An Improved Ensemble Model for Predicting Student Performance Based on Academic Achievement or Dropout Rates

Dissertation submitted in partial fulfillment of the requirements for

the award of the Degree of

Master of Technology

Department of Computer Science & Engineering

With Specialization in Artificial Intelligence & Data Science

Submitted By

GARIMA BHATIA

(2021PUSCEMCEX10528)

Supervised By

Dr. VISHNU SHARMA

Professor & Head

Department of Computer Science & Applications

(Session 2021-23)

Faculty of Computer Science & Engineering

Poornima University

Ramchandrapura, P.O. Vidhani Vatika, Sitapura Extension, Jaipur – 303905, Rajasthan

JULY, 2023

CERTIFICATE

This is to certify that **Ms. GARIMA BHATIA** registration no. **2021PUSCEMCEX10528**, student of M. Tech. (COMPUTER SCIENCE), Department of Computer Science & Engineering, Faculty of Computer Engineering has submitted this dissertation entitled **"An Improved Ensemble Model for Predicting Student Performance Based on Academic Achievement or Dropout Rates"** under the supervision of **Dr. VISHNU SHARMA** working as Professor & Head, Department of Computer Science & Applications Poornima University towards partial fulfillment of the requirements for the Degree of M. Tech. from the Poornima University.

Ms. Shikha Sharma	**Mr. Ankush Kumar Jain**
HOD	M.Tech. Coordinator
Poornima University	Poornima University
Jaipur	Jaipur

CANDIDATE'S DECLARATION

I hereby declare that the work which is being presented in this dissertation entitled **"An Improved Ensemble Model for Predicting Student Performance Based on Academic Achievement or Dropout Rates"** in the partial fulfillment for the award of the Degree of Master of Technology in Computer Science submitted to the Department of Computer Science & Engineering, Poornima University, Jaipur, is an authentic record of original work done by me under the supervision and guidance of Prof.& Head(Dr.) Vishnu Sharma, Department of Computer science & Applications, Poornima University.

I have not submitted the matter embodied in this dissertation for the award of any other degree

Date: 19-08-2023 **GARIMA BHATIA**

Place: Jaipur (2021PUSCEMCEX10528)

SUPERVISOR'S CERTIFICATE

This is to certify that this dissertation is based on original work done by the candidate under my supervision and to the best of my knowledge; this work has not been submitted elsewhere for the award of any degree.

Dated:19-08-2023 **Dr. VISHNU SHARMA**

Professor & Head

Place: Jaipur Dept. of Computer Science &Applications.

ACKNOWLEDGEMENT

I would like to pay my bottomless appreciation and obligations to **Dr. VISHNU SHARMA,** Professor, Poornima University for giving me an opportunity to work under his guidance for preparing the report of my Dissertation Work.

I would also express my sincere thanks to **Ms. Shikha Sharma**, Head of Department (Faculty of Computer Science & Engineering) and **Dr. Ankush Kumar Jain**, Poornima University for his consistent motivation & direction in this regard for his support in Dissertation Work.

I extend my sincere thanks to **Dr. Ajay Khunteta**, Dean Faculty of Engineering & Technology, Poornima University, Dean Research and Development, Poornima University, **Ms. Chandni Kirpalni**, Registrar, Poornima University, **Dr. Manoj Gupta**, Pro-President, Poornima University, for their generous guidance, helps and useful suggestions.

I would like to express my deep gratitude to **Dr. Suresh Chandra Padhy**, President, Poornima University, & **Ar. Rahul Singhi**, Director Poornima Foundation & Poornima University for providing all the necessary facilities which were indispensable in the completion of this dissertation work.

GARIMA BHATIA

(2021PUSCEMCEX10528)

\

ABSTRACT

An important topic of research in the world of education is the prediction of student performance. Educational institutions may develop focused interventions and support systems by accurately forecasting student outcomes and identifying variables that affect academic achievement or dropout rates. The goal of this study is to create a machine learning-based model for predicting student performance based on academic achievement and dropout rates. The research makes use of a wide range of elements, such as demographic data, prior academic performance, socioeconomic circumstances, and other pertinent variables. For the purpose of forecasting student performance and detecting at-risk pupils, a variety of machine learning techniques are investigated and assessed. The effectiveness of the built prediction model in identifying pupils at risk of subpar performance or dropout is evaluated. The results of this study have important ramifications for educational institutions, decision-makers, and teachers, empowering them to invest funds wisely and implement interventions on time to enhance student outcomes. The prediction model offers useful information that may guide the development of evidence-based policies and programmes to lower dropout rates and improve educational equity. Overall, by utilising machine learning approaches, this research advances the subject of predicting student performance and lays the groundwork for future developments.

TABLE OF CONTENTS

FRONT PAGE i

CERTIFICATE ii

CANDIDATE'S DECLARATION iii

SUPERVISOR'S CERTIFICATE iii

ACKNOWLEDGEMENT iv

ABSTRACT v

CHAPTER 1:INTRODUCTION 4

1.1 Background Information 2

1.2 Problem Statement 3

1.3 Research Question 4

1.4 Objectives 4

1.5 Overall aim 4

1.6 Importance of the research 5

1.7 Scope and Limitation 6

1.8 Organization of Dissertation: 6

CHAPTER 2: LITERATURE REVIEW 8

2.1 Categorical Review of Research Works Reviewed 10

CHAPTER 3:THEORETICAL ASPECTS OF PROPOSED WORK 24

3.1 Overview 24

3.2 Supervised Machine Learning Models 25

3.3 Unsupervised Machine Learning Models 29

3.4 Proposed Model 31

CHAPTER 4:IMPLEMENTATION 40

4.1 Overview 40

4.2 Data Collection 41

4.3FeatureSelection 42

4.4 Comparative Analysis 44

CHAPTER 5: RESULTS & DISSCUSION 63

CHAPTER-6 CONCLUSION AND FUTURE SCOPE 86

REFERENCES 91

LIST OF TABLES

Table No.	Name of Table	Page No.
1	Sample Visualization of the Training Dataset	36
2	Sample data of 5 Students after Standardization	46
3	KNN Sample Predicted and True Values	49
4	Multi Layer Perceptron Classification report	54

LIST OF FIGURES

Fig No.	Name of Figure	Page No
1	Types of Machine Learning Models	6
2	Linear Regression Curve	47
3	Neural Network Architecture	48
4	Support Vector Machine Representation	53
5	Clustering technique	71
6	XGBoost Confusion Matrix	82

CHAPTER 1: INTRODUCTION

INTRODUCTION

The use of machine learning techniques to forecast student performance and pinpoint variables that influence academic achievement or dropout rates has gained popularity in recent years. In order to offer the proper assistance and interventions, educational institutions have a strong interest in comprehending and predicting student results. In order to create precise prediction models, it is possible to analyse a lot of data on student traits, academic achievement, and other important variables using machine learning techniques. Academic success is a performance indicator that shows which pupils have reached particular objectives. Numerous stakeholders, including students, professors, and academic institutions, consider it crucial to predict students' success in particular courses or over a whole programme. The most crucial information for this study is student academic data since it provides the most comprehensive picture of the pupils. Likes how much he or she studies and what kinds of subjects he or she prefers and dislikes. These researches can also benefit greatly from IQ testing and a physiological examination. We could determine the sort of teacher motivation he requires if we knew how much time he spends on his academics and how much time he spends on his interests.

Many parties, including students, professors, and academic institutions, consider it crucial to predict students' success in a particular course or during a whole programme. Predicting at-risk students and dropout rates has shown to be an effective use of student performance prediction. It is furthermore utilised to provide personalised recommendations and early warning systems. This setting is where the idea of machine learning was developed. Computers can analyse digital data in ways that are too sophisticated for a human to perform in order to identify patterns and rules. A computer's ability to autonomously learn from experience is the fundamental concept behind machine learning. Machine learning has many different applications. Machine learning is used by search engines to more accurately build relationships between search terms and online pages.

1.1 BACKGROUND INFORMATION

We frequently overlook analogy-based prediction because it is so commonplace. The best instruction and tutoring are always the goals of higher education institutions. Identification of students who do poorly in courses is necessary due to the worrisome rise in drop-out rates at many institutes that has gone unrecognised. In addition to assisting weaker pupils more, this prediction approach would help teachers recognise and reward the most motivated students. To address kids' needs, student data is gathered and used. Other methods miss the pupils' performance pattern throughout the period of the semesters that pass. In order to anticipate students' bad performance throughout the course of their semesters, machine learning algorithms have shown to be a useful tool in forecasting students' performance based on a variety of indicators. The demographic information may be used to identify the pupils who are at danger. All students at educational institutions could benefit from the use of data mining techniques on datasets. It has been demonstrated that the Nave Bayes classifier is the most accurate machine learning method for making such predictions. The factors utilised for prediction were high school grades, admission exam results, and attitudes towards learning, including grades on assignments.

There were additional characteristics that were social and educational in nature. For a variety of reasons, machine learning that uses an incremental method must make adjustments to the taught system before it can be used to make predictions about the actual world. It has been demonstrated that prior grades and test scores have a significant impact on students' academic performance, but there are other important factors that may be taken into account to make reliable forecasts.

1.2 PROBLEM STATEMENT

Creating a machine learning-based model for forecasting student performance and figuring out what factors affect academic success or dropout rates is the main goal of this study. To provide precise predictions, the model will make use of a variety of factors, including demographic data, prior academic performance, socioeconomic background, and other pertinent variables. Educational institutions can undertake focused interventions to enhance student outcomes and lower dropout rates by identifying the important factors of student success. Higher education institutions' principal goal is to give their pupils a high-quality education. To identify the low performers early on, a reliable performance forecast of the pupils is helpful. The goal of this research is to find and collect information that can be used to predict both excellent and bad performances.

1.3 RESEARCH QUESTIONS

This study seeks to provide answers to the following research questions in order to solve the issue at hand:

1. What are the main elements affecting academic success and dropout rates?
2. What machine learning techniques may be used to forecast student performance?
3. How can student data be properly used to create precise prediction models?
4. What actions may be put into place in order to enhance student outcomes and lower dropout rates based on the prediction results?

1.4 OBJECTIVES

The following are the primary goals of this study:

(i) To research different prediction methods and methodologies, and to evaluate their effectiveness using certain criteria.

(ii) To evaluate the methods currently used to predict student performance based on academic success or dropout rates.

(iii) To propose an improved ensemble model for student performance prediction using machine learning techniques on the basis of academic success or dropout rates.

(iv) To find the experimental results and compare and analyze the results with the existing approaches.

1.5 OVERALL AIM

Through analysis of higher education credit data and identification of students who require study assistance, this project intends to assist students in achieving better academic results. The creation of a framework using ML modelling that can assist university personnel and students with student performance and programme completion rates would be great. This research will also examine any trends that could exist among students who finally leave the programme or have trouble finishing it. Technically speaking, the objective of this quantitative study is to compare the performance of two or more machine learning algorithms and models in terms of accuracy, precision, recall, f1 score, and prediction. The Method section provides a detailed explanation of these parameters. An ML model that recognises patterns of students' achievement based on prior data from their higher education credit is the anticipated result. This study intends to assist university students and faculty in evaluating the development of the students and forecasting future academic outcomes (dropout/completion rate) based on present performance. Technically speaking, this study offers an evaluation of the performance of several ML algorithms in relation to the previously mentioned parameters.

1.5 IMPORTANCE OF THE RESEARCH

The results of this study have profound effects on educational institutions, decision-makers, and instructors. Timely interventions and support systems can be enabled by accurate prediction of student performance and identification of variables that affect academic success or dropout rates. Educational institutions are capable of planning individualised plans and delivering targeted help to students who are at risk. The knowledge gathered from this research may be used by policymakers to create programmes and policies that are supported by data in order to lower dropout rates and enhance educational achievements. The prediction model can help teachers spot problematic children before it's too late and conduct the right interventions to improve their academic performance.

1.6 SCOPE AND LIMITATIONS

This study focuses on applying machine learning approaches to predict student performance and dropout rates based on academic success factors. The research includes a wide range of elements, such as demographic data, prior academic performance, socioeconomic circumstances, and other pertinent characteristics. It is crucial to recognise that the prediction models used in this study are based on historical data, and that other factors that were not taken into account by the research may have an impact on future results. The study is restricted to a particular academic setting or student population.

1.7 ORGANIZATION OF THE THESIS

The thesis is organized as follows:

Chapter 1: INTRODUCTION

This chapter contains the issue statement, research questions, and objectives in addition to outlining the importance of the study. It also gives a summary of the research topic.

Chapter 2: LITERATURE REVIEW

This chapter analyses the research on dropout prediction, student performance prediction, and the use of machine learning methods in these areas. It covers pertinent research and methodology and highlights the important variables affecting student results.

Chapter 3: THEORETICAL ASPECTS OF PROPOSED WORK

This chapter contains the theoretical aspects of the proposed work. The proposed model is also mentioned here with all the phases having different function. It covers the data collection and proposed model and describes about all the theoretical aspects. This chapter examines several machine learning techniques that may be used to forecast student performance and spot kids who could be at danger. It gives an overview of the chosen algorithms and how they were put into practise.

Chapter 4: METHODOLOGY

The process for creating the prediction model is described in this chapter, along with approaches for feature selection, model training, and validation. This paragraph explains how the model incorporates pertinent information and machine learning methods.

Chapter 5: RESULT AND ANALYSIS

This chapter discusses the experimental findings and assesses how well the prediction model worked. It evaluates the model's efficiency in forecasting student performance and spotting at-risk pupils by looking at accuracy, precision, recall, and other pertinent indicators. It explains the findings, explores their ramifications, and offers insights into how the discovered indicators or factors relate to the outcomes of the students. It also analyses the study's weaknesses and proposes topics for further investigation.

Chapter 6: CONCLUSION

The last chapter presents a summary of the research results, makes conclusions in light of the study's goals, and suggests future lines of inquiry into the prediction of student performance using machine learning methods based on academic achievement and dropout rates.

It is anticipated that this study will enable the development of precise student performance prediction models, improving educational outcomes and enhancing intervention tactics to aid students in attaining academic success and lowering dropout rates.

CHAPTER 2: LITERATURE REVIEW

2.1 OVERVIEW

This chapter provides a thorough assessment of the research on machine learning algorithms for predicting student performance based on academic achievement and dropout rates. The review seeks to pinpoint the most important research works, approaches, data sets, and conclusions in this field. Beginning with a summary of student performance prediction and emphasizing its importance in educational contexts, the literature review introduces the topic. It examines the different elements that affect student success, including learning styles, socioeconomic status, prior academic achievement, and demographic traits.

2.2 CATEGORICAL REVIEW OF RESEARCH WORKS REVIEWED

Albreiki, B., Zaki, N., & Alashwal, H. (2021) The improvement of the learning environment requires the use of contemporary techniques, tactics, and applications from educational data mining. The most recent study provides practical methods for assessing the learning environment of students by reviewing and exploiting educational data using machine learning and data mining methodologies. Modern academic institutions operate in a very competitive and complex environment. Universities usually struggle with performance assessment, excellent education, performance assessment techniques, and future endeavors. To deal with problems that students encounter while pursuing their education, these universities must implement plans for student intervention. The relevant EDM literature from 2009 to 2021 that relates to identifying at-risk students and dropouts among students is examined in this systematic review.

Harikumar Pallathadka, Alex Wenda, Edwin Ramirez-Asís, Maximiliano Asís-López, Judith Flores-Albornoz, Khongdet Phasinam (2023) In today's cutthroat society, it is essential for an institution to predict student performance, classify people according to their skills, and work to improve their performance in upcoming exams. To increase academic accomplishment, students should be instructed to focus their efforts on a certain subject well in advance. An institution can decrease its failure rates with the use of this kind of study. This study forecasts students' achievement in a course based on their past performance in related courses. The analysis might be put to use for classification or forecasting. Investigated are machine learning techniques including Nave Bayes, ID3, C4.5, and SVM. The experimental investigation makes use of the UCI machinery student performance data set. Algorithms are evaluated based on several criteria, including accuracy and error rate.

M. Chitti, P. Chitti and M. Jayabalan (2020) with the assistance of different Education Data Mining methodologies (EDM), the education sector is developing and expanding at an exponential rate, bringing new and enhanced options to the learning community. If the resulting analytics and the models are interpretable, they will be accepted and believed. It enables educational institutions to improve student performance and lower the dropout rate by advancing their educational practices. This research examines EDM with an emphasis on the variables impacting students' predictions, different algorithms applied, and gaps found. The paper also sheds light on how the prediction model's "black-box" judgments are produced, how different eXplainable AI (XAI) methodologies help to make the model's outcomes understandable, and how they help to provide results that are easy to explain.

Chen, Y., Zhai, L. (2023) The performance of machine learning techniques in various application situations is examined in this research using three different types of task-oriented educational data. To examine various forms of performance prediction, including binary and multi-classification prediction tasks, seven parameter-optimized machine learning techniques are specifically constructed. The experimental part includes a detailed description of the experimental findings as well as four evaluation metrics and visualizations enabling a comparative analysis of various approaches on three tasks. The experimental findings show that Random Forest has outperformed all chosen datasets in terms of generality. Additionally, given how well Decision Tree and Artificial Neural Network models performed on the chosen datasets, they appear to be viable options for tasks involving the prediction of student performance.

Altabrawee, Hussein Osama & Qaisa& Ali, r, Samir. (2019) This study focuses particularly on the impact of utilizing the internet as a learning tool and the impact of students' use of social media on their academic achievement. The classification accuracy and the ROC index performance metric have been used to compare the models. Additionally, many metrics like the classification error, accuracy, recall, and the F measure have been calculated. The ANN (completely connected feed forward multilayer ANN) model produced the greatest results, measuring 0.807 in terms of performance, and 77.04% in terms of classification accuracy. In addition, the decision tree model revealed five elements as significant influences on students' performance.

Hamsa, Hashmia & Indiradevi, Simi & Kizhakkethottam, Jubilant. (2016) The amount of research utilising data mining techniques in the educational field is rapidly expanding. The use of data mining techniques with an educational focus with the aim of exposing hidden knowledge and trends relating to student performance is known as educational data mining. This study aims to build a student's academic performance prediction model for Bachelor's and Master's degree students in the Computer Science and Electronics and Communication streams using the two selected classification methods, Decision Tree and Fuzzy Genetic Algorithm. This study's evaluation criteria included entry score, internal score, and sessional score. The average score from the two sessional exams, the attendance grade, and the assignment grade are added to determine internal marks. The admission score is a weighted mean of the grades achieved in the 10th and 12th grades as well as the entrance scores for degree-seeking students. It comprises of both entrance and degree examination scores for candidates for master's degrees. The resulting prediction model may be used to determine a student's performance for each topic. As a result, teachers may group pupils and act promptly to improve their performance. Utilizing methodical techniques, the performance may eventually be enhanced. Early prediction and solutions may raise the possibility of better results in final exams. Students can see updates and details regarding their scholastic records. Reputable companies that associate with the college may search for students who meet their requirements.

Shahiri, Amirah & Husain, Wahidah & Abdul Rashid, Nur'Aini. (2015) The abundance of data in educational databases makes predicting pupils' success more difficult. The lack of a framework to analyze and track students' development and performance is still an issue in Malaysia today. To raise student accomplishment, it is suggested to conduct a thorough literature study on data mining strategies for forecasting student performance. This paper's main goal is to give a summary of the data mining methods that have been applied to forecast student performance. The focus of this study is also on how the prediction algorithm may be applied to find the key characteristics of a student's data. Using educational data mining approaches, we might really increase student performance and accomplishment more effectively.

Xu, Jie & Moon, Kyeong & Schaar, Mihaela. (2017) We provide a novel machine learning strategy in this work that can address these significant problems for predicting student achievement in degree programs. The recommended method consists of two essential elements. A bi-layered structure composed of a number of base predictors and a cascade of ensemble predictors is initially developed in order to produce predictions based on students' shifting performance levels. Second, a data-driven approach based on latent component models and probabilistic matrix factorization is recommended to uncover course relevance, which is essential for creating powerful base predictors. Using detailed simulations and a dataset of undergraduate student data collected over three years at UCLA, we show that the suggested technique outperforms benchmark alternatives.

M. Nagy and R. Molontay (2018) In this study, we employ and evaluate a variety of machine learning algorithms to identify at-risk people and foretell student dropout from academic courses. The models are based on data from 15,825 first-year students who registered at Budapest University of Technology and Economics between 2010 and 2017 and either graduated or left school. To deal with the issue of missing data, we employ imputation. After completing feature extraction and feature selection, a variety of classifiers, including Decision Tree-based methods, Naive Bayes, k-NN, Linear Models, and Deep Learning with varying input parameters, have been trained. The techniques were evaluated using tenfold cross-validation, and the top models were Gradient Boosted Trees and Deep Learning, with AUCs of 0.808 and 0.811, respectively.

Ofori, F., Maina, E. & Gitonga, R. (2020) The research seeks to discover the most effective machine learning model for forecasting student performance as well as the most suitable machine learning model for enhancing learning through a thorough analysis of the literature. Inconclusive findings on the machine learning model that best predicts students' performance were shown by the empirical review. The study's main finding is that every educational institution in the world should place the greatest importance on forecasting students' success. It would be essential to use a variety of machine learning techniques to forecast student performance with accuracy. In order to improve learning outcomes and anticipate students' performance, it is critical to properly rank machine learning models.

Yağcı, M. (2022) The findings of the midterm test grades are the main data in this study's innovative machine learning-based model that forecasts undergraduate students' final exam scores. The outcomes of the machine learning algorithms random forests, closest neighbour, support vector machines, logistic regression, Naive Bayes, and k-nearest neighbour were computed and compared in order to predict the students' final test scores. The dataset included the academic performance grades of 1854 students who took Turkish Language-I at a public university in Turkey in the autumn semester of 2019–2020. According to the results, the proposed model has a classification accuracy that ranges from 70 to 75 percent.

Máté Baranyi, Marcell Nagy, and Roland Molontay. (2020) In this study, we attempt to predict the ultimate academic achievement of students at the Budapest University of Technology and Economics using cutting-edge machine learning methods, such as deep neural networks and gradient boosted trees. Based on the information provided at the time of enrolment, the dropout prediction was made. With the use of cutting-edge interpretable machine learning approaches like permutation significance and SHAP values, we interpret our machine learning models in addition to making predictions. The top-performing deep learning model marginally beats XGBoost, the state-of-the-art benchmark model for tabular data, with accuracy and AUC of 72.4% and 0.771, respectively.

Hassan, H., Anuar, S., Ahmad, N.B. (2019) The goal of this study is to use educational data mining tools to pinpoint the variables affecting students' performance. The datasets came from a student information system and an e-learning platform at a Malaysian public university. Eight distinct group models were developed by combining five multi-classifiers—Random Forest, Bagging, AdaBoost, Stacking, and Majority Vote classifier—with three base-classifiers—Decision Tree, Artificial Neural Network, and Support Vector Machine. Additionally, academic, demographical, economic, and behavioral e-learning aspects were utilized in this study. The best accuracy of the classifier model was then optimized. A new model for forecasting student performance was subsequently developed. The outcome demonstrates that employing a meta-classifier model with optimized hyper parameters and combining demographics and behavior gave greater accuracy to predict students' success.

Alhothali, A., Albsisi, M., Assalahi, H., & Aldosemani, T. (2022) The popularity of both massive open online courses (MOOCs) and small private online courses (SPOCs) has increased recently. In view of the growing needs and issues in online education, several academics have examined methods to predict student outcomes, such as performance and dropout in online courses. This article provides a thorough analysis of previous studies to anticipate the outcomes of online learners using machine and deep learning methodologies. In this study, the characteristics of online courses that are used to predict students' learning outcomes are identified and categorized, the prediction outputs are determined, the methodologies and feature extraction strategies are determined, and the evaluation metrics are determined, and to provide a taxonomy for the analysis of related studies.

M. M. Tamada, J. F. de Magalhães Netto and D. P. R. de Lima (2019) Our research intends to find approaches that employ Machine Learning (ML) methods to lower these high dropout rates. Method: To find, sort, and categorize primary studies, we performed a systematic review. Results: From the 199 publications that were initially found in academic databases, 13 papers were used in the study. The study details the publications' historical development, the machine learning methods employed, the features of the data collected, as well as the solutions put out to lower dropout rates in remote learning. Our research gives a comprehensive review of the current state-of-the-art approaches suggested to lower dropout rates using ML techniques, and it may help direct future research and tool development.

Azimi, S., Popa, C.-G., & Cucić, T. (2020) In this study, we demonstrate that data gathered from an online learning management system may be effectively used to forecast students' overall performance as well as to suggest timely intervention measures to raise students' performance levels. The findings of this study imply that to improve student development, effective intervention measures might be proposed as early as the middle of the course. Based on the findings of this study, we also provide an assistive pedagogical tool to help identify difficult pupils and recommend early intervention techniques.

