability of sample characteristics appearing in class Q (Likelihood). This product is then divided by the probability of sample characteristics

appearing globally (Evidence).Consequently, the formula can be expressed as follows,

$$Posterior = \frac{Likelihood * Prior}{Evidence}$$

In a sample, the evidence values remain constant for each class. Comparing the posterior value of one class with those of other classes helps determine the classification of the sample.

3.2.3 Decision Trees

The Decision Tree algorithm is a popular machine learning technique used for classification and regression tasks[13]. It constructs a tree-like model of decisions based on feature values, where each internal node represents a feature, each branch represents a decision based on that feature, and each leaf node represents a class label or prediction. The ID3 algorithm (Iterative Dichotomiser 3) is a classic Decision Tree algorithm introduced by Ross Quinlan[12]. It recursively builds the tree by selecting the best attribute at each node based on Information Gain.

ID3 is known for its simplicity and interpretability, making it suitable for tasks where model transparency is essential. However, it tends to overfit the training data, especially when dealing with noisy datasets or features with high cardinality. In this research study, the ID3 algorithm is implemented using the entropy-based Information Gain heuristic. At each node of the tree, the algorithm selects the attribute that maximizes Information Gain, which measures the reduction in entropy or impurity of the dataset after the split. The tree is grown recursively until certain stopping criteria, such as maximum tree depth or minimum samples per leaf, are met.

Here are the mathematical calculations performed in the original program code [16]:

1. **Entropy Calculation**

   The entropy (H) of a set of data is calculated using the formula:

   $$H(S) = -\sum_{i=1}^{n} p_i \log_2(p_i)$$

   Where:

   - S is the set of data
   - p_i is the probability of occurrence of class i
   - n is the number of classes

2. **Information Gain Calculation**

   The information gain (IG) of an attribute is calculated using the formula:

   $$IG(S, A) = H(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} H(S_v)$$

   Where:

   - S is the set of data
   - A is the attribute
   - $Values(A)$ is the set of all possible values of attribute A
   - S_v is the subset of S where attribute A has value v

3.2.4 Ensemble

Ensemble learning is a methodology that strategically integrates multiple models to address specific computational intelligence challenges. Its primary objective is to improve model performance across various tasks such as classification, prediction, and function approximation, while also mitigating the risk of selecting suboptimal models. Additionally, ensemble learning serves purposes such as decision validation, model selection, data fusion, incremental and non-stationary learning, as well as error correction[29].

- **Majority Voting:**

  In majority voting, the dataset is partition into several sub-sets and various classifiers are used to train the same, thereby accomplishing the learning phase. The final results of allocating a label to the sample depends on the maximum number of votes received in favor of a particular class obtained from different classifiers[30][32].

Algorithm 1 Simple Majority Voting

Require: List of classifiers: $H = \{h_1, h_2, ..., h_T\}$
Require: Unlabeled instance: x

```
procedure SimpleMajorityVoting(H, x)
    Initialize a dictionary to store the votes for each class:
    votes ← {Θ_1 : 0, Θ_2 : 0, ..., Θ_C : 0}
    for t = 1 to T do
        Let prediction ← h_t(x)
        Increment the vote count for the predicted class:
        votes[prediction] += 1
    end for
    Find the class with the highest total votes:
    max_votes_class ← argmax(votes.values())
    Return the class with the highest total votes as the final classification:
    final_class ← max_votes_class
    return final_class
end procedure
```

3.3 Results and Analysis

3.3.1 Classification Metrics

This section presents the evaluation results of the four algorithms studied in this research: Decision Tree (DT), Algorithm B, Algorithm C, and Algorithm D. The algorithms were assessed based on four key metrics: Accuracy, Precision, Recall, and F1 Score. These metrics provide a comprehensive understanding of each algorithm's performance, allowing for a fair comparison.

- **Accuracy:**

  By dividing the total number of kinds (classes) by the number of correctly predicted classes, accuracy is calculated. Consequently, it evaluates the overall quality or accuracy of the classification.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

  where:

  - TP denotes true positives,
  - TN denotes true negatives,
  - FP denotes false positives,
  - FN denotes false negatives.

- **Precision:**

  The percentage of actual negatives that are predicted to be negative is called precision. It calculates the percentage of genuine positives between the predicted true positives and false positives.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

- **Recall:**

  It is the percentage of outcomes that are actually anticipated to be positive. Recall makes ensuring that some students who fail or pass are not overlooked by the predictive model. Recall is a tool that is used to assess the real success rate of students who graduate.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

- **F1 Score:**

  The F1 score establishes the memory harmonic average and the predictive model's precision. Therefore, classification issues with imbalanced target labels are appropriate for the F1 score.

