**A new oversampling technique to solve class imbalance problem: A case study of students’ grades evaluation**

**A close up of a sign

Description automatically generated**

**International Islamic University Chittagong**

Department of Computer Science & Engineering

|  |  |  |  |
| --- | --- | --- | --- |
|  | Israt Jahan Emu | Dilshad Jahin |  |
|  | Metric ID: C171232 | Metric ID: C171250 |  |

**Supervisor:**

**Subrina Akter**

Assistant Professor

Dept. of CSE, IIUC

**DECLARATION**

With due respect, we declare that, the whole contributions we have made towards making the report are solely based on our ideologies.

|  |  |
| --- | --- |
| 10 November 2021 | Israt Jahan Emu  C171232  Dilshad Jahin  C171250 |

**SUPERVISOR’S DECLARATION**

**Acknowledgment**

At first, all praise to Almighty Allah. We would like to express our heartiest gratitude to our supervisor Assistant Professor Subrina Akter for the smooth execution of the whole thesis report. Then, would love to share utmost gratitude to department of Computer Science & Engineering because of ensuring highest level of integrity while designing the course.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | |  |
| Israt Jahan Emu  C171232 | |  | Dilshad Jahin  C171250 | |

**Abstract**

The academic performance of students is one of the most important aspects in awarding rank to educational institutions. The success of an institution, particularly at the secondary level, is determined by the performance of its students; hence, student performance is determined by the student's prior grades, demographic, socioeconomic, and school-related characteristics, and so on. If the student’s performance is not defined properly then the reputation of the institution is at risk. It is a matter of great hope that data mining has been contributing to the education sector immensely and can find meaningful knowledge from the raw data. But if the data are not accurate, include noise, and are incomplete then the performance evaluation will not be perfect, especially, because of the presence of an imbalance class label in the dataset, the accuracy of the model may be shattered which ultimately damages the overall model quality. In this paper, a new oversampling method is proposed to prepare balanced data and then classify the students' grades into a binary class with overall performance in that course. In this regard, the student performance dataset from the UCI machine learning repository is used where data related student performance of two courses are available. A detailed validation result shows that the decision tree algorithm performs better with the balanced dataset than the imbalanced dataset.

**Keywords:** Oversampling method, Semi-supervised learning, Imbalanced dataset, Binary classification, Grade evaluation.

Contents

[Introduction 8](#_Toc87451229)

[1.1 Overview 8](#_Toc87451230)

[1.2 Introduction 8](#_Toc87451231)

[1.3 Background of the Study 9](#_Toc87451232)

[1.4 Problem Statement 9](#_Toc87451233)

[1.5 Research Objective 10](#_Toc87451234)

[1.6 Contribution of thesis 10](#_Toc87451235)

[1.7 Organization of thesis 10](#_Toc87451236)

[1.8 Chapter Summary 11](#_Toc87451237)

[Literature Review & Definitions 11](#_Toc87451238)

[2.1 Overview 11](#_Toc87451239)

[2.2 Related work 11](#_Toc87451240)

[2.3 Classification methods 11](#_Toc87451241)

[2.3.1 Naive Bayes 12](#_Toc87451242)

[2.3.2 Artificial Neural Network 12](#_Toc87451243)

[2.3.3 Support Vector Machine 13](#_Toc87451244)

[2.4. Clustering methods 13](#_Toc87451245)

[2.4.1 K-means clustering Method 14](#_Toc87451246)

[2.5 Improved sampling and feature selection 14](#_Toc87451247)

[2.5.1 RUSBoost 15](#_Toc87451248)

[2.5.2 Random Forest 15](#_Toc87451249)

[2.5.3. Cost Sensitive Learning 15](#_Toc87451250)

[2.6 Decision tree 15](#_Toc87451251)

[2.7 A brief overview of the imbalanced data problems 16](#_Toc87451252)

[2.8 Chapter Summary 16](#_Toc87451253)

[Research methodology 16](#_Toc87451254)

[3.1 Overview 16](#_Toc87451255)

[3.2 Dataset 17](#_Toc87451256)

[3.3 Preprocessing 17](#_Toc87451257)

[3.4 Weight Assignment 17](#_Toc87451258)

[3.5 Generate an unlabeled dataset for preparing a balanced dataset 18](#_Toc87451259)

[3.6 Proposed method to prepare a balanced dataset 18](#_Toc87451260)

[3.7 Classification using decision tree algorithm 21](#_Toc87451261)

[3.8 Chapter Summary 22](#_Toc87451262)

[Result Analysis 22](#_Toc87451263)

[4.1 Overview 23](#_Toc87451264)

[4.2 Environment 23](#_Toc87451265)

[4.3 Summary of the results 23](#_Toc87451266)

[4.4 Chapter Summary 34](#_Toc87451267)

[Conclusion 34](#_Toc87451268)

[5.1 Conclusion 34](#_Toc87451269)

[5.2 Future work 34](#_Toc87451270)

[References 35](#_Toc87451271)

**Chapter 1**

# Introduction

## Overview

It is a source of great hope that data mining has made a significant contribution to the education sector and can extract useful information from raw data. However, if the data is inaccurate, has noise, or is inadequate, the performance evaluation will be flawed. In particular, if the dataset contains an imbalanced class label, the model's accuracy may be broken, compromising the overall model quality. This work proposes a new oversampling strategy for preparing balanced data and then classifying students' grades into a binary class based on overall performance in that course. The student performance dataset from the UCI machine learning repository is chosen in this caseas it contains data on student performance in two courses. The decision tree approach works better with the balanced dataset than with the imbalanced dataset, according to the proper use of detailed validation result.

## 1.2 Introduction

The level of education has a strong relationship with economic growth. Educational systems must improve their student’s learning processes to promote this progress [1]. Every year, educational institutions such as colleges, schools, and training centers face a complex problem: predicting students' performance. Educational systems for this institution include complicated data that can be used to disclose hidden information and thereby improve the educational system's benefit of the entire decision-making process. Experts' ability to analyze and handle the massive amounts of data generated from educational systems, as well as utilize this data to derive new usable information, is limited. As a result, processing such a massive volume of data demands a computerized approach.

Machine learning techniques have been widely used to evaluate data in a variety of fields [2,3,4]. Data should be evaluated and categorized appropriately using ML algorithms to uncover special insights [5,6]. In addition, it has been successfully implemented, particularly in the educational area. Each year, educational systems like Learning Management Systems (LMS) can store a massive amount of data. To identify the hidden knowledge and differentiate between great, good, and weak students, this data should be analyzed and classified appropriately using ML algorithms [7]. Identifying student’s performance will encourage great students to maintain their strong performance and motivates good students to keep improving; also, instructors may give weak students more attention to help them improve [8].

Above all, a high-accuracy predictive model that can distinguish passed from failed students with their overall performance in that course is necessary. However, there is a barrier in the way of achieving this goal. The problem is that when dealing with real-world engineering challenges, there is usually a class imbalance. Optimizing a single statistic may not reflect the actual performance of a predictive model when dealing with uneven data sets. For example, if 99% of the data set is made up of one class and only 1% of the other, predicting all instances as the majority class yields a 99% accuracy. Furthermore, under such situations, it is normal for the minority class to be the most significant. As a result, it is vital to address this issue by developing solutions that deal especially with class imbalance [9,10,11].

## 1.3 Background of the Study

As creating an alpha in overall student performances require a superficial effort in the era of modern technology, so data mining techniques can play a pivotal role indeed as it has lot to offer in terms of predicting the outcome with better accuracy while working and processing with the inputs. A high-accuracy predictive model that can differentiate passed from failed students based on their entire course performance is required. Because of its effectiveness and comprehensibility, the Decision Tree is one of the most extensively used prediction systems. In huge datasets with a large number of variables, decision tree classification is a speedy and effective way for identifying cases. When using data-level approaches, the training data set is balanced by deleting instances from the majority class (under sampling) and adding new examples to the minority class (oversampling) (oversampling). According to the findings, such jobs are performed at random on a regular basis. More advanced procedures, on the other hand, are required to preserve critical examples or generate important new data. We introduce a new oversampling strategy that falls within the data-level approach category in this paper.