R. Katarya, J. Gaba, A. Garg and V. Verma (2021) Universities have the opportunity to reduce their dropout rate and assist students in improving their performance by being able to predict students' academic performance in advance. In this area, research is being done to determine the optimal algorithm to utilize and the factors to take into account when forecasting students' academic achievement. Over time, there has been an increase in this type of scientific activity. This essay does a study of the methods for predicting academic success utilized in numerous research publications and also identifies any methodological flaws.

Jha, Nikhil & Ghergulescu, Ioana & Moldovan, Arghir-Nicolae. (2019) This study adds to the body of knowledge by examining the performance of machine learning models created based on several types of variables, such as demographic data, assessment data, and engagement with the VLE, in predicting student dropout and test results. According to an investigation of the OULAD dataset, models that take into account how students interact with the VLE have a high AUC of up to 0.91 for dropout prediction and 0.93 for outcome prediction when using Gradient Boosting Machines.

Sachio Hirokawa. (2018) Every possible combination of the four categories of attributes—behavioral characteristics, demographic information, academic background, and parental involvement—was used in this study to examine the accuracy of the predictions provided. The behavioral characteristics are displayed as numerical data. As an attribute name and value pair, we presented them to you instead. Data produced by this vectorization has 417 dimensions as opposed to the 68 dimensions of data that is naively represented. With accuracy 0.8096 and F-measure 0.7726, we used support vector machines and feature selection to provide the greatest feature selection prediction performance. We proved that the behavioral trait is crucial since without it, the accuracy would not have reached 0.7905. When the behavior characteristic and the demographic information were integrated, the F-measure climbed to 0.7662.

W. Nuankaew and J. Thongkam (2020) In this study, feature selection techniques that exclude occurrences that were incorrectly categorized and Synthetic Minority Over-Sampling Technique are presented as ways to enhance the prediction of student academic achievement. It assesses how well seven models—Naive Bayes, Sequential Minimum Optimization, Artificial Neural Network, k-Nearest Neighbour, REPTree, Partial decision trees, and Random Forest—perform in predicting students' academic success. 9,458 Rajabhat Maha Sarakham University students in Thailand provided the data between 2015 and 2018. Precision, recall, and F-measure were used to assess how well the model performed. The experimental findings showed that the Random Forest technique considerably enhances the accuracy, recall, and F-measure of prediction models for students' academic performance, with precision reaching 41.70 percent, 41.40 percent, and 41.6 percent, respectively.

2.3 REVIEW OF EXISTING TECHNIQUES FOR STUDENT PERFORMANCE PREDICTION

Aslam, N., Khan, I. U., Alamri, L. H., & Almuslim, R. S. (2021) The study examined data from two courses—mathematics and Portuguese—that included demographic, socioeconomic, educational, and course grade information. SMOTE (synthetic minority oversampling method) is used to address the imbalance problem in the data set. The performance model is assessed using all feature sets, with the exception of G2 and G3, using assessment criteria like as precision, recall, F-score, and accuracy. The results showed how useful the recommended DL model was for generating early predictions about the students' academic performance. The model's accuracy was 0.964 for the data set from the Portuguese course and 0.932 for the data set from the mathematics course, respectively. Similar results are obtained for mathematics, where the accuracy is 0.94, and for Portuguese, 0.99. Additionally, each trait plays a crucial role in forecasting the student's academic success.

Ankit Porwal, Aditi Tulchhia (2022) The educational dataset may be used to execute a number of data mining techniques to predict student achievement. In order to predict student performance, this study looks into a number of data mining techniques, including Naive Bayes, K-Nearest Neighbours (K-NN), Decision Trees, and Neural Networks. The primary goal of the article is to compare and contrast the effectiveness of various data mining techniques to predict student performance. The research shows that, of the methodologies discussed above, neural networks offer the best accurate prediction, with a 94.96% accuracy rate.

Dhilipan, J., Vijayalakshmi, N., Suriya, S., & Christopher, A. (2021, February) The creation of student accomplishment prediction models to predict students' performance in academic institutions is one of the key areas of research for education data mining. A prediction technique has been given using the students' grades from the previous semester, the 10th grade, and the 12th grade. The study is evaluated using KNN classifiers, entropy, decision trees, and binomial logical regression. This framework will assist the student in comprehending their final grade and improving their academic behavior to get a higher grade.

Hayder, A. (2022) The technical purpose of this study was to examine the accuracy, precision, recall, f1 score, and prediction performance of two or more machine learning algorithms. The other objective was to provide a framework for predicting and rating student achievement. These ML algorithms were legitimated since the accuracy they obtained was higher than their corresponding baseline accuracy. This implies that the results of the thesis' application point in the direction of the prediction component's success. The SVM model beat other models in terms of the specified parameters because of the nature of the SVM model, which features linearly expanding multi parameter that fits the increased inputs. This wasn't the case with the ANN's structure.

Hashim, Ali & Akeel, Wid & Khalaf, Alaa. (2020) This study assessed the performance of many supervised machine learning algorithms, including Decision Tree, Naive Bayes, Logistic Regression, Support Vector Machine, K-Nearest Neighbor, Sequential Minimal Optimization, and Neural Network. We trained a model utilizing datasets provided by courses in the bachelor study programs of the College of Computer Science and Information Technology, University of Basra, for the academic years 2017–2018 and 2018–2019 in order to predict student performance on final examinations. The findings indicated that the logistic regression classifier is the most efficient in properly predicting students' real final grades (68.7% for passed and 88.8% for failed).

Ioanna Lykourentzou, Ioannis Giannoukos, Vassilis Nikolopoulos, George Mpardis, Vassili Loumos (2009) This research proposes a dropout prediction strategy for online courses based on three well-known machine learning methods and comprehensive student data. Feed-forward neural networks, support vector machines, and probabilistic ensemble simplified fuzzy ARTMAP are the machine learning algorithms employed. Three judgement schemes, which incorporate the findings of the three machine learning approaches in various ways, were also examined because one methodology might not be able to effectively categorise some online learners while another might be successful. The method's results were discovered to be much better than those published in the pertinent literature after it was analysed in terms of overall accuracy, sensitivity, and precision.

S. Batool, J. Rashid, M. W. Nisar, J. Kim, T. Mahmood and A. Hussain (2021) It is impossible to exaggerate the value of education for both a person and the nation as a whole. Many students withdraw from various academic courses each year. This study explores the relationship between student demographics and academic achievement. The Random Forest classification algorithm is used to forecast student performance on final examinations. Three openly available datasets with varied sets of demographic data were used to analyze these traits and their impact on students' performance. Techniques like hold-out and cross-validation are used to assess the results of experiments. Using Random Forest, three different datasets were combined to give F-measures of 81.20%, 95.10%, and 84.16%.

Kabathova, J., & Drlik, M. (2021) The study's main goals are to emphasize the importance of the data understanding and data collection phases, underline the limitations of the educational datasets currently available, compare the performance of various machine learning classifiers, and show that even a small set of features can accurately predict a student's dropout if performance metrics are carefully considered. The prediction accuracy ranged from 77 to 93% for unobserved data from the future academic year. To mitigate the effect of the small dataset's size on the high performance metric values attained, the homogeneity of machine learning classifiers was compared and analysed. The outcomes demonstrated that a number of machine learning methods might be successfully used to analyse a small collection of academic data.

I. Khan, A. Al Sadiri, A. R. Ahmad and N. Jabeur (2019) The objective of this study is to see whether we can create a model that can advise students early in the semester about their expected outcomes (when examined for 15% grades) in an introductory programming course. We performed 11 machine learning algorithms (from 5 categories) using WEKA to a data source, and the results revealed that Decision Tree (J48) had the greatest degree of accuracy in terms of correctly recognized instances, F-Measure rate, and true positive detections. The findings of this study will help the students predict their final grades and change their academic behavior to improve their marks.

Rastrollo-Guerrero, J. L., Gómez-Pulido, J. A., & Durán-Domínguez, A. (2020) For learning environments like schools and colleges, predicting students' success is one of the most crucial issues since it aids in the creation of practical systems that, among other things, promote academic achievement and prevent dropout. As a result, rigorous analysis and processing of this data might provide us with insightful insights about the pupils' expertise. This data serves as the input for cutting-edge algorithms and techniques that can forecast pupils' academic achievement. About 70 papers were evaluated for this study in order to show the many modern approaches that are widely used to predict students' performance as well as the objectives they must meet in this field. Machine learning, collaborative filtering, recommender systems, and artificial neural networks are a few of the approaches and techniques connected to artificial intelligence.

2.3.1 DATA PRE-PROCESSING AND VISUALIZATIONS

C. S. K and K. S. Kumar (2022) The likelihood of receiving a job offer after graduation is influenced by both academic success and student performance over the course of their whole career. Machine learning algorithms are crucial in analyzing and forecasting the likelihood that students will be placed based on their prior academic performance. We gathered student data from a reputable technical institute for this article. The data set includes several variables that affect a student's prospects, and these variables are investigated and visualized. Before running or using machine algorithms on this data set, we sought to analyze the data and provide visualizations and insights. Our major goals in this study are to undertake data preparation, data analysis, and understanding.

A. Jain and S. Solanki (2019) Preparing the data for machine learning models is a crucial step after choosing the right datasets. The dataset is first loaded as a file with comma separated values. Null values are then examined. The final score, a numerical variable, is then changed into a category variable. Students who received scores between 15 and 20 are classified as high performers, those who had scores between 10 and 14 are classified as average performers, and those who received scores of less than or equal to 9 are classified as poor performers. Through the 1-hot-encoding approach, all nominal qualities—such as the mother's and father's occupations—were transformed into numerical attributes.

Sultana, J. & Macigi, Usha Rani & Farquad, H.. (2019) D2L, a learning management system, is the source of data used to create the proposed algorithms to predict student success. There are 1100 student entries in the dataset. There are 11 distinct characteristics. Duplicate records are removed from the dataset by pre-processing; empty fields are found and filled with the intended data. Weka and Rapid Miner tools are used to apply Deep Learning techniques like Deep Neural Net and Data Mining techniques like Random Forest, SVM, Decision Tree, and Nave Bayes to the data collection. Results are assessed using a few metrics. Deep Neural Network and Decision Tree are superior to other approaches for predicting student performance because they offer deep predictions and achieve the best outcomes in terms of high accuracy, kappa-statistic, sensitivity, and specificity.

2.3.2 FEATURE SELECTION

Gajwani, J., Chakraborty, P. (2021) To categorize a student's performance based on a subset of behavioural and academic parameters, we advise using feature selection, supervised machine learning algorithms like logistic regression, decision trees, and naive Bayes classifiers, as well as ensemble machine learning algorithms like boosting, bagging, voting, and random forest classifier. To determine the characteristics that would most likely affect and improve prediction, we created a number of graphs. The best results with our dataset are produced by ensemble machine learning algorithms, which have an accuracy of up to 75%, according to experiments with various methods. This has several uses, including helping students increase their academic performance, tailoring e-learning courses to better meet students' requirements, and offering unique solutions for various student groups.

Punlumjeak, W., Rachburee, N., & Arunrerk, J. (2017) The most significant and inherent properties should be identified using feature selection procedures prior to applying classification algorithms in this experiment. An experiment was conducted to evaluate the prediction model's effectiveness. The results of the experiment showed that the feature selection method employing mutual information and a neural network classifier produced the highest overall accuracy for the student data at Rajamangala University of Technology Thanyaburi, scoring 90.60%. This exercise will aid in the comprehension of the feature selection and classification techniques, which are the most effective ways to assess students' performance in a cloud computing environment by students, instructors, and management. They will be better able to pinpoint the issues and elements influencing pupils' performance.

Zaffar, M., Hashmani, M. A., Savita, K. S., & Rizvi, S. S. H. (2018) The feature selection process, which is regarded as a vital phase in the machine learning process, is one of the most popular and significant strategies for pre-processing data. The main emphasis of our investigation in this work is on six important FS algorithms: CfsSubsetEval, ChiSquared-AttributeEval, FilteredAttributeEval, GainRatioAttribute-Eval, Principal ComponentsAttributeEval, and ReliefAttributeEval.In order to assess the effectiveness of six feature selection strategies, this study makes use of two distinct student datasets. These algorithms' effectiveness is measured by precision, recall, F-measure, and prediction accuracy (accurately categorised occurrences). The term "Fmeasure" refers to the harmonic mean of recall and accuracy. The results from datasets 1 and 2, which were employed in this analysis, are contrasted with those from our earlier research.

2.3.2 CLASSIFICATION AND PREDICTION

Nicholas Robert Beckham, Limas Jaya Akeh, Giodio Nathanael Pratama Mitaart, Jurike V Moniaga (2023) Using Pearson correlation between each characteristic and the student G3 result, we will use this article to attempt to identify elements that might either impede or help student performance. According to the findings, prior failures will have a negative association with student grades of -0.360415 and a positive correlation with student grades of 0.217147 with respect to the mother's education. We attempt to anticipate student grades using ML models after determining the factor(s) that affect student grades to show if the factor(s) indeed affect student grades. Our MLP 12-Neuron model performs best with an RMSE of 4.32. Decision Tree comes in third with an RMSE value of 5.69, followed by Random Forest in second place with an RMSE value of 4.52.

Baig, Mirza Azam; Shaikh, Sarmad Ahmed; Khatri, Kamlesh Kumar; Shaikh, Muneer Ahmed; Khan, Muhammad Zohaib; Rauf, Mahira Abdul (2023) The effectiveness of machine learning (ML) approaches for forecasting students' academic progress is examined in this research. We describe the notion of student performance prediction in education and its many manifestations. We examine a variety of machine learning (ML) techniques, such as the Fuzzy C-Means, the Multi-Layer Perceptron (MPL), the Logistic Regression (LR), and the Random Forest (RF) algorithms, for forecasting student success in the classroom. In this study, the previously developed and recently suggested models for predicting student performance are carefully reviewed. The study examines several algorithmic pairings, including FCM-MLP, FCM-LR, and FCM-RF, and offers the complete results of each pairing. These strategies are assessed using quantitative metrics like accuracy and detection rate.

Koutina, M., Kermanidis, K.L. (2011) The goal of this study is to determine which machine learning method is more effective in forecasting postgraduate students in informatics at Ionian University's final grade. Consequently, six well-known classification methods are tested using five academic courses, each of which serves as a separate dataset. The databases are also enhanced with information related to demographics, short-term performance, and classroom conduct. The primary research problems of the current work are the limited size of the datasets and the imbalance in the distribution of class values. For the first time in a performance prediction application, a number of strategies are used to overcome these problems, including resampling and feature selection. In comparison to other comparable algorithms, Nave Bayes and 1-NN produced the best prediction results, which are highly good.

Oyedeji, A., Salami, A., Folorunsho, O., & Abolade, O. (2020, March 30) For academic institutions and education professionals, analysing student academic performance is crucial in order to identify strategies to boost each student's performance. The project analysed the prior performance of pupils and tested this information using machine learning technologies. Individual factors included age, demographic distribution, family history, and attitude towards learning. Using test and train data, three models—linear regression for supervised learning, linear regression with deep learning, and neural network—were examined. The model with the best mean average error (MAE) was linear regression for supervised learning.

2.4 RESEARCH GAPS

The evaluation of students' performance in educational settings reveals the extent of the efforts made by those settings to improve the learning of underprivileged or ordinary students. The importance of adopting EDM models is that they use student historical data to forecast future performance that has not yet occurred. A number of academics have been motivated by this concept to create classification models that forecast the as-yet-unknown labels of future cases. Numerous academic institutions and researchers began to become interested in the field of student performance prediction in order to categorize the educational level of student performance.

Although the educational sector employs a number of methods for gathering useful data on the characteristics of students who are engaged in the learning process, it is necessary to develop a model for student performance assessment to help students and faculty members advance their performance to the next level. The main objective of this research work is to examine and identify the useful rules and patterns to motivate students to handle their education and careers in a good manner, as well as to improve the and functions academics to supervise the policies for students' benefits, even though there are some existing studies related to the EDM in the educational sector.

2.5 RESEARCH OBJECTIVES

The entire thesis is divided into different objectives and they are mentioned as follows: (i) To research different prediction methods and methodologies, and to evaluate their effectiveness using certain criteria. (ii) To evaluate the methods currently used to predict student performance based on academic success or dropout rates. (iii) To propose an improved ensemble model for student performance prediction using machine learning techniques on the basis of academic success or dropout rates. (iv) To find the experimental results and compare and analyze the results with the existing approaches.

CHAPTER 3: THEORETICAL ASPECTS OF PROPOSED WORK

3.1 OVERVIEW

A mathematical representation of the results of the training process is referred to as a machine learning model. The study of various algorithms that may develop a model automatically through practice and historical data is known as machine learning. A machine learning model is comparable to software created for computers that can identify patterns or behaviors based on past experience or data. A machine learning (ML) model that captures the patterns found in the training data is produced by the learning algorithm after it analyses the training data for patterns. Machine learning models are programs that have been taught to see patterns in fresh data and anticipate outcomes. These models are trained over a set of data, and then given an algorithm to reason over the data, extract the pattern from feed data, and learn from those data; once these models get trained, they can be used to predict the unseen dataset. There are different types of machine learning.

Machine learning algorithms are trained utilizing labeled, unlabeled, or mixed data before being used to generate machine learning models. Data scientists employ various machine learning algorithms as the foundation for various models since they are suitable for various objectives, such as classification or prediction modeling. A specific algorithm is altered as more data is added, improving its ability to handle a particular task, and eventually develops into a machine learning model. A typical technique used for classification and prediction modeling, for instance, is the decision tree. A data scientist could use diverse animal photos to train a decision tree algorithm in order to build a machine learning model that can recognize numerous animal species. The data would gradually change the algorithm, making it ever more accurate at categorizing photographs of animals. This would ultimately develop into a machine learning model.

There are three learning models for algorithms that are based on various business objectives and data sources. Each algorithm for machine learning falls into one of three models:

- Supervised Learning

- Unsupervised Learning
- Reinforcement Learning

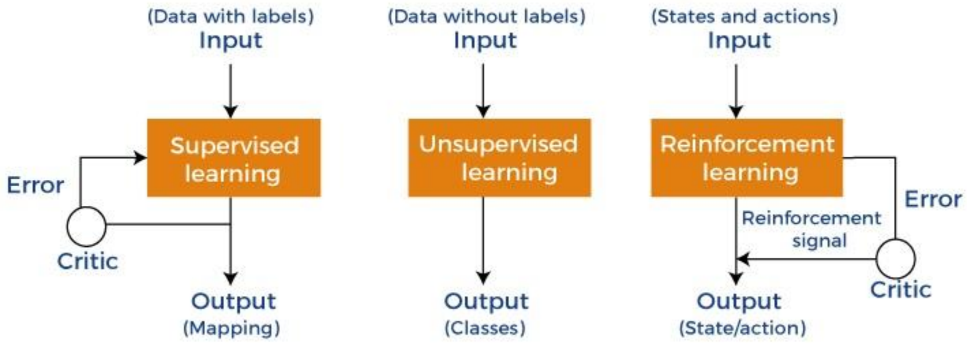


Fig. 3.1 Types of Machine Learning Models

3.2 SUPERVISED MACHINE LEARNING MODELS

The easiest machine learning model to comprehend is supervised learning, in which input data is referred to as training data and has a known label or outcome as an output. As a result, it operates on the idea of input-output pairs. In order to conduct prediction, it is necessary to develop a function that can be learned using a training set of data before being applied to unknown data. Task-based supervised learning is evaluated using labeled data sets.

On straightforward real-world issues, we can put a supervised learning model into practice. We might create a supervised learning model to predict a person's height based on their age, for instance, if we had a dataset that included both their age and height. Models for supervised learning are further divided into two groups:

3.2.1 REGRESSION

The output in regression issues is a continuous variable. Here are a few examples of frequently used regression models:

a) **Linear Regression:**

The simplest machine learning model is linear regression, which tries to predict one output variable from one or more input variables. A linear equation that combines a set of input values (x) and the anticipated output (y) for the set of those input values is how linear regression is represented. It is shown as the following line:

$$Y = bx+ c$$

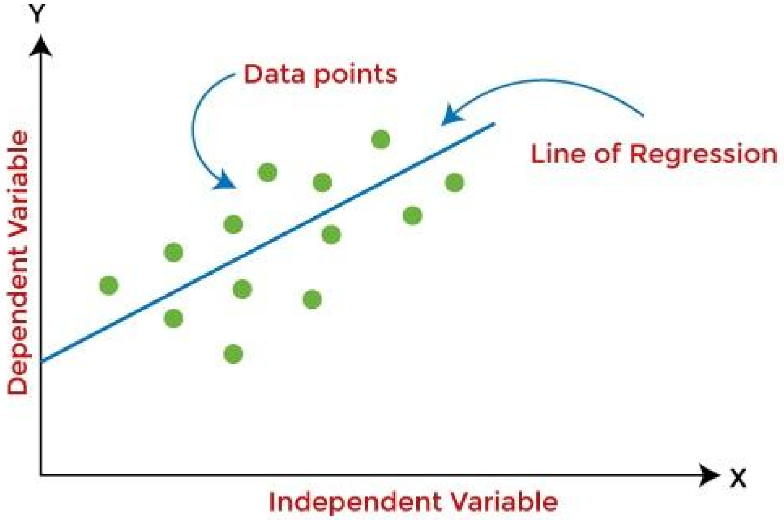


Fig. 3.2 Linear Regression Curve

Finding the line that best fits the data points is the main goal of the linear regression model. Multiple linear regression (find a best-fit plane) and polynomial regression (find the best-fit curve) are extensions of linear regression.

b) Decision Tree

The well-liked machine learning models that may be applied to both classification and regression issues are decision trees. A decision tree employs a tree-like structure to organize decisions and the potential outcomes and repercussions of those actions. Each internal node in this diagram represents a test on an attribute, and each branch the result of the test.

A decision tree's outcome will be more accurate the more nodes it contains. Decision trees have the virtue of being logical and simple to use, yet they are inaccurate. In operations research, decision trees are often utilized, particularly in machine learning, strategic planning, and decision analysis.

c) Random Forest

The ensemble learning technique called Random Forest uses several different decision trees. Each decision tree in a random forest makes a forecast, and the prediction that receives the most votes is taken as the result. A random forest model may be applied to classification and regression issues.

The majority of votes are used to determine the random forest's result for the categorization challenge. In contrast, the output of a regression task is derived from the mean or average of the predictions made by each tree.

d) Neural Networks

Neural networks, commonly referred to as artificial neural networks, are a subset of machine learning. Artificial neurons are used to create neural networks, which are fashioned after the structure and operation of the human brain. In a neural network, each artificial neuron communicates with a large number of other neurons, and with so many neurons linked, a complex cognitive structure is produced.

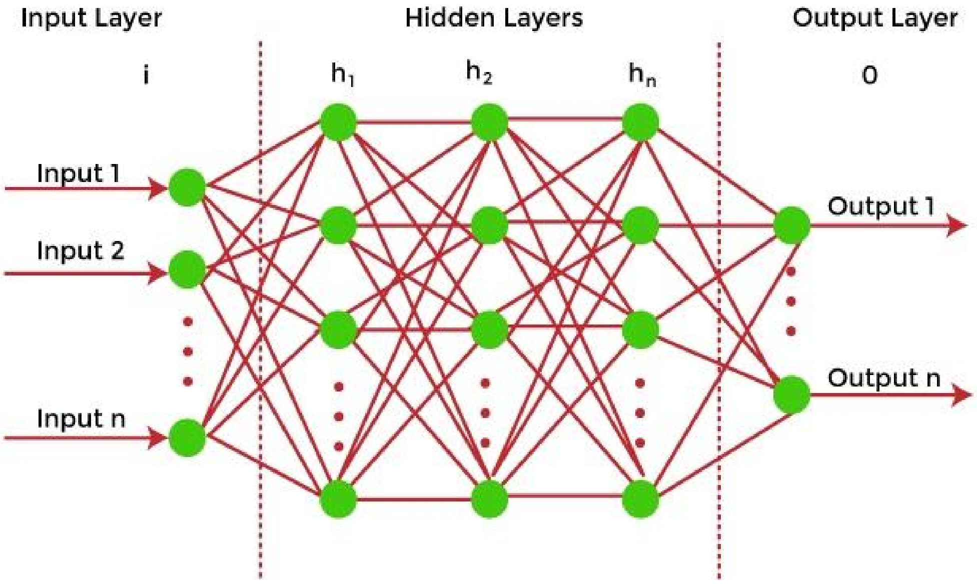


Fig. 3.3 Neural Network Architecture

One input layer, one or more hidden layers, and one output layer make up a neural network's multilayer structure. Data is transferred from one layer to the other neuron in the following layers because each neuron is connected to another neuron. The neural network's last layer, also known as the output layer, is where data finally produces an output.

