

PREDICTION OF STUDENT PERFORMANCE ON VIRTUAL PLATFORM USING MACHINE LEARNING

*A main Project report submitted in the partial fulfilment of the requirements for the
award of the degree of*

**BACHELOR OF TECHNOLOGY
In
COMPUTER SCIENCE AND ENGINEERING**

Submitted by

D.Vamsika (19471A05K8)

Y.Anjani Priya (19471A05P3)

SK.Sameena (19471A05O8)

R.Chaitra (19471A05O3)

Under the esteemed guidance of

Dr. S V N Sreenivasu M.Tech, Ph.D.
Professor



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**NARASARAOPETA ENGINEERING COLLEGE: NARASARAOPET
(AUTONOMOUS)**

Accredited by NAAC with A+ Grade and NBA(Tier -1)

NIRF rank in the band of 251-320 and an ISO 9001:2015 Certified

Approved by AICTE, New Delhi, Permanently Affiliated to JNTUK, Kakinada

**KOTAPPAKONDA ROAD, YALAMANDA VILLAGE, NARASARAOPET-
522601**

2022-2023

**NARASARAOPETA ENGINEERING COLLEGE: NARASARAOPET
(AUTONOMOUS)**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE



This is to certify that the main project entitled “**PREDICTION OF STUDENT PERFORMANCE ON VIRTUAL PLATFORM USING MACHINE LEARNING**” is a bonafide work done by “**D. Vamsika (19471A05K8), Y. Anjani Priya (19471A05P3), SK. Sameena (19471A05O8), R. Chaitra (19471A05O3)**” in partial fulfilment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** in the Department of **COMPUTER SCIENCE AND ENGINEERING** during 2022-2023.

PROJECT GUIDE

Dr. S V N Sreenivasu, M.Tech., Ph.D
Professor

PROJECT CO-ORDINATOR

Dr. M. Sireesha, M.Tech., Ph.D
Associate Professor

HEAD OF THE DEPARTMENT

Dr. S. N. Tirumala Rao, M.Tech., Ph.D
Professor

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We wish to express our thanks to carious personalities who are responsible for the completion of the project. We are extremely thankful to our beloved chairperson sir **M.V.Koteswara Rao**, B.Sc who took keen interest on us in every effort throughout this course. We owe out gratitude toour principal **Dr.M. Sreenivasa Kumar**, M.Tech., Ph.D.(UK), MISTE., FIE(I), his kind attention and valuable guidance throughout the course.

We express our deep felt gratitude to **Dr. S.N.Tirumala Rao**, M.Tech.,Ph.D, Head of CSE department and our guide **Dr. S V N Sreenivasu**, M.Tech.,Ph.D Professor of CSE department whose valuable guidance and unstinting encouragement enable us to accomplish our project successfully in time.

We extend our sincere thanks towards **Dr. M. Sireesha**, M.Tech.,Ph.D Associate professor & project coordinator of the project for extending her encouragement. Their profound knowledge and willingness have been a constant source of inspiration for us throughout this project work.

We extend our sincere thanks to all other teaching and non- teaching staff to department for their cooperation and encouragement during our B.Tech degree.We have no words to acknowledge thewarm affection, constant inspiration and encouragement that we received from our parents.

We affectionately acknowledge the encouragement received fromour friends and those who involved in giving valuable suggestions had clarifying out doubts which had really helped us in successfully completing our project.

By

D.Vamsika	(19471A05K8)
Y.AnjaniPriya	(19471A05P3)
SK.Sameena	(19471A05O8)
R.Chaitra	(19471A05O3)

ABSTRACT

Online learning platforms such as Massive Open Online Course (MOOC), Virtual Learning Environments (VLEs), and Learning Management Systems (LMS) facilitate thousands or even millions of students to learn according to their interests without spatial and temporal constraints. Besides many advantages, online learning platforms face several challenges such as students' lack of interest, high dropouts, low engagement, students' self-regulated behavior. In this project, a predictive model is implemented that analyzes the problems faced by at-risk students. Subsequently, facilitating instructors for timely intervention to persuade students to increase their study engagements and improve their study performance.

The predictive model is trained and tested using Random Forest, Support Vector Machine, K-Nearest Neighbor, Extra Tree Classifier, Ada Boost Classifier, Gradient Boosting to characterize the learning behavior of students according to their study variables. The predictive model can help instructors in identifying at-risk students early in the course for timely intervention thus avoiding student dropouts. Results have shown that students' assessment scores, engagement intensity i.e. clickstream data, and time-dependent variables are important factors. The predictive model trained using Random Forest (RF) gives the best results.

INSTITUTE VISION AND MISSION

INSTITUTION VISION

To emerge as a Centre of excellence in technical education with a blend of effective student centric teaching learning practices as well as research for the transformation of lives and community,

INSTITUTION MISSION

M1: Provide the best class infra-structure to explore the field of engineering and research

M2: Build a passionate and a determined team of faculty with student centric teaching,imbibing experiential, innovative skills

M3: Imbibe lifelong learning skills, entrepreneurial skills and ethical values in students for addressing societal problems



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION OF THE DEPARTMENT

To become a centre of excellence in nurturing the quality Computer Science & Engineering professionals embedded with software knowledge, aptitude for research and ethical values to cater to the needs of industry and society.

MISSION OF THE DEPARTMENT

The department of Computer Science and Engineering is committed to

M1: Mould the students to become Software Professionals, Researchers and Entrepreneurs by providing advanced laboratories.

M2: Impart high quality professional training to get expertize in modern software tools and technologies to cater to the real time requirements of the Industry.

M3: Inculcate team work and lifelong learning among students with a sense of societal and ethical responsibilities.



Program Specific Outcomes (PSO's)

PSO1: Apply mathematical and scientific skills in numerous areas of Computer Science and Engineering to design and develop software-based systems.

PSO2: Acquaint module knowledge on emerging trends of the modern era in Computer Science and Engineering

PSO3: Promote novel applications that meet the needs of entrepreneur, environmental and social issues.



Program Educational Objectives (PEO's)

The graduates of the programme are able to:

PEO1: Apply the knowledge of Mathematics, Science and Engineering fundamentals to identify and solve Computer Science and Engineering problems.

PEO2: Use various software tools and technologies to solve problems related to academia, industry and society.

PEO3: Work with ethical and moral values in the multi-disciplinary teams and can communicate effectively among team members with continuous learning.

PEO4: Pursue higher studies and develop their career in software industry.

Program Outcomes

- 1. Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
- 2. Problem analysis:** Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
- 3. Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
- 4. Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
- 5. Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
- 6. The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
- 7. Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Project Course Outcomes (CO'S):

CO425.1: Analyse the System of Examinations and identify the problem.

CO425.2: Identify and classify the requirements

CO425.3: Review the Related Literature

CO425.4: Design and Modularize the project

CO425.5: Construct, Integrate, Test and Implement the Project.

CO425.6: Prepare the project Documentation and present the Report using appropriate method.

Course Outcomes – Program Outcomes mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C425.1		✓											✓		
C425.2	✓		✓		✓								✓		
C425.3				✓		✓	✓	✓					✓		
C425.4			✓			✓	✓	✓					✓	✓	
C425.5					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
C425.6									✓	✓	✓		✓	✓	

Course Outcomes – Program Outcome correlation

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C425.1	2	3											2		
C425.2			2		3								2		
C425.3				2		2	3	3					2		
C425.4			2			1	1	2					3	2	
C425.5					3	3	3	2	3	2	2	1	3	2	1
C425.6									3	2	1		2	3	

Note: The values in the above table represent the level of correlation between CO's and PO's:

1. Low level

2. Medium level

3. High level

Project mapping with various courses of Curriculum with Attained PO's:

Name of the course from which principles are applied in this project	Description of the device	Attained PO
C3.2.4, C3.2.5	Gathering the information about prediction of student performance on virtual platform	PO1, PO3
CC4.2.5	Each and every requirement is critically analyzed, the process model is identified and divided into five modules	PO2, PO3
CC4.2.5	Logical design is done by using the unified modelling language which involves individual team work	PO3, PO5, PO9
CC4.2.5	Each and every module is tested, integrated, and evaluated in our project	PO1, PO5
CC4.2.5	Documentation is done by all our four members in the form of a group	PO10
CC4.2.5	Each and every phase of the work in group is presented periodically	PO10, PO11
CC4.2.5	Implementation is done and the project will be handled by the virtual platform and in future updates in our project can be done based on the score improvement.	PO4, PO7
CC4.2.8 CC4.2.	The physical design includes software components.	PO5, PO6

INDEX

S. No.	CONTENTS	PAGE NO
I	List of Figures	xv
1	Introduction	
	1.1 Introduction	1-2
	1.2 Existing System	3
	1.3 Proposed System	3
	1.4 System Requirements	4
	1.4.1 Hardware Requirements	
	1.4.2 Software Requirements	
2	Literature Survey	
	2.1 Machine Learning	5-8
	2.2 Machine learning methods	8-9
	2.3 Applications of machine learning	9
3	System Analysis	
	3.1 Architecture of the Proposed System	10
	3.2 Workflow of the Proposed System	11
	3.3 Modules to be implemented	11-13
	3.3.1 Modules Description	
	3.3.1.1 Dataset Preprocessing and merging	
	3.3.1.2 Building the machine learning models	
	3.4 Implementation of machine learning using python	14-15
	3.5 Algorithms	16-20
	3.6 Data Set	20-21
	3.7 Data Analysis	21-24
	3.8 Data Preprocessing	24
	3.8.1 Data Preprocessing	24-25
	3.8.2 Missing values	25

	3.8.3 Correlation coefficient method	25-26
	3.8.4 Confusion Matrix	26-27
	3.9 Implementation of code	27-49
	3.10 Result Analysis	50-52
4.	Output Screens	53-56
5.	Conclusion	57
6.	Future Scope	58
7	Bibliography	59-60

LIST OF FIGURES

S.NO.	LIST OF FIGURES	PAGE NO
1	Fig 3.1.1: Overall architecture	10
2	Fig 3.3.1: Schema for Merging of Data	12
3	Fig 3.8.1.1: Data Pre-processing	24
4	Fig 3.8.2.1: Missing Values	25
5	Fig 3.8.3.1: Corelation	26
6	Fig 3.8.4: Confusion matrix	27
7	Fig 3.10.1: Comparision of code modules	50
8	Fig 3.10.2: Late submissions in courses	51
9	Fig 3.10.3: Comparision of modules	52
10	Fig 4.1: Output screen for Data Uploading Screen	53
11	Fig 4.2: Output screen for prediction of Distinction Screen	53
12	Fig.4.3: Output screen for Data Uploading Screen	54
13	Fig 4.4: Output screen for Pass Screen	54
14	Fig 4.5: Output screen for Uploading Screen 3	55
15	Fig 4.6: Output screen for Prediction of Fail Screen	55
16	Fig 4.7: Output screen for Uploading Screen 4	56
17	Fig 4.8: Output screen for Prediction of Withdrawn Screen	56

CHAPTER 1

INTRODUCTION

1.1 Introduction

A massive open online course (MOOC) is an online course that has open access and interactive participation by means of the Web. MOOCs provide participants with course materials that are normally used in a conventional education setting - such as examples, lectures, videos, study materials and problem sets.

An open online course is an online course aimed at unlimited participation and open access via the Web. In addition to traditional course materials, such as filmed lectures, readings, and problem sets, many MOOCs provide interactive courses with user forums or social media discussions to support community interactions among students, professors, and teaching assistants (TAs), as well as immediate feedback to quick quizzes and assignments. MOOCs are a widely researched development in distance education, first introduced in 2008, that emerged as a popular mode of learning in 2012.

MOOCs provide an affordable and flexible way to learn new skills, advance your career and deliver quality educational experiences at scale. Millions of people around the world use MOOCs to learn for a variety of reasons, including: career development, changing careers, college preparations, supplemental learning, lifelong learning, corporate eLearning & training, and more.

The task of predicting students at different course lengths always been a challenging problem for MOOCS. The main reason behind this prediction of dropouts increases in day by day and they are probably to left the course. However, developing a predictive model that can identify the exact learning behavior of students earlier in the course by analyzing their behavior data is a challenging task. In an online learning environment, where a large amount of data is generated every day, machine learning (ML) techniques could help in analyzing the variables that define the students and come up with the results that better describe their learning behavior, thus, ML may reveal information that is beneficial for both instructors and students.

Generally, there are two ways for prediction at risk students at early stages. Fundamental analysis is one of them and relies on a student assignment submission and scores. The second

one is the technical analysis method, which concentrates on previous course completion details and values. This analysis uses histograms and patterns to predict early dropout of course.

Generally, there are two ways for prediction at risk students at early stages. Fundamental analysis is one of them and relies on a student assignment submission and scores. The second one is the technical analysis method, which concentrates on previous course completion details and values. This analysis uses histograms and patterns to predict early dropout of course.

It is clear that there are always unpredictable factors such as the student such as difficulty in course or lack of skills in understanding the concept, which affect the student course. Therefore, if the data gained from MOOCS table are efficiently pre-processed and suitable algorithms are employed, the risk of students at an early stage can be predicted.

In early prediction risk of students, machine learning and deep learning approaches can help students through their decisions to continue the course or not. These methods intend to automatically recognize and learn patterns among big amounts of information. The algorithms can be effectively self-learning, and can tackle the predicting of early risk of students to complete the course successfully.

The problem is to predict the risk of students at early stages of course using machine learning algorithms on the OULAD dataset. In doing so, we are employed to find the risk of students at different stages of the course to complete the course successfully.

The task of early prediction of risk of students helps learners to complete their courses. The main reason behind this prediction is to complete the MOOCs course more effectively. Difficulty in course and understanding are the two main challenges caused by instability of early prediction of students who are at risk and help them to complete the course.

The primary objective of this project is the earliest possible identification of students who are at-risk of dropouts by leveraging Machine Learning techniques to understand variables associated with the learning behavior of students and how they interact with the Virtual Learning Environment.