## 1.4 Problem Statement

Economic growth and educational attainment are inextricably linked. To facilitate this growth, educational systems must improve their students' learning processes. Educational institutions such as colleges, schools, and training centers face a severe dilemma every year: projecting students' performance. This institution's educational systems incorporate complex data that can be used to reveal hidden information and so improve the educational system's overall decision-making process. The ability of experts to assess and handle the huge volumes of data created by educational systems, as well as use this data to extract new usable knowledge, is limited. As a result, dealing with such a large amount of data necessitates a computerized technique which will further improve the educational better performance.

## 1.5 Research Objective

Our proposed research method aims to identify the variables that may create an impact regarding the overall student performances in the education sector. Using data mining techniques will surely give an edge as we tend to believe that, it will help us to make better decision based on high level of accuracy by using new oversampling method. As the imbalance data can’t produce high level of accuracy while predicting the conclusion, in that case balancing the data set using decision tree algorithm will make the scenario far better in terms of improving the student performances. There are other objectives that we may conclude from the research which are mentioned below.

1. SSOT method was used to balance dataset as imbalance dataset provides very poor performances indeed.
2. DTW method was initiated as better accuracy becomes the first choice to opt for while making a prediction regarding student performances.
3. To make a move aligned with the policies that may help to take right decision at time as we often need to believe that, not finding the proper reason behind the downfall of poor educational sector performances will take the nation near the edges.
4. Label encoder was opted while preprocessing the data as it requires to be preprocessed right after the collection of data.

## 1.6 Contribution of thesis

In this paper our contributions are listed below:

1. A novel oversampling method is proposed to solve the class imbalance problem.
2. A decision tree algorithm is applied to classify the student's grades into pass or fail classes with weights that express the overall performance of the students.

## 1.7 Organization of thesis

The thesis is organized as follows:

In chapter 2, We've researched at previous works. In chapter 3, We've covered over the concepts of this field so far. Besides that, we have discussed the problem statement, Background of the study, motivation & research objective. As well as the system's extensive details. In chapter 2, we have discussed about literature review where we discussed different types of methods & Related works. In a summary, chapter 3 we discussed the whole process of our work. In chapter 4, we discussed the performance analysis & results. The results came from the used algorithm is discussed here. And also used some graph here. In the final chapter, Chapter 5 contains the research work's conclusion as well as discussion of future work recommendations.

## 1.8 Chapter Summary

In this section, we have talked about the reason which is basically inspired us to work with this topic to resolve the imbalance issue and evaluate the student's grade. We worked with such a topic which is unique and nobody has worked on before. It can be used for solving a new problem or it can be the expansion of past work on any particular field. There’s been lots of work that has already done to use NB, SVM, RF, ANN, K- Means Clustering. They used different methods. Here, we have chosen Decision tree classifier to get the better performance for predicting school student performance we applied this model.

**Chapter 2**

# Literature Review & Definitions

## 2.1 Overview

We reviewed the study's background, research objectives, contribution, problem description, and other topics in Chapter 1. We covered the literature review of our work and related work, as well as associated terminologies and concepts, in this chapter.

## 2.2 Related work

Researchers are interested in educational data mining because of their sophistication and importance in the education sector. To overcome this issue, several approaches have been offered.

## 2.3 Classification methods

The prediction of student performance as a classification problem has been addressed in several research studies. Artificial Neural Network (ANN), Naive Bayes (NB), and Support Vector Machine (SVM) are three of the most used categorization methods for predicting student performance [12]. For example, Yang and Li [13] employed the ANN technique to predict high school student performance. The authors collected data from 60 high schools and proposed a student attribute model (SAM) to model the qualities they discovered. The findings suggested that the proposed method may accurately predict students' performance at an early stage. To find a model to predict student performance, Rana and Garg [14] employed two ML algorithms (NB and ANN) performed inside WEKA software. The results showed that the NB approach is more precise than the ANN method. Using SVM and NB Edumacate and Utari [15] forecast graduation time, GPA, student profile data, and senior high school are among the features chosen by the authors. SVM outperforms NB, according to their findings. Cortez and Silva [30] employed Data mining techniques such as DT, SVM, NN and RF which allow for high-level knowledge extraction from raw data, have intriguing potential for the education area.

### **2.3.1 Naive Bayes**

In data mining and computer learning, the Naive Bayes classifier is frequently used to solve classification problems. Researchers employed a Naive Bayes classifier (R-NBC) to solve two important problems in gene expression dataset classification: underflow and over-fitting. In a supervised learning scenario, Naive Bayes classifiers can be trained quickly. One of the most extensively used categorization algorithms for predicting student achievement is naive bayes. In order to forecast the category of a given sample, Naive Bayes is a family of algorithms based on using the Bayes theorem with the strong assumption that each attribute is independent of the others.

### **2.3.2 Artificial Neural Network**

The structure and function of a biological neural network are used to design ANN architecture. ANN is made up of neurons that are arranged in layers, just like neurons in the brain. Artificial neural networks (ANN) are used to model non-linear situations and estimate results based on training data for specific input parameters. In e-learning contexts, artificial neural networks are viewed as a useful tool for predicting student success. Performance predictions based on student scores are commonly produced in artificial neural network studies, but students' use of learning management systems is not highlighted.

### **2.3.3 Support Vector Machine**

In 1963, Vapnik and Chervonenkis introduced the first version of SVM. BE. Boser et al. explored the nonlinear version in the early 1990s. Although the original version of SVM was created for binary classification problems, several academics have used it to solve multi-class problems.

According to the scikit learn page, support vector machines provide the following advantages:

1. Useful in high-dimensional areas.
2. Even when the number of dimensions contains a large number of samples, the method is still effective.
3. It is memory efficient because it uses a subset of training points called support vectors in the decision function.
4. Versatile: The decision function can use a variety of Kernel functions. Common Kernels are included, however custom kernels can also be specified.

The following are some of the disadvantages of support vector machines:

1. If the number of features is substantially more than the number of samples, it is critical to avoid overfitting by selecting kernel functions and linearization terms.
2. Probability estimates are computed via a costly five-fold cross validation method rather than directly by SVMs.

## 2.4. Clustering methods

Clustering also applied in EDM by many researchers. For example, Harwati et al. [16] used the k-means clustering technique to group students by GPA, attendance ratio, grades in specific courses, and gender. Based on the students' performance, the authors categorize them into three clusters (i.e., smart, normal, and weak students). To improve overall student performance, Bharara et al. [17] used K-means clustering to extract the most valuable information and uncover the hidden correlations between these features. Morais et al. [18] used a regression methodology to predict student performance and used a k-means clustering algorithm to arrange students for an English e-learning course based on their replies. Based on the answers, the authors divided the datasets into five groups (i.e., Expert, Good, Regular, Bad, and Criticism). The authors discussed how the suggested system could help them better analyze how students perform in virtual learning environments. Trivedi et al. [19] demonstrated the value of automated assessment in reducing teaching time and improving standard text score predictions.

### **2.4.1 K-means clustering Method**

The K-means clustering algorithm is a simplistic iterative clustering method. The goal of K-means clustering is to divide data into k clusters so that data points in the same cluster are similar and data points in other clusters are further away. K-means clustering is an unsupervised learning method that is used for unlabeled data. The K cluster centroids, which can be used to label new data.

## 2.5 Improved sampling and feature selection

Underestimation of the distributional balance of a dataset has a significant impact on the categorization process outcome. There have already been various studies on how to deal with the challenges of imbalanced class distribution Nguyen et al. [20] used cost-sensitive learning (CSL) and over-sampling techniques to address the class imbalance to improve prediction/classification results.