3.2.2 CLASSIFICATION

The second category of supervised learning methods uses classification models to draw inferences from categorical values that are observed. The categorization model, for instance, may determine if an email is spam or not, whether a customer would buy a product, etc. In order to anticipate two classes and divide the output into other groups, classification algorithms are utilized. In classification, the dataset is divided into several categories using a classifier model, and each category is given a label. Some popular algorithms are:

a) **Logistic Regression**

Machine learning categorization issues are solved using logistic regression. They are utilised to predict categorical variables and are comparable to linear regression. It can forecast the result as True or False, 0 or 1, or Yes or No. However, it gives probabilistic values between 0 and 1 rather than the precise numbers.

b) **Support Vector Machine**

The well-known machine learning technique known as SVM is frequently used for classification and regression problems. But specifically, it's employed to address categorization issues. Finding the best decision boundaries in an N-dimensional space that can divide data points into classes is the basic goal of SVM, and the optimum decision boundary is referred to as a Hyperplane. Support vectors are the extreme vectors that SVM chooses to locate the hyperplane.

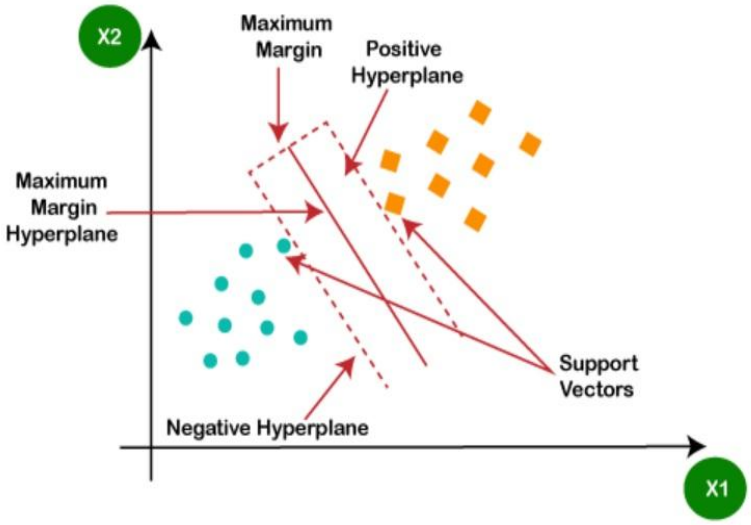


Fig. 3.4 Support Vector Machine Representation

c) **Naïve Bayes**

Another well-liked classification approach in machine learning is naive Bayes. Because it is based on the Bayes theorem and assumes that the characteristics are independent, it is given this name:

$$P(y|X) = \frac{P(X|y) * P(y)}{P(X)}$$

Fig. 3.5 Naïve Bayes Formula

Every naive Bayes classifier makes the assumption that a certain variable's value is independent of all other variables and features. For instance, if a fruit needs to be categorized according to its colour, shape, and flavour. Mango will therefore be identified as being yellow, oval, and sweet. Each feature in this case operates independently of the others.

3.3 UNSUPERVISED MACHINE LEARNING MODELS

Unsupervised Machine learning models utilize unsupervised learning instead of supervised learning, allowing the model to gain knowledge from the unlabeled training data. The model makes predictions about the results based on the unlabeled dataset. Unsupervised learning allows the model to discover hidden patterns in the dataset on its own, with no outside help. Three tasks, which are as follows, are the key uses for unsupervised learning models:

3.3.1 CLUSTERING

An unsupervised learning approach called clustering includes grouping or "groping" the data points into several clusters according to their similarities and differences. The items that have the most commonalities stay in the same group and have little to no overlap with those from other groups.

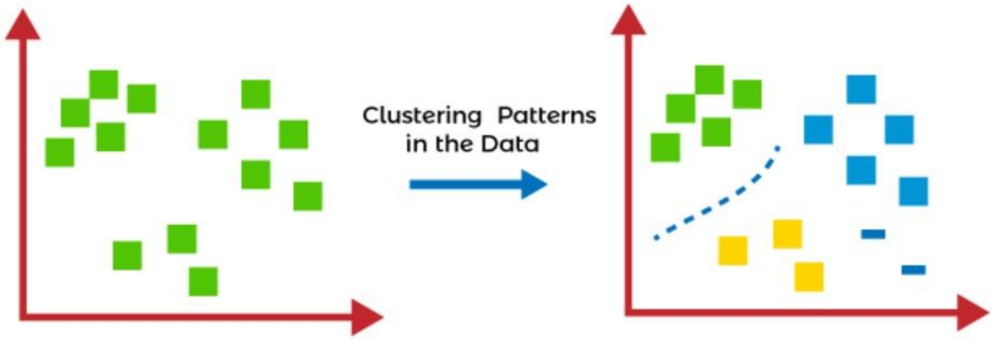


Fig. 3.6 Clustering technique

Numerous activities, including image segmentation, statistical data analysis, market segmentation, etc., can benefit from the usage of clustering techniques. K-means Clustering, hierarchical Clustering, DBSCAN, and other popular clustering methods are some examples.

3.3.2 **ASSOCIATION RULE LEARNING**

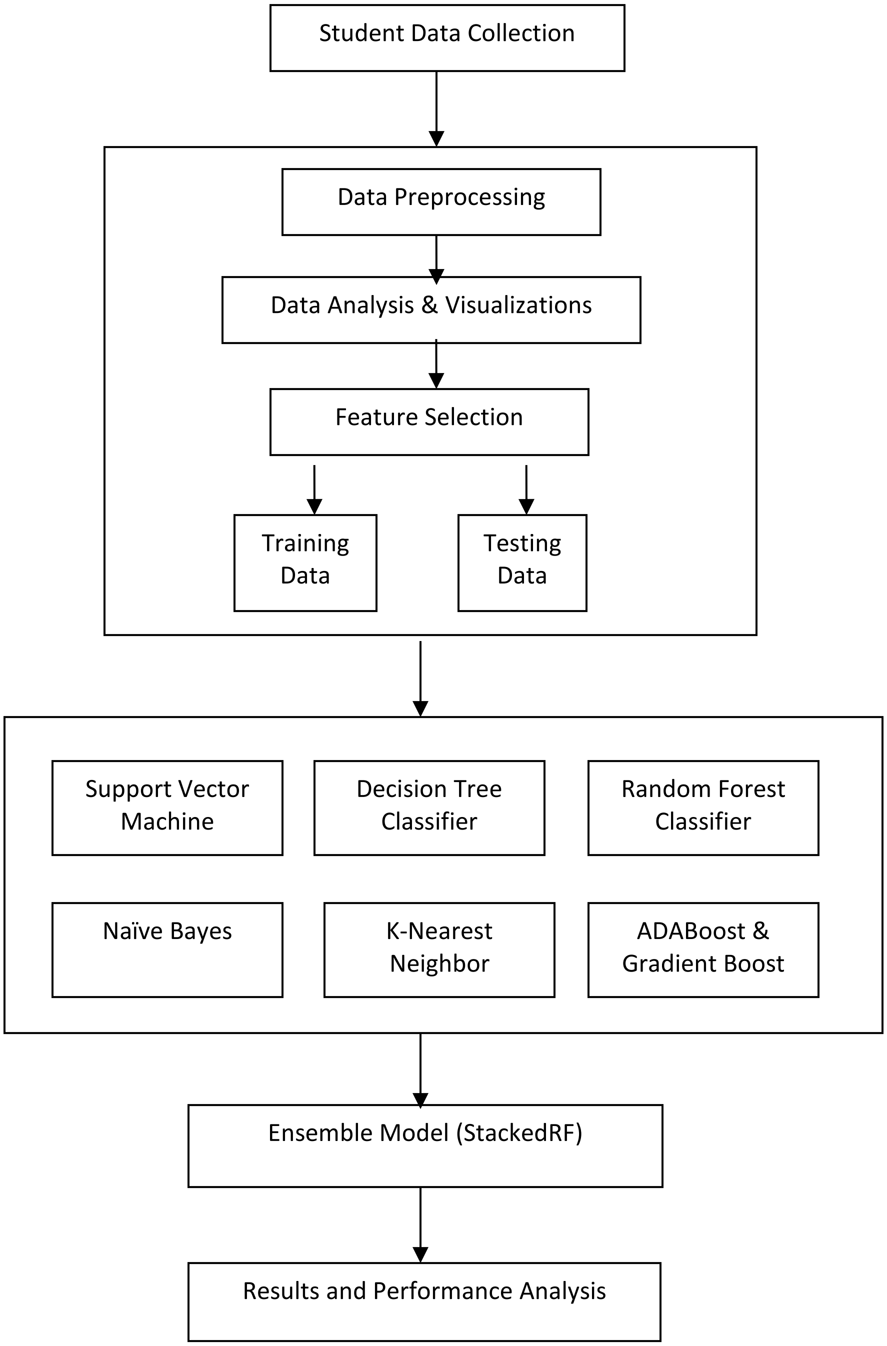
An unsupervised learning method called association rule learning identifies intriguing relationships between variables in a sizable dataset. This learning algorithm's primary goal is to identify the dependencies between data items and then map the variables in a way that maximizes profit. This method is mostly used in continuous production, web usage mining, market basket analysis, etc. Apriori, Eclat, and FP-growth algorithms are a few of the well-known algorithms for learning association rules.

3.3.3 DIMESIONALITY REDUCTION

The dimensionality of a dataset refers to how many features or variables are present, while the dimensionality reduction methodology refers to the method used to reduce the dimensionality. Although more data yields more accurate findings, it can also have negative effects on the model's or algorithm's performance due to over fitting problems. Techniques for dimensionality reduction are applied in these situations. "It is a process of converting the higher dimensions dataset into the lesser dimensions dataset while ensuring that it provides similar information". Several dimensionality reduction techniques are there, such as Singular Value Decomposition and PCA (Principal Component Analysis).

3.4 PROPOSED MODEL

The next sections describe how the aforementioned models and calculations are put to use. The following Figure 3.1 illustrates the general tasks performed during the given work stages.



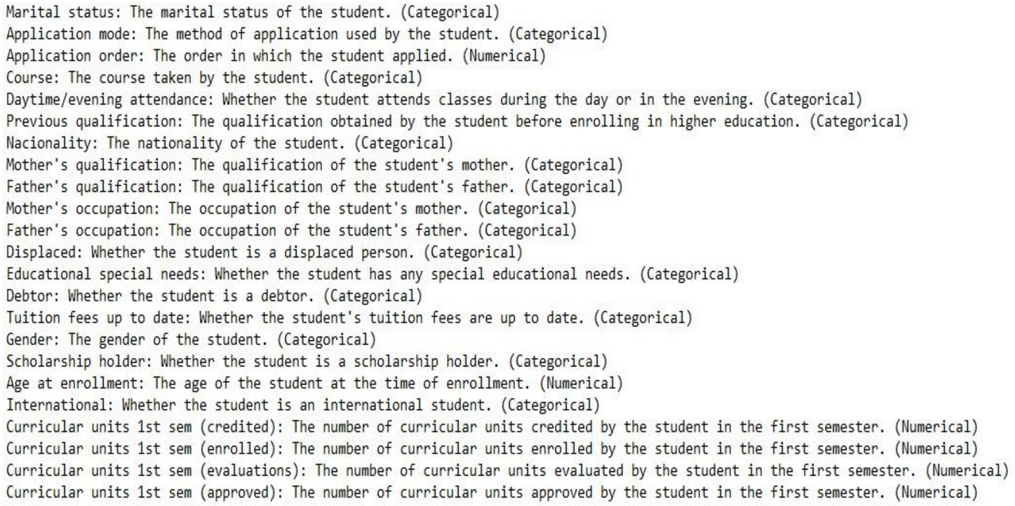
The results of the suggested model are derived in steps. The initial problem statement is crystal clear, and the necessary data is gathered from student databases. The proposed model is divided into different stages as follows:

1. Data Collection
2. Data Preprocessing
3. Data Analysis and Visualizations
4. Feature Selection
5. Spilling Data into Training and Testing Data

6. Training Different Machine Learning Models (Model Selection)
7. Prediction using Ensemble Model (StackedXRF)
8. Results and Performance Analysis

3.4.1 DATA COLLECTION

A set of student information from the university's records served as the paper's data source. The selection of the subset of all accessible data that you will be working with is the focus of this stage. Ideally, ML challenges begin with a large amount of data (examples or observations) for which you already know the desired solution. Labeled data is information for which you already know the desired outcome. To study the effects of these variables on student dropout and academic achievement, this dataset includes information from a higher education institution on a variety of undergraduate students' characteristics, social-economic determinants, and academic performance. The following image shows all the features of the dataset and their 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)
Nacionality: 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)
```

Fig. 3.8 Dataset Description

This dataset has been taken from kaggle.com and it includes the details which help to predict student’s performance on the basis of academic success or dropout rates. The dataset contains 35 attributes (including the Target) and 4425 student’s details.

3.4.2 DATA PREPROCESSING

Data conversion and format transformation are the two processes that pre-processing often entails. The format translation operation is mostly carried out when any type of data is present, such as when.xls files are converted into.csv files. Second, data transformation is calculated with the assumption that the dataset contains data in many formats, which are afterwards changed into accurate forms. Without converting them to numerical values, the majority of machine learning algorithms cannot handle categorical variables. The encoding of categorical variables affects how well many algorithms work.

In the given dataset, we check for any null or duplicate value by using methods like isnull() BS duplicated(). The isnull() methods shows the columns having null values and duplicated().sum() shows the sum of duplicated values for the column. Here in this data set, it looks like there is no null or duplicate value in any column. Only “Target” column has non-numeric value in the dataset, and this “Target” column is the output column so we need to convert it to numeric value so that we can easily find it’s correlation with other columns. There are three unique values for Target column, which we replace as Dropout->0, Enrolled->1 and Graduate->2.

3.4.3 DATA ANALYSIS AND VISUALIZATION

The study of how to visually portray data is known as data visualization. It effectively communicates findings from data by visually displaying the data. We may obtain a visual overview of our data via data visualization. The human mind processes and comprehends any given data more easily when it is presented with images, maps, and graphs. Both small and big data sets benefit from data visualization, but enormous data sets are where it really shines because it is difficult to manually view, let alone analyze, and comprehend, all of our data.

Python has a number of charting libraries, including Matplotlib, Seaborn, and many additional data visualization tools with a variety of capabilities for building educational, unique, and visually appealing charts to show data in the simplest and most powerful manner.

Python libraries used for data visualization include Matplotlib and Seaborn. They contain built-in modules for generating various graphs. While Seaborn is generally used for statistical graphs, Matplotlib is used to integrate graphs into programs.

3.4.4 FEATURE SELECTION

The attributes that are considered for the feature selection are those which are required for effective prediction of student performance based on the academic success or dropout rates. Additionally, the attribute selection procedure is carried out using data from samples of high school and college students in a variety of settings. Based on their academic achievement and demographic information, students' profiles are established. For accurate prediction of student performance, the Target column has the three unique values: Dropout, Enrolled and Graduate.

The data which is required for the prediction is selected at this stage and is as follows: Application mode, displaced, debtor, Tution fees up to date, Gender, Scholarship holder, age at enrollment, Curricular units 1st sem (enrolled), Curricular units 1st sem (approved), Curricular units 1st sem (grade), Curricular units 2nd sem (enrolled), Curricular units 2nd sem (approved), Curricular units 2nd sem (grade) and Target.

3.4.5 SPLITTING DATA INTO TRAINING & TESTING DATA

A method for assessing a machine learning algorithm's performance is the train-test split. It may be applied to issues involving classification or regression as well as any supervised learning technique. The process entails splitting the dataset into two subgroups. The training dataset is the initial subset, which is used to fit the model. The model is not trained using the second subset; rather, it is given the input element of the dataset, and its predictions are then produced and contrasted with the expected values. The test dataset is the second dataset in question.

For fitting the machine learning model, use the train dataset. Test Dataset: Used to assess how well a machine learning model fits the data.

The goal is to gauge the machine learning model's performance on fresh data—data that were not used to train the model. We anticipate applying the model in this way. In other words, to fit it to the data that is now available with inputs and outputs that are known, then to forecast on fresh examples in the future when we do not have the anticipated output or goal values. When a suitable size dataset is provided, the train-test technique is acceptable. This entails taking a random sample without replacement of around 80% of the rows and adding them to your training set. You add the final 20 percent to your test set. Note that for a certain train test split, the colours in "Features" and "Target" indicate where their data will go ("X_train," "X_test," "y_train," "y_test").

3.4.6 MODEL SELECTION

The process of choosing a single machine learning model out of a group of potential candidates for a training dataset is known as model selection. Model selection is a technique that may be used to compare models of the same type that have been set with various model hyper parameters (for example, different kernels in an SVM) as well as models of other types (such as logistic regression, SVM, KNN, etc.).

The ideal kind of model that may be applied to a certain issue can be predicted. For instance, it is quite probable that deep learning-based predictive models would outperform statistical-based models if you are modeling a problem in natural language processing. Instead of looking for the best model, the goal is to choose one that fits our needs and other criteria like performance, robustness, complexity, etc.

Depending on variables including the type of data available, data noise, and the kind of predicting issue, different models perform differently. When choosing a model, the greater contextual world around the model must be taken into account. a model that is flawless in terms of computing but cannot be explained.

Different machine learning models like Support Vector Machine, decision tree classifier, Random forest classifier, Naïve Bayes, K-Nearest Neighbor, ADABoost and Gradient Boosting were trained and their performance was compared. The model which gave the best performance was seleted.

3.4.7 ENSEMBLE MODEL (StackedRF)

Using a variety of modeling techniques or training data sets, ensemble modeling is the process of building numerous varied models to predict a result. The ensemble model then combines each base model's forecast into a single overall prediction for the unobserved data. The goal of employing ensemble models is to lower the prediction's generalization error. The ensemble technique reduces the model's prediction error as long as the base models are varied and independent. The strategy looks on the collective knowledge of people to make a forecast. The ensemble model behaves and functions as a single model even when it has numerous basis models.

In order to increase the precision of predictive analytics and data mining applications, ensemble modeling is the act of executing two or more related but separate analytical models and then combining the results into a single score or spread. A single model based on a single data sample might include biases, excessive variability, or plain mistakes that impair the validity of its analytical results in predictive modeling and other forms of data analytics. Similar problems might result from using particular modeling methodologies. Data scientists and other data analysts can lessen the effects of such restrictions and give corporate decision makers greater information by combining various models or analyzing several samples.

Here we select the best performing machine learning model in the previous stage of model selection. To get more efficient results, we build an improved ensemble model using the selected model as the base model. Various ensemble techniques are available for the same. Bagging, stacking, and boosting are the three primary categories of ensemble learning techniques.

Bagging entails averaging the predictions from many decision trees that have been fitted to various samples of the same dataset. When numerous distinct model types are fitted to the same data, stacking is used to learn how to combine the predictions most effectively. A weighted average of the predictions is produced through boosting, which entails adding ensemble members in a sequential manner that correct the predictions provided by earlier models.

3.4.8 RESULTS AND PERFORMANCE ANALYSIS

One of the crucial phases in creating a successful machine learning model is evaluating its performance. Different metrics—also referred to as performance metrics or evaluation metrics—are used to assess the effectiveness or quality of the model. These performance indicators enable us to evaluate how successfully our model handled the supplied data.

By adjusting the hyper-parameters, we can make the model perform better. Performance indicators assist measure how successfully a machine learning (ML) model generalizes on new or previously unexplored data.

The four basic performance analysis parameters are:

a) **Recall**

It is comparable to the Precision metric and attempts to determine the percentage of genuine positives that were mistakenly detected. It can be measured as True Positive, or forecasts that really match the overall number of positives, either properly forecasted as positive or wrongly anticipated as negative (true Positive and false negative). The following is the calculation method for recall:

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

Fig. 3.9 Recall Formula

b) **Precision**

The accuracy measure's restriction is overridden by the precision metric. The fraction of positive predictions that were accurate is determined by precision. It may be measured as the proportion of True Positives, or forecasts, to all positive predictions (True Positive and False Positive), which are truly accurate.

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

Fig. 3.10 Precision Formula

c) **F1 Score**

A binary classification model is evaluated using the F-score or F1 Score metric based on the predictions provided for the positive class. With the use of Precision and Recall, it is computed. It is a particular kind of score that combines Precision and Recall. As a result, the F1 Score may be determined by taking the harmonic mean of both accuracy and recall and giving each variable equal weight. The following is the formula for determining the F1 score:

$$F1 - score = 2 * \frac{precision * recall}{precision + recall}$$

Fig. 3.11 F1 Score Formula

d) Accuracy

One of the easiest Classification metrics to use is accuracy, which is calculated as the proportion of accurate predictions to all other predictions. The formula to calculate accuracy is:

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ number\ of\ predictions}$$

Fig. 3.12 Accuracy Formula

This chapter explained the theoretical aspects of the proposed work. The proposed model is also discussed in this chapter, which includes 8 different stages for the prediction of student academic performance. Here, we discussed various types of machine learning models and the design of the proposed model. In the next chapter we will discuss the implementation of the proposed model in which the mentioned machine learning models will be trained and their performance will be analyzed to choose the best model out of them. The best model then will be ensemble to get better prediction accuracy.

CHAPTER 4: IMPLEMENTATION

4.1 OVERVIEW

The technique of knowledge discovery in educational data mining is established using student data from the institutions. The major goal of EDM is to give evaluation results of student performance that help to frame the quality of education, improve the student pass rate, and improve the institution's overall outcomes. The student log is used in this instance to gather big data samples that are stored in databases of educational institutions. The collected student records are given in a variety of formats, together with the student's personal information and academic logs.

The institution's findings can be more accurate and better thanks to the effective student categorization approach. Based on the learned knowledge patterns, student characteristics are developed for the purpose of determining the outcomes and made available to academics for informed decision-making. The instructional strategies are changed in response to their choices.

The computation of a student's success in higher grades is based on their personal information, academic performance, family situation, conduct, etc. For improved outcomes, an improved ensemble model Stacked Random Forest (StackedRF) is suggested in this chapter for computing student performance and student categorization. The model makes use of ensemble and simple classification techniques for that. The suggested study makes use of relevant and significant student qualities that have a substantial impact on the outcomes to increase classification accuracy and model performance.

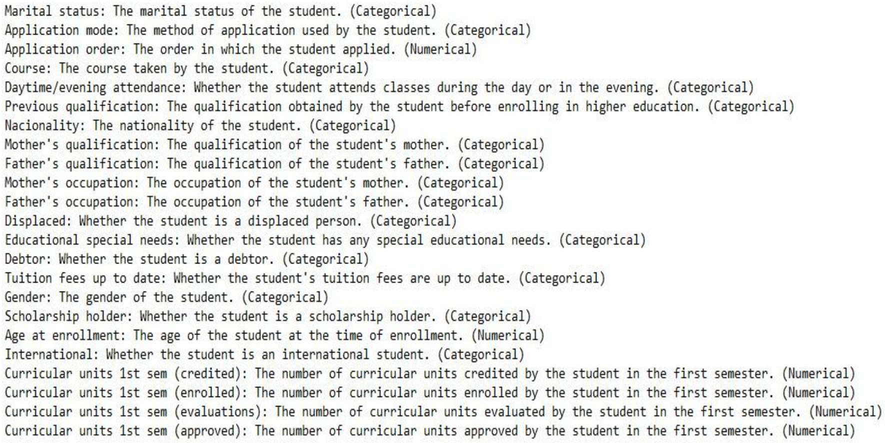
In this chapter, an Ensemble model StackedRF is proposed for student performance prediction on the basis of academic success or dropout rates. StackedRF is developed from a student data sample of 4425 and compared with the algorithms such as Naïve Bayes, Logistic Regression, Random Forest, XGBoost Classifier, Support Vector Machine and Multi layer Perceptron. The results of the suggested model are derived in steps. The initial problem statement is crystal clear, and the necessary data is gathered from student databases. The proposed model is divided into different stages as follows:

9. Data Collection

10. Data Preprocessing
11. Data Analysis and Visualizations
12. Feature Selection
13. Spilling Data into Training and Testing Data
14. Training Different Machine Learning Models (Model Selection)
15. Prediction using Ensemble Model (StackedRF)
16. Results and Performance Analysis\

4.2 DATA COLLECTION AND PREPROCESSING

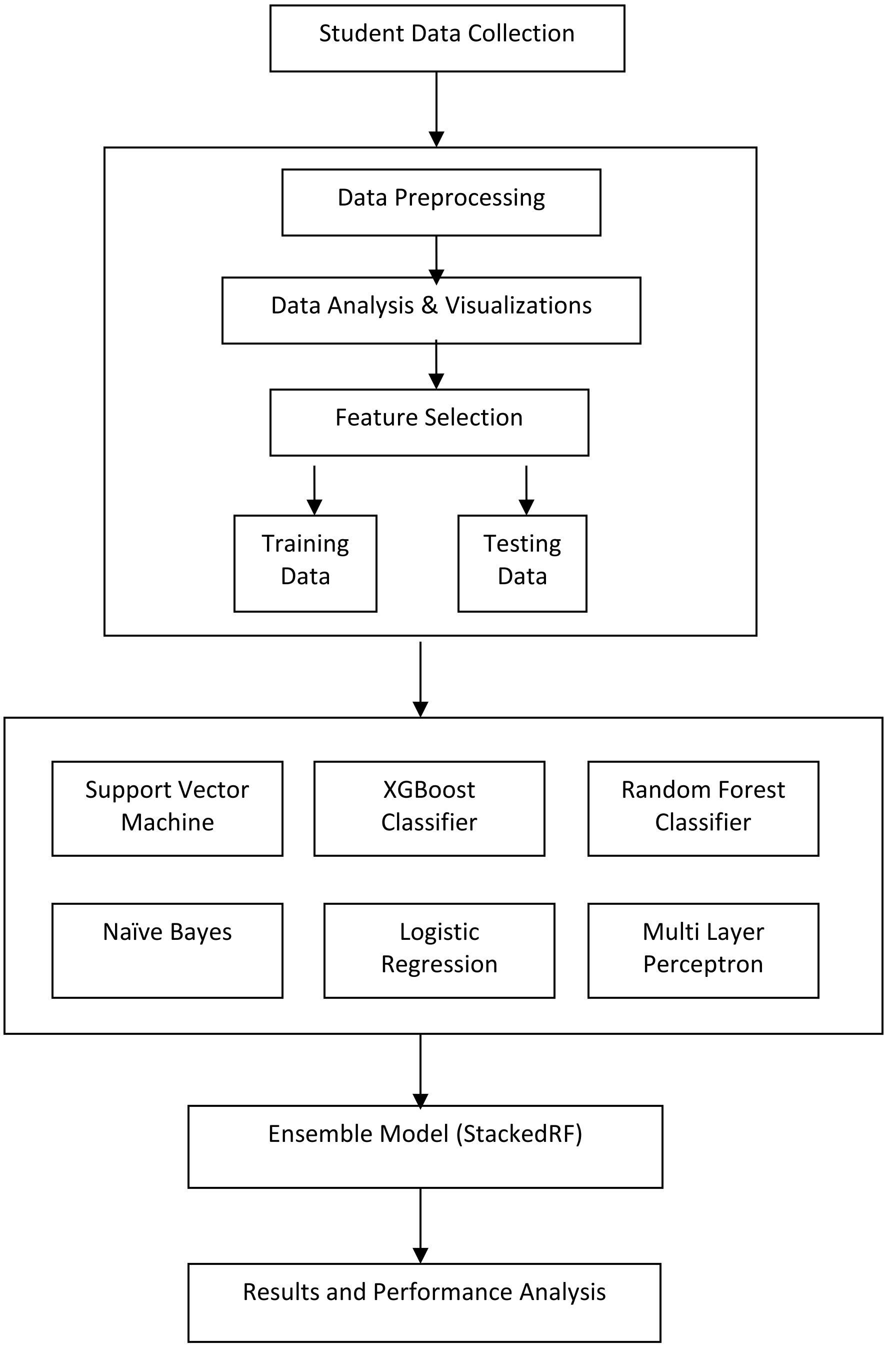
The paper's data came from a collection of student records kept by the institution. This step focuses on choosing the subset of all accessible data that you will be using. It's ideal for ML problems to start with a lot of data (examples or observations) for which you already know the ideal outcome. Labeled data is information for which you already know the desired outcome. To study the effects of these variables on student dropout and academic achievement, this dataset includes information from a higher education institution on a variety of undergraduate students' characteristics, social-economic determinants, and academic performance. The following image shows all the features of the dataset and their 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)
Nacionality: 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)
```