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

3.3.2 Logistic Regression:

A) **Data Pre-processing:**

The data preprocessing pipeline for the dropout prediction analysis encompassed several key steps to ensure the integrity and relevance of the dataset:

a) **Data Filtering:**
Students who were already enrolled were filtered out from the dataset, focusing the analysis solely on students who were at risk of dropping out.

b) **Target Variable Encoding:**
The target variable, denoting whether a student dropped out or not, was encoded into a binary format. This transformation facilitated the classification task, with 'Dropout' assigned a value of 1 and other outcomes assigned a value of 0.

c) **Feature Selection:**
Certain columns deemed extraneous or redundant for dropout prediction were removed from the dataset. These included demographic factors like nationality and parental occupation, as well as economic indicators such as GDP and inflation rate.

d) **Data Randomization:**
To mitigate potential biases stemming from the order of entries, the dataset was randomly shuffled. This ensured that the model training process was not influenced by any systematic patterns in the data arrangement.

e) **Data Standardization:**
Input features were standardized to have a mean of 0 and a standard deviation of 1. This normalization technique ensured consistent scaling across features, aiding in the convergence of the optimization algorithm during model training.

Each of these preprocessing steps played a critical role in preparing the dataset for subsequent analysis, contributing to the robustness and reliability of the dropout prediction model employed in the study.

Experiment Analysis:
After conducting logistic regression training and classification, a series of overlaid plots depicting the training and validation log losses across epochs were generated. In Figure 5, a learning rate of 3.0 was employed, resulting in noticeable oscillations in both training and validation losses. This oscillatory behavior indicates that the learning rate was excessively high, hindering effective generalization. To address this issue and promote better generalization, the learning rate was reduced to 0.1, as illustrated in Figure 6.

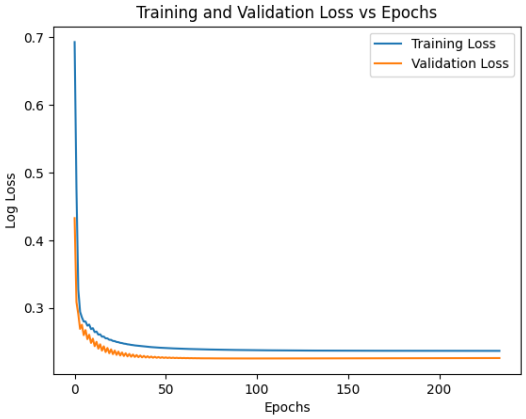

Figure. 5. Logistic Regression - Learning Rate = 3.0

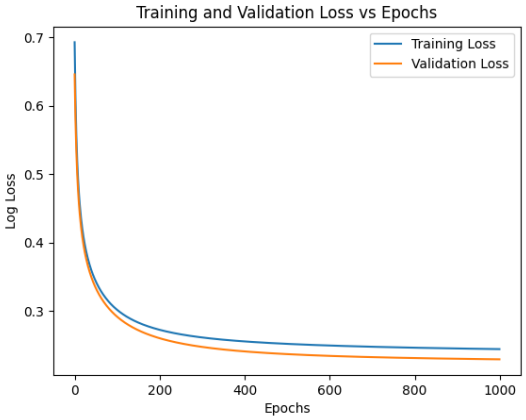

Figure. 6. Logistic Regression - Learning Rate = 0.1

Figure 6 demonstrates an improvement in generalization without the presence of oscillations in the log loss. This refined learning rate was subsequently utilized for classification and predicting target values. Conversely, when the learning rate was decreased to 0.01 (Figure 7), there was an initial increase observed in both training and validation losses compared to the scenario with a learning rate of 0.1.

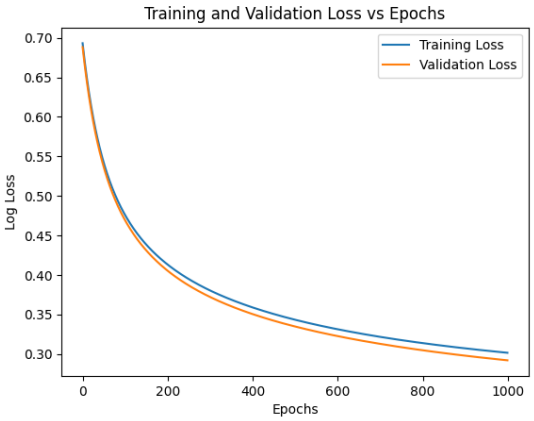

Figure 7. Logistic Regression - Learning Rate = 0.01

Comparing the two graphs: In Figure 6 (learning rate 0.1), both Training and Validation Loss decrease sharply initially and then level off, with Training Loss being consistently lower. In Figure 7 (learning rate 0.01), Training Loss starts higher but decreases more rapidly than Validation Loss. By around epoch 200, both losses stabilize with Training Loss being lower than Validation
The logistic regression confusion matrix is shown on Figure 8. This model has a relatively high number of true positives and true negatives, indicating a good ability to predict the target class.