Amal et al. [21] proposed a new method Multi-IM to deal with an imbalanced multiclass data problem. Multi-IM is based on the probabilistic relational technique (PRMs IM), which was designed for learning from imbalanced relational data in the two-class problem. For multiclass imbalanced learning in relational and non-relational domains that builds on PRMs-IM, Multi-IM is a generalized framework. Akram et al. [22] proposed a method for balancing training data and selecting the best boundary between classes by combining the over-sampling SMOTE technique with the thresholding technique. To identify the noisy instances, we used a noise detection approach. When the two proposed methods were combined with existing class-imbalance techniques, the accuracy scores improved significantly. Yoga et al. [23] used a combination of SMOTE and OSS to handle imbalanced classes on multiclass EDM datasets. The SMOTE and OSS methods provide a balancing mechanism for the distribution of the dataset, resulting in improved classification performance. Hybrid sampling and feature selection, for example, are two key concepts presented by [24]. Oversampling and under sampling are used together in hybrid sampling. Despite being employed only a few times in the literature, feature selection can assist models to enhance predictive performance by deleting irrelevant features and lowering the danger of omitting key cases that might be considered noisy. When it comes to dealing with unbalanced data sets, binary segmentation, according to [25] is the most developed branch. In classification methods, learning problems are created not only by the unbalance ratio but also by the presence of difficult cases (mostly in the minority group), such as overlapping distributions and misfits. For better comprehension of learning concerns, an examination of the neighborhood of minority class occurrences is recommended. Victor and Gilberto [26] emphasizes the use of a well-known ensemble learning method, random forest (RF), as well as an imbalanced learning specific technique, random under sampling boosting (RUSBoost), and the use of multi-objective optimization design (MOOD) in the various stages of ensemble learning.

### **2.5.1 RUSBoost**

RUSBoost is an algorithm for dealing with class imbalance in data with discrete class labels. To better model the minority class, it utilizes a combination of RUS (random under-sampling) and the standard boosting procedure AdaBoost. RUSBoost makes use of RUS, a technique that removes examples from the majority class at random. Simplicity, speed, and performance are the reasons for including RUS into the boosting process.

### **2.5.2 Random Forest**

A random forest is a machine learning method of solving regression and classification tasks.It employs ensemble learning, which is an approach for solving complicated problems that includes multiple classifiers. For text classification, the random forests technique is an ensemble learning method. T.kam Ho was the first to introduce this approach, which employed trees in parallel, in 1995. L.Breiman later created this technique in 1999, which they discovered converged for RF as a margin measure.

### **2.5.3. Cost Sensitive Learning**

Cost-sensitive learning is a subfield of machine learning in which costs are explicitly defined and used when training machine learning algorithms. Data resampling, algorithm modifications, and ensemble methods are three types of cost-sensitive techniques.

## 2.6 Decision tree

The decision tree is a quick and effective classification method for identifying instances in large datasets with a large number of variables [27]. Originally, decision trees were used to analyze categorized data, but in practice, continuous variables are more common [28]. In this paper, the information gain is measured by using the Gini index whose formula can be defined as:

Gini Index = (1)



Where Pj = The probability of class J.

## 2.7 A brief overview of the imbalanced data problems

A lot of research has been done on imbalanced learning[29].To cope with unbalanced data challenges Bartosz [25] proposes three approaches: data-level, algorithm-level, and hybrid solutions. The training data set is balanced when employing data-level methods by removing instances from the majority class(under-sampling) and adding new instances to the minority class(oversampling). According to the research, such jobs are regularly performed at random. To preserve essential cases or generate valuable new data; however, more sophisticated strategies are required. Methods at the algorithmic level aim on altering the existing learning algorithms to eliminate majority group bias. This is typically accomplished by implementing a cost-sensitive strategy, in which each class is penalized differently while computing the learner's loss function. Another method is one-class learning, in which only a specific group is chosen. Hybrid approaches combine the two previously described methods, as the name suggests. In this paper, we have proposed a new oversampling technique that falls in the category of data-level approach.

## 2.8 Chapter Summary

This chapter is made up of Several research papers have looked at student performance prediction as a classification problem. This section also covers Artificial Neural Network (ANN), Naive Bayes (NB), and Support Vector Machine (SVM), Decision tree (DT) categorization approaches for predicting student performance. Clustering was also utilized to predict student performance. as well as a variety of methods that researchers employed to improve classification results.

**Chapter 3**

# Research methodology

## 3.1 Overview

This chapter provides a brief description of the complete study technique, starting with data collecting and progressing to the implemented approach for the proposed method. We discussed about weight assignment and how to prepare a balanced dataset using an unlabeled dataset. Aside from that, Semi-supervised oversampling technique (SSOT) and decision tree algorithm with weight assignment (DTW) are two algorithms for preparing a balanced dataset and predicting student performance, respectively.

To overcome the class imbalance problem, we have proposed a new method and have considered two datasets of student performance.

## 3.2 Dataset

We have collected Student Performance Dataset from the UCI machine learning repository. This data was about the performance of the students of secondary education of two Portuguese schools. The database was built by incorporating student grades, demographic, social, and school-related features where student grades were collected from school reports and the other features were from questionnaires. Two datasets were developed. One is a mathematics dataset where 396 samples with 33 attributes are given and the other one is a Portuguese language dataset with 650 samples and 33 attributes. Among these, 32 attributes are inputs and one attribute is output. The type of the input attributes is different; 13 are binary type, 15 are numeric and 4 attributes are nominal type. The target attribute named G3 is numeric.

## 3.3 Preprocessing

Both datasets must be preprocessed after they have been collected. We started by looking for any missing values, which we didn't find. The majority of the attributes in the dataset were categorical, while the rest were numeric. Using a label encoder, we converted categorical attributes to numerical attributes. The Portuguese and maths grades have been divided into binary classes. As a result, the target attribute G3 has been transformed to a binary class. If G3>=10, the students will pass; otherwise, they will fail.

## 3.4 Weight Assignment

The term weight we have used here expresses the overall performance (in percentage) of a student in that course.

The following formula is used to assign the weight:

Wi=+2.5 (2)



Where Wi=Weight of the ith student in a course.

G1i=1st period grade of the ith student.

G2i=2nd period grade of the ith student

We have analyzed that the most prevalent features in this sample are G1 and G2. Both characteristics have a significant influence on the target attribute. We can acquire a fairly close estimate of the average of the three grades (in percentage) G1, G2, and G3 by adding at least 2.5 to the average of the first two period grades (in percentage). The weight helps a student to know about his/her development in this course, if he/she needs further improvement and to what extent.

**Table 1: Weight assignment**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Serial | age | schoolsup | goout | absences | …….. | G1 | G2 | G3 | Label | Weight | Avg%(G1, G2, G3) |
| 1 | 18 | 1 | 4 | 6 |  | 5 | 6 | 6 | -1 | 30 | 28.33333 |
| 2 | 16 | 0 | 2 | 4 |  | 6 | 10 | 10 | 1 | 42.5 | 43.33333 |

## 3.5 Generate an unlabeled dataset for preparing a balanced dataset

When we have built the initial classifier using the original dataset by the decision tree algorithm then the significant attributes are recorded using the information gain of the nodes. For example, G2 is the root node having the highest information gain of 0.433. In this way, G2, G1, schoolsup, fjob, age, mjob, fedu, medu, goout, absences, studytime, activities, famsize, freetime, walc features are recorded as the significant features. Moreover, Pearson correlation was also used for an approximate assumption. According to Pearson correlation, G2 and G1 are strongly related to the target class and goout, higher, failure, age, medu and fedu have trivial relation with the target class.

The derived attributes from both assumptions are merged for generating the new dataset. The dataset has been generated by taking all possible combinations of the unique values of these attributes. For example, suppose the unique values of G1=1,2,3 and G2=1,4,5. Then the dataset will look like the following: [(1, 1), (1, 4), (1, 5), (2, 1), (2, 4), (2, 5), (3, 1), (3, 4), (3, 5)]

After generating the dataset, we remove the duplicate rows (the rows that exist in the original dataset) from the generated dataset.