Fig. 4.1 Dataset Description

Data conversion and format transformation are the two processes that pre-processing often entails. The format translation operation is mostly carried out when any type of data is present, such as when.xls files are converted into.csv files. Second, data transformation is calculated with the assumption that the dataset contains data in many formats, which are afterwards changed into accurate forms. Without converting them to numerical values, the majority of machine learning algorithms cannot handle categorical variables. The encoding of categorical variables affects how well many algorithms work.



In the given dataset, we check for any null or duplicate value by using methods like isnull() BS duplicated(). The isnull() methods shows the columns having null values and duplicated().sum() shows the sum of duplicated values for the column. Here in this data set, it looks like there is no null or duplicate value in any column. Only "Target" column has non-numeric value in the dataset, and this "Target" column is the output column so we need to convert it to numeric value so that we can easily find it's correlation with other columns. There are three unique values for Target column, which we replace as Dropout->0, Enrolled->1 and Graduate->2.

4.3 FEATURE SELECTION

The attributes that are considered for the feature selection are those which are required for effective prediction of student performance based on the academic success or dropout rates. Additionally, the attribute selection procedure is carried out using data from samples of high school and college students in a variety of settings. Based on their academic achievement and demographic information, students' profiles are established. For accurate prediction of student performance, the Target column has the three unique values: Dropout, Enrolled and Graduate.

At this point, the necessary information is chosen, and it is as follows: application in progress, displaced, debtor, current tuition expenses, Gender, scholarship holder, age at enrollment, first semester curriculum (enrolled), first semester curriculum (approved), first semester curriculum (grade), second semester curriculum (enrolled), second semester curriculum (approved), second semester curriculum (grade), and target.

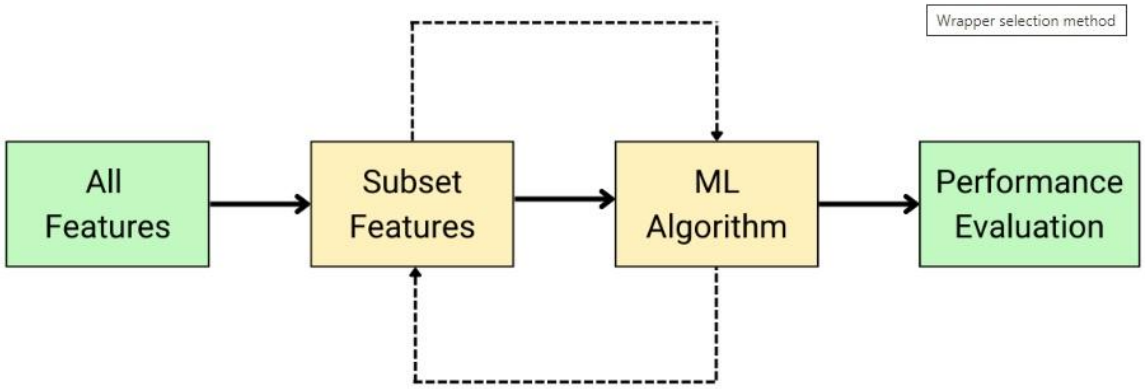


Fig. 4.3 Wrapper Selection Method for Feature Selection

Working with a dataset that has a lot of characteristics is typical while solving a machine learning challenge. All of these traits, however, may not always be beneficial for creating the greatest model. In fact, employing irrelevant characteristics might hurt the model's performance and reduce its capacity to generalize to new data. A model's complexity might rise and the generalization error can rise as a result of adding too many features. For the model-building phase, it is crucial to carefully choose the important elements. The objective is to have the fewest features feasible while yet delivering excellent performance. The merits and downsides of various feature selection approaches will be covered in this blog.

Here in this research, Wrapper Selection method is used for Feature Selection. When using the wrapper strategy for feature selection, an algorithm must be used to evaluate the model's performance over all possible feature subsets. A performance comparison of the model with different feature sets is the outcome, which compares the efficacy of learning with various subsets of features to the evaluation criterion. The user may then select the appropriate set of characteristics for which the model works best.

The greedy approach used by the wrapper technique is characterized by the model's performance being assessed over all feasible feature combinations until a certain criterion is met. Consider a huge dataset with more than 50 features; each feature subset would need at least 1275 model fits. It is a critical flaw in the wrapper approach. When compared to feature selection strategies based on the filter method, the wrapper method, however, yields superior results.

4.4 COMPARATIVE ANALYSIS OF MACHINE LEARNING ALGORITHMS

Machine learning algorithms are used by programs to find hidden patterns in data, anticipate outcomes, and improve performance based on prior performance. For different tasks in machine learning, a variety of algorithms may be used, such as simple linear regression for prediction problems like stock market forecasting and the KNN algorithm for classification problems.

In this research, we have performed comparative analysis of a number of machine learning algorithms. This comparative analysis helps us to identify which machine learning algorithms is working best on our dataset. We choose that algorithm for further improvement of the performance metrics. The following is the detailed description of those algorithms which we have compared in this research.

1) Support Vector Machine

Classification and regression problems are resolved using Support Vector Machine, or SVM, one of the most used supervised learning techniques. It is mostly used, nevertheless, in Machine Learning Classification problems. In order to swiftly categorize new data points in the future, the SVM algorithm aims to define the best line or decision boundary that can split n-dimensional space into classes. The name of this best choice boundary is a hyperplane. The extreme vectors and points that help create the hyperplane are chosen via SVM. The SVM approach is based on support vectors, which are utilized to represent these extreme situations.

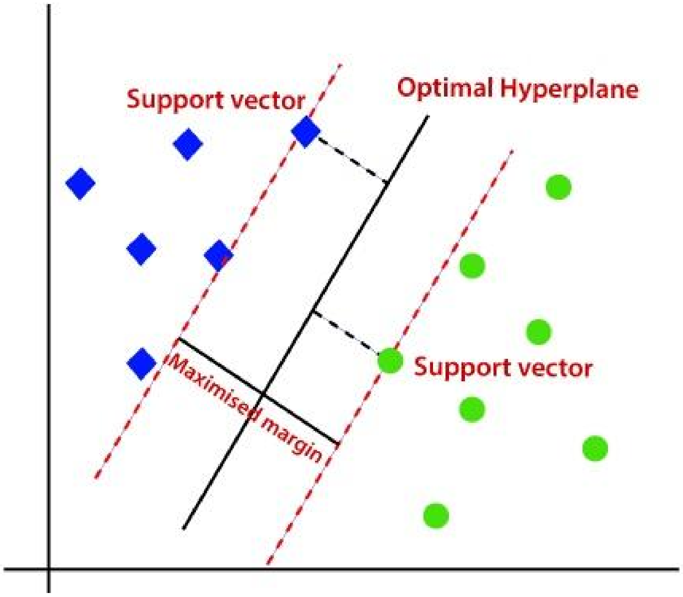


Fig. 4.4 Support Vector Machine

Support Vector: Support vectors are the spots that are most near the hyperplane. A separation line will be established using these data points.

Margin: Margin is the separation between the hyperplane and the observations (support vectors) that are closest to the hyperplane. Large margins are regarded as favorable margins in SVM. Hard margins and soft margins are the two sorts of margins.

We don't need to worry about additional observations since SVM is defined only in terms of the support vectors because the margin is determined using the points that are closest to the hyperplane (support vectors), which eliminates the requirement for other observations. Therefore, SVM gains from certain organic speedups.

2) **Decision Tree Classifier**

Classification and regression problems can be resolved using the supervised learning technique known as a decision tree, however this approach is frequently preferred. It is a tree-structured classifier, where each leaf node represents the classification outcome and inside nodes represent the features of a dataset. The two nodes in a decision tree are the Decision Node and Leaf Node. Decision nodes are used to make decisions and have many branches, whereas Leaf nodes are the outcomes of decisions and do not have any more branches. The features of the submitted dataset are utilized to run the test or get the conclusions.

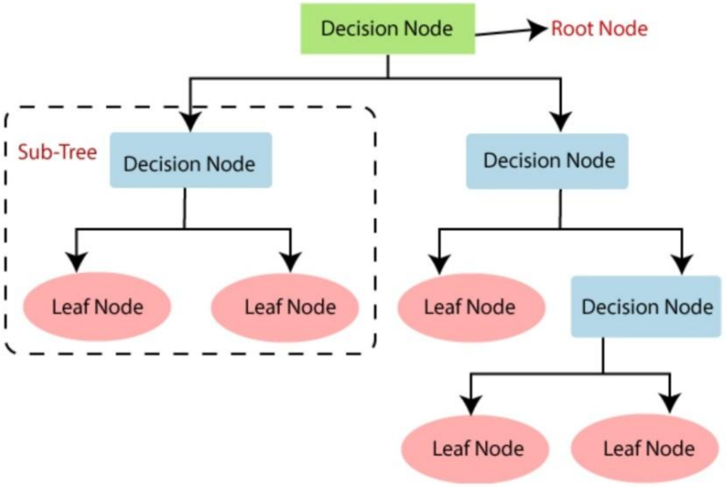


Fig. 4.5 Decision Tree Classifier

The procedure in a decision tree starts at the root node and moves upward to predict the class of the supplied dataset. By comparing the values of the record (actual dataset) attribute with those of the root attribute, this method tracks the branch and moves to the next node.

Before moving on to the next node, the algorithm double-checks the attribute value with the other sub-nodes. The process is continued until the leaf node of the tree is reached.

3) Random Forest Classifier

Preferred algorithm for machine learning A component of the supervised learning approach is Random Forest. It may be used to solve classification and regression-related ML problems. It is based on the concept of ensemble learning, a technique for combining several classifiers to solve complex problems and improve model performance.

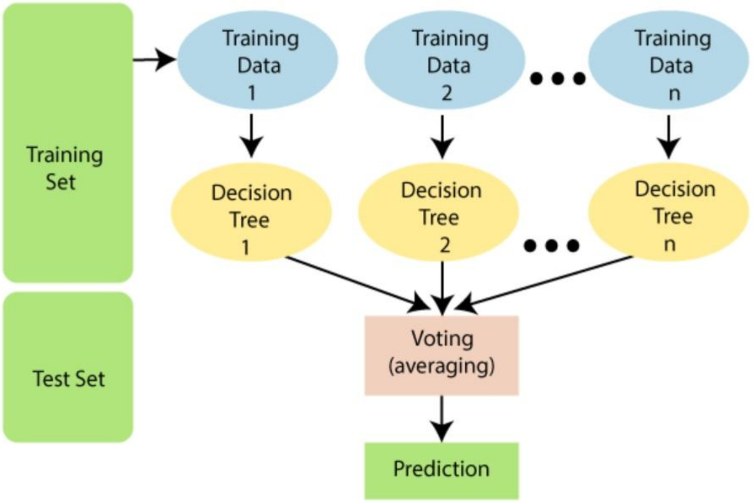


Fig. 4.6 Random Forest Classifier

According to what its name implies, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." The random forest takes predictions from each decision tree and predicts the outcome based on the votes of the majority of projections rather than relying just on one decision tree. The greater number of trees in the forest prevents higher accuracy and over fitting.

To create the random forest, N decision trees are joined initially. Then, predictions are generated for each tree that was created during the first stage.

4) Naïve Bayes

The Nave Bayes algorithm is a supervised learning method for classification issues that is based on the Bayes theorem. It is mostly employed in text categorization with a large training set. The Naive Bayes Classifier is one of the most straightforward and efficient classification algorithms available today. It aids in the development of rapid machine learning models capable of making accurate predictions.

It provides predictions based on the likelihood that an object will occur since it is a probabilistic classifier. A few applications for Naive Bayes algorithms include spam filtration, sentiment analysis, and article categorization. The Nave Bayes algorithm is formulated as follows:

$$P(A|B) = \underline{P(B \mid A)\ P(A)}$$

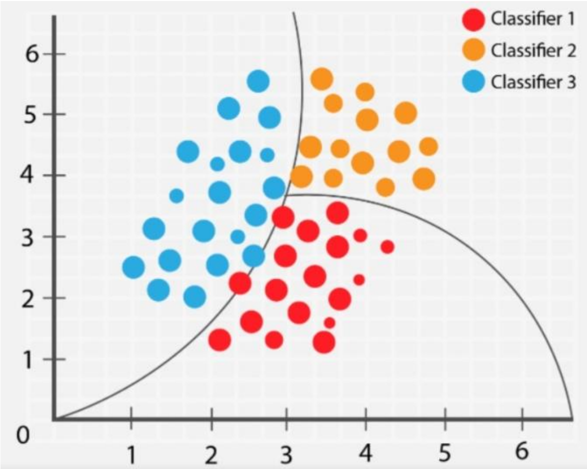


Fig. 4.7 Naïve Bayes Curve

It is referred regarded as naïve since it assumes that the existence of one attribute is unconnected to the prevalence of other traits. For instance, if a red, spherical, sweet fruit is recognized as an apple based on its color, shape, and flavor. Consequently, each trait contributes to identifying it as an apple without depending on the others. It is called Bayes because it relies on the Bayes' Theorem premise.

5) **K-Nearest Neighbors**

One of the most fundamental but crucial classification methods in machine learning is K-Nearest Neighbors. It falls under the category of supervised learning and is widely used in intrusion detection, data mining, and pattern recognition.

Since it is non-parametric and makes no underlying assumptions about the distribution of the data (unlike other algorithms like GMM, which assume a Gaussian distribution of the input data), it is extensively applicable in real-world applications. We are provided some prior information (also known as training data), which organizes coordinates according to an attribute. Take the following table of data points with two characteristics as an illustration:

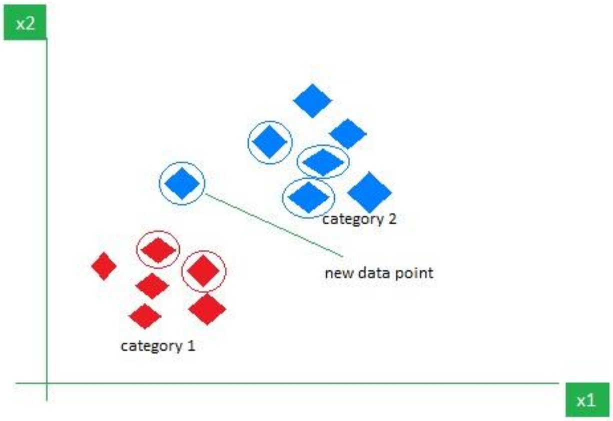


Fig. 4.8 K-Nearest Neighbor

Assign these points to a group by examining the training set now that you have another set of data points (also known as testing data). The unclassified locations are designated as "White," as you'll see. We might be able to find some clusters or groupings if we plot these points on a graph. Now that a point has been declassified, we may classify it by looking at which group its closest neighbors is a part of. This indicates that a point's likelihood of being labeled as "Red" increases the closer it is to a group of points with the same color designation.

6) **Logistic Regression**

One of the most often used Machine Learning algorithms, within the category of Supervised Learning, is logistic regression. Using a predetermined set of independent factors, it is used to predict the categorical dependent variable.

In a categorical dependent variable, the output is predicted via logistic regression. As a result, the result must be a discrete or categorical value. Rather of providing the precise values of 0 and 1, it provides the probabilistic values that fall between 0 and 1. It can be either Yes or No, 0 or 1, true or false, etc.

With the exception of how they are applied, logistic regression and linear regression are very similar. While logistic regression is used to solve classification difficulties, linear regression is used to solve regression problems.

7) XGBoost Classifier

A machine learning approach called XGBoost classifier is used for structured and tabular data. A gradient boosted decision tree implementation created for speed and performance is called XGBoost. An extreme gradient boost algorithm is XGBoost. Therefore, it is a large machine learning method with several components.

8) Multi Layer Perceptron

The most complicated architecture of artificial neural networks is defined by the multi-layer perceptron. It is largely constructed from several perceptron layers. This notebook will walk you through creating a neural network using the widely used deep learning framework TensorFlow. Back propagation's algorithm is another name for a typical learning method for MLP networks.

A feed-forward artificial neural network that produces a set of outputs from a collection of inputs is called a multilayer perceptron (MLP). A directed graph connecting the input nodes in several layers of input nodes connected between the input and output layers is an MLP's defining feature. Backpropagation is used by MLP to train the network. A deep learning technique is MLP.

4.5 PROPOSED TECHNIQUE

After comparative analysis of several techniques: Naïve Bayes, Logistic Regression, Random Forest, XGBoost Classifier, Support Vector Machine and Multi layer Perceptron we choose the algorithm which gives best performance among all. The performance is measured on the basis of accuracy, precision, recall and f1_score.

Here in this section, we will discuss about the proposed technique. The algorithm which performs best out of the algorithms mentioned above is Random Forest with a good accuracy score of 92.01%. Thus to improve this algorithm we use ensemble method Stacking with Random Forest as Meta Classifier. The algorithms used in stacking are: K-Nearest Neighbor, Naïve Bayes and Random Forest. Thus the improved ensemble algorithm is named as StackedRF (Stacking with Random Forest as Meta Classifier).

4.5.1 STACKING

It is an ensemble strategy that combines many models (classification or regression) using a meta-model (meta-classifier or meta-regression). After the base models have been trained on the complete dataset, the meta-model is constructed utilizing features returned (as output) by the basis models. The underlying models in stacking are typically different. The meta-model facilitates in the finding of features from base models to achieve the best accuracy. Stacking differs from the basic assembly approaches since it uses first- and second-level models. Stacking characteristics are initially collected by training the dataset with each first-level model. A first-level model employs test stacking features to predict the result after training using train stacking features. The algorithm which is followed for stacking ensemble technique is as follows:

1) Make n sections of the train dataset.
2) In order to provide predictions for the nth part, a base model (let's say linear regression) is fitted on n-1 parts. This is carried out for each of the train set's n components.
3) The entire train dataset is then fitted with the basic model.
4) The test dataset is predicted using this model.
5) For a different base model, Steps 2 through 4 are
6) The new model is built with the predictions on the train data set as repeated, yielding a different set of predictions for the train and test dataset a feature.

7) On the test dataset, this final model is utilized to generate predictions.

Contrary to bagging and boosting, which utilized homogeneous weak learners for ensemble, stacking usually takes into consideration heterogeneous weak learners, learns them in parallel, and combines them by training a meta-learner to output a prediction based on the multiple weak learners' predictions. Using the input features from the predictions and the data's ground truth values, a meta learner tries to figure out the best method to combine the input predictions to get a better output prediction. The architecture of stacking is depicted in the diagram below:

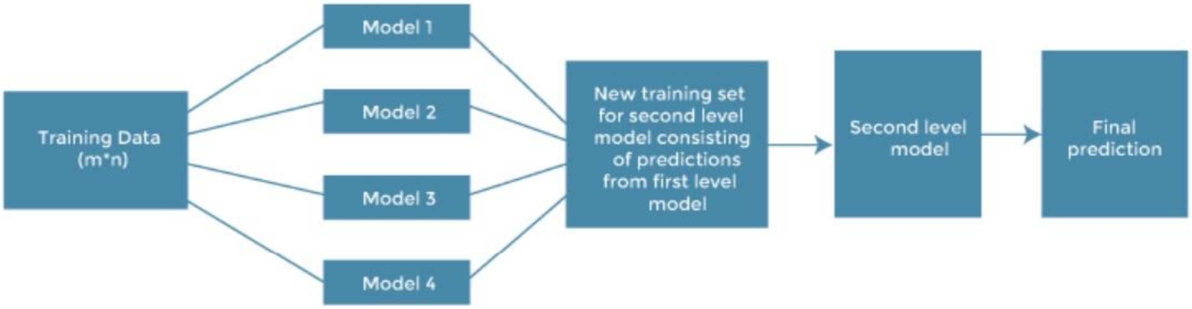

Training Data (m*n)
Model 1
Model 2
Model 3
Model 4
New training set for second level model consisting of predictions from first level model
Second level model
Final prediction

Fig. 4.9 Stacking Architecture

When utilizing an average ensemble, such as Random Forest, the model takes into account the predictions from other trained models. An issue with this method is that each model contributes the same amount to the ensemble forecast, regardless of how well the model performs. An alternate approach is to use a weighted average ensemble, which distributes the input of each ensemble member according to how confident one is in their capacity to provide the most accurate forecasts. The weighted average ensemble performs better than the model average ensemble.

By replacing the linear weighted sum with either linear regression (for regression issues) or logistic regression (for classification problems), this strategy may be further generalized by combining the predictions of the sub-models with any learning methodology. Stacking is the term for this tactic.

4.5.2 PERFORMANCE METRICS

Performance indicators assist measure how successfully a machine learning (ML) model generalizes on new or previously unexplored data. Evaluation of a machine learning model's performance is one of the key stages in its development. The efficiency or caliber of the model is evaluated using a variety of measures, often known as performance metrics or evaluation metrics. These performance metrics allow us to assess how well our model processed the given data. We can improve the model's performance by modifying the hyper-parameters. Performance indicators provide a way to assess how well a machine learning (ML) model generalizes to fresh or unstudied data.

The following four fundamental performance analysis criteria:

e) Recall

It measures the proportion of real positives that were incorrectly identified and is equivalent to the Precision metric. It may be quantified using the terms True Positive and False Negative, which refer to projections that accurately reflect the total amount of positives, whether they were correctly expected as positive or falsely anticipated as negative. The recall calculation procedure is as follows:

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

Fig. 3.9 Recall Formula

f) Precision

The precision metric prevails over the restriction of the accuracy measure. Precision is the measure of the proportion of correct positive forecasts. It may be calculated as the ratio of all genuinely accurate positive predictions (True Positive and False Positive) to all True Positives, or forecasts.