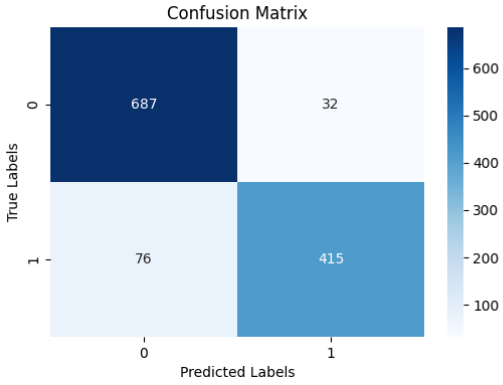

Figure. 8. Logistic Regression - Confusion Matrix

The logistic regression model achieved an accuracy of 0.9107, indicating that it accurately classified 91.07% of the instances in the dataset.

Metric	Validation results
Precision	0.8930817610062893
Recall	0.8676171079429735
F-1 Score	0.8801652892561984
Accuracy	0.9041322314049587

Figure. 9. Logistic Regression - Metrics

The dataset includes features that are highly indicative of student dropout behavior. These features could encompass various aspects such as academic performance, socio-economic background, and demographic information. By incorporating these pertinent features into the model, it can discern patterns that distinguish between students who drop out and those who don't.

Logistic regression is a well-suited model for binary classification tasks like dropout prediction, particularly when the relationship between features and the target variable is approximately linear. This model's simplicity and interpretability make it effective for analyzing relationships between predictors and the likelihood of dropout.

The choice of optimization algorithm and hy-

perparameters significantly influences model performance. stochastic gradient descent (SGD) optimization with a carefully chosen learning rate of 0.1 ensures stable convergence without oscillations or divergence. This optimal tuning of hyperparameters enables the model to efficiently learn from the data and make accurate predictions.

Standardizing the features to have a mean of 0 and a standard deviation of 1 ensures that they are on a similar scale. This preprocessing step facilitates the convergence of optimization algorithms, prevents dominance by features with larger scales, and improves the stability and performance of the model.

The accuracy metric, which measures the proportion of correctly classified instances out of the total instances, indicates that the model is making accurate predictions for a significant portion of the dataset. However, it's essential to complement accuracy with other evaluation metrics such as precision, recall, and F1-score to gain a more comprehensive understanding of the model's performance, particularly in imbalanced datasets.

The implementation of an early stopping criterion, as demonstrated in the code with a predefined tolerance threshold, prevents overfitting by halting the training process when the improvement in performance becomes negligible. This ensures that the model generalizes well to unseen data and doesn't memorize the training set.

In summary, the high accuracy achieved by the logistic regression model can be attributed to a combination of relevant features, effective model selection, optimal tuning of hyperparameters, high-quality data, standardized feature scaling, careful evaluation, and the implementation of an early stopping mechanism. These factors collectively contribute to the model's ability to accurately predict student dropout and facilitate proactive interventions in higher education institutions.

3.3.3 Naïve Bayes:

A) **Data Pre-processing:**

The following Data pre-processing is performed to clean, transform, and prepare the data for Naïve Bayes algorithm by handling missing values, encoding categorical variables, selecting relevant features, and splitting the dataset into training and testing sets

a) **Loading the Dataset:**
The dataset is loaded from a CSV file and read using pandas library function. The separator ';' is specified as the delimiter.

b) **Data Filtering:**
Rows where the 'Target' variable is equal to 'Enrolled' are filtered out of the dataset. This is done using boolean indexing

c) **Target Variable Encoding:**
The 'Target' variable is converted into a binary format. 'Dropout' is encoded as 1, and other classes are encoded as 0.

d) **Dropping Columns:**
Several columns that are not pertinent to the for the classification task are dropped from the dataset. This is done using the drop() method.

e) **Splitting the Data:**
The dataset is split into features and the target variable. This is done using the 'values' attribute to convert the data into numpy arrays.

Overall, these pre-processing steps involve cleaning the data, encoding categorical variables, dropping columns which are not pertinent, and preparing the data for training the Naive Bayes classifier.