1.2 Existing System

In existing systems, the prediction is done at the end of the course length, The predictive models were not validated with additional databases, the variables related to student engagement are not used,.In Existing System they make use of Deep Learning Models.

Disadvantages

- 1.Doesn't generate accurate and efficient results.
- 2.Computation time is very high.
- 3.Lacking of accuracy may result in lack of efficient further Prediction.

1.3 Proposed System

By Using this Proposed System we can reduce the errors,enhance student performance correctly by dividing the course length into different categories. It is easier to predict the performance of the students.

Advantages

1. Generates accurate and efficient results.
2. Computation time is greatly reduced.
3. Reduces manual work.
4. Efficient further prediction

1.4. System Requirements

1.4.1 Hardware Requirements:

- Processor : intel®core™i7-7500UCPU@2.70gh
- Speed : 2.4GHz
- RAM : 12GB
- Hard Disk : 8GB
- Cache memory : 4 MB

1.4.2 Software Requirements

- Operating System : Windows or MacOS
- Language : Python
- Python distribution : Anaconda, Flask

CHAPTER 2

LITERATURE SURVEY

2.1 Machine Learning

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

There are many proposed frameworks for prediction of risk of students using many differential algorithms and different classifiers. Every individual framework proposed at that time have certain advantages over previous models and methodologies and certain drawbacks with present proposed robust frame works.

Models trained on particular Dataset(Platform Dependency) :

Only the particular dataset is used for predicting the students but the following studies are unable to predict on different datasets. The following frameworks that are platform dependent and based on the binary classification.

C. C. Gray and D. Perkins[6], proposed “Utilizing early engagement and machine learning to predict student outcomes,” This paper examines the literature surrounding current methods and measures in use in Learning Analytics. Their work defines a new descriptive statistic for student attendance and applies modern machine learning tools and techniques to create a predictive model. It demonstrates how students can be identified as early as week 3 (of the Fall semester) with approximately 97% accuracy. It identifies the possible failing students at week 3 of the full semester and the algorithm used is the k-Nearest-Neighbor with 97% accuracy. It situates this result within an appropriate pedagogical context to support its use as part of a more comprehensive student support mechanism. It shows the better results but it is platform dependent that it is stable and only applicable to the Bangor university students.

J. Y. Chung and S. Lee [9] proposed “Dropout early warning systems for high school students using machine learning” In this study, we use the random forests in machine learning to predict

students at risk of dropping out. The data used in this study are the samples of 165,715 high school students from the 2014 National Education Information System (NEIS), algorithm used is the Random Forest (RF). Performance achieved by the algorithm of accuracy 95%. *It* showed an excellent performance in predicting students' dropouts in terms of various performance metrics for binary classification. The problem addressed is the students' binary classification which makes better accuracy, but the limitation for this classification is it suffers from the potential inaccuracy in calculating the weights of the feature that are considered.

S. Lee and J. Y. Chung,[17] presented the study “The machine learning-based dropout early warning system for improving the performance of dropout prediction,”. This study aimed to improve the performance of a dropout early warning system: (a) by addressing the class imbalance issue using the synthetic minority oversampling techniques (SMOTE) and the ensemble methods in machine learning; and (b) by evaluating the trained classifiers with both receiver operating characteristic (ROC) and precision–recall (PR) curves. This model improves the performance of dropout prediction by using the machine learning based early warning system. It uses algorithms of the Random Forest, boosted decision tree. The BDT shows the highest accuracy of the 99%. As it shows the highest accuracy but it failed for platform dependence. Here the limited NEIS database is used, all the features were not included in creating a predictive model.

Consideration of Important Variables:

For predicting the students at earlier stage, we need to consider the Demographic data which is an important variable for early intervention. Along with Demographic data Clickstream data and Assessment scores are important time dependent variable. The following studies doesn't consider the important variables:

R. Al-Shabandar, A. J. Hussain, P. Liatsis, and R. Keight[16], “Detecting at-risk students with early interventions using machine learning techniques”. In this paper, the early identification of students who are at risk of withdrew and failure is provided. Therefore, two models are constructed namely at-risk student model and learning achievement model. The models have the potential to detect the students who are in danger of failing and withdrawal at the early stage of the online course. The proposed frameworks can be used to assist instructors in delivering intensive intervention support to at-risk student's. It identifies of at-risk students with intensive earlier intervention in online courses by using an Gradient Boosting Model with

the accuracy of 95%. In this the temporal features are not considered to predict the students. This predictive model was not validated with additional database that is considered.

A. Behr, M. Giese, and K. Theune,[1] “Early prediction of university dropouts—A random forest approach,” It uses to classify based on the Binary Classification of modelling student’s dropout. It evaluates how predictive performance changes over the three models, and observe a substantially increased performance when including variables from the first study experiences. The algorithm used for achieving the better performance is the Random Forest with AUC (area under curve) of 0.86. In this model, the students’ satisfaction (wishes and needs) features are not considered. This may impact the model’s performance.

L.C. B. Martins, R. N. Carvalho, R. S. Carvalho, M. C. Victorino, and M. Holanda,[12] “Early prediction of college attrition using data mining,” it predicts the at earliest possible attrition. The algorithms were used for this model are Gradient Boosting Machine, Distributed Random Forest. Deep Learning Model outperformed the other models by achieving the highest True Positive Rate of 71.1%. This model doesn’t consider about the first semester data which includes the Demographic data which is an important feature to be selected.

N. Mduma, K. Kalegele, and D. Machuve,[15] “Machine learning approach for reducing students dropout rates,” This paper, presents a thorough analysis of four supervised learning classifiers that represent linear, ensemble, instance and neural networks, identifies the at-risk students using a machine learning method. The goal of the study is to provide data-driven algorithm recommendations to current researchers on the topic. It uses Linear Regression with ROC score of 0.88. It doesn’t use the under-sampling approach with a penalized model was not used.

Predicting performance at end of course length:

Predicting the students at the end of the course length is not useful for the early intervention and it doesn’t have any significance for the improvement of a student during the course.

A. S. Imran, F. Dalipi, and Z. Kastrati,[3] “Predicting student dropout in a MOOC: An evaluation of a deep neural network model,” study on the application of feed-forward deep neural network architectures to address the problem. This model achieves not only high accuracy, but also low false negative rate while predicting dropouts on the MOOC data. Moreover, it also provide an in-depth comparison of the proposed architectures concerning precision, recall, and F1 measure. it predicts and explained about the student dropout. A feed-forward neural network is used with an accuracy greater than the 90%. In this model the

prediction is done after the course completion which doesn't predict the students earlier of the course length.

A. A. Mubarak, H. Cao, and S. A. M. Ahmed,[4]“Predictive learning analytics using deep learning model in MOOCs' courses videos,” , This paper exploits a temporal sequential classification problem by analyzing video clickstream data and predict learner performance, which is a vital decision-making problem, by addressing their issues and improving the educational process. This paper employs a deep neural network (LSTM) on a set of implicit features extracted from video clickstreams data to predict learners' weekly performance and enable instructors to set measures for timely intervention. it used the deep neural network to predict the learning analytics in MOOCs course videos. It uses the algorithms like LSTM,ANN,SVM, Logistic Regression. LSTM performs with the highest accuracy of 93%. This current model only employs' learners interaction patterns with videos, A complete learningactivity pattern of learners is missing.

J. Xu, K. H. Moon, and M. van der Schaar,[11] “A machine learning approach for tracking and predicting student performance in degree programs,” In this paper, developed a novel machine learning method for predicting student performance in degree programs that is able to address these key challenges. The proposed method has two major features. First, a bilayered structure comprising multiple base predictors and a cascade of ensemble predictors is developed for making predictions based on students' evolving performance states. Second, a data-driven approach based on latent factor models and probabilistic matrix factorization is proposed to discover course relevance, which is important for constructing efficient base predictors. It tracks and predicts the student performance using ML techniques. Linear Regression, Logistic Regression, Random Forest, KNN,EPP, Ensemble based progressive prediction showed the best result having the lowest mean square error. Courses prediction to the students was not carriedout. There was no interventiontechniqueis employed. “Student performance prediction and classification using machine learning algorithms,” In this paper, two datasets have been considered for the prediction and classification of student performance respectively using five machine learning algorithms.

2.2 Machine Learning Methods

Machine Learning algorithms are often categorized as supervised and unsupervised.

Supervised machine learning algorithms can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known

training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly

Unsupervised machine learning algorithms are used when the information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system doesn't figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.

Reinforcement machine learning algorithms is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behaviour within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best. This is known as the reinforcement signal

2.3 Applications of machine learning

1. Virtual Personal Assistants
2. Predictions while Commuting
3. Videos Surveillance
4. Social Media Services
5. Email Spam and Malware Filtering
6. Online Customer Support
7. Search Engine Result Refining
8. Product Recommendations
9. Online Fraud Detection

CHAPTER 3

SYSTEM ANALYSIS

Six ML algorithms were selected for training/testing the predictive models during different stages of the course. For modeling Open University Learning Analytics Dataset (OULAD), these algorithms were designated for classifying students' performance into four categories i.e. Withdrawn (students who were not able to complete the course), Fail (students who completed the course but were not able to secure passing marks), Pass (completed courses with a passing score), Distinction (completed courses with excellent grades). Our study includes two different approaches for inputs, numerical data and categorical data, to investigate the effect of pre-processing.

3.1 Architecture of the Proposed System

For the earliest possible prediction of student's performance, divide the course length into 20%,40%,60%,80% and 100% of course completed. Initially, the future prediction of the student performance solely depends on demographic data. After the course starts at 20% length of course assessment data, VLE i.e the click stream data can be used.

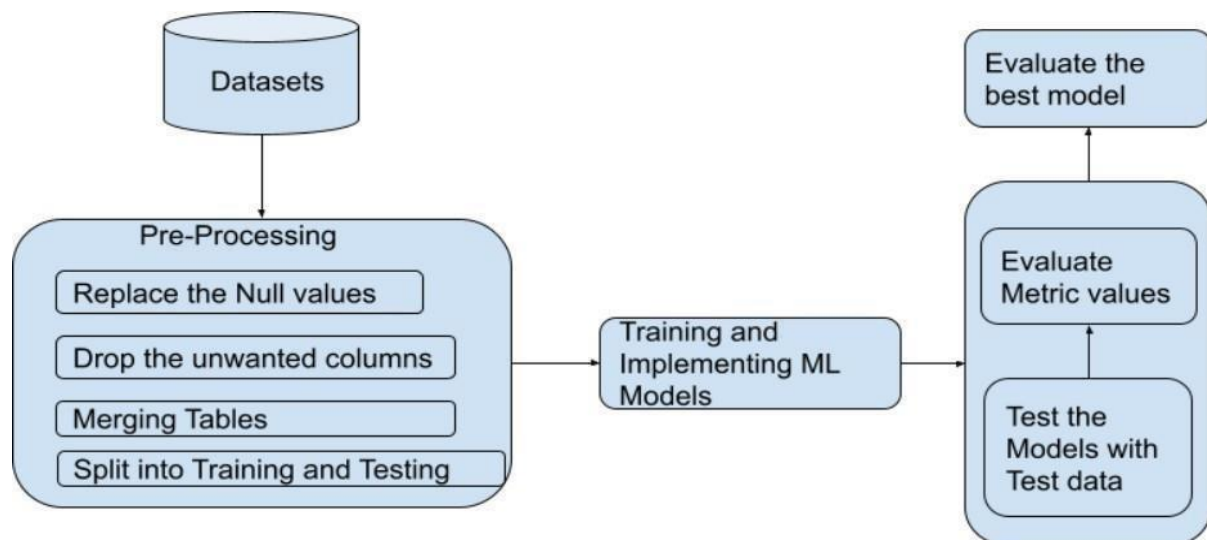


Fig.3.1.1 Overall architecture

3.2 Workflow of the Proposed System

A workflow consists of an orchestrated and repeatable pattern of business activity enabled by the systematic organization of resources into processes that transform materials, provide services, or process information. It can be depicted as a sequence of operations, the work of a person or group, the work of an organization of staff, or one or more simple or complex mechanisms.

Workflows may be viewed as one fundamental building block to be combined with other parts of an organization's structure such as information technology, teams, projects and hierarchies.

The workflow illustrates different modules in student prediction. In this, proposed a predictive model that analyzes the problems faced by at-Risk students, to persuade students to improve their study performance. Initially to improve the performance of the model, use the Data Preprocessing. Data Preprocessing- It is a technique that is used to improve the quality of data before applied mining. Example: Missing values of Date are replaced by mean Date mean value. Students' demographic data is merged with student's assessment data. To know the students interaction with VLE demographic data is combined with VLE information. For the better performance, the triplet that is student-assessment-clickstream data table is generated.

3.3 Modules to be Implemented

Initially, Data pre-processing, merging of tables and Feature Engineering are the main phases in this project.

3.3.1 Modules Description

The modules used for analysis are:

1. Data Pre-processing and Merging of Tables
2. Building the Machine Learning models.
3. Testing the model.

3.3.1.1 Dataset Preprocessing and Merging:

We are taken the datasets from Open University Learning Analytics Dataset

(OULAD), it is provided by the Open University. Student data is collected in a structured manner in 7 tables. It contains the details of demographics (Student info & registration), students' Virtual Learning Environment (VLE) interaction, assessments, course registration, and courses offered. Students' daily activities and VLE interaction are represented as clickstreams data (number of clicks) stored in the student VLE table. Students' assessment scores are stored in a dataset triple called student-module presentation. The OULAD was generated for the year 2013 and 2014 containing 7 courses, 22 module-presentations with 32,593 registered students.

The effective preprocessing of the dataset and the selection of the right classifier highly affect the result and accuracy of the machine learning-based approach. Data preprocessing is a data mining technique, which cleans the raw data and delivers more relevant data. Raw data contains numerous numbers of missing values, outliers, and unnecessary features. Our dataset contains large numbers of infinite and NaN (Not a number) values that are replaced by the mean value using the NumPy library in Python.

The important variables like Clickstream data, Assessment Scores were obtained by merging of the data. These improve the performance of the models. The merging takes place based on the schema.