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

Fig. 3.10 Precision Formula

g) F1 Score

Based on the predictions offered for the positive class, the F-score or F1 Score measure is used to assess a binary classification model. It is calculated using Precision and Recall. A specific type of score that combines Precision and Recall is this one. As a result, the harmonic mean of accuracy and recall, with equal weights for both variables, may be used to calculate the F1 Score. The formula for calculating the F1 score is as follows:

$$F1 - score = 2 * \frac{precision * recall}{precision + recall}$$

Fig. 3.11 F1 Score Formula

h) Accuracy

The proportion of correct forecasts to all other predictions, which is calculated, is one of the simplest Classification metrics to utilize. The accuracy calculation formula is:

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ number\ of\ predictions}$$

Fig. 3.12 Accuracy Formula

4.5.3 PYTHON LIBRARIES

The libraries that come with Python are numerous. Python is a beginner-friendly programming language because of its simplicity and ease of use. In all facets of development, deployment, and maintenance, Python wants its developers to be more productive. Python's portability is another factor contributing to its huge appeal. Python's programming syntax is simple to learn and has a high degree of abstraction compared to C, Java, and C++. The libraries that we have used in our implementation are as follows:

1) Pandas

Pandas is a BSD (Berkeley Software Distribution) licensed open-source library. This well-known library is extensively used in the field of data science. They are often used to modify, clean up, and analyze data. Simple data modeling and analysis activities may be completed with Pandas without switching to another language, such as R.

2) Matplotlib

It is this library's responsibility to plot numerical data. For this reason, it is used in data analysis. An open-source library presents high-quality data in the form of diagrams, scatterplots, boxplots, and pie charts, among other things.

3) Seaborn

You may build statistical graphics with the Seaborn library in Python. It is based on Matplotlib and strongly integrated with Pandas data structures.

Using Seaborn, you may inspect and understand your data. Its charting methods do the internal semantic mapping and statistical aggregation necessary to produce usable graphs. They operate with data frames and arrays that include whole datasets. Thanks to its dataset-oriented, declarative API, you can focus on what the various elements of your plots indicate rather than the details of how to generate them.

4) Sklearn

Skearn (Skit-Learn) is the most efficient and dependable Python machine learning toolkit. It provides a number of efficient techniques for statistical modeling and machine learning, including as classification, regression, clustering, and dimensionality reduction, through a Python consistency interface. This library was mostly developed in Python and is based on NumPy, SciPy, and Matplotlib. A Python-based open-source machine learning toolkit called Scikit-learn is also available. This library supports both supervised and unsupervised learning methods. This library comes pre-installed with the SciPy, NumPy, and Matplotlib packages as well as a number of well-known algorithms. The most used Scikit-most-learn tool is Spotify's music recommendations.

5) Time

Objects, integers, and strings are just a few of the coding representations for time that are available in the Python time module. It furthermore gives you the ability to assess the efficiency of your code and wait while it runs in addition to other features outside time representation.

6) MLextend

The MLxtend library (Machine Learning extensions) contains a wealth of helpful functions for basic data analysis and machine learning tasks. Although there are several machine learning libraries for Python, like scikit-learn, TensorFlow, Keras, PyTorch, etc., MLxtend adds additional features and may be a beneficial addition to your data science toolset.

4.6 IMPLEMENTATION CODE

Student.ipynb

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib.gridspec import GridSpec
```

Load Data

```
df = pd.read_csv('dataset.csv')
print(df.shape)
df.head(5)
df['Target'].head(5)
df.columns

df.nunique()

df['Target'].head(5)
```

Data Analysis & Visualization

```
 Categorical_features = ['Marital status', 'Application mode', 'Course', 'Daytime/evening attendance',
        'Previous qualification', 'Nacionality', 'Mother\'s qualification', 'Father\'s qualification',
        'Mother\'s occupation', 'Father\'s occupation', 'Displaced', 'Educational special needs',
        'Debtor', 'Tuition fees up to date', 'Gender', 'Scholarship holder', 'International']
Numeric_features = list(set(df.columns)-set(Categorical_features)-set(['Target']))

print(len(Categorical_features))
print(Categorical_features)
print(len(Numeric_features))
print(Numeric_features)
```

```
num = []
for t in df['Target'].unique():
    num.append(df['Target'].tolist().count(t))
plt.pie(num, labels=df['Target'].unique(), autopct="%.2f%%", labeldistance=1.,
        wedgeprops = {'linewidth':1, 'edgecolor':'white'}, textprops={'color':'darkgreen', 'fontsize':10},
        colors=sns.color_palette('pastel'))
plt.title('Target')

for feature in Categorical_features:
    print(feature, ";", df[feature].unique())

fig, axes = plt.subplots(3, 6, figsize=(20,10))
plt.title("Category_feature")
for k in range(len(Categorical_features)):
  num = []
  for t in df[Categorical_features[k]].unique():
    num.append(df[Categorical_features[k]].tolist().count(t))
  axes[k//6][k%6].pie(num, labels=df[Categorical_features[k]].unique(), autopct="%.2f%%",
labeldistance=1.,
            wedgeprops = {'linewidth':1, 'edgecolor':'white'}, textprops={'color':'lightgreen', 'fontsize':10},
            colors=sns.color_palette('Blues_r'))
  axes[k//6][k%6].set_title(Categorical_features[k])
axes[-1][-1].axis('off')
plt.tight_layout()
plt.show()


fig, axes = plt.subplots(3, 6, figsize=(20,10))
plt.title("Category_features")
for k in range(len(Categorical_features)):
  sns.countplot(ax=axes[k//6][k%6], data=df, x=Categorical_features[k],
palette=sns.color_palette('pastel'))
  axes[k//6][k%6].set_title(Categorical_features[k])

axes[-1][-1].axis('off')
plt.tight_layout()
plt.show()

fig, axes = plt.subplots(3, 6, figsize=(20,10))
plt.title("Category_features")
for k in range(len(Categorical_features)):
    ax = sns.countplot(ax=axes[k//6][k%6], data=df, x=Categorical_features[k],
palette=sns.color_palette('pastel'))
    axes[k//6][k%6].set_title(Categorical_features[k])
    for label in ax.containers:
```

```
    ax.bar_label(label)
axes[-1][-1].axis('off')
plt.tight_layout()
plt.show()


fig, axes = plt.subplots(3, 6, figsize=(20,10))
for k in range(len(Categorical_features)):
   sns.countplot(ax=axes[k//6][k%6], data=df, x=Categorical_features[k], hue='Target')
   axes[k//6][k%6].set_ylabel('Number of Students')
plt.tight_layout
plt.show()

fig, axes = plt.subplots(3, 6, figsize=(20,10))
for k in range(len(Numeric_features)):
   sns.histplot(ax=axes[k//6][k%6], data=df, x=Numeric_features[k], bins=10, element="bars",
shrink=0.8, stat="percent")
   axes[k//6][k%6].set_title(Numeric_features[k])
plt.tight_layout()
plt.show()


plt.figure(figsize=(30,30))
sns.heatmap(df.corr(), cmap="Blues", annot=True)
plt.show()

fig, axes = plt.subplots(1, 2, figsize=(10,5))
sns.scatterplot(ax=axes[0], data=df, x='Curricular units 1st sem (grade)', y='Curricular units 2nd sem
(grade)', hue='Target', size="Age at enrollment", sizes=(10, 200), alpha=0.2)
axes[0].legend(loc='center left', prop={'size': 8})
markers = {'Dropout':'X', 'Graduate':'s', 'Enrolled':'*'}
sns.scatterplot(ax=axes[1], data=df, x='Curricular units 1st sem (grade)', y='Curricular units 2nd sem
(grade)', style='Target', alpha=0.2)
plt.tight_layout()
plt.show()

fig, axes = plt.subplots(1,2, figsize=(10,5))
sns.violinplot(ax=axes[0], data=df, x='Target', y='Curricular units 1st sem (grade)',
    hue='Tuition fees up to date', inner="quartile", color='c')
sns.violinplot(ax=axes[1], data=df, x='Target', y='Curricular units 2nd sem (grade)',
    hue='Tuition fees up to date')
sns.stripplot(ax=axes[1], data=df, x='Target', y='Curricular units 2nd sem (grade)',
    hue='Tuition fees up to date', size=5, marker="*", dodge=True, alpha=0.2, palette="pastel")
for i in range(2):
   axes[i].set_ylim(-5,23)
```

```
plt.show()

s_df = df[['Target', 'Curricular units 1st sem (approved)',
        'Curricular units 2nd sem (approved)']]
# change the font size of all elements in the plot.
sns.set(font_scale=0.7)
sns.pairplot(data=s_df, hue='Target')
```

Preprocessing and Feature Selection

```
 features = df.columns.drop('Target')
features


from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()
df[features] = scaler.fit_transform(df[features])

for k in features:
    df[k] = (df[k] - df[k].mean())/(df[k].max()-df[k].min())

df.head(5)


from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
df[features] = scaler.fit_transform(df[features])
df.head(50)

x = df.iloc[:, :34].values
x = StandardScaler().fit_transform(x)
x
y = df['Dropout'].values
y
```

Train & Test Splitting the data

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state = 1)
```

Function to measure performance

```
 def perform(y_pred):
    print("Precision : ", precision_score(y_test, y_pred, average = 'micro'))
    print("Recall : ", recall_score(y_test, y_pred, average = 'micro'))
    print("Accuracy : ", accuracy_score(y_test, y_pred))
    print("F1 Score : ", f1_score(y_test, y_pred, average = 'micro'))
cm = confusion_matrix(y_test, y_pred)
    print("\n", cm)
    print("\n")
    print("**"*27 + "\n" + " "* 16 + "Classification Report\n" + "**"*27)
    print(classification_report(y_test, y_pred))
    print("**"*27+"\n")

    cm = ConfusionMatrixDisplay(confusion_matrix = cm, display_labels=['Non-Dropout', 'Dropout'])
    cm.plot()
```

Gaussian Naive Bayes

```
model_nb = GaussianNB()
model_nb.fit(x_train, y_train)
y_pred_nb = model_nb.predict(x_test)
perform(y_pred_nb)
```

Logistic Regression

```
model_lr = LogisticRegression()
model_lr.fit(x_train, y_train)
y_pred_lr = model_lr.predict(x_test)
perform(y_pred_lr)
```

Random Forest

```
model_rf = RandomForestClassifier()
model_rf.fit(x_train, y_train)
y_pred_rf = model_rf.predict(x_test)
perform(y_pred_rf)
```

XGBoost Classifier

```
model_xgb = XGBClassifier()
model_xgb.fit(x_train, y_train)
y_pred_xgb = model_xgb.predict(x_test)
perform(y_pred_xgb)
```

Support Vector Classifier

```
model_svc = SVC()
model_svc.fit(x_train, y_train)
y_pred_svc = model_svc.predict(x_test)
```

```
perform(y_pred_svc)
```

Multi-Layer Perceptron

```
model_mlp = MLPClassifier()
model_mlp.fit(x_train, y_train)
y_pred_mlp = model_mlp.predict(x_test)
perform(y_pred_mlp)
```

Precision Recall Curve

```
fig, ax = plt.subplots()
plt.title("Precision-Recall Curve")
PrecisionRecallDisplay.from_predictions(y_test, y_pred_nb, ax = ax, name = "GNB", color='orange')
PrecisionRecallDisplay.from_predictions(y_test, y_pred_lr, ax = ax, name = "LR", color='blue')
PrecisionRecallDisplay.from_predictions(y_test, y_pred_rf, ax = ax, name = "RF", color='cyan')
PrecisionRecallDisplay.from_predictions(y_test, y_pred_xgb, ax = ax, name = "XGB", color='lime')
PrecisionRecallDisplay.from_predictions(y_test, y_pred_svc, ax = ax, name = "SVC", color='black')
PrecisionRecallDisplay.from_predictions(y_test, y_pred_mlp, ax = ax, name = "MLP", color='red')
```

ROC Curve

```
fig, ax = plt.subplots()
plt.title("ROC Curve")
RocCurveDisplay.from_predictions(y_test, y_pred_nb, ax = ax, name = "GNB", color='orange')
RocCurveDisplay.from_predictions(y_test, y_pred_lr, ax = ax, name = "LR", color='blue')
RocCurveDisplay.from_predictions(y_test, y_pred_rf, ax = ax, name = "RF", color='cyan')
RocCurveDisplay.from_predictions(y_test, y_pred_xgb, ax = ax, name = "XGB", color='lime')
RocCurveDisplay.from_predictions(y_test, y_pred_svc, ax = ax, name = "SVC", color='black')
RocCurveDisplay.from_predictions(y_test, y_pred_mlp, ax = ax, name = "MLP", color='red')
```

StackedRF

```
import six
import sys
sys.modules['sklearn.externals.six'] = six

from mlxtend.classifier import StackingClassifier
from sklearn.neighbors import KNeighborsClassifier
clf_stack = StackingClassifier(classifiers =[KNeighborsClassifier(), GaussianNB(), model_rf, model_
mlp, model_lr], meta_classifier =model_rf, use_probas = True, use_features_in_secondary = True)

model_stack = clf_stack.fit(x_train, y_train)   # training of stacked model
pred_stack = model_stack.predict(x_test)

acc_stack = accuracy_score(y_test, pred_stack)  # evaluating accuracy
recall_st=recall_score(y_test, pred_stack, average='macro')
precision_st = precision_score(y_test, pred_stack, average='macro')
f1_st = f1_score(y_test, pred_stack, average='macro')
print('accuracy score of Stacked model:', acc_stack * 100)
print('recall score of Stacked model:', recall_st)
print('precision score of Stacked model:', precision_st)
print('f1 score of Stacked model:', f1_st)
```

CHAPTER-5

RESULTS & DISSCUSION

5.1 OVERVIEW

The performance validation of several models included in the thesis is examined in this chapter. The definitions of the set of performance metrics utilized in the research have been presented at the outset. Additionally, a thorough analysis of the outcomes of the three suggested models is provided, along with findings of a clear visualization. The dataset used, implementation setup, obtained findings, and discussion is all included in each experimental validation of the proposed technique part.

5.2 PERFORMANCE METRICS

Below is a representation of a set of performance measures used to assess how well the suggested models performed. As shown below, accuracy metric is primarily used to determine the overall classifier result of the prediction process:

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

Recall is determined by the proportion of TP to the sum of TP and FN.

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

The proportion of TP to the sum of TP and FP is the measure of precision.

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

Through the harmonic mean, F-measure is typically used to combine accuracy and recall:

$$F - \text{measure} = \frac{2TP}{2TP + FP + FN}$$

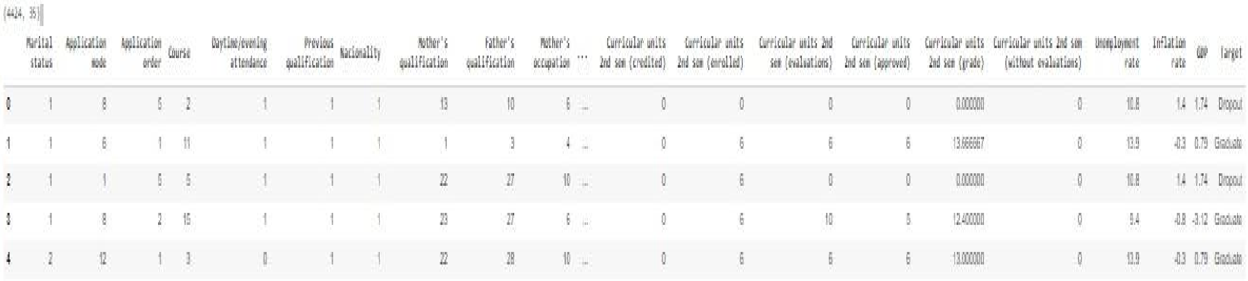
5.3 PERFORMANCE EVALUATION OF THE ENSEMBLE MODEL (StackedRF)

5.3.1 IMPLEMENTATION DATA

The presented ensemble method (StackedRF) is simulated on the basis of charts, on Google Colab using Python (Version 3.6.9). The dataset used is from Kaggle.com, it includes the details which help to predict student's performance on the basis of academic success or dropout rates. The dataset contains 35 attributes (including the Target) and 4425 student's details. To study the effects of these variables on student dropout and academic achievement, this dataset includes information from a higher education institution on a variety of undergraduate students' characteristics, social-economic determinants, and academic performance.

5.3.2 PREDICTIVE RESULTS ANALYSIS OF STUDENTS

The sample training dataset contains information on 05 students, including whether they graduated / enrolled / drop out. The following table displays some student’s data:



```
(4424, 35)
```

	Marital status	Application mode	Application order	Course	Daytime/evening attendance	Previous qualification	Nacionality	Mother's qualification	Father's qualification	Mother's occupation	...	Curricular units 2nd sem (credited)	Curricular units 2nd sem (enrolled)	Curricular units 2nd sem (evaluations)	Curricular units 2nd sem (approved)	Curricular units 2nd sem (grade)	Curricular units 2nd sem (without evaluations)	Unemployment rate	Inflation rate	GDP	Target
0	1	8	5	2	1	1	1	13	10	6	...	0	0	0	0	0.000000	0	10.8	1.4	1.74	Dropout
1	1	6	1	11	1	1	1	1	3	4	...	0	6	6	6	13.666667	0	13.9	-0.3	0.79	Graduate
2	1	1	5	5	1	1	1	22	27	10	...	0	6	0	0	0.000000	0	10.8	1.4	1.74	Dropout
3	1	8	2	15	1	1	1	23	27	6	...	0	6	10	5	12.400000	0	9.4	-0.8	-3.12	Graduate
4	2	12	1	3	0	1	1	22	28	10	...	0	6	6	6	13.000000	0	13.9	-0.3	0.79	Graduate

Fig. 5.1 Sample Visualization of the Training Dataset

The ensemble model StackedRF predicts the academic success or dropout rates of the students on the basis of students’ social-economic determinants and academic performance. The results produced by the StackedRF model on the training dataset employed, comprising of a test dataset of 885 students. The experiments done gave the result as the dataset contains 49.93% graduated students, 17.95 enrolled students and 32.12 drop out students. The following pie chart shows the Target in the form of classification into Graduated, enrolled and drop outs. The below chart shows the students who have either graduated or enrolled or dropped out.

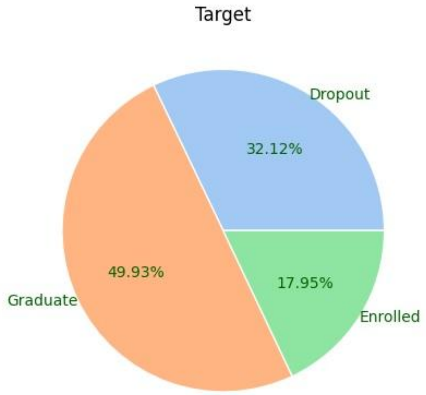


Fig. 5.2 Pie Chart Representing Target Feature

5.4 DATA ANALYSIS AND VISUALIZATIONS

5.4.1 CATEGORICAL FEATURES

The dataset contains two types of features: categorical and numerical. Statistical information made up of categorical variables—data that have been divided into categories—is known as categorical data. A set of grouped data is one of the examples. More specifically, countable qualitative data or quantitative data clustered within predetermined intervals might be used to create category data. The information is condensed into a probability table. The below figure shows the details of categorical features in the form of pie charts:

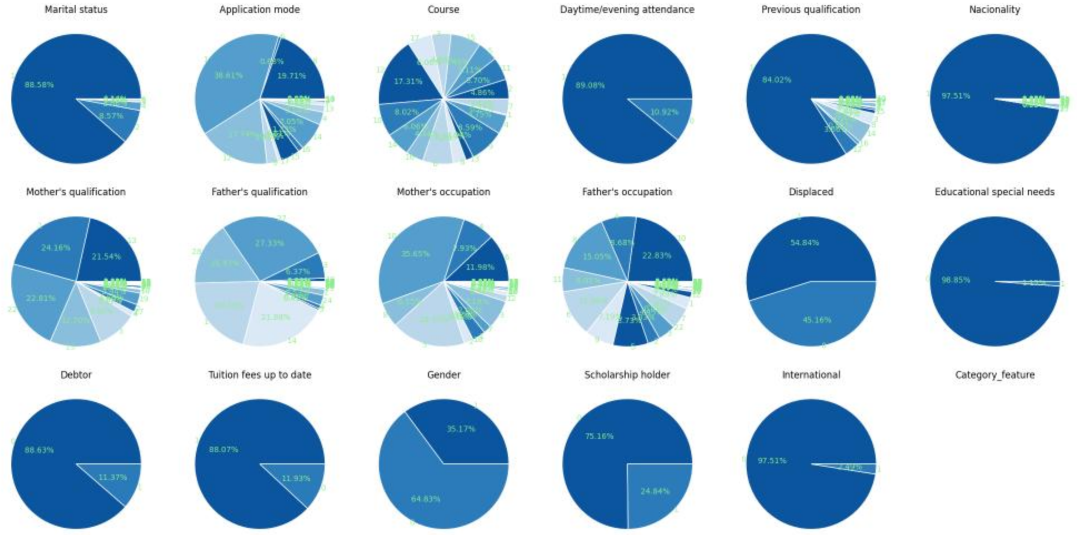


Fig. 5.3 Categorical Features Details in Pie Charts

These pie charts show the data in the form of percentages for each categorical feature. Now, the same categorical features are described on the basis of counts of the data using countplot().Use bars to visually represent the numbers of observations in each category bin. A count plot resembles a histogram over a categorical variable as opposed to a quantitative one. You can compare counts across nested variables since the fundamental API and settings are the same as those for barplot(). The below figure shows the count plot of each categorical feature:

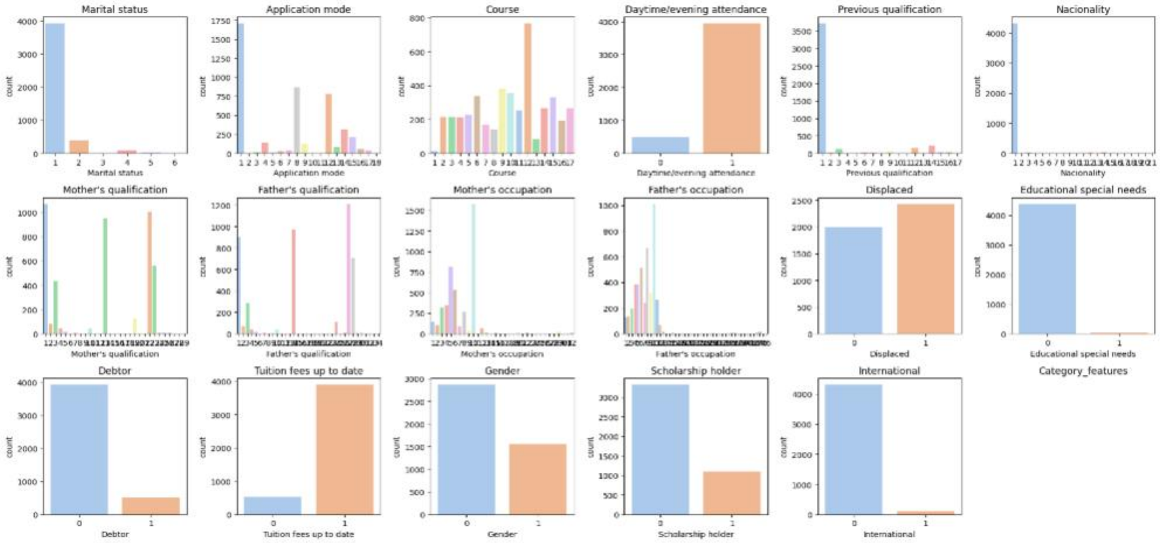


Fig. 5.4 Count Plot of each Categorical Feature

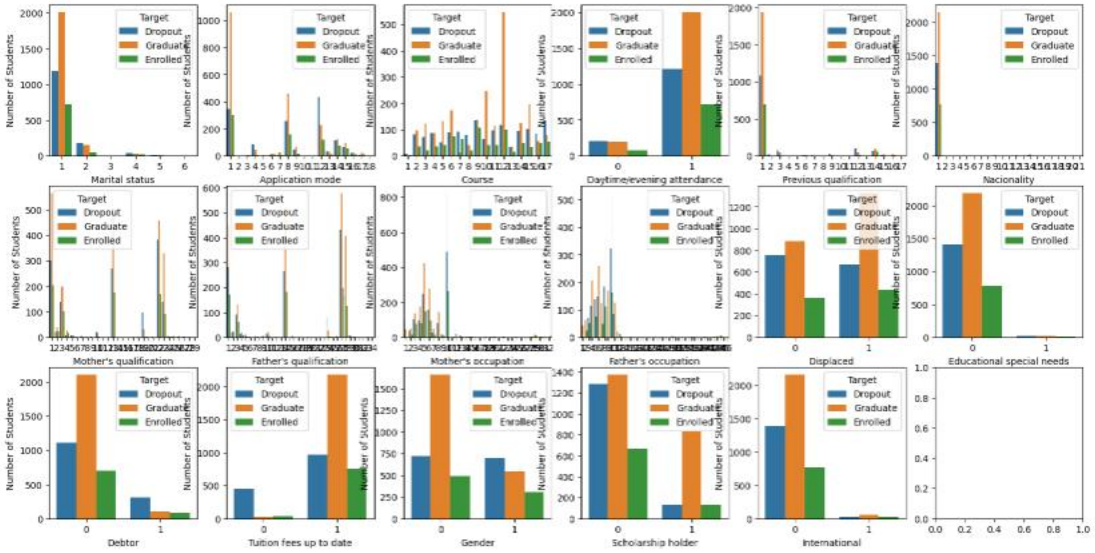


Fig. 5.5 Number of Students (Graduated/ Enrolled/ Dropped out) in Each Categorical Feature

As for each categorical data, there is some number of students who have got either graduated or enrolled or drop out. So, the above figure 5.5 shows the number of students graduated/ enrolled/ drops out on the basis of each and every categorical feature.