## 3.6 Proposed method to prepare a balanced dataset

The main idea is that the given original dataset Xor(L) is arbitrarily split into a training set, Xtrain(L), and a testing set, Xtest(L). A primary classifier C is trained with Xtrain(L) by the decision tree algorithm. Then the primary classifier is tested by classifying each of the samples from Xtest(L). After this, the initial decision tree is pruned and the pruned tree is used to classify again each test data and calculate their weights. Then a new unlabeled dataset Xn(U) is generated and the pruned tree is used to classify a portion, Xnp(U) of this unlabeled dataset, and calculated their weights. As the target class is a fail class, samples of the pass class are deleted from the newly labeled dataset Xnp(U). Observing the magnitude of the weights of the data in Xnp(U) a threshold has been set and based on this threshold samples are categorized into Xnp(U)accept and Xnp(U)reject samples. Only the samples with accepted weight, Xnp(U)accept, are added to the actual training set. Then the classifier is re-educated with these new training samples and continues this process until making the balanced data. The full view of this process is presented in Algorithm 1. Here, we have denoted the proposed method Semi-supervised Oversampling Technique(SSOT).

**Algorithm 1: SSOT**

**Input:** Dataset, p(Total samples of pass class), f(Total samples of fail class)

**Output:** Make a balanced data where p = f

1: Arbitrarily separate the original labeled dataset, Xor(L) into a training dataset Xtrain(L) and a testing dataset Xtest(L).

2: Train the classifier C, according to the decision tree algorithm using the training dataset Xtrain(L).

3: Get the training and testing specificity respectively trn\_ specificityB, tes\_ specificityB.

4:Prune the tree

5:Again train the classifier C by the pruned tree and update the training and testing specificity respectively trn\_ specificityB, tes\_ specificityB.

while f != p do

5: Select a portion, Xnp(U) from the newly generated unlabeled dataset, Xn(U), and classify the samples of Xnp(U) using C.

6: Delete the samples categorize as pass class from the newly labeled dataset Xnp(U).

7: Calculate the weights of each sample in Xnp(U) by Wi=+2.5



8: Set threshold value = 30 and reject the samples from Xnp(U) whose threshold value is more than 30.

9: Update the original training dataset by adding the filtered data into it, Xtrupd = Xor(L) ∪ Xnp(U)

10: Update the unlabeled dataset, Xn(U)=Xn(U)-Xnp(U)

12: Retrain an updated classifier Cupd according to the given training algorithm with Xtrupd.

13: Again record the training and testing specificity respectively trn\_ specificityA,

tes\_ specificityA by classifier Cupd with Xtrupd.

14: If the tes\_ specificityA > tes\_ specificityB then repeat the process with Cupd and Xtrupd and replace trn\_ specificityB and tes\_ specificityB with the value of trn\_specificityA, tes\_ specificityA respectively.

15: end while.

**Flowchart 1**

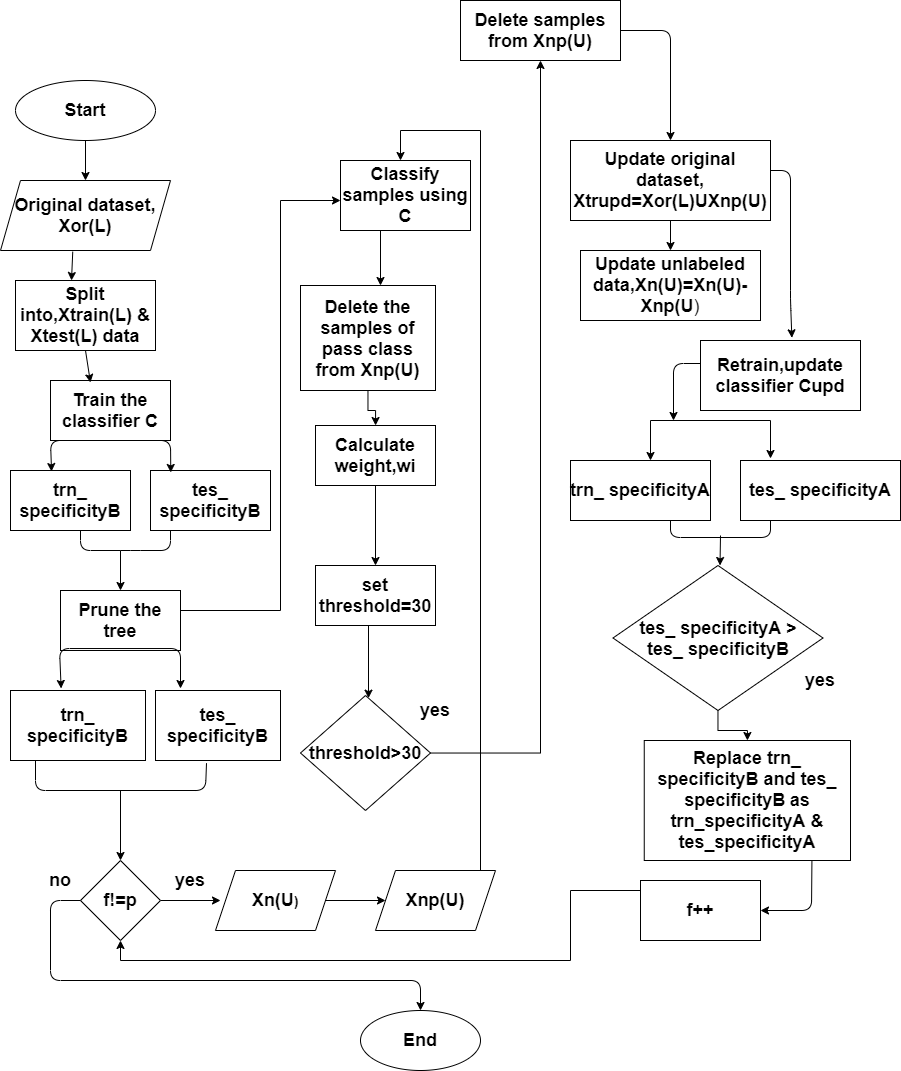


Figure : flowchart of the 1st experiment.

## 3.7 Classification using decision tree algorithm

Using 5 fold cross-validation, the entire balanced datasets are initially separated into two sections, X train, and X test. The first decision tree is modeled with X train, and the performance of the tree is tested with the X test. The initial tree has an alpha of 0.0125 and a mean accuracy of 83%. The next step is to prune the tree to prevent it from overfitting. Using 5 fold cross-validation, we have determined the parameter ccp alpha= 0.022165 and have pruned the tree using the cost complexity pruning technique. Using this alpha value, we have divided the dataset into 5 folds and classify the test samples of each fold with a weight. Finally, the average accuracy is set at 87%. The whole process has depicted in algorithm 2, the decision tree algorithm with weight assignment(DTW) for predicting student performance.

**Algorithm 2 : (DTW)**

**Input:** Dataset

**Output**: Class of test data with weight.

1: i=1

2. Split the balanced labeled dataset into a training dataset Xtrain(L) and a testing dataset Xtest(L) using 5 fold cross-validation.

While i <=5 do

3. Train the classifier C, according to the decision tree algorithm using the training dataset Xtrain(L)

4: Prune the tree

5: Again train the classifier C by the pruned tree and record the classification report and accuracy.

6: Calculate the weights of each test sample.

7: i=+1

end while

8: Calculate mean accuracy.

**Flowchart 2:**

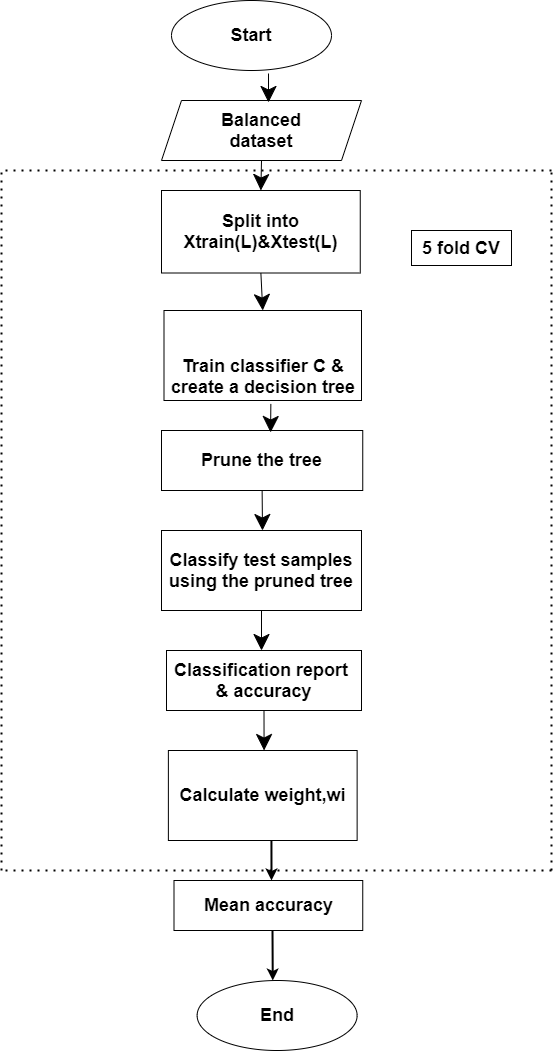
****

Figure 2: flowchart of 2nd experiment

## 3.8 Chapter Summary

The characteristics of the dataset, from its collection through the implementation procedure, are explained in depth in this chapter. Pre-processing techniques were also addressed, each with a full description and processing procedures. The process of balanced data from a new unlabeled dataset and predicting student performance technique are also discussed.