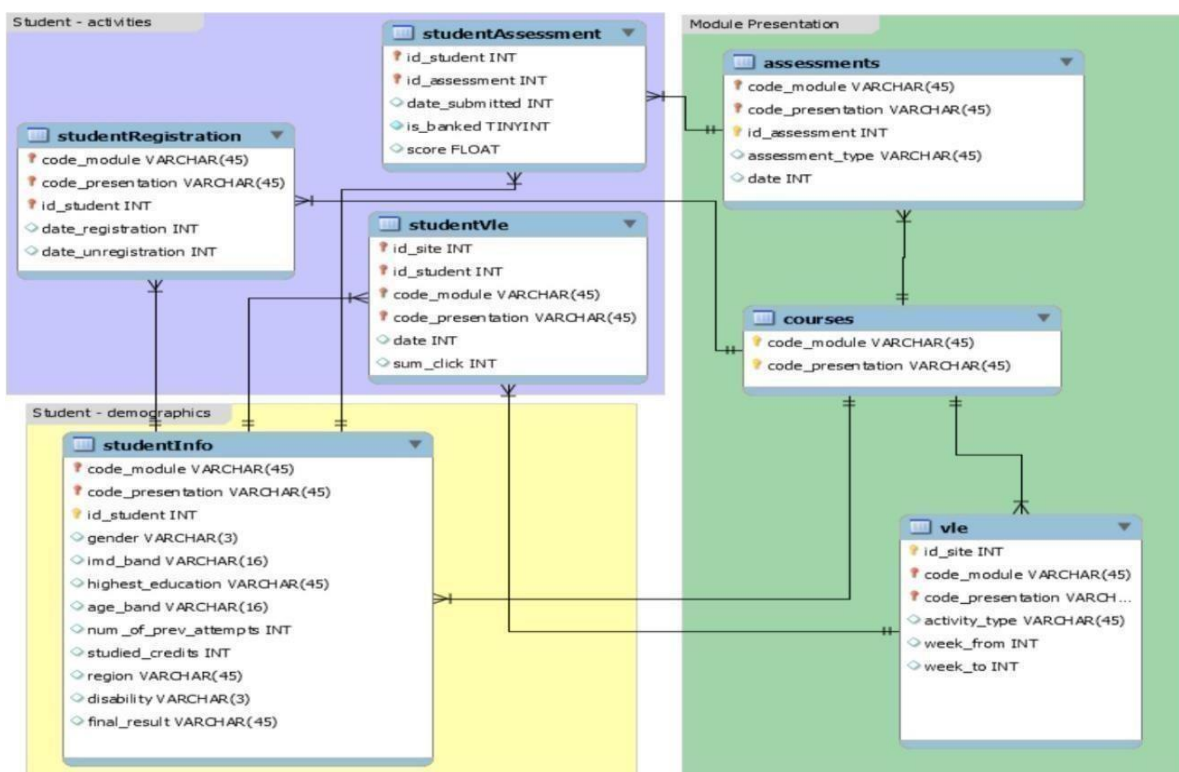


Fig.3.3.1 Schema for Merging of data

3.3.1.1. Building the Machine Learning models:

To build the machine learning model we have tried using different algorithms in machine learning like:

1. K Nearest Neighbors
2. Support Vector Machine
3. Ada Boost Classifier
4. Random Forest Classifier
5. Gradient Boosting Classifier
6. Extra Tree Classifier

Python is used to load the dataset, pre-process the dataset, extract features, build the model, split the dataset into train and test parts, train the model, test the model. Dataset is divided in 20:80 ratio. Where 20% of the data goes into the testing part of data and 80% goes into the training part of data. After splitting the data, machine learning models were trained using the training dataset.

3.3.1.2. Testing the Machine Learning Model:

The model was tested using the test data and the final accuracy is calculated. To obtain correct results, machine learning models were trained and tested by splitting the dataset in different ways. Training and testing can be done by 4 phases:

PHASE-1: Using DEMOGRAPHIC DATA

The results when trained only on demographic data indicates the very low performance. Moreover, the performance of all predictive models for fail position is degraded. So to improve the performance of predictive model, we consider the VLE data.

PHASE-2: Using DEMOGRAPHIC and CLICK STREAM DATA

In the case of Pass class, Random Forest, Extra tree classifier, Ada Boost Classifier and Gradient Boost Classifier showing satisfactory results. The performance scores of predictive model are better when trained only on the demographics.

PHASE-3: Considering DEMOGRAPHIC, CLICK STREAM and ASSESSMENT

When trained there is a substantial improvement in performance of pass, withdrawn, distinction, and fail classes. But the performance results of SVM and K-NN were still very low.

FEATURE ENGINEERING:

To further improve the performance results, merge the different classes like Distinction-Pass were combined into Pass class and Withdrawn-Fail class as Fail class. outperformed all other models.

3.4 Implementation of machine learning using Python

Python is a popular programming language. It was created in 1991 by Guido van Rossum. It is used for:

- 1.web development (server-side),
- 2.software development,
- 3.mathematics,
- 4.system scripting

The most recent major version of Python is Python 3. However, Python 2, although not being updated with anything other than security updates, is still quite popular.

It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse, Anaconda which are particularly useful when managing larger collections of Python files.

Python was designed for its readability. Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses. Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose. In the older days, people used to perform Machine Learning tasks manually by coding all the algorithms and mathematical and statistical formula. This made the process time consuming, tedious and inefficient. But in the modern days, it is become very much easy and efficient compared to the olden days by various python libraries, frameworks, and modules.

Python is one of the most popular programming languages for this task and it has replaced many languages in the industry, one of the reason is its vast collection of libraries. Python libraries that used in Machine Learning are:

- 1.Numpy
- 2.Scipy

3.Scikit-learn

4.Pandas

5.Matplotlib

NumPy is a very popular python library for large multi-dimensional array and matrix processing, with the help of a large collection of high-level mathematical functions. It is very useful for fundamental scientific computations in Machine Learning. It is particularly useful for linear algebra, Fourier transform, and random number capabilities. High-end libraries like TensorFlow uses NumPy internally for manipulation of Tensors.

SciPy is a very popular library among Machine Learning enthusiasts as it contains different modules for optimization, linear algebra, integration and statistics. There is a difference between the SciPy library and the SciPy stack. The SciPy is one of the core packages that make up the SciPy stack. SciPy is also very useful for image manipulation.

Skikit-learn is one of the most popular Machine Learning libraries for classical Machine Learning algorithms. It is built on top of two basic Python libraries, NumPy and SciPy. Scikit-learn supports most of the supervised and unsupervised learning algorithms. Scikit learn can also be used for data-mining and data-analysis, which makes it a great tool who is starting out with Machine Learning.

Pandas is a popular Python library for data analysis. It is not directly related to Machine Learning. As we know that the dataset must be prepared before training. In this case, Pandas comes handy as it was developed specifically for data extraction and preparation. It provides high-level data structures and wide variety tools for data analysis. It provides many inbuilt methods for groping, combining and filtering data.

Matpoltlib is a very popular Python library for data visualization. Like Pandas, it is not directly related to Machine Learning. It particularly comes in handy when a programmer wants to visualize the patterns in the data. It is a 2D plotting library used for creating 2D graphs and plots. A module named pyplot makes it easy for programmers for plotting as it provides features to control line styles, font properties, formatting axes, etc. It provides various kinds of graphs and plots for data visualization, histogram, error charts, bar chats, etc..

3.5 Algorithms:

In this study, they used six machine learning methods (Random Forest, Support Vector Machine, KNN, Extra Tree Classifier, Adaboost Classifier, Gradient Boosting)

RANDOM FOREST

Great number of decision trees make a random forest model. The model basically averages the forecast result of trees, which is named a forest. Also, the algorithm includes three random ideas, selecting training data randomly when forming trees, randomly choosing some subsets of variables when dividing nodes and deeming only a subset of all variables for splitting every node in each basic decision tree. Every basic tree learns from a random sample of the dataset during the training process of a random forest.

The random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random Forest classifier:

- There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.
- The predictions from each tree must have very low correlations.

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase. The Working process can be explained in the below steps:

Step-1: Select random K data points from the training set.

Step-2: Build the decision trees associated with the selected data points (Subsets).

Step-3: Choose the number N for decision trees that you want to build.

Step-4: Repeat Step 1 & 2.

Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes

SUPPORT VECTOR MACHINE(SVM)

Support Vector Machines (SVMs) are a set of supervised learning approaches that can be employed for classification and regression problems. The classifier version is named SVC. The method's purpose is finding a decision boundary between two classes with vectors. The boundary must be far from any point in the dataset, and support vectors are the sign of observation coordinates with a gap named margin. SVM is a boundary that best separates two classes with employing a line or hyperplane. The decision boundary is defined in Equation 1 where SVMs can map input vectors $x_i \in \mathbb{R}^d$ into a high dimensional feature space $\Phi(x_i) \in H$, and Φ is mapped by a kernel function $K(x_i, x_j)$.

$$f(x) = \text{sgn}\left(\sum_{i=1}^n \alpha_i y_i \cdot K(x, x_i) + b\right) \quad (1)$$

SVMs can perform a linear or non-linear classification efficiently, but for non-linear, they must use a kernel trick which map inputs to high dimensional feature spaces. SVMs convert non-separable classes to separable ones by kernel functions such as linear, non-linear, sigmoid, radial basis function (RBF) and polynomial. The formula of kernel functions is shown in Equations 2-4 where the constant of radial basis function and d is the degree of polynomial function. Indeed, there are two adjustable parameters in the sigmoid function, the slope α and the intercepted constant c .

$$RBF : K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \quad (2)$$

$$Polynomial : K(x_i, x_j) = (x_i \cdot x_j + 1)^d \quad (3)$$

$$Sigmoid : K(x_i, x_j) = \tanh(\alpha x_i^T y + c) \quad (4)$$

SVMs are often effective in high dimensional spaces and cases where the number of dimensions is greater than the number of samples, but to avoid over-fitting in selecting regularization term and kernel functions, the number of features should be much greater than the number of samples.

K NEAREST NEIGHBOR

Two properties usually are suggested for KNN, lazy learning and nonparametric algorithm, because there is not any assumption for underlying data distribution by KNN. The method follows some steps to find targets: Dividing dataset into training and test data, selecting the value

of K , determining which distance function should be used, choosing a sample from test data (as a new sample) and computing the distance to its n training samples, sorting distances gained and taking k -nearest data samples, and finally, assigning the test class to the sample on the majority vote of its k neighbours.

The K-NN working can be explained on the basis of the below steps:

Step-1: Select the number K of the neighbors

Step-2: Calculate the Euclidean distance of K number of neighbors

Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.

Step-4: Among these k neighbors, count the number of the data points in each category.

Step-5: Assign the new data points to that category for which the number of the neighbors is maximum.

Step-6: Model is ready.

EXTRA TREE CLASSIFIER

Extremely randomized tree classifier (extra tree classifier) is a type of ensemble learning technique which aggregates the results of multiple decorrelated decision trees collected in a “forest” to output its classification result. Each decision tree in Extra Trees Forest is constructed from the original training sample. In concept, it is very similar to a Random Forest Classifier and only differs from it in the manner of construction of the decision trees in the forest. Each tree is provided with a random sample of k features from the feature-set, leads to the creation of multiple de-correlated decision trees. Extra tree classifier working can be explained below

Step 1: Importing the required libraries.

Step 2: Loading and Cleaning the Data.

Step 3: Building the Extra Trees Forest and computing the individual feature importances.

Step 4: Visualizing and Comparing the results.

ADABOOST CLASSIFIER

The process of converting some weak learners to a powerful one is named Boosting method. AdaBoost is a specific type of Boosting that is an ensemble model to progress the predictions of every learning technique. The goal of boosting is to train weak learners sequentially for

adjusting their previous predictions. This model is a meta-predictor which starts by fitting a model on the basic dataset before fitting additional copies of it on the same dataset. During the process of training, samples' weights are modified based on the current forecasting error; therefore, the consequent model focuses on tough items.

Procedure:

Step 1 – Creating First Base Learner: The algorithm takes the first feature, i.e., feature 1, and creates the first stump f_1 . It will create the same number of stumps as the number of features. From all these stumps it will create decision trees and can be called stumps base learner model. Out of these all models, the algorithm selects only one. For selecting a base learner, there are two properties, those are, Gini and Entropy. The stump that has the least value will be the first base learner.

Step 2 – Calculating the Total Error (TE): The total error is the sum of all the errors in the classified record for sample weights.

Step 3 – Calculating Performance of Stump: Formula for calculating Performance of Stump is: –

Performance of stump = $\frac{1}{2} \ln \left[\frac{1-TE}{TE} \right]$ where, \ln is natural log and TE is Total Error.

Step 4 – Updating Weight: For incorrectly classified records the formula is:

New Sample Weight = Sample Weight * $e^{(\text{Performance})}$ And for correctly classified records, we use the same formula with a negative sign with performance, so that the weight for correctly classified records will reduce compared to the incorrect classified ones.

The formula is:

New Sample Weight = Sample Weight * $e^{- (\text{Performance})}$

Step 5 – Creating New Dataset: Now, it's time to create a new dataset from our previous one. In the new dataset, the frequency of incorrectly classified records will be more than the correct ones.

While considering these normalized weights, have to create a new dataset and that dataset is based on normalized weights. It will probably select the wrong records for training purposes. That will be the second decision tree/stump. To make a new dataset based on normalized weight, the algorithm will divide it into buckets

GRADIENT BOOSTING

Gradient boosting classifiers are a group of machine learning algorithms that combine many weak learning models together to create a strong predictive model. Decision trees are usually

used when doing gradient boosting. The objective of Gradient Boosting classifiers is to minimize the loss, or the difference between the actual class value of the training example and the predicted class value. Gradient Boosting Machines, every time a new weak learner is added to the model, the weights of the previous learners are frozen or cemented in place, left unchanged as the new layers are introduced. This is distinct from the approaches used in adaboosting where the values are adjusted when new learners are added.