Experiment Analysis:

The Naïve Bayes classifier trained on the provided dataset demonstrates promising performance in predicting student dropout.When employing a training ratio of 2/3 and testing ratio of 1/3, the Naive Bayes classifier demonstrates a notable performance in predicting student dropout rates. In figure 10, the testing accuracy of 82.22% reflects the classifier's ability to effectively generalize to unseen data. Additionally, the precision metric for the testing data, standing at 80.73%, underscores the classifier's capability to accurately identify true dropout cases among the predicted instances. However, the recall metric at 73.55% indicates a moderate ability to capture all actual dropout cases. This balance between precision and recall is further highlighted by the F-measure of 76.97%. The confusion matrix in the figure 11, depicts 629 true negative predictions and 356 true positive predictions, with 85 false positive instances and 128 false negative instances.Overall, while the classifier demonstrates commendable accuracy and precision, there is room for improvement in recall, which is due to overfitting and imbalanced class distribution

Metric	Validation Result
Accuracy	0.82220367278798
Precision	0.80725623582766
Recall	0.73553719008264
F-measure	0.76972972972972

Figure. 10.Naïve Bayes - Metrics

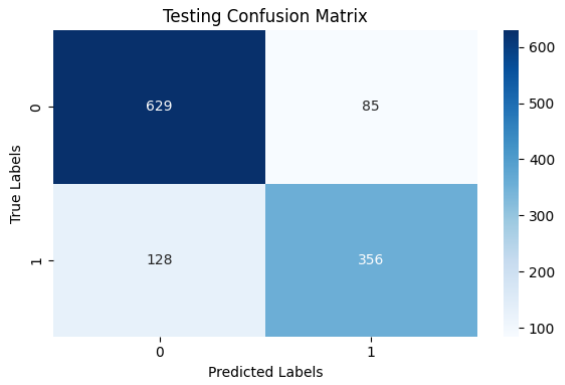

Figure. 11.Naïve Bayes - Confusion Matrix

Additionally, employed a experimental analysis conducted on the Naïve Bayes classifier for predicting student dropout rates shares similar methods of experiments conducted by Haviluddin et al. [28]. The performance metrics including accuracy, precision, recall, and F-measure were systematically evaluated for training data ratios ranging from 10% to 90%.

In the Figure 11, while examining the training accuracy, it is observed that as the proportion of training data increases, there is a tendency for the accuracy to improve slightly, although with fluctuations. The highest training accuracy of 85.7% was achieved with a 10% training data ratio, while the lowest was recorded at 82.49% with a 70% training data ratio. This suggests that the classifier effectively learns from the available training data, but the extent of improvement diminishes as more data is added.

Similarly, testing accuracy demonstrates a similar trend, with slight fluctuations across different training data ratios. The highest testing accuracy of 83.19% was attained with a 60% training data ratio, indicating the classifier's ability to generalize well to unseen data. However, it is noteworthy that the testing accuracy does not consistently mirror the training accuracy, underscoring the importance of robust evaluation on unseen data.

Training Data Ratio	Training Accuracy	Testing Accuracy	Training Precision	Training Recall	Training F-measure	Testing Precision	Testing Recall	Testing F-measure
10%	0.8457	0.8292	0.8478	0.7697	0.8068	0.8041	0.7407	0.7711
20%	0.8375	0.8230	0.8210	0.7782	0.7782	0.8015	0.7217	0.7595
30%	0.8503	0.8252	0.8493	0.7569	0.8004	0.8079	0.7229	0.7630
40%	0.8493	0.8252	0.75694	0.8004	0.8503	0.8079	0.7229	0.7630
50%	0.8296	0.8264	0.7464	0.7858	0.8435	0.8167	0.7275	0.7695
60%	0.8286	0.8319	0.7349	0.7790	0.8374	0.8228	0.7307	0.7740
70%	0.8249	0.8301	0.7357	0.7777	0.8378	0.8200	0.7437	0.7800
80%	0.8347	0.8415	0.7266	0.7769	0.8402	0.8439	0.7702	0.8054
90%	0.8429	0.8484	0.7290	0.7818	0.8423	0.8333	0.8064	0.8196

Figure. 12. Metrics of Different Training Data Ratio

Precision and recall metrics offer insights into the classifier's ability to correctly identify dropout cases and avoid false positives. Across the experiments, precision values ranged from approximately 72.29% to 84.78%, while recall values varied between approximately 75.95% and 80.04%. These metrics highlight the trade-off between precision and recall, emphasizing the classifier's performance in correctly identifying true positives while minimizing false positives.

Furthermore, the F-measure, which combines precision and recall into a single metric, provides a comprehensive assessment of the classifier's overall performance. F-measure values ranged from approximately 76.30% to 80.79%, indicating a balance between precision and recall across different training data ratios.

Overall, the experimental analysis underscores the effectiveness of the Naive Bayes classifier in predicting student dropout rates. The results, as depicted in the figure 6, suggest that with an adequate amount of training data, the classifier can achieve robust performance, offering valuable insights for educational institutions to identify and potentially intervene with students at risk of dropping out. These findings provide a solid foundation for further research and practical applications in educational data analytics.