5.4.2 NUMERICAL FEATURES

In addition to categorical features, we also have some numerical features. Continuous values that may be measured on a scale are known as numerical characteristics. The following are some examples of numerical features: age, height, weight, and income.

Machine learning algorithms can use numerical features directly. A feature vector may easily express a numeric feature. Utilizing a linear predictor function (connected to the perceptron) with a feature vector as input is one method for achieving binary classification. By computing the scalar product of the feature vector and a vector of weights, the approach selects observations whose result is greater than a predetermined threshold. The below figure 5.6 displays the histogram of each numeric feature:

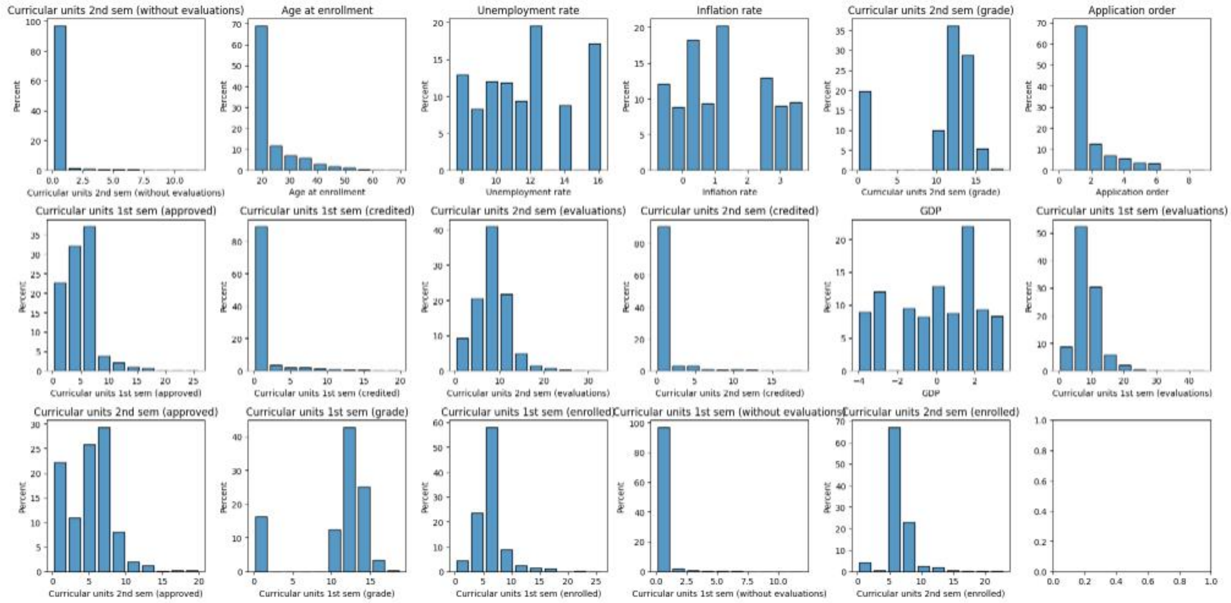


Fig. 5.6 Histogram of each Numeric Feature

Now, the heatmap is plotted in the below figure 5.7. A heatmap is a graphical representation of data that uses color to show the matrix's value. In this, brighter, mostly reddish colours are utilized to symbolize more common values or higher levels of activity, whereas darker colours are used to represent less common or activity values. The term "heatmap" is also used to describe the shading matrix. The seaborn.heatmap() method may be used to plot heatmaps in Seaborn. The below figure 5.7 shows the heatmap:

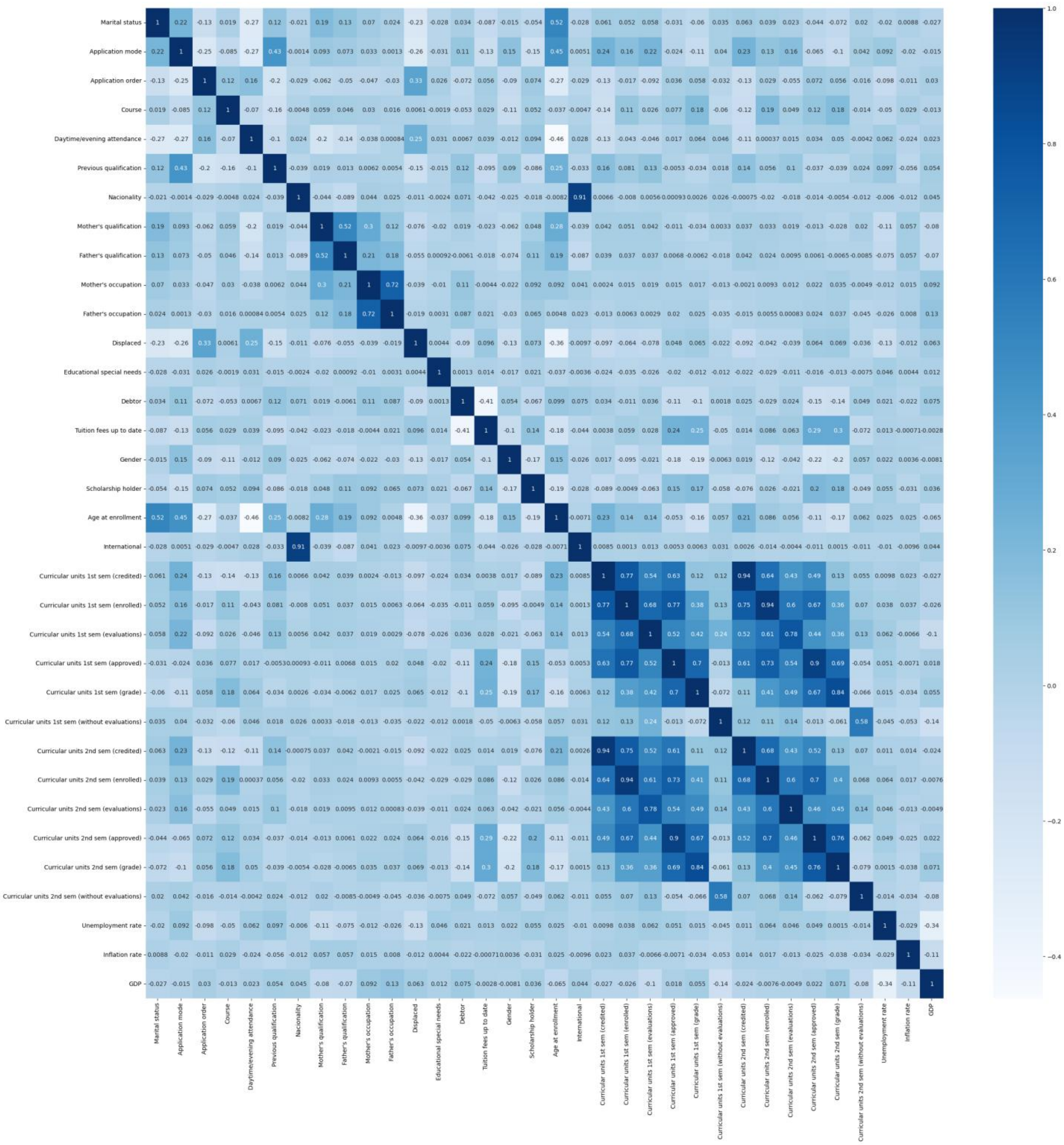


Fig. 5.7 Heatmap

There are 2 features Curricular Unit 1st Sem (Grade) and Curricular Unit 2nd Sem (Grade), which are important for the prediction of academic success or dropout rates of the students. A scatter chart, often referred to as a scatter plot, is a graph that displays the correlation between two variables. They are a very effective sort of chart because they enable viewers to see relationships or trends that would be nearly hard to observe in any other form right away. So analyzing these two features, scatter chart is plotted representing the Target. The below figure shows the Scatter plot:

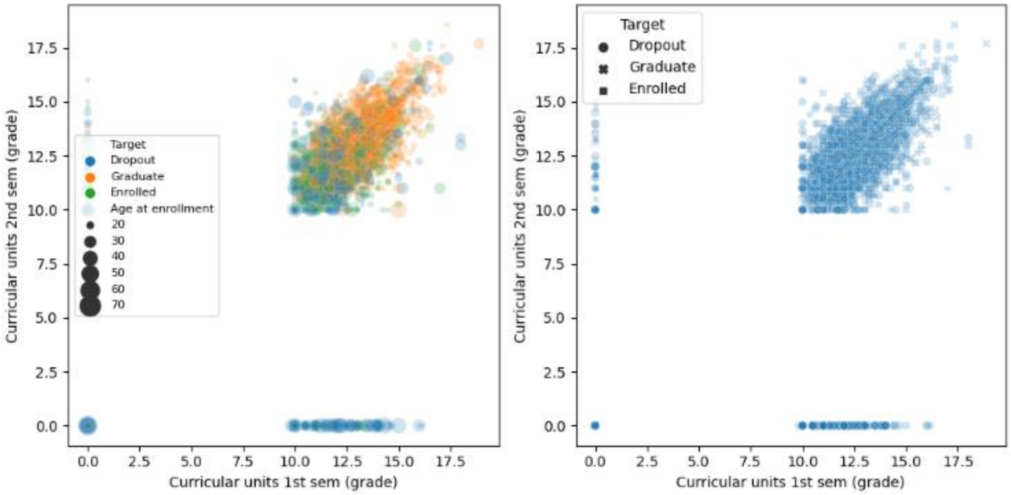


Fig. 5.8 Scatter plot for Target

By viewing the below violin plot, the Target is related to the Curricular Unit 1st Sem (Grade) and Curricular Unit 2nd Sem (Grade) and the relationship is analyzed on the basis of whether the Tution fees is up to date or not.

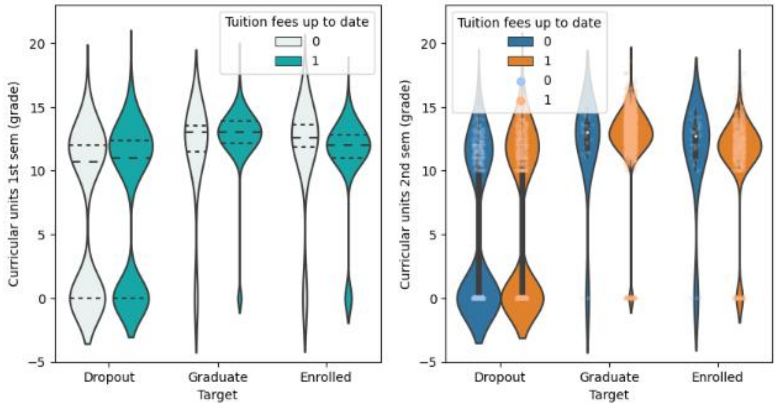


Fig. 5.9 Violin Plot

A technique for visualizing the distribution of numerical data for various variables is the violin plot. With a rotated plot on either side that provides additional details about the density estimate on the y-axis, it is comparable to a box plot. A violin-like picture is produced by mirroring and flipping the density, then filling in the resultant form. The benefit of a violin plot is that it can reveal subtleties in the distribution that a boxplot cannot. On the other hand, the boxplot makes the outliers in the data more obvious. Box plots are more common, although violin plots include more information. The above figure 5.9 shows the violin plot showing the relationship between Target with Curricular Unit 1st Sem (Grade) and Curricular Unit 2nd Sem (Grade). Now two more features are there which are also important in the prediction of academic success or dropout rates of the students. They are Curricular Unit 1st Sem (Approved) and Curricular Unit 2nd Sem (Approved). In a dataset, it depicts pairwise associations. Each numeric variable in the data will be spread over the y-axes across a single row and the x-axes across a single column by default, according to the axes grid created by this function. A univariate distribution plot is created to display the marginal distribution of the data in each column for the diagonal plots, which are handled differently. Additionally, you may display only a portion of the data or plot other variables in distinct rows and columns.

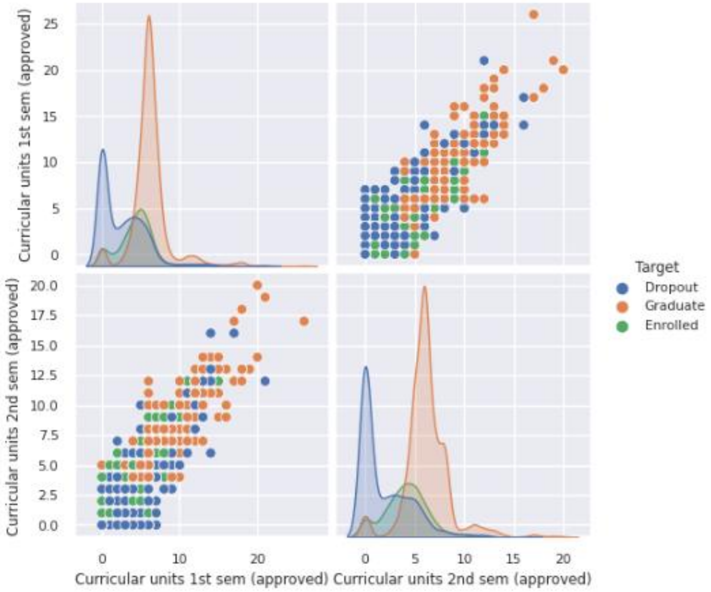


Fig. 5.10 Pair Plot

This high-level PairGrid interface is designed to make it simple to create a few popular designs. If you want greater flexibility, you should utilise PairGrid directly. The above figure 5.10 pairplot shows the relationship between Target and Curricular Unit 1st Sem (Approved) and Curricular Unit 2nd Sem (Approved).

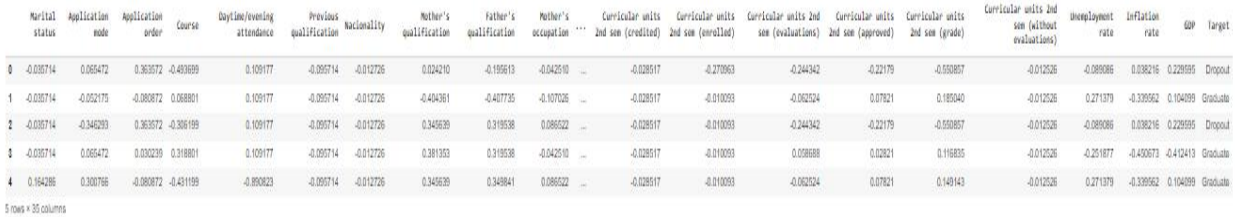
5.5 DATA PREPROCESSING

When the range of the numerical input variables is scaled to a standard, several machine learning algorithms perform better. This covers methods like linear regression that weight the input, as well as techniques like k-nearest neighbors that employ distance metrics. Prior to modeling, the two most often used methods for scaling numerical data are normalization and standardization. Each input variable is scaled individually to a range of 0–1, which is the range for floating-point values where we have the most accuracy. Standardization shifts the distribution to have a mean of zero and a standard deviation of one by scaling each input variable independently by removing the mean (a process known as centering) and dividing by the standard deviation.

Data conversion and format transformation are the two processes that pre-processing often entails. The format translation operation is mostly carried out when any type of data is present, such as when .xls files are converted into.csv files. Second, data transformation is calculated with the assumption that the dataset contains data in many formats, which are afterwards changed into accurate forms. Without converting them to numerical values, the majority of machine learning algorithms cannot handle categorical variables. The encoding of categorical variables affects how well many algorithms work.

In the given dataset, we check for any null or duplicate value by using methods like isnull() BS duplicated(). The isnull() methods shows the columns having null values and duplicated().sum() shows the sum of duplicated values for the column. Here in this data set, it looks like there is no null or duplicate value in any column. Only “Target” column has non-numeric value in the dataset, and this “Target” column is the output column so we need to convert it to numeric value so that we can easily find it’s correlation with other columns. There are three unique values for Target column, which we replace as Dropout->0, Enrolled->1 and Graduate->2.

As shown in the below figure 5.11 preprocessing is done using MinMaxScaler from Sklearn, By scaling each feature to a predetermined range, features are transformed. This estimator scales and translates each feature separately so that it falls inside the training set's predefined range, such as between zero and one.



	Marital status	Application mode	Application order	Course	Daytime/evening attendance	Previous qualification	Nacionality	Mother's qualification	Father's qualification	Mother's occupation	...	Curricular units 2nd sem (credited)	Curricular units 2nd sem (enrolled)	Curricular units 2nd sem (evaluations)	Curricular units 2nd sem (approved)	Curricular units 2nd sem (grade)	Curricular units 2nd sem (without evaluations)	Unemployment rate	Inflation rate	GDP	Target
0	-0.035714	0.065472	0.363572	-0.493699	0.109177	-0.095714	-0.012726	0.024210	-0.195613	-0.042510	...	-0.028517	-0.270963	-0.244342	-0.22179	-0.550857	-0.012526	-0.089086	0.038216	0.229595	Dropout
1	-0.035714	-0.052175	-0.080872	0.068801	0.109177	-0.095714	-0.012726	-0.404361	-0.407735	-0.107026	...	-0.028517	-0.010093	-0.062524	0.07821	0.185040	-0.012526	0.271379	-0.339562	0.104099	Graduate
2	-0.035714	-0.346293	0.363572	-0.306199	0.109177	-0.095714	-0.012726	0.345639	0.319538	0.086522	...	-0.028517	-0.010093	-0.244342	-0.22179	-0.550857	-0.012526	-0.089086	0.038216	0.229595	Dropout
3	-0.035714	0.065472	0.030239	0.318801	0.109177	-0.095714	-0.012726	0.381353	0.319538	-0.042510	...	-0.028517	-0.010093	0.058688	0.02821	0.116835	-0.012526	-0.251877	-0.450673	-0.412413	Graduate
4	0.164286	0.300766	-0.080872	-0.431199	-0.890823	-0.095714	-0.012726	0.345639	0.349841	0.086522	...	-0.028517	-0.010093	-0.062524	0.07821	0.149143	-0.012526	0.271379	-0.339562	0.104099	Graduate

5 rows × 35 columns

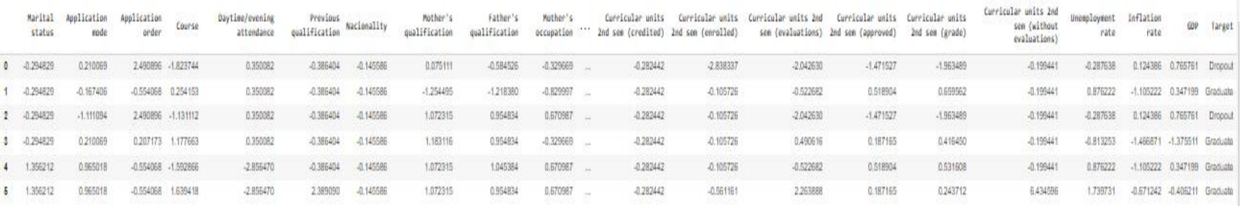
Fig. 5.11 Sample data of 5 Students after MinMaxScaler Preprocessing

After MinMaxScaler Preprocessing, the standardization of the dataset is done. Another way of scaling is standardization, which centers the numbers on the mean and uses a unit standard deviation. As a result, the attribute's mean becomes zero, and the distribution that results has a unit standard deviation. The method for standardization is as follows:

$$X' = \frac{X - \mu}{\sigma}$$

Fig. 5.12 Standardization Formula

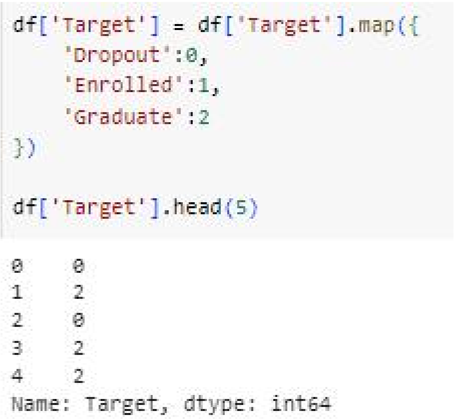
μ represents the mean of the values of feature and σ represents the standard deviation of the values of the feature. The below figure 5.13 shows the sample data of 5 students after standardization:



	Marital status	Application mode	Application order	Course	Daytime/evening attendance	Previous qualification	Nacionality	Mother's qualification	Father's qualification	Mother's occupation	...	Curricular units 2nd sem (credited)	Curricular units 2nd sem (enrolled)	Curricular units 2nd sem (evaluations)	Curricular units 2nd sem (approved)	Curricular units 2nd sem (grade)	Curricular units 2nd sem (without evaluations)	Unemployment rate	Inflation rate	GDP	Target
0	-0.294829	0.210069	2.490896	-1.823744	0.350082	-0.386404	-0.145586	0.075111	-0.584526	-0.329669	...	-0.282442	-2.838337	-2.042630	-1.471527	-1.963489	-0.199441	-0.287638	0.124386	0.765761	Dropout
1	-0.294829	-0.167406	-0.554068	0.254153	0.350082	-0.386404	-0.145586	-1.254495	-1.218380	-0.829997	...	-0.282442	-0.105726	-0.522682	0.518904	0.659562	-0.199441	0.876222	-1.105222	0.347199	Graduate
2	-0.294829	-1.111094	2.490896	-1.131112	0.350082	-0.386404	-0.145586	1.072315	0.954834	0.670987	...	-0.282442	-0.105726	-2.042630	-1.471527	-1.963489	-0.199441	-0.287638	0.124386	0.765761	Dropout
3	-0.294829	0.210069	0.207173	1.177663	0.350082	-0.386404	-0.145586	1.183116	0.954834	-0.329669	...	-0.282442	-0.105726	0.490616	0.187165	0.416450	-0.199441	-0.813253	-1.466871	-1.375511	Graduate
4	1.356212	0.965018	-0.554068	-1.592866	-2.856470	-0.386404	-0.145586	1.072315	1.045384	0.670987	...	-0.282442	-0.105726	-0.522682	0.518904	0.531608	-0.199441	0.876222	-1.105222	0.347199	Graduate
5	1.356212	0.965018	-0.554068	1.639418	-2.856470	2.389090	-0.145586	1.072315	0.954834	0.670987	...	-0.282442	-0.561161	2.263888	0.187165	0.243712	6.434596	1.739731	-0.671242	-0.406211	Graduate

Fig. 5.13 Sample data of 5 Students after Standardization

Using techniques like isnull() BS duplicated(), we search the provided dataset for any null or duplicate values. The duplicated() and isnull() methods display the columns with null values.The total of the duplicate values for the column is displayed via sum(). It appears that no column in this data collection has a null value or a duplicate value. Only the "Target" column in the dataset has a non-numeric value; since it is the output column, we must convert it to a numeric value so that we can quickly determine how it correlates with other columns. Target column has three distinct values that we change with Dropout->0, Enrolled->1 and Graduate->2. The below figure 5.14 shows the transformation of Target from categorical to numerical values:



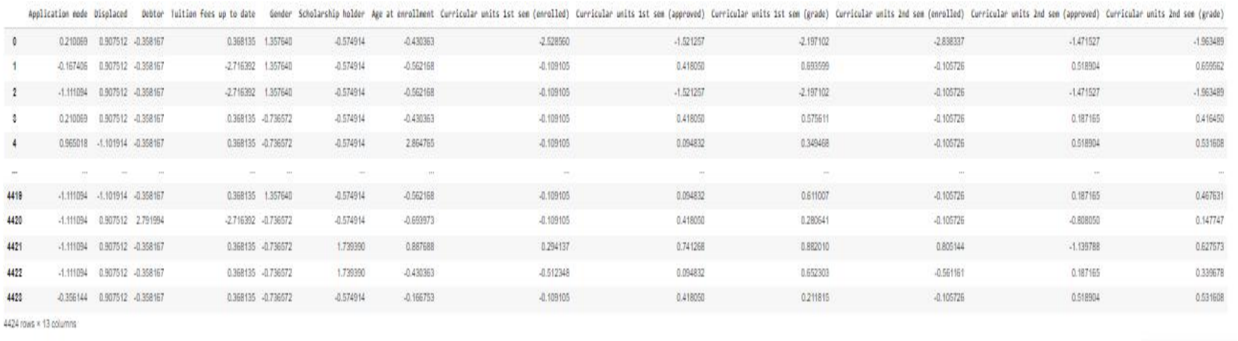
```
df['Target'] = df['Target'].map({
    'Dropout':0,
    'Enrolled':1,
    'Graduate':2
})

df['Target'].head(5)
```

```
0    0
1    2
2    0
3    2
4    2
Name: Target, dtype: int64
```

Fig. 5.14 Transformation of Target

All the features are important in some or the other way, but some features which do not contribute much in the prediction are excluded from the feature set.. So, the selected features are only used to predict the academic success or dropout rates on the basis of academic performance of the students. The below figure 5.15 shows the sample data of students on the basis of selected features:



	Application mode	Displaced	Debtor	Tuition fees up to date	Gender	Scholarship holder	Age at enrollment	Curricular units 1st sem (enrolled)	Curricular units 1st sem (approved)	Curricular units 1st sem (grade)	Curricular units 2nd sem (enrolled)	Curricular units 2nd sem (approved)	Curricular units 2nd sem (grade)
0	0.210069	0.907512	-0.358167	0.368135	1.357640	-0.574914	-0.430363	-2.528560	-1.521257	-2.197102	-2.838337	-1.471527	-1.963489
1	-0.167406	0.907512	-0.358167	-2.716392	1.357640	-0.574914	-0.562168	-0.109105	0.418050	0.693599	-0.105726	0.518904	0.659562
2	-1.111094	0.907512	-0.358167	-2.716392	1.357640	-0.574914	-0.562168	-0.109105	-1.521257	-2.197102	-0.105726	-1.471527	-1.963489
3	0.210069	0.907512	-0.358167	0.368135	-0.736572	-0.574914	-0.430363	-0.109105	0.418050	0.575611	-0.105726	0.187165	0.416450
4	0.965018	-1.101914	-0.358167	0.368135	-0.736572	-0.574914	2.864765	-0.109105	0.094832	0.349468	-0.105726	0.518904	0.531608
...	...	...	...	...	...	...	...	...	...	...	...	...	...
4419	-1.111094	-1.101914	-0.358167	0.368135	1.357640	-0.574914	-0.562168	-0.109105	0.094832	0.611007	-0.105726	0.187165	0.467631
4420	-1.111094	0.907512	2.791994	-2.716392	-0.736572	-0.574914	-0.693973	-0.109105	0.418050	0.280641	-0.105726	-0.808050	0.147747
4421	-1.111094	0.907512	-0.358167	0.368135	-0.736572	1.739390	0.887688	0.294137	0.741268	0.882010	0.805144	-1.139788	0.627573
4422	-1.111094	0.907512	-0.358167	0.368135	-0.736572	1.739390	-0.430363	-0.512348	0.094832	0.652303	-0.561161	0.187165	0.339678
4423	-0.356144	0.907512	-0.358167	0.368135	-0.736572	-0.574914	-0.166753	-0.109105	0.418050	0.211815	-0.105726	0.518904	0.531608

4424 rows × 13 columns

Fig. 5.15 Sample data after Selecting Features

After selecting the features, this research continues with the comparative analysis of the machine learning algorithms. Programs that use machine learning algorithms are able to discover hidden patterns in data, forecast results, and enhance performance based on past performance.