Procedure:

Step 1: Install libraries, xgboost, margittr, Matrix

Step 2: Create a Matrix for train and test datasets- with the use of xgb.DMatrix() function

Step 3: Set parameters, for params and watchlist

Step 4: Build model using xgb.train() function

Step 5: Use xgb.importance() function for feature analysis

Step 6: Make prediction with the use of predict() function

Scope of the Project

The scope of this system is to maintain student details in datasets, train the model using the large quantity of data present in datasets and predict the performance of the students whether he was pass, fail or withdrawn.

3.6 Data Set

We are taken the datasets from Open University Learning Analytics Dataset (OULAD), it is provided by the Open University. Student data is collected in a structured manner in 7 tables. It contain the details of demographics(Student info & registration), students' Virtual Learning Environment (VLE) interaction, assessments, course registration, and courses offered. Students' daily activities and VLE interaction are represented as clickstreams data (number of clicks) stored in the student VLE table. Students' assessment scores are stored in a dataset triplet called student-module presentation. The OULAD was generated for the year 2013 and 2014 containing 7 courses, 22 module-presentations with 32,593 registered students

1	code_mod	code_pres	id_student	gender	region	highest_ec	imd_band	age_band	num_of_p	studied_cr	disability	final_result
2	AAA	2013J	11391	M	East Anglia	HE Qualific	90-100%	55<=	0	240	N	Pass
3	AAA	2013J	28400	F	Scotland	HE Qualific	20-30%	35-55	0	60	N	Pass
4	AAA	2013J	30268	F	North Wes	A Level or	30-40%	35-55	0	60	Y	Withdrawn
5	AAA	2013J	31604	F	South East	A Level or	50-60%	35-55	0	60	N	Pass
6	AAA	2013J	32885	F	West Midl	Lower Tha	50-60%	0-35	0	60	N	Pass
7	AAA	2013J	38053	M	Wales	A Level or	80-90%	35-55	0	60	N	Pass
8	AAA	2013J	45462	M	Scotland	HE Qualific	30-40%	0-35	0	60	N	Pass
9	AAA	2013J	45642	F	North Wes	A Level or	90-100%	0-35	0	120	N	Pass
10	AAA	2013J	52130	F	East Anglia	A Level or	70-80%	0-35	0	90	N	Pass

Fig 3.6.1 Dataset

3.7 Data Analysis

The dataset contains 32593 rows 12 attributes which are used to predict the performance of the students using datasets such as

1. code_module
2. code_presentation
3. id_student
4. gender
5. region
6. highest_education
7. imd_band
8. age_band
9. num_of_prev_attempts
10. studied_credits
11. disability
12. final_result

- code_module Name of course, for which student registered
- code_presentation Name of semester, for which student registered
- id_student Unique integer identifying each student
- gender Students gender
- region UK region, in which student lives
- Region values East Anglian Region, Scotland, North Western Region, South East Region, West Midlands Region, Wales, North Region, South Region, Ireland, South West Region, East Midlands Region, Yorkshire Region, London Region
- highest_education Highest education student achieved before taking course.
- Highest education values HE Qualification - awarded after one year full-time study at the university or higher education institution
- A Level or Equivalent - secondary school leaving qualification
- Lower Than A Level - did not completed secondary school
- Post Graduate Qualification - equal to Master degree more or less
- No Formal quals - no previous formal education
- imd_band Index of Multiple Deprivation percentile, students with imd_band lower than 20 comes from the most deprived regions
- age_band Age band of student
- num_of_prev_attempts Number of student previous attempts on the selected course
- studied_credits Total credits student is studying at the Open University during period of the course
- disability Student claims disability of any type, logical
- final_result Student final result in the course
- Final result values Pass - passed the course
- Withdrawn - withdrawn the course before official end
- Fail - failed the course after taking final exam
- Distinction - passed course with outstanding results.

F-Score, Accuracy, Precision and Recall metrics are employed to evaluate the performance of our models. For Computing F-score and Accuracy, Precision, Recall, Specificity and Sensitivity must be evaluated by Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN).

Accuracy:

The accuracy metric is one of the simplest Classification metrics to implement, and it can be determined as the number of correct predictions to the total number of predictions.

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

Precision:

The precision metric is used to overcome the limitation of Accuracy. The precision determines the proportion of positive prediction that was actually correct. It can be calculated as the True Positive(TP) or predictions that are actually true to the total positive predictions (True Positive and False Positive(FP)).

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

Recall:

It is also similar to the Precision metric; however, it aims to calculate the proportion of actual positive that was identified incorrectly. It can be calculated as True Positive (TP) or predictions that are actually true to the total number of positives, either correctly predicted as positive or incorrectly predicted as negative (true Positive and false negative(FN)).The formula for calculating Recall is given below:

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

F-Score:

It is the harmonic mean of a system's precision and recall values. It can be calculated by the following formula:

$$\text{F-Score} = 2 \times \text{Precision} \times \text{Recall} / \text{Precision} + \text{Recall}$$

Specificity:

It can be described as the algorithm/model's ability to predict a true negative of each category available. Formally it can be calculated by the following formula:

$$\text{Specificity} = \text{TN} / \text{TN} + \text{FP}$$

Sensitivity:

It can be described as the metric used for evaluating a model's ability to predict the true positives of each available category.

$$\text{Sensitivity} = \text{TP} / \text{TP} + \text{FN}$$

Among classification metrics, Accuracy is a good metric, but it is not enough for all classification problems. It is often necessary to look at some other metrics to make sure that a model is reliable. F-Score might be a better metric to employ if results need to achieve a balance between Recall and Precision, especially when there is an uneven class distribution.

3.8 Data Pre-processing

Before feeding data to an algorithm we have to apply transformations to our data which is referred as pre-processing. By performing pre-processing the raw data which is not feasible for analysis is converted into clean data. In-order to achieve better results using a model in Machine Learning, data format has to be in a proper manner. The data should be in a particular format for different algorithms. For example, if we consider Random Forest algorithm it does not support null values. So that those null values have to be managed using raw data.

3.8.1 Data Pre-processing:

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data Pre-processing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis.

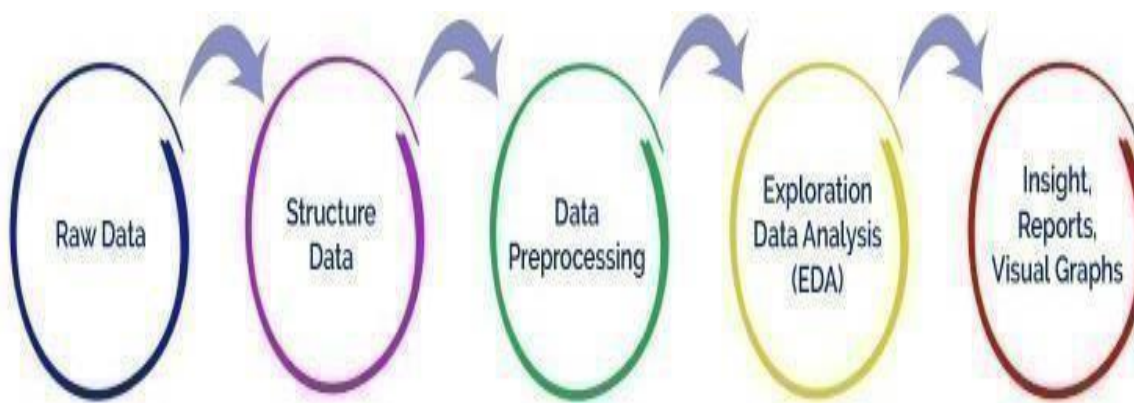


Fig 3.8.1.1 Data Pre-processing

Need of Data Preprocessing:

For achieving better results from the applied model in Machine Learning projects the format of the data has to be in a proper manner. Some specified Machine Learning model needs information in a specified format. For example, Random Forest algorithm does not support null values, therefore to execute random forest algorithm null values have to be managed from the original raw data set. Another aspect is that data set should be formatted in such a way that more than one Machine Learning are executed in one data set, and best out of them is chosen.

3.8.2 Missing values

Filling missing values is one of the pre-processing techniques. The missing values in the dataset is represented as '?' but it a non-standard missing value and it has to be converted into a standard missing value NaN. So that pandas can detect the missing values. The heat map representing the missing values. We have filled that missing values with 0 .

```
studentInfo_df.isnull().sum()
code_module      0
code_presentation 0
id_student       0
gender           0
region           0
highest_education 0
imd_band        1111
age_band         0
num_of_prev_attempts 0
studied_credits  0
disability       0
final_result     0
dtype: int64

[17] studentInfo_df['imd_band'] = studentInfo_df['imd_band'].fillna(studentInfo_df['imd_band'].mode()[0])
studentInfo_df1['imd_band'] = studentInfo_df1['imd_band'].fillna(studentInfo_df1['imd_band'].mode()[0])

studentInfo_df.isnull().sum()
code_module      0
code_presentation 0
id_student       0
gender           0
region           0
highest_education 0
imd_band         0
age_band         0
num_of_prev_attempts 0
studied_credits  0
disability       0
final_result     0
dtype: int64
```

Fig 3.8.2.1 Missing Values

3.8.3 Correlation coefficient method

We can find dependency between two attributes p and q using Correlation coefficient method using the formula.

$$r_{p,q} = \frac{\sum (p_i - \bar{p})(q_i - \bar{q})}{n \sigma_p \sigma_q}$$

$$= \frac{\sum (p_i q_i) - n \bar{p} \bar{q}}{n \sigma_p \sigma_q}$$

n is the total number of patterns, p_i and q_i are respective values of p and q attributes in patterns i, \bar{p} and \bar{q} are respective mean values of p and q attributes, σ_p , σ_q are respective

standard deviations values of p and q attributes. Generally, $-1 \leq r_{p,q} \leq +1$. If $r_{p,q} < 0$, then p and q are negatively correlated. If $r_{p,q} = 0$, then p and q are independent attributes and there is no correlation between them. If $r_{p,q} > 0$, then p and q are positively correlated. We can drop the attributes that are having correlation coefficient value as 0 as it indicates that the variables are independent with respect to the prediction attribute. Fig:3.8.2 is the correlation heat map. After applying correlation the attributes are PR interval , QRS duration , QT interval , QTc interval, P wave , T wave , QRS wave and problem . The attribute Vent_rate got dropped.



Fig 3.8.3.1 Correlation

3.8.4 Confusion Matrix

A confusion matrix is a table that is often used to describe the performance of a classification model(or “ classifier”) on a set of test data for which the true values .

		Predicted class	
		<i>P</i>	<i>N</i>
Actual Class	<i>P</i>	True Positives (TP)	False Negatives (FN)
	<i>N</i>	False Positives (FP)	True Negatives (TN)

A true positive (tp) is a result where the model predict the positive class correctly. Similarly

A true negative (tn) is an outcome where the model correctly predicts the negative class.

A false positive (fp) is an outcome where the model the incorrectly predicts the positive class and

A false negative (fn) is an outcome where the model incorrectly predicts the negative class.

3.9 Implementation of Code

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.svm import SVC
from sklearn.linear_model import SGDClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier as RFC
from sklearn.ensemble import ExtraTreesClassifier as ETC
from sklearn.neighbors import KNeighborsClassifier as KNN
from sklearn.ensemble import AdaBoostClassifier as ABC
from sklearn.ensemble import GradientBoostingClassifier as GBC
from sklearn.neural_network import MLPClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report,
precision_score, recall_score, f1_score
```

```

from sklearn.pipeline import Pipeline
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import KFold

#connect to drive
from google.colab import drive
drive.mount('/content/drive')

# Reading Datasets
assessment_df=pd.read_csv('/content/drive/MyDrive/studentdataset/assessments.csv')
courses_df=pd.read_csv('/content/drive/MyDrive/studentdataset/courses.csv')
studentAssessment_df=pd.read_csv('/content/drive/MyDrive/studentdataset/studentAssessment.csv')
studentInfo_df=pd.read_csv('/content/drive/MyDrive/studentdataset/studentInfo.csv')
studentInfo_df1=pd.read_csv('/content/drive/MyDrive/studentdataset/studentInfo.csv')
studentRegistration_df=pd.read_csv('/content/drive/MyDrive/studentdataset/studentRegistration.csv')
studentVle_0=pd.read_csv('/content/drive/MyDrive/studentVle/studentVle_0.csv')
studentVle_1=pd.read_csv('/content/drive/MyDrive/studentVle/studentVle_1.csv')
studentVle_2=pd.read_csv('/content/drive/MyDrive/studentVle/studentVle_2.csv')
studentVle_3=pd.read_csv('/content/drive/MyDrive/studentVle/studentVle_3.csv')
studentVle_4=pd.read_csv('/content/drive/MyDrive/studentVle/studentVle_4.csv')
studentVle_5=pd.read_csv('/content/drive/MyDrive/studentVle/studentVle_5.csv')
studentVle_6=pd.read_csv('/content/drive/MyDrive/studentVle/studentVle_6.csv')
studentVle_7=pd.read_csv('/content/drive/MyDrive/studentVle/studentVle_7.csv')
vle_df=pd.read_csv('/content/drive/MyDrive/studentdataset/vle.csv')
studentVle_df=pd.concat([studentVle_0,studentVle_1,studentVle_2,studentVle_3,studentVle_4,studentVle_5,studentVle_6,studentVle_7])

#for plotting
def stacked_plot(data, column_one, column_two, agg_column, plot_size=(10, 5)):
    pal = sns.color_palette("colorblind")
    grouped = data.groupby([column_one, column_two])[agg_column].count()
    grouped = grouped.groupby(level=0).apply(lambda x:100 * x / float(x.sum()))
    grouped = grouped.unstack(column_two).fillna(0)
    print(grouped)
    unique_list = list(data[column_two].unique())

```