3.3.4 Decision Trees:

A) **Data Pre-processing:**
Before applying the Decision Tree algorithm, the dataset undergoes preprocessing to ensure its suitability for the task. The following preprocessing steps are performed:

(a) **Loading the Data:** The dataset is loaded into memory from CSV files using pandas $read_csv$ function [20].

(b) **Data Formatting:** Categorical target variables, representing students' dropout status or academic success, are encoded into numerical values for ease of processing. For example, 'Dropout' is encoded as 0, 'Graduate' as 1, and 'Enrolled' as 2 [14][15].

(c) **Feature Selection:** Irrelevant or redundant features are removed from the dataset to improve the efficiency and effectiveness of the Decision Tree algorithm. This may involve dropping columns that have little predictive power or are highly correlated with other features.

(d) **Data Scaling:** The features in the dataset are scaled to a similar range using a StandardScaler class, which is a manual simulation of sklearn preprocessing. This step ensures that features with larger scales do not dominate the learning process [17].

(e) **Partitioning the Data:** The dataset is partitioned into training and testing sets to evaluate the performance of the Decision Tree algorithm. This partitioning is typically done such that two-third data is used for training and one-third for validation.

B) **Parameter Hyper-tuning:**

Parameter hyper-tuning is a crucial step in optimizing the performance of the Decision Tree algorithm. It involves finding the optimal values for the hyperparameters of the Decision Tree model to improve its predictive accuracy and generalization ability. In this research study, the following hyperparameters of the Decision Tree algorithm are tuned:

(a) **Max Depth:** The maximum depth of the Decision Tree determines the maximum number of levels in the tree. A deeper tree may capture more complex patterns in the data but risks overfitting. To find the optimal value for the max depth hyperparameter, a range of values is typically explored, and cross-validation techniques, such as k-fold cross-validation, are used to evaluate the model's performance on different subsets of the training data as described in [19].

(b) **Min Samples Split:** The min samples split hyperparameter determines the minimum number of samples required to split an internal node as described in [21]. Setting a higher value for this hyperparameter prevents the Decision Tree from splitting nodes that have fewer samples, which can help prevent overfitting. Similar to max depth, the optimal value for min samples split is found through cross-validation.

(c) **Min Samples Leaf:** The min samples leaf hyperparameter specifies the minimum number of samples required to be at a leaf node. It controls the size of the leaf nodes and affects the granularity of the Decision Tree. By setting a higher value for min samples leaf, the Decision Tree avoids creating leaf nodes with very few samples, which can lead to overfitting.

(d) **Random State:** The random state hyperparameter controls the random number generator used for randomizing certain aspects of the Decision Tree algorithm, such as the randomness in feature selection during tree building. Setting a fixed value for random state ensures reproducibility of results across multiple runs of the algorithm.

The parameter hyper-tuning process involves systematically searching the hyperparameter space, typically using techniques such as grid search or randomized search. Grid search exhaustively searches through a specified grid of hyperparameter values, while randomized search randomly samples from the hyperparameter space. Cross-validation is used to evaluate the performance of different hyperparameter configurations and select the optimal one based on a predefined evaluation metric, such as accuracy, precision, or F1 score.

C) **Evaluation Results:**

The Decision Tree model demonstrated exceptional performance, achieving an accuracy of 88.82%[19]. This high level of accuracy indicates that the model is highly effective in correctly predicting the dropout status of students, thus showcasing its reliability in educational settings. Precision, a measure of the model's ability to identify true positive instances among all positive predictions, was recorded at 98.06%. This suggests that the model is remarkably precise in distinguishing students who are at risk of dropping out.

Furthermore, the model achieved a recall of 95.22%, indicating its strength in identifying nearly all actual dropout cases within the dataset. This capability is vital for interventions in educational environments, ensuring that at-risk students are not overlooked. The F-1 Score, which harmonizes precision and recall into a single metric, stood at 90.88%. This balance between precision and recall underscores the model's overall effectiveness in identifying true dropout cases while minimizing false positives and negatives.

Metric	Result
Accuracy	88.82%
Recall	95.22%
Precision	86.92%
F1 score	90.88%

Figure. 13. Decision Trees - Metrics

Analyzing the confusion matrix as shown in figure 14 further elucidates the model's predictive capabilities. The matrix, [[402, 102], [34,678]], reveals that the model correctly identified 402 true negatives and 678 true positives. Meanwhile, it misclassified only 102 instances as false positives and 34 as false negatives. This balance between sensitivity and specificity underlines the model's robustness in accurately identifying both classes of the target variable