5.6 COMPARATIVE ANALYSIS OF ALGORITHMS

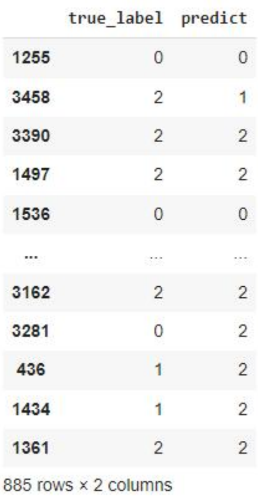
In machine learning, several algorithms may be employed for various tasks, such as basic linear regression for prediction issues like stock market forecasting and the KNN algorithm for categorization issues.

In this research, we have performed comparative analysis of a number of machine learning algorithms. This comparative analysis helps us to identify which machine learning algorithms is working best on our dataset. We choose that algorithm for further improvement of the performance metrics.

Model selection is the process of selecting one machine learning model from a selection of candidates for a training dataset. Model selection is a technique that may be used to compare models of different types (such as logistic regression, SVM, KNN, etc.) as well as models of the same type that have been set with different model hyper parameters (for example, different kernels in an SVM). It is possible to forecast the best sort of model that may be used to address a specific problem. The objective is to select a model that meets our goals and other criteria, such as performance, robustness, complexity, etc., rather than searching for the best model.

5.6.1 PERFORMANCE ANALYSIS OF ALGORITHMS

This section describes the performance analysis of the machine learning algorithms. The research involves the given machine learning algorithms for comparison such as Naïve Bayes, Logistic Regression, Random Forest, XGBoost Classifier and Support Vector Machine. The above mentioned algorithms were compared on the basis of their performance in terms of accuracy, precision, recall and f-score. These performance indicators assist measure how successfully a machine learning (ML) model generalizes on new or previously unexplored data. The performance is measured on the basis of true value and the predicted value and relationship between these values gives the result in the form of accuracy, precision, recall and f-score. The following figure shows the true and predicted value for KNN:



	true_label	predict
1255	0	0
3458	2	1
3390	2	2
1497	2	2
1536	0	0
...	...	...
3162	2	2
3281	0	2
436	1	2
1434	1	2
1361	2	2

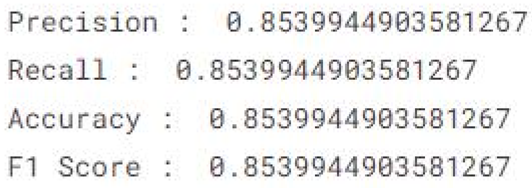
885 rows × 2 columns

Fig. 5.16 KNN Sample Predicted and True Values

The below figure indicates the comparison table including accuracy, precision, recall and f-score of the above mentioned machine learning algorithms. These values show what the performance of the algorithms is individually. Out of these 5 algorithms, we choose the algorithm which gives the best performance.

1) **Gaussian Naïve Bayes**:

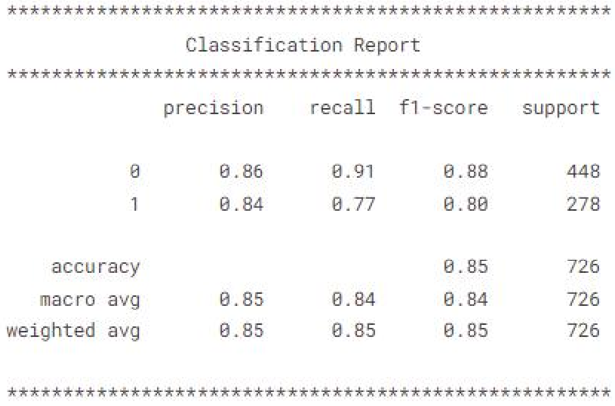
   The Naïve Bayes algorithm gives the following performance in terms of accuracy, Precision, Recall and F1 score. Ther below figure shows the values of the performance parameters for Naïve Bayes algorithm:



```
Precision :  0.8539944903581267
Recall :  0.8539944903581267
Accuracy :  0.8539944903581267
F1 Score :  0.8539944903581267
```

Fig. 5.17 Naïve Bayes Performance Evaluation

In this section, we also discuss the classification report and confusion matrix of the student dataset for academic success predicted by Naïve Bayes algorithm. The below figure shows the classification report and the confusion matrix of Naïve Bayes:



```
*****************************************************
              Classification Report
*****************************************************
              precision    recall  f1-score   support

           0       0.86      0.91      0.88       448
           1       0.84      0.77      0.80       278

    accuracy                           0.85       726
   macro avg       0.85      0.84      0.84       726
weighted avg       0.85      0.85      0.85       726

*****************************************************
```

Fig. 5.18 Naïve Bayes Classification Report

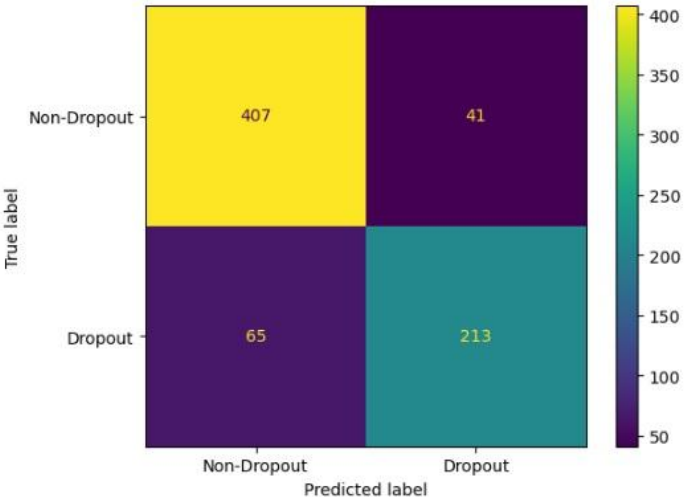
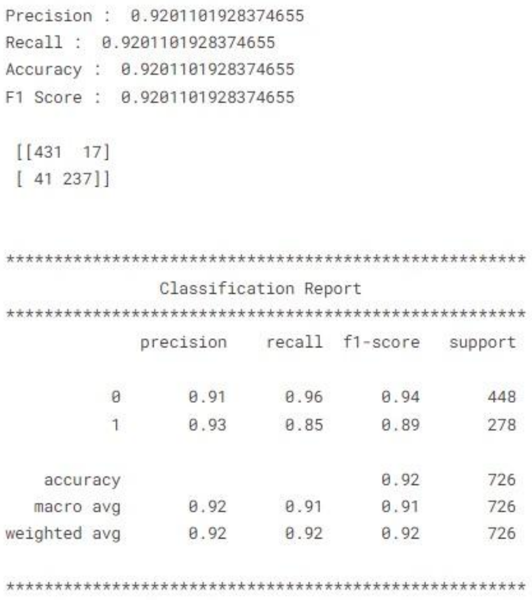


Fig. 5.19 Naïve Bayes Confusion Matrix

2) **Logistic Regression**:

The second algorithm we are working on is Logistic Regression. This algorithm gave a result as accuracy of 92%, precision of 92.01 %, recall value 92.01% and f1-score value 92.01%. The below figure shows the performance parameters and classification report of Logistic Regression on the given training dataset:



```
Precision :  0.9201101928374655
Recall :  0.9201101928374655
Accuracy :  0.9201101928374655
F1 Score :  0.9201101928374655

 [[431  17]
 [ 41 237]]


******************************************************
               Classification Report
******************************************************
              precision    recall  f1-score   support

           0       0.91      0.96      0.94       448
           1       0.93      0.85      0.89       278

    accuracy                           0.92       726
   macro avg       0.92      0.91      0.91       726
weighted avg       0.92      0.92      0.92       726

******************************************************
```

Fig. 5.20 Logistic Regression Classification Report

The below figure shows the confusion matrix for Logistic Regression algorithm prediction of the student academic performance dataset.

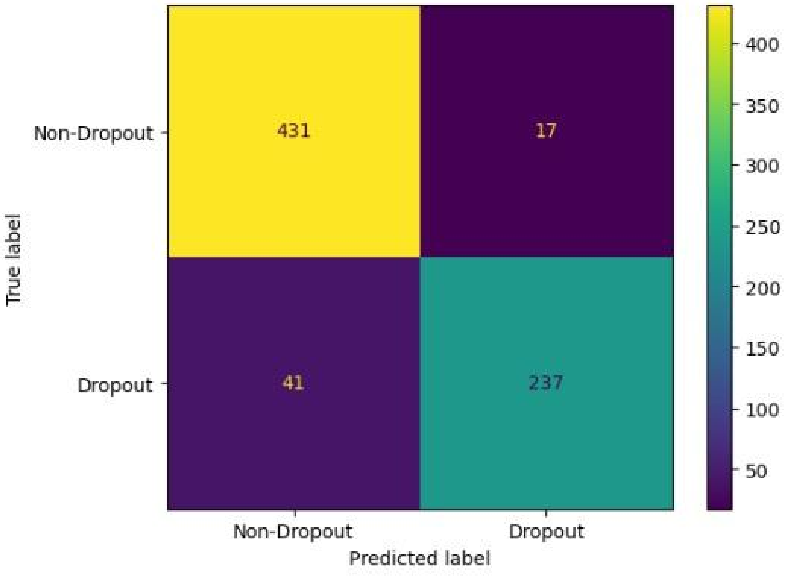
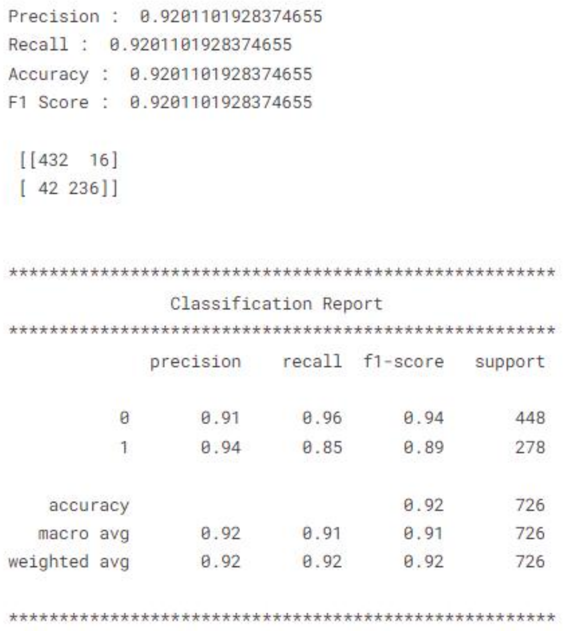


Fig. 5.21 Logistic Regression Confusion Matrix

3) Random Forest:

The next algorithm we preferred for comparison is Random Forest Classifier. This Random Forest is a collection of several decision trees working in parallel. The below figure shows the performance parameters of Random Forest algorithm to predict student’s performance and dropout rates.



```
Precision :  0.9201101928374655
Recall :  0.9201101928374655
Accuracy :  0.9201101928374655
F1 Score :  0.9201101928374655

 [[432  16]
 [ 42 236]]


*****************************************************
                Classification Report
*****************************************************
              precision    recall  f1-score   support

           0       0.91      0.96      0.94       448
           1       0.94      0.85      0.89       278

    accuracy                           0.92       726
   macro avg       0.92      0.91      0.91       726
weighted avg       0.92      0.92      0.92       726

*****************************************************
```

Fig. 5.22 Random Forest Classification report

The below figure shows the confusion matrix for the Random Forest algorithm:

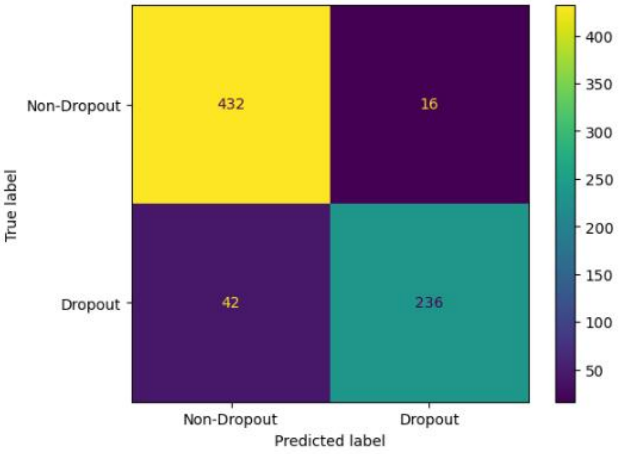
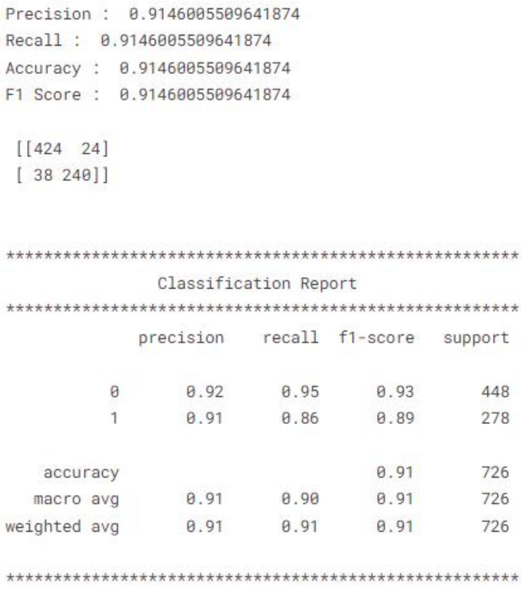


Fig. 5.23 Random Forest Confusion Matrix

4) XGBoost Classifier:

XGBoost Clasifier gave the following result for prediction of student academic success or dropout rates. These parameters show the performance of the XGBoost Classifier:



```
Precision :  0.9146005509641874
Recall :  0.9146005509641874
Accuracy :  0.9146005509641874
F1 Score :  0.9146005509641874

 [[424  24]
 [ 38 240]]


*****************************************************
                Classification Report
*****************************************************
              precision    recall  f1-score   support

           0       0.92      0.95      0.93       448
           1       0.91      0.86      0.89       278

    accuracy                           0.91       726
   macro avg       0.91      0.90      0.91       726
weighted avg       0.91      0.91      0.91       726

*****************************************************
```

Fig. 5.24 XGBoost Classifier Classification Report

The below figure shows the confusion matrix for the XGBoost algorithm:

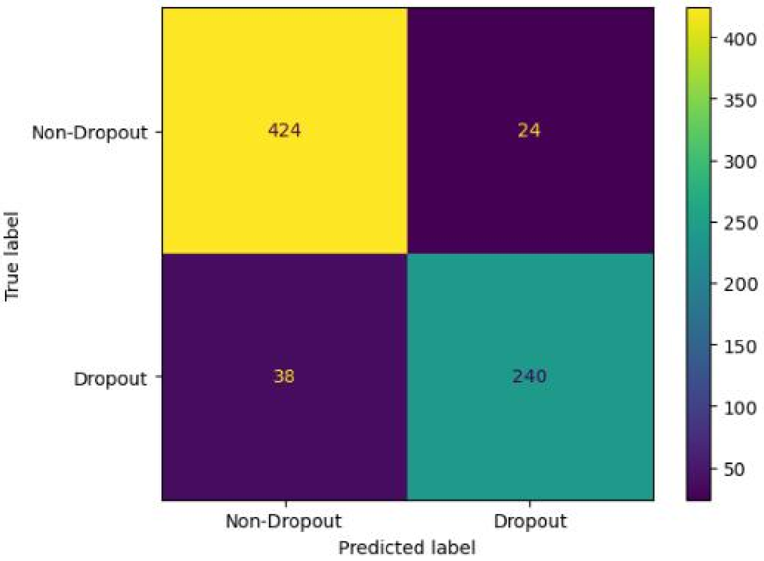
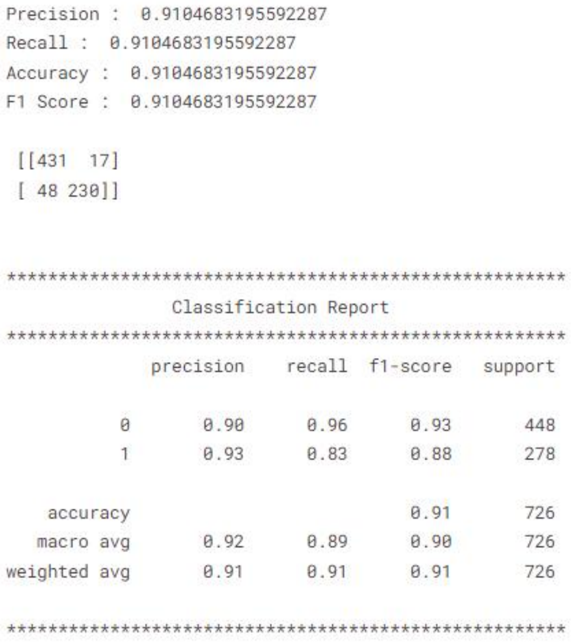


Fig. 5.25 XGBoost Confusion Matrix

5) Support Vector Machine:

The next algorithm we preferred for comparison is Support Vector Machine Classifier. This Support Vector Machine is a collection of several decision trees working in parallel. The below figure shows the performance parameters of Support Vector Machine algorithm to predict student's performance and dropout rates.



```
Precision :  0.9104683195592287
Recall :  0.9104683195592287
Accuracy :  0.9104683195592287
F1 Score :  0.9104683195592287

 [[431  17]
 [ 48 230]]


******************************************************
                Classification Report
******************************************************
              precision    recall  f1-score   support

           0       0.90      0.96      0.93       448
           1       0.93      0.83      0.88       278

    accuracy                           0.91       726
   macro avg       0.92      0.89      0.90       726
weighted avg       0.91      0.91      0.91       726

******************************************************
```

Fig. 5.26 Support Vector Machine Classification Report

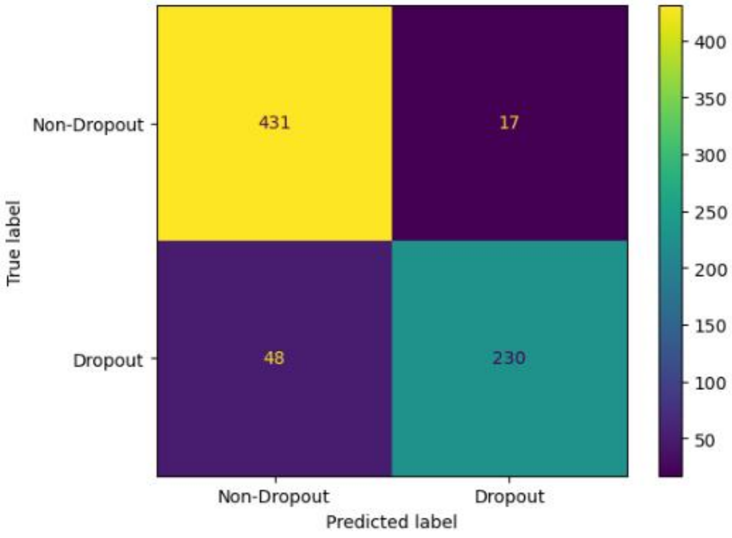
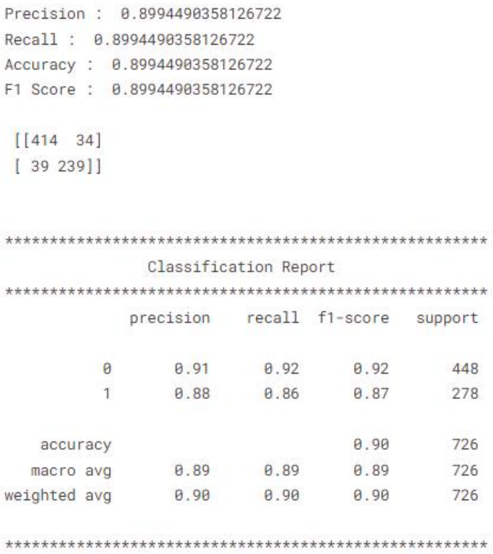


Fig. 5.27 Support Vector Machine Confusion Matrix

6) Multi Layer Perceptron

The last algorithm we preferred for comparison is Multi Layer Perceptron. This Multi Layer Perceptron is a collection of several decision trees working in parallel. The below figure shows the performance parameters of Multi Layer Perceptron algorithm to predict student's performance and dropout rates.



```
Precision :  0.8994490358126722
Recall :  0.8994490358126722
Accuracy :  0.8994490358126722
F1 Score :  0.8994490358126722

 [[414  34]
 [ 39 239]]


******************************************************
                Classification Report
******************************************************
              precision    recall  f1-score   support

           0       0.91      0.92      0.92       448
           1       0.88      0.86      0.87       278

    accuracy                           0.90       726
   macro avg       0.89      0.89      0.89       726
weighted avg       0.90      0.90      0.90       726

******************************************************
```

Fig. 5.28 Multi Layer Perceptron Classification report

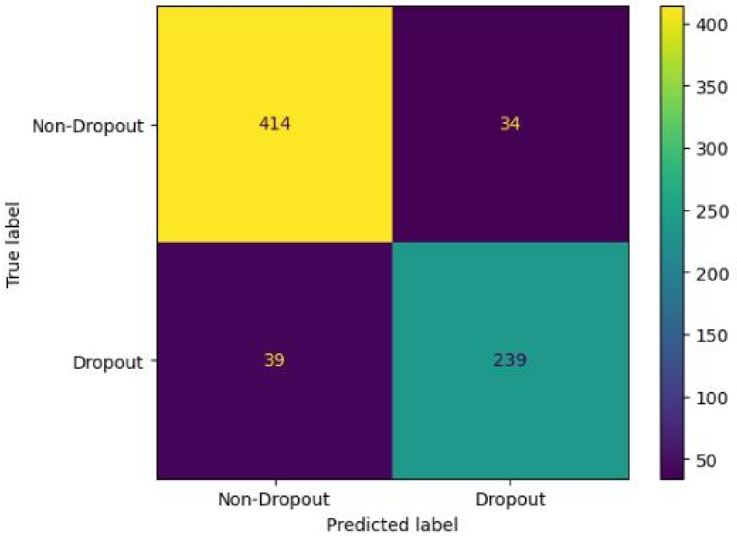


Fig. 5.29 Multi Layer Perceptron confusion matrix

The feed forward neural network is supplemented by the multi layer perceptron (MLP). The input layer, output layer, and hidden layer are the three different kinds of layers that make it up. The input layer is where the input signal for processing is received.