```

    grouped[unique_list].plot(kind='bar', stacked=True, color=pal, figsize=plot_size)
#confusion matrix

def specificity(tn,fp,fn,tp):
    return tn/(tn+fp)
def sensitivity(tn,fp,fn,tp):
    return tp/(fn+tp)
assessment_df['date'] =
    assessment_df['date'].fillna(int(assessment_df['date'].astype(float).mean()))
studentAssessment_df.isnull().sum()
studentAssessment_df = studentAssessment_df.dropna()
studentRegistration_df.isnull().sum()
studentRegistration_df['date_unregistration'] = ['0' if pd.isnull(days) else '1' for days in
    studentRegistration_df['date_unregistration']]
studentRegistration_df['date_registration'] =studentRegistration_df['date_registration'].fill
    na(0).astype(float).apply(abs)
studentInfo_df.isnull().sum()
studentInfo_df['imd_band'] =
    studentInfo_df['imd_band'].fillna(studentInfo_df['imd_band'].mode()[0])
studentInfo_df1['imd_band'] =
    studentInfo_df1['imd_band'].fillna(studentInfo_df1['imd_band'].mode()[0])
studentVle_df.isnull().sum()
vle_df.isnull().sum()
vle_df = vle_df.drop(['week_from', 'week_to'], axis = 1)
studentInfo_df.drop(['id_student'],axis=1,inplace=True)
studentInfo_df.drop(['imd_band'],axis=1,inplace=True)
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
label=le.fit_transform(studentInfo_df['code_module'])
label1=le.fit_transform(studentInfo_df['code_presentation'])
label2=le.fit_transform(studentInfo_df['gender'])
label3=le.fit_transform(studentInfo_df['region'])
label4=le.fit_transform(studentInfo_df['highest_education'])
#label5=le.fit_transform(studentInfo_df['imd_band'])

```

```

label6=le.fit_transform(studentInfo_df['age_band'])
#label7=le.fit_transform(studentInfo_df['num_of_prev_attempts'])
#label8=le.fit_transform(studentInfo_df['studied_credits'])
label9=le.fit_transform(studentInfo_df['disability'])
label10=le.fit_transform(studentInfo_df['final_result'])
studentInfo_df['code_module']=label
studentInfo_df['code_presentation']=label1
studentInfo_df['gender']=label2
studentInfo_df['region']=label3
studentInfo_df['highest_education']=label4
#studentInfo_df['imd_band']=label5
studentInfo_df['age_band']=label6
#studentInfo_df['num_of_prev_attempts']=label7
#studentInfo_df['studied_credits']=label8
studentInfo_df['disability']=label9
studentInfo_df['final_result']=label10
studentInfo_df.head(20)
#corelation
plt.figure(figsize=(15,8))
sns.heatmap(studentInfo_df.corr(),annot=True)
plt.show()
X=studentInfo_df.loc[:, "code_module": "disability":1]
#print(X)
target_names=["Distinction", "Fail", "Pass", "Withdrawn"]
Y=studentInfo_df["final_result"]
print(Y)
#split
x_train,x_test,y_train,y_test=train_test_split(X,Y,test_size=0.2,stratify=Y)
#Modelling
SVC:
model_svm=SVC()
model_svm.fit(x_train, y_train)
y_pred = model_svm.predict(x_test)
print(classification_report(y_test,y_pred,target_names=target_names,digits=4))

```

RFC:

```
model_rfc=RFC()
model_rfc.fit(x_train,y_train)
y_pred = model_rfc.predict(x_test)
print(classification_report(y_test,y_pred,target_names=target_names,digits=4))
```

#KNN

```
model_knn=KNN()
model_knn.fit(x_train,y_train)
y_pred = model_knn.predict(x_test)
print(classification_report(y_test,y_pred,target_names=target_names,digits=4))
```

#ETC

```
model_etc=ETC()
model_etc.fit(x_train,y_train)
y_pred = model_etc.predict(x_test)
print(classification_report(y_test,y_pred,target_names=target_names,digits=4))
```

#ABC

```
model_abc=ABC()
model_abc.fit(x_train,y_train)
y_pred = model_abc.predict(x_test)
print(classification_report(y_test,y_pred,target_names=target_names,digits=4))
```

#GBC

```
model_gbc=GBC()
model_gbc.fit(x_train,y_train)
y_pred = model_gbc.predict(x_test)
print(classification_report(y_test,y_pred,target_names=target_names,digits=4))
```

##merging

```
student_vle_merge_vle = studentVle_df.merge(vle_df, on=['id_site', 'code_module',
        'code_presentation'],how = 'left')
student_vle_merge_vle[(student_vle_merge_vle['id_student']==28400) &
        (student_vle_merge_vle['date']<0)].head(10)
student_vle_merge_vle['Click_Timing'] = ['Before' if date < 0 else 'After' for date in
        student_vle_merge_vle['date']]
```

```

student_vle_merge_vle['After_Clicks'] = np.where(student_vle_merge_vle['Click_Timing']
    == 'After', student_vle_merge_vle['sum_click'], 0)
student_vle_merge_vle['Before_Clicks'] =
    np.where(student_vle_merge_vle['Click_Timing']
    == 'Before', student_vle_merge_vle['sum_click'], 0)

student_vle_merge_vle_group = student_vle_merge_vle.groupby(['code_module',
    'code_presentation', 'id_student'], as_index=False)['sum_click', 'After_Clicks',
    'Before_Clicks'].sum()
student_vle_merge_vle_group.isnull().sum()
student_vle_merge_vle.head()

##merge studentRegistration table with the courses table to understand the related between
    registrations and length of course
student_registration_merge_courses = studentRegistration_df.merge(courses_df, on =
['code_module', 'code_presentation'], how = 'left')
student_registration_merge_courses['Year'] =
student_registration_merge_courses['code_presentation'].str[0:4]
student_registration_merge_courses['Starting_Month'] = ['February' if code[-1] == 'B' else
    'October' for code in student_registration_merge_courses['code_presentation']]
student_registration_merge_courses

#course length doesn't have much difference for the student who withdrawn and students
    who completed the course
student_registration_merge_courses.groupby('date_unregistration',
    as_index=False)['module_presentation_length'].mean()

#merge assessments Table with the studentAssessment Table to understand the relationship
    between assessment and student performance
student_assessment_merge_assessment = studentAssessment_df.merge(assessment_df, on =
    ['id_assessment'], how='left')

## we can check whether there was a late submission or not (0 : Late, 1:OnTime)
student_assessment_merge_assessment['Late_submission'] = ['0' if
    int(student_assessment_merge_assessment['date_submitted'].iloc[i]) >
    int(student_assessment_merge_assessment['date'].iloc[i]) else '1'
    for i in

```

```

range(len(student_assessment_merge_assessment))]

print('Percentage of Late Submissions From Students are : ')
print(((len(student_assessment_merge_assessment[student_assessment_merge_assessment['
Late_submission'] == '0'])/len(student_assessment_merge_assessment)*100))
print('We can see that approximately 30 percent of students submitted their assignments
late')

#There are three types of assessments :- Tutor Marked Assessment (TMA), Computer
Marked Assessment (CMA), Exams

# Following Plot Shows us the Percentage of Late Submission by Assessment_Type
stacked_plot(student_assessment_merge_assessment, 'assessment_type', 'Late_submission',
'id_student', plot_size=(10, 7))

# There are 7 course modules. 4 are from STEM and 3 from Social Sciences
# Social Sciences :- AAA, BBB, GGG
# STEM :- CCC, DDD, EEE, FFF
stacked_plot(student_assessment_merge_assessment, 'code_module', 'Late_submission',
'id_student', plot_size=(12, 8))

student_assessment_merge_assessment['Code_Category'] = ['Social_Science' if
student_assessment_merge_assessment['code_module'].iloc[i] in ['AAA', 'BBB', 'GGG']
else 'STEM' for i in range(len(student_assessment_merge_assessment))]

# As from the description of table we know that score less than 40 is considered as Fail and
above that is pass

student_assessment_merge_assessment['Result'] = ['Fail' if
int(student_assessment_merge_assessment['score'].iloc[i]) < 40 else 'Pass' for i in
range(len(student_assessment_merge_assessment))]

## Weightage of Assignment can have impact on the submissions and Result of
students.categorized the weight into Low, Medium and High Weightage.
print(student_assessment_merge_assessment['weight'].unique())
percentage_segment = []
for percent in student_assessment_merge_assessment['weight']:
    if percent <= 10:
        percentage_segment.append('Low_Weightage')
    elif percent > 10 and percent <= 30:
        percentage_segment.append('Medium_Weightage')

```



```

else:
    percentage_segment.append('High_Weightage')
student_assessment_merge_assessment['Weighthage'] = percentage_segment
#MERGING VLE DATA WITH THE STUDENT INFO DATA
student_info = studentInfo_df1.merge(student_vle_merge_vle_group,
                                     on = ['code_module', 'code_presentation', 'id_student'],
                                     how = 'left')
student_info['sum_click'] =
    student_info['sum_click'].fillna(student_info['sum_click'].mean())
student_info['After_Clicks'] =
    student_info['After_Clicks'].fillna(student_info['After_Clicks'].mean())
student_info['Before_Clicks'] =
    student_info['Before_Clicks'].fillna(student_info['Before_Clicks'].mean())
student_info['highest_education'] = [0 if education in ['A Level or Equivalent', 'Lower Than
    A Level', 'No Formal quals'] else 1 for education in student_info['highest_education']]
#Combining Student Info with the Student Registration Table
student_registration_merge_courses = student_registration_merge_courses.drop('date_unre
    gistration', axis = 1)
student_info = student_info.merge(student_registration_merge_courses,on = ['code_module',
'code_presentation', 'id_student'],how = 'left')
student_info['num_of_prev_attempts'] = [0 if attempts == 0 else 1 for attempts in student_inf
o['num_of_prev_attempts']]
student_info['Code_Category'] = ['Social_Science' if student_info['code_module'].iloc[i] in ['
AAA', 'BBB', 'GGG']else 'STEM' for i in range(len(student_info))]
# In this step, I will remove code_module, code_presentation, id_student and Year as those w
on't have impact on the result
student_info = student_info.drop(['code_presentation', 'id_student', 'Year'], axis = 1)
student_info['date_registration'] = student_info['date_registration'].astype(float)
student_info['date_registration'].describe()
student_info.head()
def categorical_encoding(df, column_name_list=[])

```

```

for column_name in column_name_list:
    print(df[column_name].unique())
    categorical_columns = pd.get_dummies(df[column_name], prefix = column_name,
                                         prefix_sep = '_', drop_first = False)
    df = pd.concat([df, categorical_columns], axis = 1)
    df = df.drop(column_name, axis = 1)
return df

#to label
def labelEncoder(data, columns_list):
    for col in columns_list:
        encoder = LabelEncoder()
        data[col] = encoder.fit_transform(data[col])
    return data

# There are two types of categorical variables in the data.
# 1. NOMINAL :- Here there is no order in the categories
# 2. ORDINAL :- When there is order in the category
nominal_columns = ['gender', 'region', 'disability', 'Starting_Month', 'code_module', 'Code_Category']
ordinal_columns = ['highest_education', 'imd_band', 'age_band']
data = labelEncoder(student_info, ordinal_columns)
data = categorical_encoding(student_info, nominal_columns)
data.head()

#splitting the data
temp_df=data
X=temp_df.drop(['final_result'],axis=1)
target_names=["Distinction","Fail","Pass","Withdrawn"]
Y=data["final_result"]
x_train,x_test,y_train,y_test=train_test_split(X,Y,test_size=0.2,random_state=42,stratify=Y)

#SVC
model_svm=SVC()
model_svm.fit(x_train, y_train)
y_pred = model_svm.predict(x_test)

```

```

print(classification_report(y_test,y_pred,target_names=target_names,digits=4))
#KNN
model_knn=KNN()
model_knn.fit(x_train,y_train)
y_pred = model_knn.predict(x_test)
print(classification_report(y_test,y_pred,target_names=target_names,digits=4))
#ETC
model_etc=ETC()
model_etc.fit(x_train,y_train)
y_pred = model_etc.predict(x_test)
print(classification_report(y_test,y_pred,target_names=target_names,digits=4))
#ABC
model_abc=ABC()
model_abc.fit(x_train,y_train)
y_pred = model_abc.predict(x_test)
print(classification_report(y_test,y_pred,target_names=target_names,digits=4))
#GBC
model_gbc=GBC()
model_gbc.fit(x_train,y_train)
y_pred = model_gbc.predict(x_test)
print(classification_report(y_test,y_pred,target_names=target_names,digits=4))
#RFC
model_rfc=RFC()
model_rfc.fit(x_train,y_train)
y_pred = model_rfc.predict(x_test)
print(classification_report(y_test,y_pred,target_names=target_names,digits=4))
#MLP
mlp = MLPClassifier(random_state=1,hidden_layer_sizes=(15,))
mlp.fit(x_train,y_train)
y_pred=mlp.predict(x_test)
print(classification_report(y_test,y_pred,target_names=target_names,digits=4))
data2=data[data['final_result']=='Withdrawn']
data1 = data.drop(data2.index, axis=0)
data1.head(20)

```

#feature engineering:First case,assuming Withdrawn Class as Fail and building a model whether student will pass or fail.