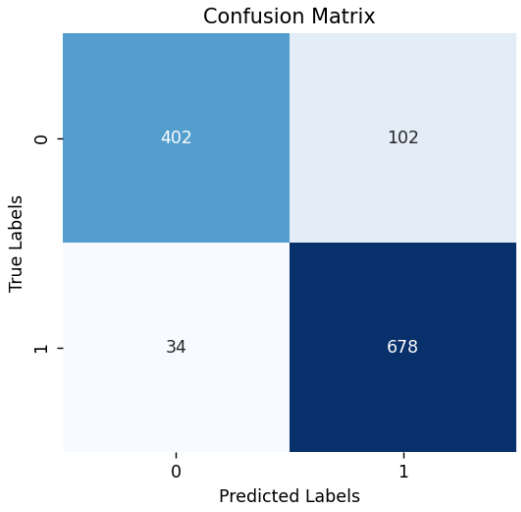

Figure. 14. Decision Trees - Confusion Matrix

Metric	Validation
Accuracy	0.86060
Precision	0.94647
Recall	0.69421
F1 Score	0.80095

Table. 1. Ensemble Metrics

Overall, the ensemble model demonstrated strong performance in predicting student dropout, achieving high accuracy, recall, precision, and F1 score. This suggests that the ensemble approach, leveraging the strengths of multiple classifiers, can be effective in addressing the complex task of identifying at-risk students and potentially improving retention efforts in educational institutions.

3.3.5 Ensemble:

The ensemble model achieved an accuracy of approximately 86.1%, indicating that it correctly classified the majority of the samples in the validation set. This suggests that the ensemble approach, combining predictions from multiple classifiers, effectively captured the underlying patterns in the data.

The lower-than-desired recall score of approximately 69.4% suggests that the ensemble model may have missed a considerable proportion of actual positive cases. This could be attributed to the complexity of the dropout prediction task and the inherent difficulty in capturing all relevant patterns from the available data. Additionally, the ensemble's reliance on individual classifiers, each with its own limitations, may have contributed to the suboptimal recall performance.

The precision of the ensemble model was approximately 94.6%. Precision measures the proportion of positive predictions made by the model that were correct. A high precision value suggests that the model made few false positive predictions, accurately identifying most of the predicted positive cases.

The F1 score, which is the harmonic mean of precision and recall, was approximately 80.1%. The F1 score provides a balanced measure of a model's performance, taking into account both precision and recall. The relatively high F1 score indicates that the ensemble model achieved a good balance between precision and recall, effectively identifying dropout cases while minimizing false positives.

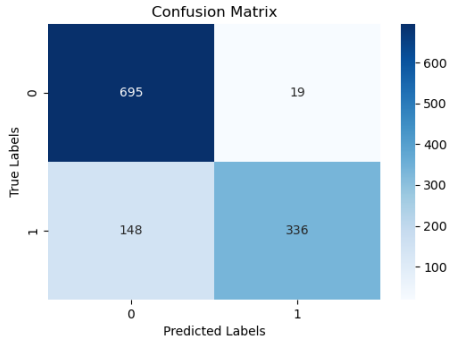

Figure. 15. Ensemble - Confusion Matrix

4 Conclusion:

In this study, we conducted a comprehensive analysis to assess the effectiveness of three machine learning models: Logistic Regression, Naive Bayes, and Decision Trees, using the same dataset for predicting student dropout. Based on the performance metrics—accuracy, precision, recall, and F-1 score—we have drawn several conclusions regarding the models' predictive capabilities.

Our findings reveal that Decision Trees exhibit promising performance, achieving an accuracy of 88.82%, recall of 95.22%, precision of 86.92%, and an F1 score of 90.88%. Despite slightly lower accuracy compared to Logistic Regression, Decision Trees demonstrate superior recall, emphasizing their proficiency in correctly identifying students at risk of dropout.

Logistic Regression, with an accuracy of 90.41%, recall of 86.76%, precision of 89.30%, and an F1 score of 88.01%, presents a competitive alternative for student dropout prediction. While it excels in overall accuracy and precision, its comparatively lower recall suggests a higher tendency to miss identifying students who may potentially dropout.

On the other hand, Naive Bayes, although less accurate than the other models with an accuracy of 82.22%,

recall of 73.53%, precision of 80.72%, and an F1 score of 76.97%, still exhibits reasonable predictive performance. Its simplicity and efficiency make it an attractive option for preliminary analysis or scenarios with computational constraints, despite its marginally inferior performance.

The ensemble majority voting method presented in this study exhibited promising performance, particularly highlighted by its high precision, reaching approximately 94.6%. However, the observed lower recall rate, approximately 69.4%, indicates a potential limitation in capturing all instances of positive cases. This discrepancy might stem from the inherent complexity of dropout prediction tasks, where certain nuanced patterns may not be adequately captured by the employed classifiers.

After a comprehensive evaluation, it is evident that Decision Trees emerge as the most effective model for student dropout prediction within the scope of our study. Their balanced performance across all metrics, particularly their high recall rate, positions them as a robust tool for identifying at-risk students. However, it's essential to acknowledge the strengths and weaknesses of each model, as Logistic Regression and Naive Bayes offer valuable insights and may be suitable in specific contexts or for different objectives.