**Chapter 4**

# Result Analysis

## 4.1 Overview

We will explain the environment setup required for this proposed system in this chapter, as well as the Summary of Results for the language dataset and mathematics dataset.

## 4.2 Environment

The Algorithm was implemented in Python 3.8.8 software, and The studies are conducted in a personal computer running Windows 10, 64-bit, with an Intel Core i5-4590 CPU running at 3.30 GHz and 4 GB of RAM. We have used the 5-fold cross-validation using Stratified KFold,conda version 4.10.1& conda build version 3.21.4. & also used numpy 1.20.1,pandas 1.2.4,sklearn 0.24.1. we will need python 2.7 or 3.8 & have at least these versions for each of the following modules:

pandas >= 0.25.1

numpy >= 1.172

sklearn >= 0.22.1

## 4.3 Summary of the results

A quick rundown of the results from the two datasets is delivered in the following tables.

**Table 2: Summary of Results for language dataset**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| The output of unpruned tree (imbalanced dataset) | **alpha**  0.0045 | **Fold** | **Sensitivity** | **Specificity** | **Precision** | **F1 score** | **Accuracy** | **Mean accuracy** |
| 1 | 0.91 | 0.47 | 0.89 | 0.90 | 0.83 | 0.856 |
| 2 | 0.93 | 0.61 | 0.92 | 0.93 | 0.88 |
| 3 | 0.92 | 0.52 | 0.9 | 0.91 | 0.85 |
| 4 | 0.91 | 0.66 | 0.95 | 0.93 | 0.88 |
| 5 | 0.87 | 0.4 | 0.91 | 0.89 | 0.82 |
| Output of pruned tree (imbalanced dataset)  Pass:549  Fail:100 | **alpha**    0.0094 | **Fold** | **sensitivity** | **Specificity** | **Precision** | **F1 score** | **Accuracy** | **Mean accuracy** |
| 1 | 0.94 | 0.7 | 0.94 | 0.94 | 0.90 | 0.918 |
| 2 | 0.96 | 0.65 | 0.93 | 0.95 | 0.91 |
| 3 | 0.97 | 0.77 | 0.95 | 0.96 | 0.93 |
| 4 | 0.93 | 0.85 | 0.98 | 0.95 | 0.92 |
| 5 | 0.94 | 0.7 | 0.94 | 0.94 | 0.90 |
| The output of pruned tree  Imbalanced ratio:42%  Pass:549  Fail:220 | **alpha**  0.009784 | **Fold** | **sensitivity** | **Specificity** | **Precision** | **F1 score** | **Accuracy** | **Mean accuracy** |
| 1 | 0.89 | 0.73 | 0.89 | 0.89 | 0.85 | 0.869 |
| 2 | 0.89 | 0.88 | 0.96 | 0.93 | 0.89 |
| 3 | 0.9 | 0.78 | 0.91 | 0.90 | 0.87 |
| 4 | 0.88 | 0.93 | 0.98 | 0.93 | 0.89 |
| 5 | 0.83 | 0.82 | 0.95 | 0.89 | 0.83 |
| Output of pruned tree  Imbalanced ratio:24%  Pass:549  Fail:340 | **alpha**  0.029303 | **Fold** | **sensitivity** | **Specificity** | **Precision** | **F1 score** | **Accuracy** | **Mean accuracy** |
| 1 | 0.84 | 0.89 | 0.94 | 0.89 | 0.85 | 0.892 |
| 2 | 0.96 | 0.76 | 0.86 | 0.91 | 0.88 |
| 3 | 0.95 | 0.85 | 0.91 | 0.93 | 0.91 |
| 4 | 0.87 | 0.96 | 0.98 | 0.92 | 0.89 |
| 5 | 0.88 | 0.91 | 0.95 | 0.92 | 0.89 |
| Output of pruned tree  Imbalanced ratio:12%  Pass:549  Fail:440 | **alpha**  0.030925 | **Fold** | **sensitivity** | **Specificity** | **Precision** | **F1 score** | **Accuracy** | **Mean accuracy** |
| 1 | 0.881356 | 0.924051 | 0.945455 | 0.913405 | 0.898477 | 0.902 |
| 2 | 0.890756 | 0.949367 | 0.963636 | 0.927196 | 0.914141 |
| 3 | 0.853659 | 0.933333 | 0.954545 | 0.904102 | 0.883838 |
| 4 | 0.850394 | 0.971429 | 0.981818 | 0.916106 | 0.893401 |
| 5 | 0.912281 | 0.939759 | 0.954128 | 0.933205 | 0.923858 |
| Output of pruned tree  Balanced dataset  Pass:549  Fail:549 | **alpha**  0.023314 | **Fold** | **sensitivity** | **Specificity** | **Precision** | **F1 score** | **Accuracy** | **Mean accuracy** |
| 1 | 0.87395 | 0.940594 | 0.945455 | 0.909702 | 0.904545 | 0.911 |
| 2 | 0.890756 | 0.960396 | 0.963636 | 0.927196 | 0.922727 |
| 3 | 0.853659 | 0.947917 | 0.954545 | 0.904102 | 0.894977 |
| 4 | 0.870968 | 0.978947 | 0.981818 | 0.926393 | 0.917808 |
| 5 | 0.888889 | 0.95098 | 0.954128 | 0.921509 | 0.917808 |
|  | 0.87395 | 0.940594 | 0.945455 | 0.909702 | 0.904545 |