```
data['Result'] = [0 if result in ['Pass', 'Distinction'] else 1 for result in data['final_result']]
```

```
feature = data.drop(['final_result', 'Result'], axis = 1)
```

```
target = data['Result']
```

```
target_names=["Pass","Fail"]
```

```
x_train, x_test, y_train, y_test = train_test_split(feature, target, test_size = 0.2, random_state = 123, stratify=data.final_result)
```

```
#SVC
```

```
model_svm=SVC()
```

```
model_svm.fit(x_train, y_train)
```

```
y_pred = model_svm.predict(x_test)
```

```
sa=accuracy_score(y_test,y_pred)
```

```
sp=precision_score(y_test,y_pred)
```

```
sr=recall_score(y_test,y_pred)
```

```
sf=f1_score(y_test,y_pred)
```

```
print(sa)
```

```
tn, fp, fn, tp = confusion_matrix(y_test,y_pred).ravel()
```

```
print(classification_report(y_test,y_pred,target_names=target_names,digits=4))
```

```
print("Sensitivity:",sensitivity(tn,fp,fn,tp))
```

```
print("Specificity:",specificity(tn,fp,fn,tp))
```

```
#kNN
```

```
model_knn=KNN()
```

```
model_knn.fit(x_train,y_train)
```

```
y_pred = model_knn.predict(x_test)
```

```
ka=accuracy_score(y_test,y_pred)
```

```
kp=precision_score(y_test,y_pred)
```

```
kr=recall_score(y_test,y_pred)
```

```
kf=f1_score(y_test,y_pred)
```

```
tn, fp, fn, tp = confusion_matrix(y_test,y_pred).ravel()
```

```
print(classification_report(y_test,y_pred,target_names=target_names,digits=4))
```

```
print("Sensitivity:",sensitivity(tn,fp,fn,tp))
```

```
print("Specificity:",specificity(tn,fp,fn,tp))
```

```
#ETC
```

```

model_etc=ETC()
model_etc.fit(x_train,y_train)
y_pred = model_etc.predict(x_test)
ea=accuracy_score(y_test,y_pred)
ep=precision_score(y_test,y_pred)
er=recall_score(y_test,y_pred)
ef=f1_score(y_test,y_pred)
tn, fp, fn, tp = confusion_matrix(y_test,y_pred).ravel()
print(classification_report(y_test,y_pred,target_names=target_names,digits=4))
print("Sensitivity:",sensitivity(tn,fp,fn,tp))
print("Specificity:",specificity(tn,fp,fn,tp))
#ABC
model_abc=ABC()
model_abc.fit(x_train,y_train)
y_pred = model_abc.predict(x_test)
aa=accuracy_score(y_test,y_pred)
ap=precision_score(y_test,y_pred)
ar=recall_score(y_test,y_pred)
af=f1_score(y_test,y_pred)
tn, fp, fn, tp = confusion_matrix(y_test,y_pred).ravel()
print(classification_report(y_test,y_pred,target_names=target_names,digits=4))
print("Sensitivity:",sensitivity(tn,fp,fn,tp))
print("Specificity:",specificity(tn,fp,fn,tp))
#GBC
model_gbc=GBC()
model_gbc.fit(x_train,y_train)
y_pred = model_gbc.predict(x_test)
ga=accuracy_score(y_test,y_pred)
gp=precision_score(y_test,y_pred)
gr=recall_score(y_test,y_pred)
gf=f1_score(y_test,y_pred)
tn, fp, fn, tp = confusion_matrix(y_test,y_pred).ravel()
print(classification_report(y_test,y_pred,target_names=target_names,digits=4))
print("Sensitivity:",sensitivity(tn,fp,fn,tp))

```

```

print("Specificity:",specificity(tn,fp,fn,tp))
#RFC
model_rfc=RFC()
model_rfc.fit(x_train,y_train)
y_pred = model_rfc.predict(x_test)
ra=accuracy_score(y_test,y_pred)
rp=precision_score(y_test,y_pred)
rr=recall_score(y_test,y_pred)
rf=f1_score(y_test,y_pred)
tn, fp, fn, tp = confusion_matrix(y_test,y_pred).ravel()
print(classification_report(y_test,y_pred,target_names=target_names,digits=4))
print("Sensitivity:",sensitivity(tn,fp,fn,tp))
print("Specificity:",specificity(tn,fp,fn,tp))
#MLP
mlp = MLPClassifier(random_state=1,hidden_layer_sizes=(15,))
mlp.fit(x_train,y_train)
y_pred=mlp.predict(x_test)
ma=accuracy_score(y_test,y_pred)
mp=precision_score(y_test,y_pred)
mr=recall_score(y_test,y_pred)
mf=f1_score(y_test,y_pred)
tn, fp, fn, tp = confusion_matrix(y_test,y_pred).ravel()
print(classification_report(y_test,y_pred,target_names=target_names,digits=4))
print("Sensitivity:",sensitivity(tn,fp,fn,tp))
print("Specificity:",specificity(tn,fp,fn,tp))
import numpy as np
import matplotlib.pyplot as plt
model_acc=[sa,ka,ea,aa,ga,ra,ma]
model_pre=[sp,kp,ep,ap,gp,rp,mp]
model_rec=[sr,kr,er,ar,gr,rr,mr]
model_f1=[sf,kf,ef,af,gf,rf,mf]
names=['Support Vector','KNearest','extraTree','AdaBoost','GradientBoost','RandomForest','DeepFeedForward']
#fig = plt.figure(figsize = (10, 5))

```

```

plt.bar(names,model,width = 0.4)
barWidth = 0.2
fig = plt.subplots(figsize =(15, 8))
br1 = np.arange(len(model_acc))
br2 = [x + barWidth for x in br1]
br3 = [x + barWidth for x in br2]
br4 = [x + barWidth for x in br3]
plt.bar(br1, model_acc, color ='r', width = barWidth,edgecolor ='grey', label ='Accuracy',Line
width=1)
plt.bar(br2, model_pre, color ='g', width = barWidth,edgecolor ='grey', label ='precision',Line
width=1)
plt.bar(br3, model_rec, color ='b', width = barWidth,edgecolor ='grey', label ='recall',Linewid
th=1)
plt.bar(br4, model_f1, color ='y', width = barWidth,edgecolor ='grey', label ='f1score',Linewi
dth=1)
plt.xlabel('models', fontweight ='bold', fontsize = 15)
plt.ylabel('values', fontweight ='bold', fontsize = 15)
plt.xticks([r + barWidth for r in range(len(model_acc))], ['SupportVector','KNearest','extraTre
e','AdaBoost','GradientBoost','RandomForest',' DeepFeedForward'])
plt.legend()
plt.show()
"""plt.xlabel("models")
plt.ylabel("Accuracy")
plt.title("Accuracy comparsion of different models")
plt.show()"""
import pandas as pd
from matplotlib import pyplot as plt
plt.rcParams["figure.figsize"] = [7.50, 3.50]
plt.rcParams["figure.autolayout"] = True
d = {'model_acc': [i for i in range(10)],
      'model_pre': [i * i for i in range(10)],
      'model_rec': [i * i for i in range(10)],
      'model_f1': [i * i for i in range(10)]
    }

```

```

df = pd.DataFrame(d)
df.plot(kind='bar', edgecolor='white', linewidth=1)
plt.legend(loc="upper left")
plt.show()

data1['Result'] = [0 if result in ['Pass', 'Distinction'] else 1 for result in data1['final_result']]
feature = data.drop(['final_result', 'Result'], axis = 1)
target = data['Result']
target.value_counts()
target_name=["Pass","Fail"]
x_train, x_test, y_train, y_test = train_test_split(feature, target, test_size = 0.2, random_state =
123, stratify=data.final_result)
#SVC
model_svm=SVC()
model_svm.fit(x_train, y_train)
y_pred = model_svm.predict(x_test)
tn, fp, fn, tp = confusion_matrix(y_test, y_pred).ravel()
print(classification_report(y_test, y_pred, target_names=target_name, digits=4))
print("Sensitivity:", sensitivity(tn, fp, fn, tp))
print("Specificity:", specificity(tn, fp, fn, tp))
#KNN
model_knn=KNN()
model_knn.fit(x_train, y_train)
y_pred = model_knn.predict(x_test)
tn, fp, fn, tp = confusion_matrix(y_test, y_pred).ravel()
print(classification_report(y_test, y_pred, target_names=target_name, digits=4))
print("Sensitivity:", sensitivity(tn, fp, fn, tp))
print("Specificity:", specificity(tn, fp, fn, tp))
#ETC
model_etc=ETC()
model_etc.fit(x_train, y_train)
y_pred = model_etc.predict(x_test)
tn, fp, fn, tp = confusion_matrix(y_test, y_pred).ravel()
print(classification_report(y_test, y_pred, target_names=target_name, digits=4))

```



```

print("Sensitivity:",sensitivity(tn,fp,fn,tp))
print("Specificity:",specificity(tn,fp,fn,tp))
#ABC
model_abc=ABC()
model_abc.fit(x_train,y_train)
y_pred = model_abc.predict(x_test)
tn, fp, fn, tp = confusion_matrix(y_test,y_pred).ravel()
print(classification_report(y_test,y_pred,target_names=target_name,digits=4))
print("Sensitivity:",sensitivity(tn,fp,fn,tp))
print("Specificity:",specificity(tn,fp,fn,tp))
#GBC
model_gbc=GBC()
model_gbc.fit(x_train,y_train)
y_pred = model_gbc.predict(x_test)
tn, fp, fn, tp = confusion_matrix(y_test,y_pred).ravel()
print(classification_report(y_test,y_pred,target_names=target_name,digits=4))
print("Sensitivity:",sensitivity(tn,fp,fn,tp))
print("Specificity:",specificity(tn,fp,fn,tp))

#RFC
model_rfc=RFC()
model_rfc.fit(x_train,y_train)
y_pred = model_rfc.predict(x_test)
tn, fp, fn, tp = confusion_matrix(y_test,y_pred).ravel()
print(classification_report(y_test,y_pred,target_names=target_name,digits=4))
print("Sensitivity:",sensitivity(tn,fp,fn,tp))
print("Specificity:",specificity(tn,fp,fn,tp))
#MLP
mlp = MLPClassifier(random_state=1,hidden_layer_sizes=(15,))
mlp.fit(x_train,y_train)
y_pred=mlp.predict(x_test)
tn, fp, fn, tp = confusion_matrix(y_test,y_pred).ravel()
print(classification_report(y_test,y_pred,target_names=target_name,digits=4))
print("Sensitivity:",sensitivity(tn,fp,fn,tp))

```

```
print("Specificity:",specificity(tn,fp,fn,tp))
```

```
#Save MI model
```

```
import joblib
```

```
joblib.dump(model_rfc,"studentInfo.pkl")
```

```
model_rfc.fit(x_train,y_train)
```

```
model=joblib.load("studentInfo.pkl")
```

3.9.1 Model.py

```
from flask import Flask, render_template, request
```

```
import numpy as np
```

```
import pickle
```

```
app = Flask(__name__,template_folder='templates')
```

```
model = pickle.load(open("model.pkl", "rb"))
```

```
@app.route('/')
```

```
def home():
```

```
    return render_template('upload.html')
```

```
@app.route("/", methods=["POST"])
```

```
def predict():
```

```
    if request.method == "POST":
```

```
        float_features = [x for x in request.form.values()]
```

```
        print(float_features)
```

```
        float_features.pop(2)
```

```
        float_features.pop(5)
```

```
        c = list(map(int,float_features))
```

```
        features=[np.array(c)]
```

```
        print(features)
```

```
        prediction = model.predict(features)
```

```
        classes = ['Distinction', 'Fail', 'Pass', 'Withdrawn']
```

```
        return render_template('predict.html',prediction_text=classes[prediction[0]])
```

```
if __name__ == "__main__":
```

```
    app.run(debug=True)
```

```
home()
```

3.9.2 Upload.html

```
<html>
<head>
<title>Student Performance Prediction</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>jQuery UI Dialog - Default functionality</title>
<link rel="stylesheet" href="//code.jquery.com/ui/1.13.2/themes/base/jquery-ui.css">
<link rel="stylesheet" href="/resources/demos/style.css">
<script src="https://code.jquery.com/jquery-3.6.0.js"></script>
<script src="https://code.jquery.com/ui/1.13.2/jquery-ui.js"></script>
<script>
window.onerror = function (e) {
console.log('Error: ', e);
};
</script>
<script>
typeof({ {prediction_text_} })
$( function() {
$( "#dialog" ).dialog();
})
</script>
</head>
<style>
body{
background-image:
url("https://tse3.mm.bing.net/th?id=OIP.aeM9vp76Iw4XRywEuSShBAHaDs&pid=Api&P=0
");
background-size: cover;
background-position: center;
background-attachment: fixed;
background-repeat: no-repeat;
```

```

}
.heading{
    color:#ff0066;
    font-size: 40px;
    text-align: center;
}
.form-dropdown field{
    font-size: 60px;
}
.table{
    background-color:#e6ffff;
}
</style>
<body>
<h1 class="heading">STUDENT PERFORMANCE PREDICTION</h1>
<table class="table " align="center" cellpadding = "15" >
<form action="/" method="POST">
<tr>
<div>
<td>Code_Module</td>
<td>
<select class="form-dropdown field" name="code">
    <option value=0> AAA </option>
    <option value=1> BBB</option>
    <option value=2> CCC</option>
    <option value=3> DDD</option>
    <option value=4> EEE</option>
    <option value=5> FFF</option>
    <option value=6> GGG</option>
</select>
</td>
</div>
<tr>
<div>

```

```

<td>Code_Presentation</td>
<td>
<select class="form-dropdown field" name="presentation">
    <option value=1> 2013J </option>
    <option value=3> 2014J</option>
    <option value=0> 2013B</option>
    <option value=2> 2014B</option>
</select>
</td>
</div>
</tr>
<tr>
<td>id_student</td>
<td><input type="text" name="id_student" placeholder="11391" />
</td>
</tr>
<tr>
<td>Gender</td>
<td>
    <input type="radio" name="Gender" value=1 /> Male
    <input type="radio" name="Gender" value=0 /> Female
</td> </tr>
<tr>
<td>Region</td>
<td>
<select class="form-dropdown field" name="region">
    <option value=" "></option>
    <option value=0>East Anglian Region</option>
    <option value=1>East Midlands Region</option>
    <option value=2>Ireland</option>
    <option value=3>London Region</option>
    <option value=4>North Region </option>
    <option value=5>North Western Region</option>
    <option value=6>Scotland</option>

```