5.6.2 COMPARATIVE PERFORMANCE EVALUATION

Now, as shown in the above below, the accuracy of Naïve Bayes is 85.39%, Logistic Regression is 92.01%, Random Forest Classifier is 92.01, XGBoost Classifier is 91.4, Multi Layer Perceptron is 89.94% and Support Vector Machine is 91.04.

The tradeoff between accuracy and recall for various thresholds is depicted by the precision-recall curve. High accuracy is correlated with a low false positive rate, while high recall is correlated with a low false negative rate. A big area under the curve denotes both high recall and high precision. The below figure shows the Precision-Recall Curve:

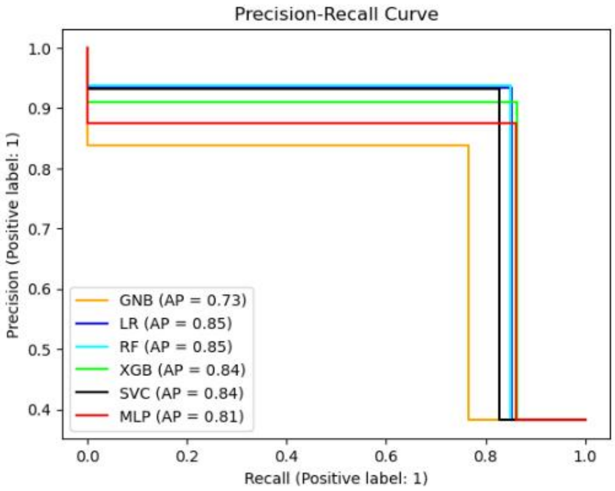


Fig. 5.30 Precision-Recall Curve

In machine learning, creating an ML model alone is insufficient since we also need to assess how effectively it is working. It means that after creating an ML model, we must assess and verify its quality. In these situations, we employ several Evaluation Metrics. Such an assessment metric is the AUC-ROC curve, which is used to show the effectiveness of a classification model. It is a well-liked and significant indicator for assessing the effectiveness of the classification model. An evaluation statistic for a classification model's performance at various threshold levels is the AUC-ROC curve. The Receiver Operating Characteristic (ROC) curve is a probability graph that displays how well a classification model performs at various threshold levels. The below figure shows the ROC Curve:

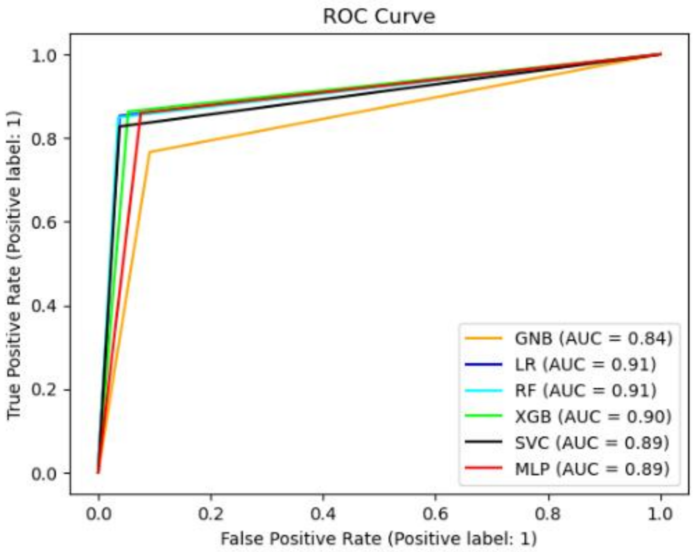


Fig. 5.31 ROC Curve

As the results show that Random Forest Classifier performs best out of the 6 machine learning algorithms. Thus, we choose Random Forest Classifier with an accuracy of 92.01% for ensemble modeling. The ensemble model is explained in the next section.

5.7 ENSEMBLE MODEL (StackedRF)

Ensemble modeling is the process of creating multiple diverse models to predict a result using a range of modeling approaches or training data sets. The ensemble model then incorporates the forecast from each base model to create a final, unified prediction for the unobserved data. Using ensemble models aims to reduce the generalization error of the forecast. When the base models are diverse and independent, the ensemble approach lowers the model's prediction error. The approach uses people's collective wisdom to generate a forecast. Even though it has a large number of base models, the ensemble model acts and performs as a single model.

After comparative analysis of several techniques: Naïve Bayes, Logistic Regression, Random Forest, XGBoost Classifier, Mutli Layer Perceptron and Support Vector Machine, we choose the algorithm which gives best performance among all. The performance is measured on the basis of accuracy, precision, recall and f1_score.

Here in this section, we will discuss about the proposed technique. The algorithm which performs best out of the algorithms mentioned above is Random Forest with a good accuracy score of 92.01%. Thus to improve this algorithm we use ensemble method Stacking with Random Forest as Meta Classifier. The algorithms used in stacking are: K-Nearest Neighbor, Naïve Bayes and Random Forest. Thus the improved ensemble algorithm is named as StackedRF (Stacking with Random Forest as Meta Classifier).

The algorithm which is followed for stacking ensemble technique is as follows:

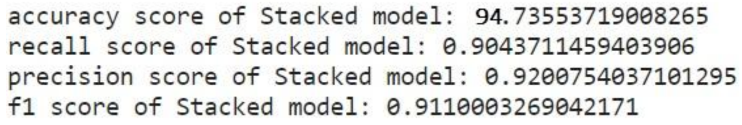
8) Make n sections of the train dataset.
9) In order to provide predictions for the nth part, a base model (let's say linear regression) is fitted on n-1 parts. This is carried out for each of the train set's n components.
10) The entire train dataset is then fitted with the basic model.
11) The test dataset is predicted using this model.
12) For a different base model, Steps 2 through 4 are
13) The new model is built with the predictions on the train data set as repeated, yielding a different set of predictions for the train and test dataset a feature.
14) On the test dataset, this final model is utilized to generate predictions.

5.7.1 PERFORMANCE ANALYSIS OF StackedRF

The model incorporates the forecasts from other trained models when using an average ensemble, such as Random Forest. The fact that each model contributes the same amount to the ensemble forecast, regardless of how well the model performs, is a drawback of this technique. A weighted average ensemble is an alternative strategy that weights each ensemble member's input according to the confidence in their ability to produce the best forecasts. The model average ensemble is outperformed by the weighted average ensemble.

This method may be further generalized by combining the predictions of the sub-models with any learning technique by substituting the linear weighted sum with either linear regression (for regression problems) or logistic regression (for classification problems). This strategy is known as stacking.

The StackedRF model gives the following result for prediction of student's academic success and dropout rates: it gives an accuracy of 94.73%, recall score value 90.4%, precision score is 92.0% and f1 score value is 91.1%. The below figure shows the performance parameters:



```
accuracy score of Stacked model:  94.73553719008265
recall score of Stacked model: 0.9043711459403906
precision score of Stacked model: 0.9200754037101295
f1 score of Stacked model: 0.9110003269042171
```

Fig. 5.32 StackedRF Performance Parameters

Thus the ensemble model performs better than all other algorithms mentioned above. This shows the supreme outcome of StackedRF over other existing algorithm. In this chapter, data analysis and visualization is done, followed by preprocessing of the dataset and feature selection. Finally, comparative analysis is done and ensemble model is analyzed for performance. The performance attributes used were accuracy, recall, precision and f1 score. These are valid performance criteria for Classification problems. And hence, the results were analyzed and discussed in the form of charts and tables. The goal of this study is to create a machine learning-based model for predicting student performance based on academic achievement and dropout rates. The research makes use of a wide range of elements, such as demographic data, prior academic performance, socioeconomic circumstances, and other pertinent variables. For the purpose of forecasting student performance and detecting at-risk pupils, a variety of machine learning techniques are investigated and assessed. The effectiveness of the built prediction model in identifying pupils at risk of subpar performance or dropout is evaluated.

The results of this study have important ramifications for educational institutions, decision-makers, and teachers, empowering them to invest funds wisely and implement interventions on time to enhance student outcomes. The prediction model offers useful information that may guide the development of evidence-based policies and programs to lower dropout rates and improve educational equity. Overall, by utilizing machine learning approaches, this research advances the subject of predicting student performance and lays the groundwork for future developments. In the next chapter, conclusion of this research will be discussed.

CHAPTER-6

CONCLUSION AND FUTURE SCOPE

A useful tool in education can be machine learning predictions of student performance based on academic achievement or dropout rates. Machine learning algorithms may create prediction models to calculate the chance of a student's success or possible dropout by analyzing numerous criteria such prior academic achievements, attendance records, socioeconomic backgrounds, and other pertinent data. There are various advantages of using machine learning to predict student performance. In the first place, it can assist educational institutions in identifying pupils who may be at danger of having academic problems or quitting school. The chance of these pupils' academic achievement is increased by early identification of these children, which enables prompt interventions and the implementation of focused support systems.

Additionally, machine learning models can offer insightful information on the underlying causes of student success or failure. These models can find patterns and correlations by studying vast volumes of data that might not be immediately obvious using conventional techniques. With the use of this knowledge, evidence-based programs and policies may be created with the goal of enhancing overall educational results.

Additionally, machine learning's predictive skills can help teachers tailor their lessons to the needs of certain pupils. Better learning results result from teachers customizing their instruction and offering the right assistance to each student's individual requirements and problems. Additionally, students who can benefit from extra support or specialized programs can be identified using machine learning models, ensuring that educational resources are used efficiently. It's critical to recognize the constraints and difficulties involved in utilizing machine learning to forecast student success. The high dependence on past data for model training is a serious drawback. The predictions made by the models may be unfair or erroneous if the training data is skewed or lacking. In order to reduce these biases and guarantee that the models are strong and dependable, it is essential to regularly analyze and update the training data.

The ethical issues pertaining to data protection and privacy provide another difficulty. The collection and analysis of student data must be done with the utmost care in order to ensure confidentiality and adhere to applicable laws. Furthermore, it is not advisable to see machine learning models as a substitute for human discretion and knowledge. Education professionals and administrators are crucial in evaluating and using the findings to make wise decisions, even though they can offer insightful analysis and forecasts. The human element is essential for comprehending the context, nuance, and complexity of student performance and should not be overlooked in favor of relying entirely on predictions made by computers.

Thus in this research The presented ensemble method (StackedRF) is simulated on the basis of charts, on Google Colab using Python (Version 3.6.9). The dataset used is from Kaggle.com, it includes the details which help to predict student's performance on the basis of academic success or dropout rates. The dataset contains 35 attributes (including the Target) and 4425 student's details. In the comparative analysis of machine learning algorithms, the accuracy of Naïve Bayes is 85.39%, Logistic Regression is 92.01%, Random Forest Classifier is 92.01%, XGBoost Classifier is 91.4, Multi Layer Perceptron is 89.94% and Support Vector Machine is 91.04%.

As the results show that Random Forest Classifier performs best out of the 6 machine learning algorithms. Thus, we choose Random Forest Classifier with an accuracy of 92.01% for ensemble modeling. Thus to improve this algorithm we use ensemble method Stacking with Random Forest as Meta Classifier. The algorithms used in stacking are: K-Nearest Neighbor, Naïve Bayes and Random Forest. Thus the improved ensemble algorithm is named as StackedRF (Stacking with Random Forest as Meta Classifier). The StackedRF model gives the following result for prediction of student's academic success and dropout rates: it gives an accuracy of 94.73%, recall score value 90.4%, precision score is 92.0% and f1 score value is 91.1%. Thus, StackedRF is performing supreme over other existing models.

The prediction model offers useful information that may guide the development of evidence-based policies and programs to lower dropout rates and improve educational equity. Overall, by utilizing machine learning approaches, this research advances the subject of predicting student performance and lays the groundwork for future developments. The model incorporates the forecasts from other trained models when using an average ensemble, such as Random Forest

In conclusion, utilizing machine learning to predict student performance based on academic achievement or dropout rates has enormous potential to enhance educational results. It may be used to find at-risk pupils, provide interventions that are specific to them, and personalize education. Addressing this approach's drawbacks and difficulties is necessary, though, since these include biases in training data, privacy issues, and the requirement for human experience. We can leverage the advantages of predictive analytics to build a more efficient and inclusive educational system by fusing the strength of machine learning with the knowledge and expertise of educators.

REFERENCES

1. Albreiki, B., Zaki, N., & Alashwal, H. (2021). A Systematic Literature Review of Student' Performance Prediction Using Machine Learning Techniques. *Education Sciences*, *11*(9), 552. https://doi.org/10.3390/educsci11090552
2. Harikumar Pallathadka, Alex Wenda, Edwin Ramirez-Asís, Maximiliano Asís-López, Judith Flores-Albornoz, Khongdet Phasinam (2023). Classification and prediction of student performance data using various machine learning algorithms. Materials Today: Proceedings, Volume 80, Part 3, 2023, Pages 3782-3785, ISSN 2214-7853, https://doi.org/10.1016/j.matpr.2021.07.382.
3. M. Chitti, P. Chitti and M. Jayabalan, "Need for Interpretable Student Performance Prediction," 2020 13th International Conference on Developments in eSystems Engineering (DeSE), Liverpool, United Kingdom, 2020, pp. 269-272, doi: 10.1109/DeSE51703.2020.9450735.
4. Chen, Y., Zhai, L. A comparative study on student performance prediction using machine learning. Educ Inf Technol (2023). https://doi.org/10.1007/s10639-023-11672-1
5. Altabrawee, Hussein Osama & Qaisa& Ali, r, Samir. (2019). Predicting Students' Performance Using Machine Learning Techniques. JOURNAL OF UNIVERSITY OF BABYLON for pure and applied sciences. 27. 194-205. 10.29196/jubpas.v27i1.2108.

6. Hamsa, Hashmia & Indiradevi, Simi & Kizhakkethottam, Jubilant. (2016). Student Academic Performance Prediction Model Using Decision Tree and Fuzzy Genetic Algorithm. Procedia Technology. 25. 326-332. 10.1016/j.protcy.2016.08.114.
7. Shahiri, Amirah & Husain, Wahidah & Abdul Rashid, Nur'Aini. (2015). A Review on Predicting Student's Performance Using Data Mining Techniques. Procedia Computer Science. 72. 414-422. 10.1016/j.procs.2015.12.157.
8. Xu, Jie & Moon, Kyeong & Schaar, Mihaela. (2017). A Machine Learning Approach for Tracking and Predicting Student Performance in Degree Programs. IEEE Journal of Selected Topics in Signal Processing. PP. 1-1. 10.1109/JSTSP.2017.2692560.
9. M. Nagy and R. Molontay, "Predicting Dropout in Higher Education Based on Secondary School Performance," 2018 IEEE 22nd International Conference on Intelligent Engineering Systems (INES), Las Palmas de Gran Canaria, Spain, 2018, pp. 000389-000394, doi: 10.1109/INES.2018.8523888.
10. Ofori, F., Maina, E. & Gitonga, R. (2020). Using Machine Learning Algorithms to Predict Students’ Performance and Improve Learning Outcome: A Literature Based Review, Journal of Information and Technology, Vol. 4(1), 33-55.
11. Yağcı, M. Educational data mining: prediction of students' academic performance using machine learning algorithms. Smart Learn. Environ. 9, 11 (2022).
12. Máté Baranyi, Marcell Nagy, and Roland Molontay. 2020. Interpretable Deep Learning for University Dropout Prediction. In Proceedings of the 21st Annual Conference on Information Technology Education (SIGITE '20). Association for

Computing Machinery, New York, NY, USA, 13–19. https://doi.org/10.1145/3368308.3415382

13. Hassan, H., Anuar, S., Ahmad, N.B. (2019). Students' Performance Prediction Model Using Meta-classifier Approach. In: Macintyre, J., Iliadis, L., Maglogiannis, I., Jayne, C. (eds) Engineering Applications of Neural Networks. EANN 2019. Communications in Computer and Information Science, vol 1000. Springer, Cham. https://doi.org/10.1007/978-3-030-20257-6_19.
14. Alhothali, A., Albsisi, M., Assalahi, H., & Aldosemani, T. (2022). Predicting Student Outcomes in Online Courses Using Machine Learning Techniques: A Review. Sustainability, 14(10), 6199. https://doi.org/10.3390/su14106199
15. M. M. Tamada, J. F. de Magalhães Netto and D. P. R. de Lima, "Predicting and Reducing Dropout in Virtual Learning using Machine Learning Techniques: A Systematic Review," *2019 IEEE Frontiers in Education Conference (FIE)*, Covington, KY, USA, 2019, pp. 1-9, doi: 10.1109/FIE43999.2019.9028545.
16. Azimi, S., Popa, C.-G., & Cucić, T. (2020). Improving Students Performance in Small-Scale Online Courses - A Machine Learning-Based Intervention. International Journal of Learning Analytics and Artificial Intelligence for Education (iJAI), 2(2), pp. 80–95. https://doi.org/10.3991/ijai.v2i2.19371
17. R. Katarya, J. Gaba, A. Garg and V. Verma, "A review on machine learning based student's academic performance prediction systems," *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, Coimbatore, India, 2021, pp. 254-259, doi: 10.1109/ICAIS50930.2021.9395767.

18. Jha, Nikhil & Ghergulescu, Ioana & Moldovan, Arghir-Nicolae. (2019). OULAD MOOC Dropout and Result Prediction using Ensemble, Deep Learning and Regression Techniques. 154-164. 10.5220/0007767901540164.
19. Sachio Hirokawa. 2018. Key attribute for predicting student academic performance. In Proceedings of the 10th International Conference on Education Technology and Computers (ICETC '18). Association for Computing Machinery, New York, NY, USA, 308–313. https://doi.org/10.1145/3290511.3290576
20. W. Nuankaew and J. Thongkam, "Improving Student Academic Performance Prediction Models using Feature Selection," *2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, Phuket, Thailand, 2020, pp. 392-395, doi: 10.1109/ECTI-CON49241.2020.9158286.
21. Aslam, N., Khan, I. U., Alamri, L. H., & Almuslim, R. S. (2021). An Improved Early Student's Academic Performance Prediction Using Deep Learning. International Journal of Emerging Technologies in Learning (iJET), 16(12), pp. 108–122.
22. Ankit Porwal, Aditi Tulchhia "A Comparative Analysis of Decision Tree, KNN, Naïve Bayes and Neural Network for Student Performance Prediction", Mukt Shabd Journal (ISSN: 2347-3150) Volume XI, Issue III, March 2022
23. Dhilipan, J., Vijayalakshmi, N., Suriya, S., & Christopher, A. (2021, February). Prediction of students performance using machine learning. In IOP conference series: Materials science and engineering (Vol. 1055, No. 1, p. 012122). IOP Publishing.

24. Hayder, A. (2022). Predicting Student Performance Using Machine Learning: A Comparative Study Between Classification Algorithms.

25. Hashim, Ali & Akeel, Wid & Khalaf, Alaa. (2020). Student Performance Prediction Model based on Supervised Machine Learning Algorithms. IOP Conference Series: Materials Science and Engineering. 928. 032019. 10.1088/1757-899X/928/3/032019.

26. Ioanna Lykourentzou, Ioannis Giannoukos, Vassilis Nikolopoulos, George Mpardis, Vassili Loumos,Dropout prediction in e-learning courses through the combination of machine learning techniques, Computers & Education,Volume 53, Issue 3,2009, Pages 950965,ISSN03601315,https://doi.org/10.1016/j.compedu.2009.05.010.(https://www.sciencedirect.com/science/article/pii/S0360131509001249)

27. S. Batool, J. Rashid, M. W. Nisar, J. Kim, T. Mahmood and A. Hussain, "A Random Forest Students' Performance Prediction (RFSPP) Model Based on Students' Demographic Features," 2021 Mohammad Ali Jinnah University International Conference on Computing (MAJICC), Karachi, Pakistan, 2021, pp. 1-4, doi: 10.1109/MAJICC53071.2021.9526239.

28. Kabathova, J., & Drlik, M. (2021). Towards Predicting Student's Dropout in University Courses Using Different Machine Learning Techniques. Applied Sciences, 11(7), 3130. ttps://doi.org/10.3390/app11073130

29. I. Khan, A. Al Sadiri, A. R. Ahmad and N. Jabeur, "Tracking Student Performance in Introductory Programming by Means of Machine Learning," 2019

4th MEC International Conference on Big Data and Smart City (ICBDSC), Muscat, Oman, 2019, pp. 1-6, doi: 10.1109/ICBDSC.2019.8645608.

30. Rastrollo-Guerrero, J. L., Gómez-Pulido, J. A., & Durán-Domínguez, A. (2020). Analyzing and Predicting Students' Performance by Means of Machine Learning: A Review. Applied Sciences, 10(3), 1042. https://doi.org/10.3390/app10031042
31. C. S. K and K. S. Kumar, "Data Preprocessing and Visualizations Using Machine Learning for Student Placement Prediction," 2022 2nd International Conference on Technological Advancements in Computational Sciences (ICTACS), Tashkent, Uzbekistan, 2022, pp. 386-391, doi: 10.1109/ICTACS56270.2022.9988247.
32. A. Jain and S. Solanki, "An Efficient Approach for Multiclass Student Performance Prediction based upon Machine Learning," 2019 International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2019, pp. 1457-1462, doi: 10.1109/ICCES45898.2019.9002038.
33. Sultana, J. & Macigi, Usha Rani & Farquad, H.. (2019). Student's Performance Prediction using Deep Learning and Data Mining methods. International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8, Issue-1S4, June 2019.
34. Gajwani, J., Chakraborty, P. (2021). Students' Performance Prediction Using Feature Selection and Supervised Machine Learning Algorithms. In: Gupta, D., Khanna, A., Bhattacharyya, S., Hassanien, A.E., Anand, S., Jaiswal, A. (eds) International Conference on Innovative Computing and Communications. Advances in Intelligent Systems and Computing, vol 1165. Springer, Singapore. https://doi.org/10.1007/978-981-15-5113-0_25

35. Punlumjeak, W., Rachburee, N., & Arunrerk, J. (2017). Big Data Analytics: Student Performance Prediction Using Feature Selection and Machine Learning on Microsoft Azure Platform. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 9(1-4), 113–117. Retrieved from https://jtec.utem.edu.my/jtec/article/view/1791.

36. Zaffar, M., Hashmani, M. A., Savita, K. S., & Rizvi, S. S. H. (2018). A study of feature selection algorithms for predicting students academic performance. International Journal of Advanced Computer Science and Applications, 9(5).

37. Nicholas Robert Beckham, Limas Jaya Akeh, Giodio Nathanael Pratama Mitaart, Jurike V Moniaga, Determining factors that affect student performance using various machine learning methods, Procedia Computer Science, Volume 216, 2023, Pages 597-603, ISSN 1877-0509,https://doi.org/10.1016/j.procs.2022.12.174.

38. Baig, Mirza Azam; Shaikh, Sarmad Ahmed; Khatri, Kamlesh Kumar; Shaikh, Muneer Ahmed; Khan, Muhammad Zohaib; Rauf, Mahira Abdul Prediction of Students Performance Level Using Integrated Approach of ML Algorithms. International Journal of Emerging Technologies in Learning . 2023, Vol. 18 Issue 1, p216-234. 19p.

39. Koutina, M., Kermanidis, K.L. (2011). Predicting Postgraduate Students' Performance Using Machine Learning Techniques. In: Iliadis, L., Maglogiannis, I., Papadopoulos, H. (eds) Artificial Intelligence Applications and Innovations. EANN AIAI 2011 2011. IFIP Advances in Information and Communication

Technology, vol 364. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-23960-1_20

40. Oyedeji, A., Salami, A., Folorunsho, O., & Abolade, O. (2020, March 30). Analysis and Prediction of Student Academic Performance Using Machine Learning. JITCE (Journal of Information Technology and Computer Engineering), 4(01), 10-15. https://doi.org/https://doi.org/10.25077/jitce.4.01.10-15.2020.