**Table 3: Summary of the results for mathematics dataset**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Output of unpruned tree (imbalanced dataset) | **alpha**  0.0125 | **Fold** | **Sensitivity** | **Specificity** | **Precision** | **F1 score** | **Accuracy** | **Mean accuracy** |
| 1 | 0.81 | 0.64 | 0.83 | 0.82 | 0.75 | 0.772 |
| 2 | 0.84 | 0.69 | 0.84 | 0.84 | 0.79 |
| 3 | 0.81 | 0.6 | 0.81 | 0.81 | 0.74 |
| 4 | 0.83 | 0.8 | 0.87 | 0.87 | 0.82 |
| 5 | 0.78 | 0.6 | 0.80 | 0.80 | 0.73 |
| Output of pruned tree (imbalanced dataset) | **alpha**  0.019943 | **Fold** | **Sensitivity** | **Specificity** | **Precision** | **F1 score** | **Accuracy** | **Mean accuracy** |
| 1 | 0.81 | 0.75 | 0.90 | 0.85 | 0.79 | 0.80 |
| 2 | 1 | 0.61 | 0.69 | 0.84 | 0.79 |
| 3 | 0.95 | 0.61 | 0.71 | 0.83 | 0.78 |
| 4 | 0.88 | 0.68 | 0.83 | 0.85 | 0.81 |
| 5 | 1 | 0.65 | 0.73 | 0.86 | 0.82 |
| Output of pruned tree  Imbalanced ratio:20%  Pass:265  Fail:175 | **alpha**  0.015973 | **Fold** | **Sensitivity** | **Specificity** | **Precision** | **F1 score** | **Accuracy** | **Mean accuracy** |
| 1 | 1 | 0.64 | 0.64 | 0.82 | 0.78 | 0.819 |
| 2 | 0.90 | 0.81 | 0.86 | 0.88 | 0.86 |
| 3 | 0.84 | 0.86 | 0.92 | 0.88 | 0.85 |
| 4 | 0.79 | 0.84 | 0.92 | 0.85 | 0.80 |
| 5 | 0.95 | 0.71 | 0.75 | 0.85 | 0.82 |
| Output of pruned tree  Imbalanced ratio:13%  Pass:265  Fail:204 | **alpha**  0.017459 | **Fold** | **Sensitivity** | **Specificity** | **Precision** | **F1 score** | **Accuracy** | **Mean accuracy** |
| 1 | 0.92 | 0.84 | 0.86 | 0.89 | 0.88 | 0.855 |
| 2 | 0.96 | 0.65 | 0.60 | 0.78 | 0.76 |
| 3 | 0.88 | 0.81 | 0.84 | 0.86 | 0.85 |
| 4 | 0.85 | 0.94 | 0.96 | 0.90 | 0.88 |
| 5 | 0.93 | 0.84 | 0.86 | 0.90 | 0.89 |
| Output of pruned tree  Imbalanced ratio:6%  Pass:265  Fail:236 | **alpha**  0.0164 | **Fold** | **Sensitivity** | **Specificity** | **Precision** | **F1 score** | **Accuracy** | **Mean accuracy** |
| 1 | 0.78 | 0.87 | 0.90 | 0.84 | 0.82 | 0.846 |
| 2 | 0.88 | 0.82 | 0.83 | 0.85 | 0.85 |
| 3 | 0.84 | 0.82 | 0.84 | 0.84 | 0.84 |
| 4 | 0.98 | 0.63 | 0.75 | 0.86 | 0.82 |
| 5 | 0.89 | 0.91 | 0.92 | 0.90 | 0.9 |
| Output of pruned tree  Balanced dataset  Pass265  Fail:265 | **alpha**  0.0125 | **Fold** | **Sensitivity** | **Specificity** | **Precision** | **F1 score** | **Accuracy** | **Mean accuracy** |
| 1 | 0.86 | 0.86 | 0.85 | 0.85 | 0.85 | 0.878 |
| 2 | 0.83 | 0.83 | 0.84 | 0.84 | 0.86 |
| 3 | 0.88 | 0.88 | 0.89 | 0.89 | 0.90 |
| 4 | 0.84 | 0.83 | 0.88 | 0.88 | 0.93 |
| 5 | 0.87 | 0.86 | 0.90 | 0.90 | 0.93 |
|  |  |  |  |  |  |  |

**Table 4: Performance of Language dataset according to imbalanced ratio.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Dataset** | **Pass %** | **Fail%** | **Sensitivity** | **Specificity** |
| Original dataset unpruned tree | 85% | 15% | 0.91 | 0.53 |
| Original dataset pruned tree | 85% | 15% | 0.95 | 0.73 |
| New\_dataset\_1 | 71% | 29% | 0.88 | 0.83 |
| New\_dataset\_2 | 62% | 38% | 0.9 | 0.87 |
| New\_dataset\_3 | 56% | 44% | 0.87 | 0.94 |
| New\_dataset\_4 | 50% | 50% | 0.87 | 0.95 |

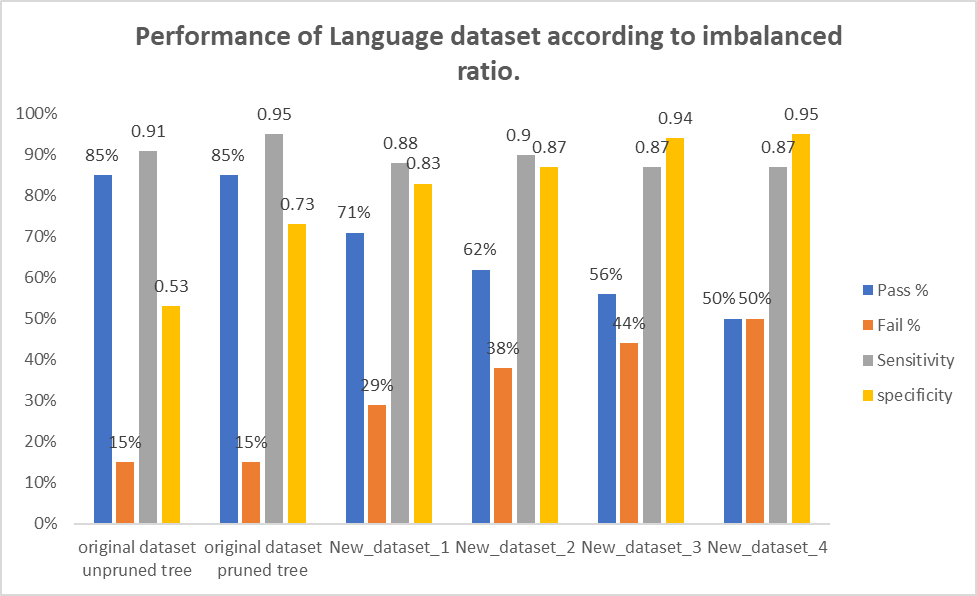


Figure 3: Performance of Language dataset according to imbalanced ratio

**Table 5 : Performance of Mathematics dataset according to imbalanced ratio.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Dataset** | **Pass%** | **Fail%** | **Sensitivity** | **Specificity** |
| Original dataset unpruned tree | 67% | 33% | 0.81 | 0.66 |
| Original dataset pruned tree | 67% | 33% | 0.92 | 0.66 |
| New\_dataset\_1 | 60% | 40% | 0.89 | 0.77 |
| New\_dataset\_2 | 57% | 43% | 0.91 | 0.81 |
| New\_dataset\_3 | 53% | 47% | 0.87 | 0.81 |
| New\_dataset\_4 | 50% | 50% | 0.89 | 0.86 |

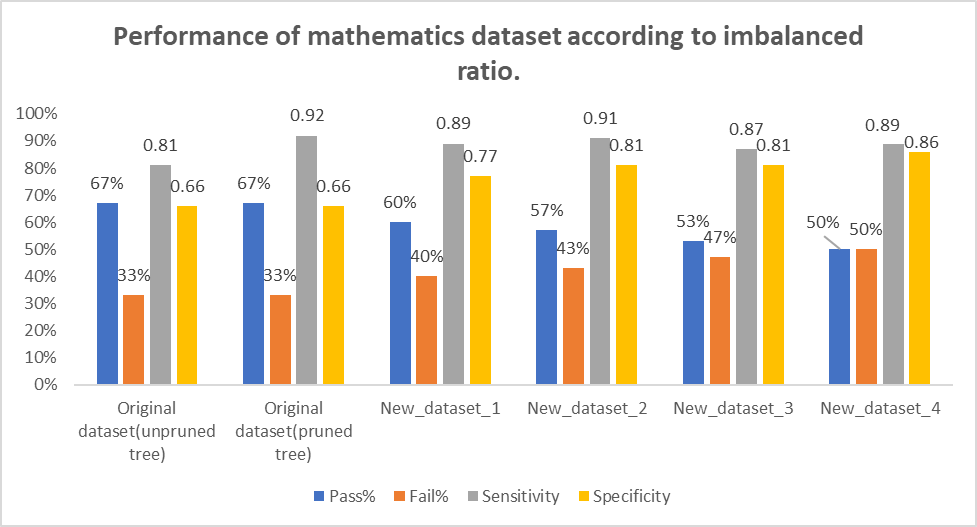


Figure 4: Performance of Mathematics dataset according to imbalanced ratio.

From Figure 3 and Figure 4, we have seen that the specificity increases gradually as the ratio of the pass class increases and the sensitivity is also satisfactory.