```

        <option value=7>South East Region</option>
        <option value=8>South Region</option>
        <option value=9>South West Region</option>
        <option value=10>Wales</option>
        <option value=11>West Midlands Region</option>
        <option value=12>Yorkshire Region</option>
    </select>
</td>
</tr>
<tr>
<td>Higher Education</td>
<td>
<select class="form-dropdown field" name="edu">
    <option value=0> A Level or Equivalent </option>
    <option value=1> HE Qualification </option>
    <option value=2>Lower Than A Level </option>
    <option value=3>No Formal Equals </option>
    <option value=4>Post Graduate Qualification </option>
</select>
</td>
</tr>
<tr>
<td>imd_band</td>
<td>
<select class="form-dropdown field" name="band">
    <option value="0-10%">0-10%</option>
    <option value="20-30%">20-30%</option>
    <option value="30-40%">30-40%</option>
    <option value="40-50%">40-50%</option>
    <option value="50-60%">50-60%</option>
    <option value="60-70%">60-70%</option>
    <option value="70-80%">70-80%</option>
    <option value="80-90%">80-90%</option>
    <option value="90-100%">90-100%</option>

```

```

</select>
</td>
</tr>
<tr>
<td>Age</td>
<td>
<select class="form-dropdown field" name="age">
    <option value=0>0-35</option>
    <option value=1>35-55</option>
    <option value=2>55<=</option>
</select>
</td>
</tr>
<tr>
<td>Number of previous attempts</td>
<td><input type="text" name="Number of attempts" placeholder="0" />
</td>
</tr>
<tr>
<td>Number of credits</td>
<td><input type="text" name="Number of Credits" placeholder="0" />
</td>
</tr>
<tr>
<td>Disability</td>
<td>
<input type="radio" name="Disability" value=1 />Y
<input type="radio" name="Disability" value=0 />N
</td> </tr>
<tr>
<div class="input-field">
<td>
<button type="submit" class="btn-block btn-large">PREDICT</button>
</td>

```

```

</div>
</form>
</div>
</table>
</body>
</html>

```

3.9.3 Predict.html

```

<html>
<style>
  body{
    background-image: url("https://wallpapercave.com/wp/wp4780150.jpg");
    background-size: cover;
    background-position: center;
    background-attachment: fixed;
    background-repeat: no-repeat;
  }

  .heading{
    color:black;
    font-size: 49px;
    text-align: center;
  }
</style>
<body>
<div id="dialog" title="Basic dialog">
  <br>
  <br>
  <br><br><br><br><br><br><br><br><br><br>
  <h1 class="heading"> Student's Performance: { { prediction_text_ } }</h1>