In conclusion, while Decision Trees stand out as the preferred model for student dropout prediction in our research, the selection of the most appropriate model should consider the specific requirements, constraints, and objectives of the educational institution or research context. Future investigations could delve deeper into ensemble techniques or feature engineering to further enhance the predictive capabilities of these models.

5 Future Work/Extensions

Expanding on the findings presented, future research avenues could involve integrating additional machine learning techniques such as support vector machines, alternative ensemble methods, or neural networks. A comparative analysis could then be conducted to assess whether these methods yield superior performance metrics compared to those observed in the current study.

Furthermore, extending the dataset or incorporating entirely new datasets could offer valuable insights. The current dataset may not encompass the full spectrum of offline learning data, and augmenting it with more samples could enhance the applicability of models derived from this research.

Moreover, considering that the UCI dataset used in this study was simulated, investigating the performance of these models on real-world data would be intriguing. This could provide a deeper understanding of how well the models generalize beyond simulated scenarios.

5.1 Algorithmic Enhancements

Future studies could focus on refining the algorithms employed in this research by exploring:

- Advanced optimization techniques to enhance model performance and efficiency.
- Integration of newer, more robust algorithms to tackle the limitations observed in current models.
- Development of hybrid models that combine the strengths of various algorithms to improve accuracy and reliability.

5.2 Data and Feature Exploration

There is significant potential in expanding the datasets and features used in our models:

- Incorporating more diverse and comprehensive datasets to validate the models across different contexts and scenarios.
- Investigating the impact of additional feature sets and data augmentation techniques on model performance.
- Enhancing data preprocessing and cleaning methodologies to further improve model accuracy and efficiency.

5.3 Application to New Domains

Extending the application of our findings to new domains offers a promising direction for future research:

- Applying the models and methodologies developed in this study to other industries or fields to assess their versatility and adaptability.
- Exploring cross-disciplinary applications, particularly where Students' dropout intersects with social, economic, or environmental factors.

5.4 Ethical and Societal Considerations

As the application of Students' Dropout Prediction expands, it is critical to address the ethical implications and societal impacts:

- Conducting thorough ethical reviews to identify and mitigate potential biases in the models and their applications.
- Developing guidelines and frameworks for the responsible use of AI and machine learning in sensitive applications.

5.5 Technological Advancements

Looking forward, technological advancements will play a key role in advancing the field of Prediction capabilities in Educational areas:

- Leveraging emerging technologies to enhance computational efficiency and model scalability.
- Incorporating state-of-the-art hardware and software solutions to address current limitations in processing power and data storage.

In conclusion, the pathways for future work and extensions identified above not only aim to address the current study's limitations but also to expand the scope and impact of our research. As the field of Prediction capabilities in Educational field continues to evolve, these efforts will be crucial in advancing our understanding and application of these complex systems.

References

[1] P. M. Moreno-Marcos, C. Alario-Hoyos, P. J. Munoz-Merino, and C. D. Kloos, "Prediction in MOOCs: A review and future research directions," *IEEE Transactions on Learning Technologies*, vol. 12, no. 3, pp. 384–401, Jul. 2019.

[2] V. Tinto, "Dropout from higher education: A theoretical synthesis of recent research," *Review of Educational Research*, vol. 45, no. 1, pp. 89–125, 1975.

[3] R. Baker and G. Siemens, "Educational data mining and learning analytics," *The Cambridge Handbook of the Learning Sciences*, pp. 253–272, Sep. 2014.

[4] A. Carducci, I. Federigi, L. Dasheng, R.T. Julian, & V. Marco, "Making waves: Coronavirus detection, presence and persistence in the water environment: State of the art and knowledge needs for public health", *Water Research*, 179, 2020.

[5] L. Zhang, H. Rangwala, "Early identification of at-risk students using iterative logistic regression", *International Conference on Artificial Intelligence in Education*, pp. 613-626. Springer, Cham, 2018

[6] A. Behr, M. Giese, H. D. Teguim K, and K. Theune, "Early prediction of university dropouts – A random forest approach," *Jahrbücher für Nationalökonomie und Statistik*, vol. 240, no. 6, pp. 743–789, Feb. 2020.

[7] A. J. Fernandez-Garcia, J. C. Preciado, F. Melchor, R. Rodriguez-Echeverria, J. M. Conejero, and F. Sanchez- Figueroa, "A real-life machine learning experience for predicting university dropout at different stages using academic data," *IEEE Access*, vol. 9, pp. 133076–133090, 2021, doi: 10.1109/access.2021.3115851.