**Table 6: Sensitivity and Specificity for both datasets**

|  |  |  |  |
| --- | --- | --- | --- |
| **Dataset** | **Imbalanced ratio** | **Sensitivity** | **Specificity** |
| Language | 70% | 0.91 | 0.53 |
| 42% | 0.88 | 0.83 |
| 24% | 0.9 | 0.87 |
| 12% | 0.87 | 0.94 |
| 0% | 0.87 | 0.95 |
| Mathematics | 34% | 0.81 | 0.66 |
| 20% | 0.89 | 0.77 |
| 14% | 0.91 | 0.81 |
| 6% | 0.87 | 0.81 |
| 0% | 0.89 | 0.86 |
|  |  |  |

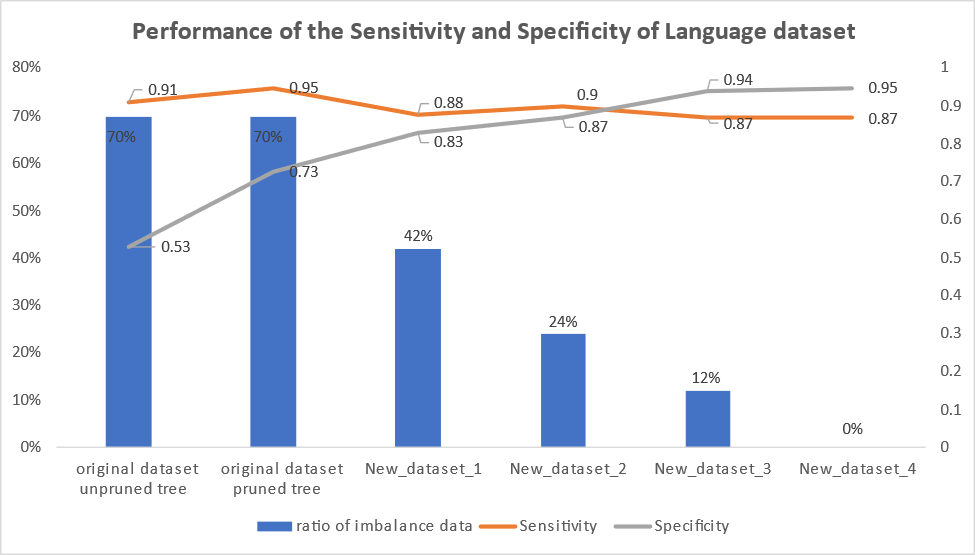


Figure 5: Sensitivity and Specificity for language dataset.

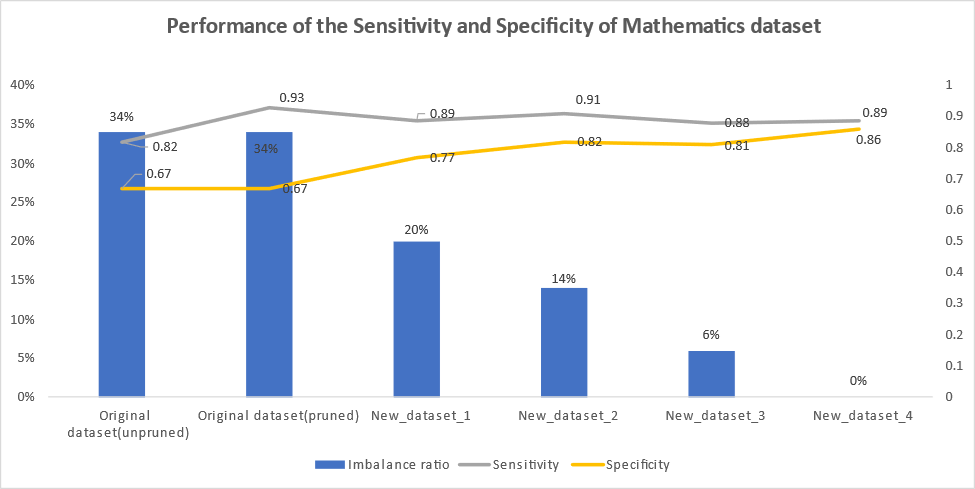


Figure 6: Sensitivity and Specificity for mathematics dataset.

From Figure 5 and Figure 6, we have seen that the Specificity of the classifier increases as the imbalanced ratio of the dataset decreases.

**Table 7: Output of Classification of 5 samples**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Serial | age | schoolsup | goout | absences | …….. | | G1 | G2 | G3 | Label | Weight |
| 1 | 18 | 1 | 4 | 6 |  | 5 | | 6 | 6 | -1 | 30 |
| 2 | 17 | 0 | 3 | 4 |  | 5 | | 5 | 6 | -1 | 27.5 |
| 3 | 15 | 1 | 2 | 10 |  | 7 | | 8 | 10 | 1 | 40 |
| 4 | 15 | 0 | 2 | 2 |  | 15 | | 14 | 15 | 1 | 75 |
| 5 | 16 | 0 | 2 | 4 |  | 6 | | 10 | 10 | 1 | 42.5 |

**Table 8: Performance of the classifier for the different imbalanced ratios of the language dataset.**

|  |  |  |
| --- | --- | --- |
| **Serial no.** | **Imbalanced ratio of the dataset** | **Accuracy of the classifier** |
| 1 | 70% | 0.86 |
| 2 | 42% | 0.87 |
| 3 | 24% | 0.89 |
| 4 | 12% | 0.9 |
| 5 | 0% | 0.91 |

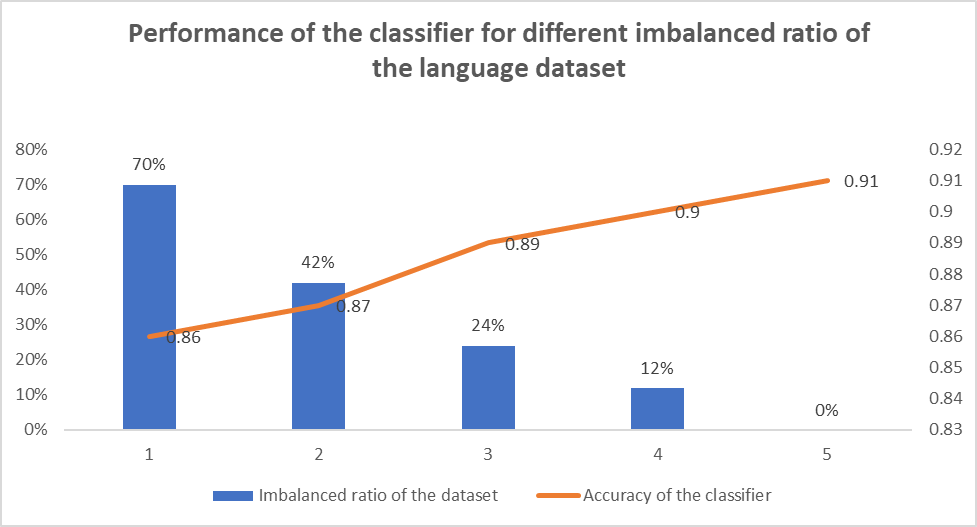


Figure 7: Performance of the classifier for different imbalanced ratios of the language dataset.

**Table 9: Performance of the classifier for different imbalanced ratios in mathematics dataset.**

|  |  |  |
| --- | --- | --- |
| **Serial no.** | **Imbalanced ratio of the dataset** | **Accuracy of the classifier** |
| 1 | 34% | 0.8 |
| 2 | 20% | 0.82 |
| 3 | 14% | 0.85 |
| 4 | 6% | 0.85 |
| 5 | 0% | 0.88 |

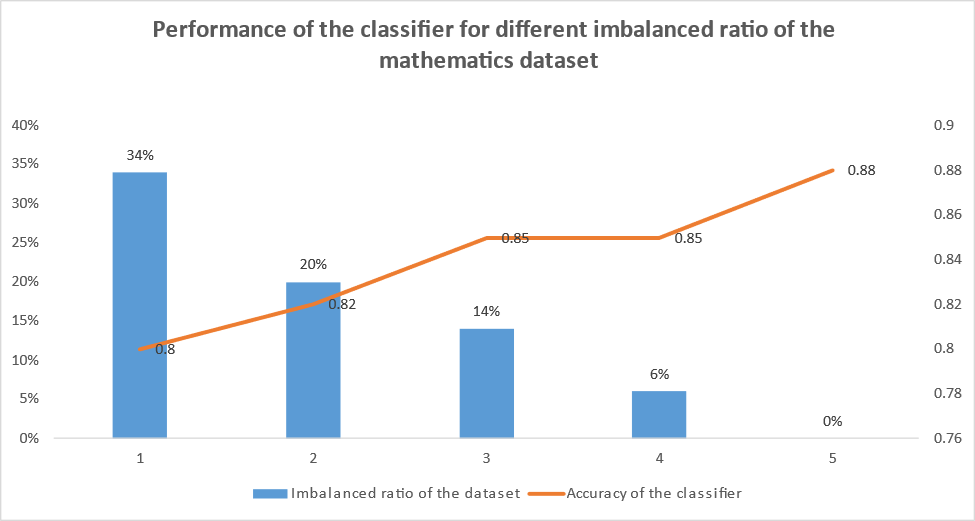


Figure 8: Performance of the classifier for the different imbalanced ratios of the mathematics dataset

From the Figure 7 and Figure 8, we have seen that the accuracy of the classifier increases as the imbalanced ratio of the dataset decreases.

## 4.4 Chapter Summary

We can observe that as the ratio of the pass class increases, the specificity increases progressively, and the sensitivity is likewise acceptable. In addition, as the dataset's imbalanced ratio decreases, the classifier's specificity increases. As the dataset's imbalanced ratio reduces, the classifier's accuracy improves.

**Chapter 5**

# Conclusion

## 5.1 Conclusion

Dataset can be considered as a milestone for the academic field to properly define a student's grade as well as to highlight the reputation of an institution. If used properly this dataset can achieve the highest peak of its own by being a unique and complete system. We focused on two primary challenges in this paper: the problem of class imbalance and binary classification with overall performance. We looked at the many solutions presented to these issues. We presented a method based on this research that can manage all of these difficulties at the same time. This method has solved the class imbalance problem using a new oversampling method. Moreover assigning the weight for understanding overall performance, could be a useful piece of feedback for the learners.

## 5.2 Future work

Our extensive test on a benchmark dataset demonstrates the immense potential of this integrated solution. In the future, we'd like to show the performance of this new oversampling method for extreme imbalanced datasets and which type of classification is ideal for evaluating student achievement.

# References

[1] Kemper L, Vorhoff G, Wigger BU, Predicting student dropout: a machine learning approach, in: Eur J Higher Educ, 2020,pp. 10:28–47.

[2]Patwary MJ, Wang XZ. Sensitivity analysis on initial classifier accuracy in fuzziness based semi-supervised learning. Information Sciences. 2019 Jul 1; 490: 93-112.

[3]Liu J, Patwary MJ, Sun X, Tao K. An experimental study on symbolic extreme learning machine. International Journal of Machine Learning and Cybernetics. 2019 Apr; 10(4):787-97.

[4] Patwary MJ, Wang XZ, Yan D. Impact of fuzziness measures on the performance of semi-supervised learning. International Journal of Fuzzy Systems. 2019 Jul; 21(5):1430-42.

[5] Patwary MJ, Akter S, Alam MB, Karim AR. Bank Deposit Prediction Using Ensemble Learning. Artificial Intelligence Evolution. 2021 Aug 30:42-51.

[6] Patwary, Muhammed JA, James NK Liu, and Honghua Dai. "Recent advances of statistics in computational intelligence (RASCI)." International Journal of Machine Learning and Cybernetics. 2018: 9(1): 1-3.

[7] Hussain S, Atallah R, Kamsin A, Hazarika J, Classification, clustering and association rule mining in educational datasets using data mining tools, in: Cybermetics and Algorithm in intelligent System, 2018,pp. 196-211.

[8] Khare K, Lam H, Khare A, Educational Data Mining (EDM): Researching impact on online business education, in: Springer International Publishing, Cham, 2018, pp. 37–53.

[9] Georgios , D. ,Fernando,B., Effective data generation for imbalanced learning using conditional generative adversarial networks, in: Expert system application , 2018, pp. 464-471.

[10] G. Lemaitre, Nogueira .F, & Christos K. Aridas , Imbalanced learn A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning, 2017, Journal of Machine Learning Research, pp.1-5.

[11] Taghi M.K , Jason.V.H,Amri .N , Comparing Boosting and Bagging Techniques With Noisy and Imbalanced Data, in : [IEEE Transactions on Systems,](https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=3468) 2010,pp. 552 – 568.

[12] Hamza .T, Sana Al.A , Mahmoud .R, Enhanced Harris Hawks optimization as a feature selection for the prediction of student performance, 2020.

[13] Yang F, Li FW, Study on student performance estimation, student progress analysis, and student potential prediction based on data mining,in : Computers & education,2018,pp. 97-108.

[14] Rana S, Garg R, Student’s performance evaluation of an institute using various classification algorithms. in: Information and communication technology for sustainable development,2018, pp. 229-238.

[15] Kesumawati A, Utari DT, Predicting patterns of student graduation rates using naïve bayes classifier and support vector machine, in: AIP Conf Proc,2018.

[16] Harwatia, A.P. Alfiani , Wulandari FA, Mapping student’s performance based on data mining approach, in : international Conference on Agro-industry,2015, pp.173-177.

[17] A. Bharara S, Sabitha S, Bansal A, Application of learning analytics using clustering data mining for students’ disposition analysis,in : Educ Inf Technol,2018,pp. 957–984.

[18] De Morais AM, Araújo JMFR, Costa EB, Monitoring student performance using data clustering and predictive modelling, in : IEEE frontiers in education conference,2014,pp. 1-8.

[19] Trivedi S, Pardos ZA, Heffernan NT, Clustering students to generate an ensemble to improve standard test score predictions,in: Artificial Intelligence in Education,2011, pp 377–384.

[20] Nguyen T, A. Busche, Lars S, Improving Academic Performance Prediction by Dealing with Class Imbalance, in : Ninth International Conference on Intelligent Systems Design and Applications,2019.

[21] Amal S. G, Svetha V, Geoff W, Multi-Class Pattern Classification in Imbalanced Data, in : International Conference on Pattern Recognition,2010.

[22] Akram M, Radwan,Zehra C, Improving Performance Prediction on Education Data with Noise and Class Imbalance. in: Intelligent Automation & Soft Computing,2017.

[23] Yoga P, I Pratama,Anggit F. N, Data Level Approach for Imbalanced Class Handling on Educational Data Mining Multiclass Classification. in: ICOIACT,2018.

[24] [G. Haixiang ,](https://www.sciencedirect.com/science/article/abs/pii/S0957417416307175#!) Li yijing[,](https://www.sciencedirect.com/science/article/abs/pii/S0957417416307175" \l "!), J.shang , Learning from class-imbalanced data: Review of methods and applications, in : Expert Systems with Applications, 2017,pp. 220-239.

[25] [Bartosz. K](https://link.springer.com/article/10.1007/s13748-016-0094-0?TB_iframe=true&error=cookies_not_supported&code=a3e33168-782e-41e5-8585-e731754069d2#auth-Bartosz-Krawczyk), Learning from imbalanced data: open challenges and future directions, 2016, pp.221-232.

[26] Victor Henrique, Gilberto. R.M , Ensemble learning by means of a multi-objective optimization design approach for dealing with imbalanced data sets, in: Expert system applications,2020.

[27] Dewan Md. F , Li Zhang, Chowdhury M.R, M.A. Hossain , Rebecca S, Hybrid decision tree and naïve Bayes classifiers for multi-class classification tasks, in: Expert Systems with Applications,2014, pp.1937–1946.

[28] S R Jiao,J Song, B Liu , A Review of Decision Tree Classification Algorithms for Continuous Variables,in : ICAITA , 2020.

[29] A.Singh , A.Purohit , A Survey on Methods for Solving Data Imbalance Problem for Classification, in : International Journal of Computer Applications, 2015.

[30] Cortez P, Silva A (2008) Using data mining to predict secondary school student performance.

[31] Dua D, Graff C, "UCI machine learning repository", 2021. [Online]. Available: <http://archive.ics.uci.edu/ml>.