[8] J. Kabathova and M. Drlik, "Towards predicting student's dropout in university courses using different machine learning techniques," *Applied Sciences*, vol. 11, no. 7, p. 3130, Apr. 2021, doi: 10.3390/app11073130.

[9] J. Niyogisubizo, L. Liao, E. Nziyumva, E. Murwanashyaka, and P. C. Nshimyumukiza, "Predicting student's dropout in university classes using two-layer ensemble machine learning approach: A novel stacked generalization," *Computers and Education: Artificial Intelligence*, vol. 3, p. 100066, 2022, doi: 10.1016/j.caeai.2022.100066.

[10] C. Taylor, K. Veeramachaneni, and U. M. O'Reilly, "Likely to stop? predicting stopout in massive open online courses," *arXiv preprint arXiv:1408.3382*, August 2014.

[11] K. A. L. Wilson, R. Jeffrey, "Introduction to Binary Logistic Regression," *Modeling Binary Correlated Responses using SAS, SPSS and R*, Springer International Publishing, 2015.

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

[13] Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). Classification and regression trees. CRC press.

[14] Han, J., Pei, J., & Kamber, M. (2011). Data mining: concepts and techniques. Elsevier.

[15] Blake, C. L., & Merz, C. J. (1998). UCI repository of machine learning databases. [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, Department of Information and Computer Science.

[16] Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2), 179-188.

[17] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(Oct), 2825-2830.

[18] Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, 63(1), 3-42.

[19] Zhang, J. (2005). A detailed study of the performance of the ID3 algorithm. *Computer Science and Engineering*, 5(3), 20-29.

[20] Frank, A., & Asuncion, A. (2010). UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences.

[21] Loh, W. Y., & Shih, Y. S. (1997). Split selection methods for classification trees. *Statistica Sinica*, 815-840.

[22] H. Muhamad et al., “Optimasi Naive Bayes Classifier dengan Menggunakan Particle Swarm Optimization pada Data Iris,” Teknol. Inf. Dan Pendidik., vol. 4, no. 3, pp. 180–184, 2017[https://doi.org/10.25126/jtiik.201743251]

[23] M. Dhanashree S, B. Mayur P, and D. Shruti D, “Prediction System For Heart Disease Using Naive Bayes,” Int. J. Adv. Comput. Math. Sci., vol. 3, no. 3, pp. 290–294, 2012.

[24] A. Jamain and D. J. Hand, “The Naive Bayes Mystery: A classification detective story,”Pattern Recognit. Lett., vol. 26, no. 11, pp. 1752–1760, 2005.

[25] C. Science and S. Engineering, “Comparative analysis of Naive Bayes and J48 Classification Algorithms,” IJARCSSE, vol. 5, no. 12, pp. 813–817, 2015.

[26] R. P. Rajeswari, K. Juliet, and Aradhana, “Text Classification for Student Data Set using Naive Bayes Classifier and KNN Classifier,” Int. J. Comput. Trends Technol., vol. 43, no.1, pp. 8–12, 2017.[https://doi.org/10.14445/22312803/ijctt-v43p103]

[27] Bustami, “Penerapan Algoritma Naive Bayes Untuk Mengklasifikasi Data Nasabah Asuransi,” J. Inform., vol. 8, no. 1, pp. 884–898, 2014.

[28] Haviluddin, N. Dengen, E. Budiman, M. Wati and U. Hairah, ”Student Academic Evaluation using Naïve Bayes Classifier Algorithm,” 2018 2nd East Indonesia Conference on Computer and Information Technology (EIConCIT), Makassar, Indonesia, 2018, pp. 104-107, doi: 10.1109/EIConCIT.2018.8878626

[29] Polikar, R. (2009). ”Ensemble learning”. Scholarpedia, 4(1), 2776.

[30] Bamhdi, A. M., Abrar, I., & Masoodi, F. (2021). ”An ensemble based approach for effective intrusion detection using majority voting”. TELKOMNIKA Telecommunication, Computing, Electronics and Control, 19(2), 664-671.

[31] UCI Machine Learning Repository.Predict Students' Dropout and Academic Success.

[32] E. Mulyani, I. Hidayah and S. Fauziati, ”Dropout Prediction Optimization through SMOTE and Ensemble Learning,” 2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), Yogyakarta, Indonesia, 2019, pp. 516-521

[33] C. Cechinel, L. A. B. Macarini, M. F. B. Machado, V. F. C. Ramos, and R. Munoz, “Predicting students success in blended learning Evaluating different interactions inside learning management systems,” Appl. Sci.,vol. 9, no. 24, p. 5523, Dec. 2019, doi: 10.3390/app9245523

[34] G. V. & B. U. W. Lorenz Kemper, ”Predicting student dropout: A machine learning approach,” European Journal of Higher Education, 2020.