</div>
</body>
</html>

```


3.10 Result Analysis

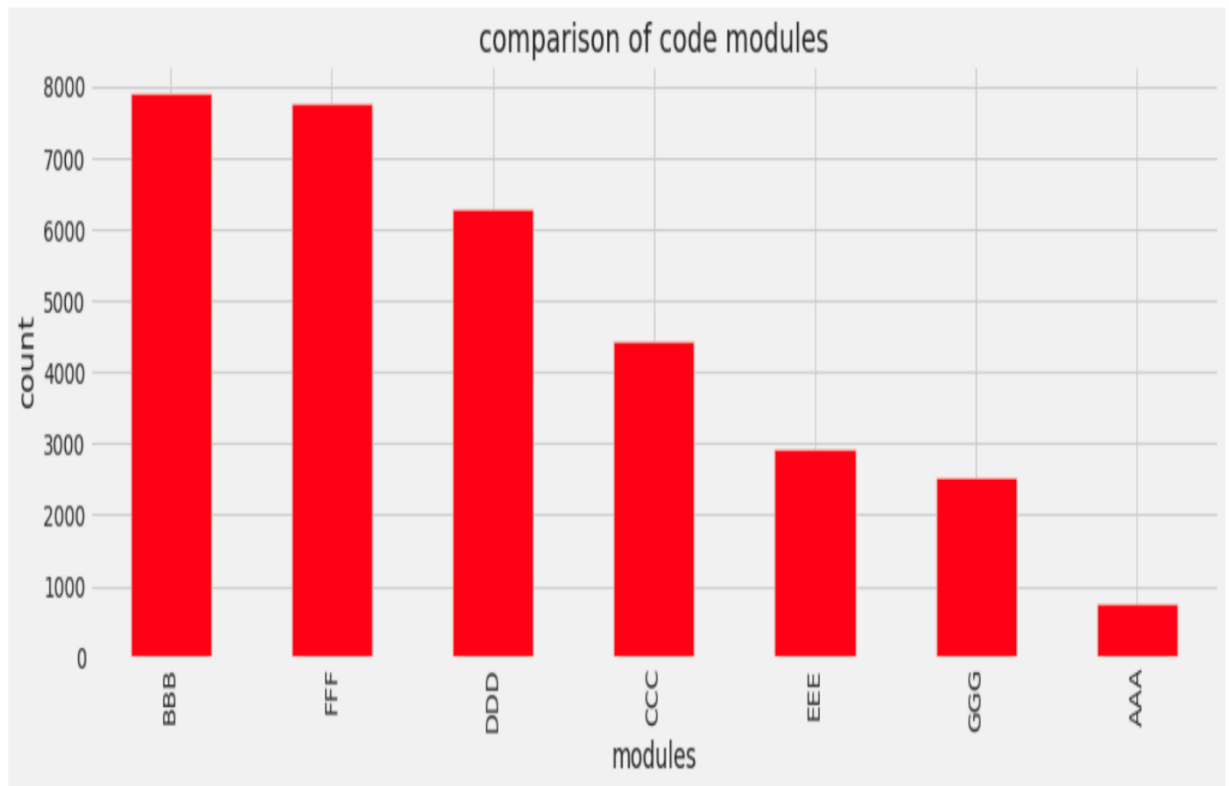


Fig 3.10.1 Comparison of Code modules

The above graph specifies about the comparison of code modules. Based on this graph we can predict the highly popular module. Based on the graph we predict the students are very much interested towards the BBB, FFF modules.

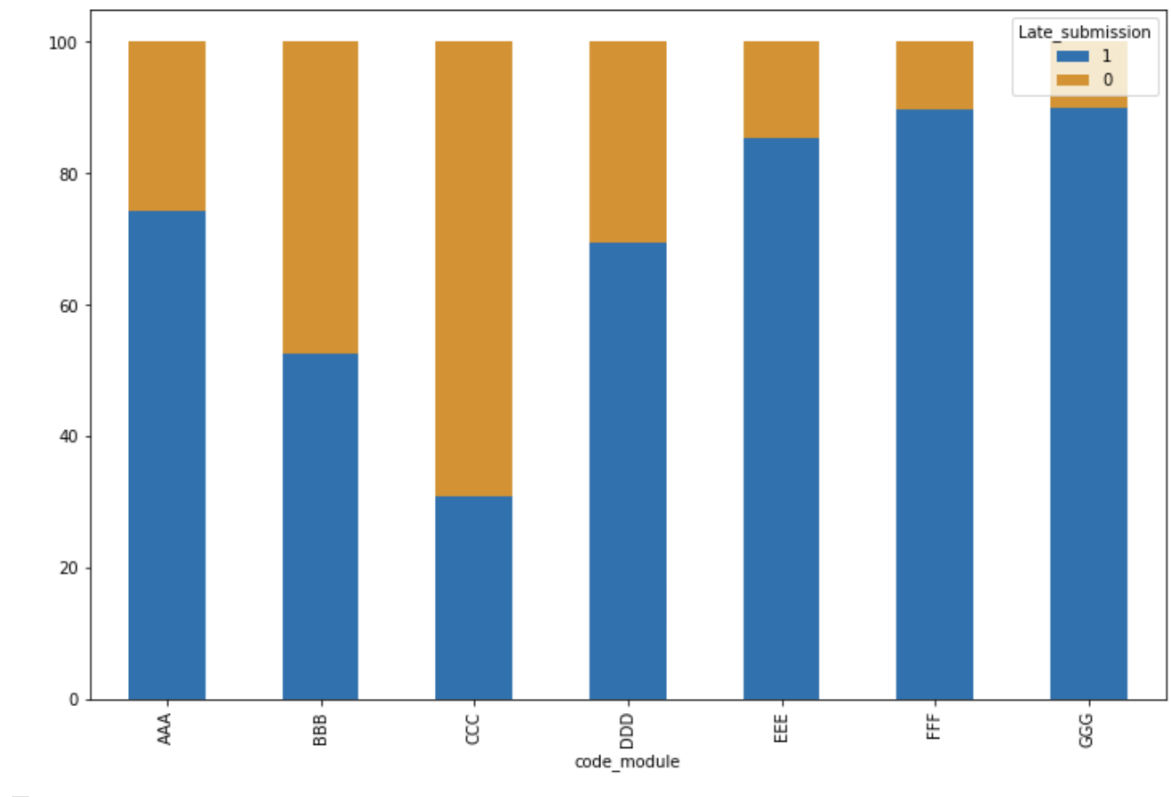


Fig 3.10.2 Late Submissions in courses

The above graph specifies about the late submissions in the courses. In this graph “0” representing the late submission of assessment and “1” representing the submission of courses by students. Most of the students in FFF and GGG are submitted their assessment.

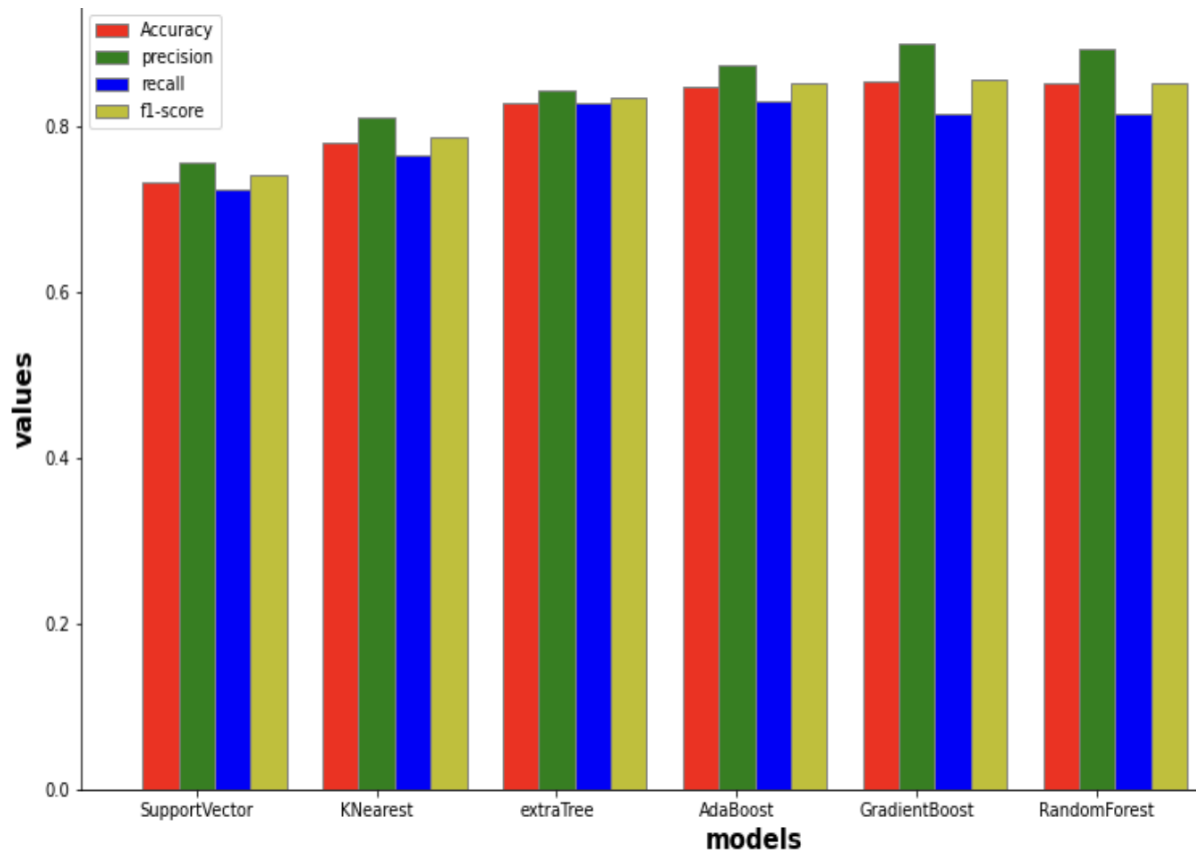
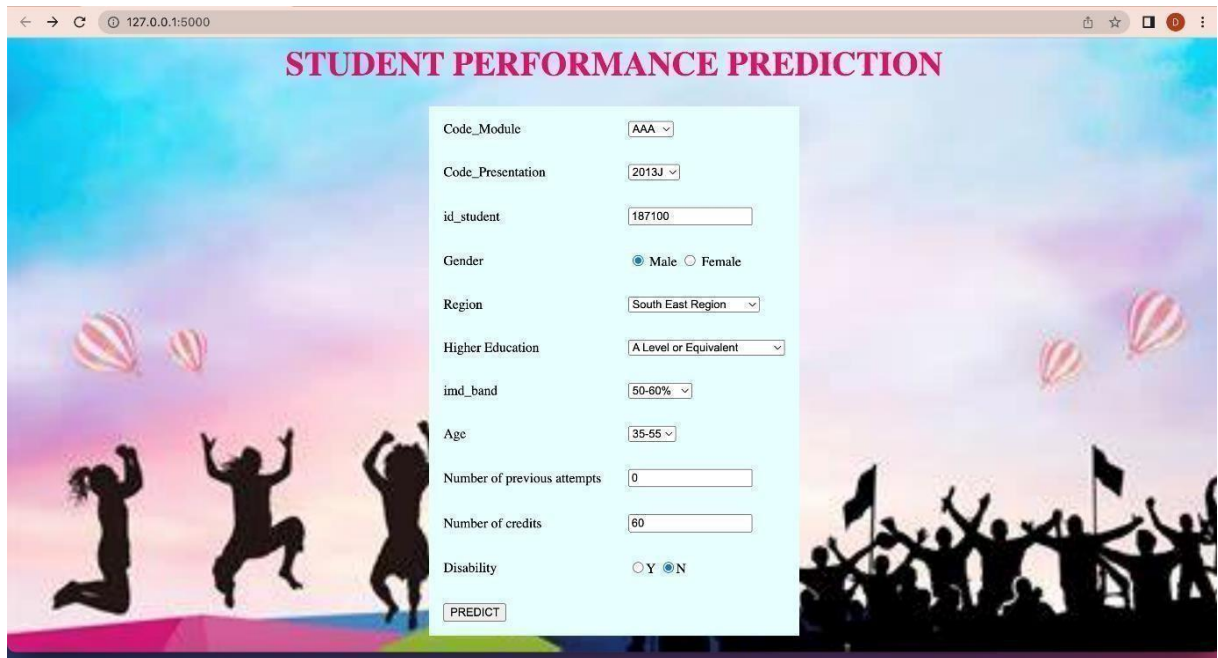


Fig 3.10.3 Comparision of models

The above graph specifies the comparison between modules and their accuracy. After the performance of 6 models (Random Forest Classifier, Support Vector Machine, Extra Tree Classifier, Knearest Classifier, Adaboost Classifier) Accuracy, Precision, Recall, F1score are predicted. Based on the prediction graphs are plotted.

4. OUTPUT SCREENS



The screenshot displays a web browser window with the address bar showing "127.0.0.1:5000". The page title is "STUDENT PERFORMANCE PREDICTION". The background features a colorful sky with hot air balloons and silhouettes of people jumping and celebrating. A central form contains the following fields and options:

- Code_Module: AAA (dropdown)
- Code_Presentation: 2013J (dropdown)
- id_student: 187100 (text input)
- Gender: ☒ Male ☐ Female
- Region: South East Region (dropdown)
- Higher Education: A Level or Equivalent (dropdown)
- imd_band: 50-60% (dropdown)
- Age: 35-55 (dropdown)
- Number of previous attempts: 0 (text input)
- Number of credits: 60 (text input)
- Disability: ☐ Y ☒ N
- Predict button

Fig-4.1 Data Uploading Screen



Fig 4.2 Prediction of Distinction Screen

STUDENT PERFORMANCE PREDICTION

Code_Module	AAA
Code_Presentation	2013J
id_student	65002
Gender	<input type="radio"/> Male <input checked="" type="radio"/> Female
Region	East Anglian Region
Higher Education	A Level or Equivalent
imd_band	70-80%
Age	0-35
Number of previous attempts	0
Number of credits	60
Disability	<input type="radio"/> Y <input checked="" type="radio"/> N

PREDICT

Fig 4.3 Data Uploading Screen



Fig 4.4 Prediction of Pass Screen

STUDENT PERFORMANCE PREDICTION

Code_Module	AAA
Code_Presentation	2013J
id_student	444677
Gender	<input checked="" type="radio"/> Male <input type="radio"/> Female
Region	London Region
Higher Education	HE Qualification
imd_band	30-40%
Age	0-35
Number of previous attempts	0
Number of credits	60
Disability	<input type="radio"/> Y <input checked="" type="radio"/> N
<input type="button" value="PREDICT"/>	

Fig 4.5 Data Uploading Screen 3

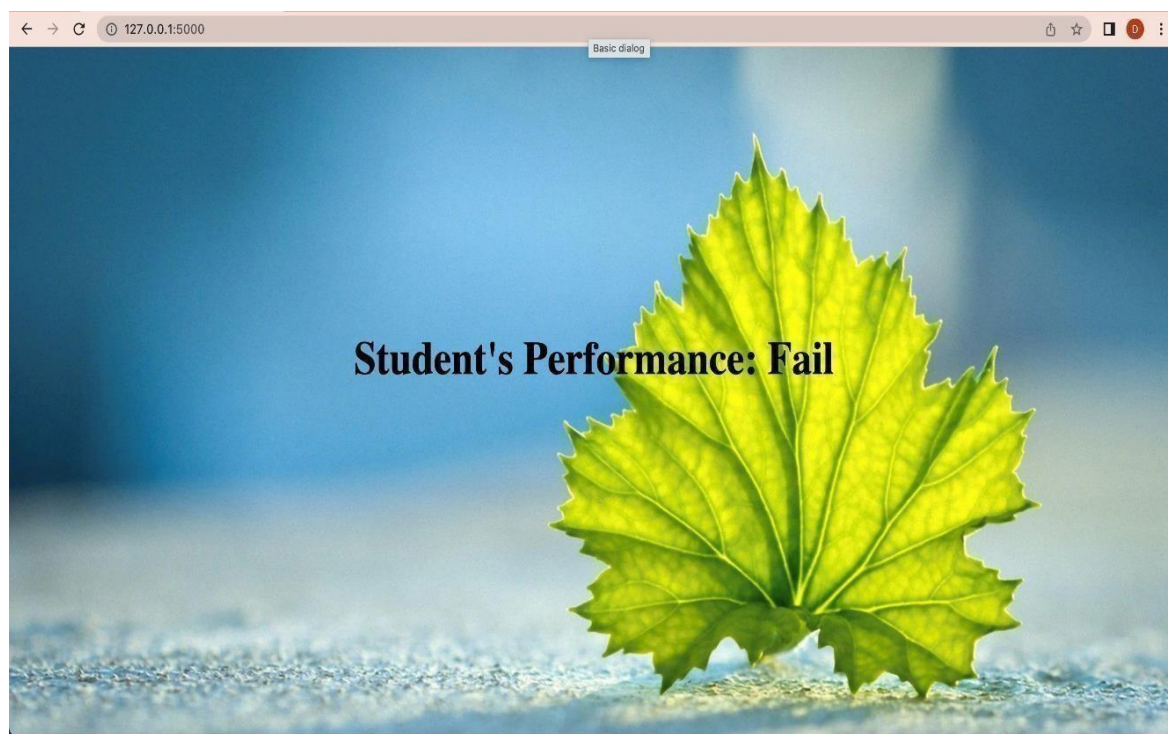


Fig 4.6 Prediction of Fail Screen

STUDENT PERFORMANCE PREDICTION

Code_Module: AAA

Code_Presentation: 2013J

id_student: 65002

Gender: ☐ Male ☒ Female

Region: East Anglian Region

Higher Education: A Level or Equivalent

imd_band: 70-80%

Age: 0-35

Number of previous attempts: 0

Number of credits: 60

Disability: ☐ Y ☒ N

PREDICT

Fig 4.7 Data Uploading Screen 4



Fig 4.8 Prediction of Withdrawn Screen

5. CONCLUSION

Conclusion

The purpose of the study was the prediction of risk of students at different course lengths by machine learning and deep learning algorithms. The study and employed four classification metrics for evaluations. The study showed that there was a significant improvement in the performance of models when they used the clickstream data and assignment scores. Random Forest predictive model (85%) with the highest performance scores selected for predicting students' performance. Out of many variables clickstream data and assessment scores have the most significant impact on the final result.

6. FUTURE WORK

Future Work

The prediction accuracy of machine learning models can be increased by using various other algorithms like Decision Tree, Random Forest, Adaboost, XGBoost, SVC, NaïveBayes, KNN, Logistic Regression and ANN and two deep learning methods (RNN and LSTM). This can be made into mobile application so that students can track their performance and improve their scores.

7. BIBLIOGRAPHY

- [1]A. Behr, M. Giese, and K. Theune, “Early prediction of university dropouts—A random forest approach,” *J. Nat. Stat.*, vol. 1, pp. 743–789, Feb. 2020.
- [2] A. Ortigosa, R. M. Carro, J. Bravo-Agapito, D. Lizcano, J. J. Alcolea, and O. Blanco, “From lab to production: Lessons learnt and real-life challenges of an early student-dropout prevention system,” *IEEE Trans. Learn. Technol.*, vol. 12, no. 2, pp. 264–277, Apr. 2019.
- [3]A. S. Imran, F. Dalipi, and Z. Kastrati, “Predicting student dropout in a MOOC: An evaluation of a deep neural network model,” in *Proc. 5th Int. Conf. Comput. Artif. Intell. (ICCAI)*, 2019, pp. 190–195.
- [4] A. A. Mubarak, H. Cao, and S. A. M. Ahmed, “Predictive learning analytics using deep learning model in MOOCs’ courses videos,” *Edu. Inf. Technol.*, vol. 6, pp. 1–22, Jul. 2020.
- [5] B. Sekeroglu, K. Dimililer, and K. Tuncal, “Student performance prediction and classification using machine learning algorithms,” in *Proc. 8th Int. Conf. Educ. Inf. Technol.*, Mar. 2019, pp. 7–11.
- [6] C. C. Gray and D. Perkins, “Utilizing early engagement and machine learning to predict student outcomes,” *Comput. Edu.*, vol. 131, pp. 22–32, Apr. 2019.
- [7]C. P. Rosé, E. A. McLaughlin, R. Liu, and K. R. Koedinger, “Explanatory learner models: Why machine learning (alone) is not the answer,” *Brit. J. Educ. Technol.*, vol. 50, no. 6, pp. 2943–2958, 2019.
- [8]H. L. Fwa and L. Marshall, “Modeling engagement of programming students using unsupervised machine learning technique,” *GSTF J. Comput.*, vol. 6, no. 1, pp. 1–6, 2018. [9]J. Y. Chung and S. Lee, “Dropout early warning systems for high school students using machine learning,” *Children Youth Services Rev.*, vol. 96, pp. 346–353, Jan. 2019.
- [10] J. Figueroa-Cañas and T. Sancho-Vinuesa, “Predicting early dropout student is a matter of checking completed quizzes: The case of an online statistics module,” in *Proc. LASI-SPAIN*, 2019, pp. 100–111.
- [11]J. Xu, K. H. Moon, and M. van der Schaar, “A machine learning approach for tracking and predicting student performance in degree programs,” *IEEE J. Sel. Topics Signal Process.* [12]L. C. B. Martins, R. N. Carvalho, R. S. Carvalho, M. C. Victorino, and M. Holanda, “Early prediction of college attrition using data mining,” in *Proc. 16th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2017, pp. 1075–1078.

- [13] M. Hussain, W. Zhu, W. Zhang, S. M. R. Abidi, and S. Ali, "Using machine learning to predict student difficulties from learning session data," *Artif. Intell. Rev.*, vol. 52, no. 1, pp. 381–407, Jun. 2019.
- [14] N. Wu, L. Zhang, Y. Gao, M. Zhang, X. Sun, and J. Feng, "CLMS-net: Dropout prediction in MOOCs with deep learning," in *Proc. ACM Turing Celebration Conf.*, May 2019, pp. 1–6
- [15] N. Mduma, K. Kalegele, and D. Machuve, "Machine learning approach for reducing students dropout rates," *Int. J. Adv. Comput. Res.*, vol. 9, no. 42, 2019, doi: 10.19101/IJACR.2018.83904
- [16] R. Al-Shabandar, A. J. Hussain, P. Liatsis, and R. Keight, "Detecting atrisk students with early interventions using machine learning techniques," *IEEE Access*, vol. 7, pp. 149464–149478, 2019.
- [17] S. Lee and J. Y. Chung, "The machine learning-based dropout early warning system for improving the performance of dropout prediction," *Appl. Sci.*, vol. 9, no. 15, p. 3093, Jul. 2019.
- [18] S. N. Liao, D. Zingaro, K. Thai, C. Alvarado, W. G. Griswold, and L. Porter, "A robust machine learning technique to predict lowperforming students," *ACM Trans. Comput. Edu.*, vol. 19, no. 3, pp. 1–19, Jun. 2019.
- [19] Y. Mao, et al., "Deep learning vs. Bayesian knowledge tracing: Student models for interventions," *J. Educ. Data Mining*, vol. 10, no. 2, pp. 1–27, 2018.
- [20] Z. Iqbal, J. Qadir, A. Noor Mian, and F. Kamiran, "Machine learning based student grade prediction: A case study," 2017, arXiv:1708.08744. [Online]. Available: <http://arxiv.org/abs/1708.08744>

Prediction of Student Performance on Virtual Platform Using Machine Learning Algorithm

D.Vamsika¹, Y.Anjani Priya², Sk.Sameena³, R.Chaitra⁴, S V N Sreenivasu⁵

^{1,2,3,4}Student, Department of CSE, Narasaraopeta Engineering College, Narasaraopeta, Guntur (D.T) A.P, India

⁵Professor, Department of CSE, Narasaraopeta Engineering College, Narasaraopeta, Guntur (D.T) A.P, India

vamsika.danda@gmail.com¹, anjaniPriya93@gmail.com², sameenashaik2801@gmail.com³, regalagaddachaitra@gmail.com⁴, drsvnsrinivasu@gmail.com⁵

1. ABSTRACT- Predictive analysis is a machine learning analytical technique that is the focus of this research. The problem of reliable performance prediction is addressed by a number of online learning systems, including number of Courses and number of learning platforms. We are recommending this work by the contrasting techniques like the regression and the classification, which are useful for the prediction modelling to obtain the most accurate outcomes. The prediction model is trained the data and tested the data by using random forest and different models to explain the learning behaviour of the students in connection with their study factors. These predictive model was trained with random forest and it has the higher accuracy.

2. KEYWORDS: Supervised learning, Predictive analysis, Performance prediction, Machine learning, and feature selection.

3. INTRODUCTION

Machine learning ML is the superset of the deep learning, while Artificial intelligence is the superset of ML. ML is helpful in model creation as data is fed to the machine, employing algorithms for additional training and testing on those enormous data so that the machine can conduct operations on its own when given fresh data. There are 3 different categories:

Supervised learning: Throughout the machine's learning phase, supervision is needed. It contains both the input and the desired output, and the model is set up to forecast the desired outcome. Example: Classification and Regression.

Unsupervised learning: The model learns by itself by identifying the pattern inside the dataset without any supervision. Only input is provided, the model self-trains, and output results. Ex. Clustering and Association.

Because it best fits the needs of predictive analysis, we adopted the supervised learning approach in this study. Prediction is carried out as historical data is gathered, and the model is trained to handle fresh input and forecast the de

This paper's main goal is to use machine learning approaches to identify characteristics related with students' learning development and how they engaged with the virtual learning environment in order to identify students who are at danger of dropping out as early as feasible.

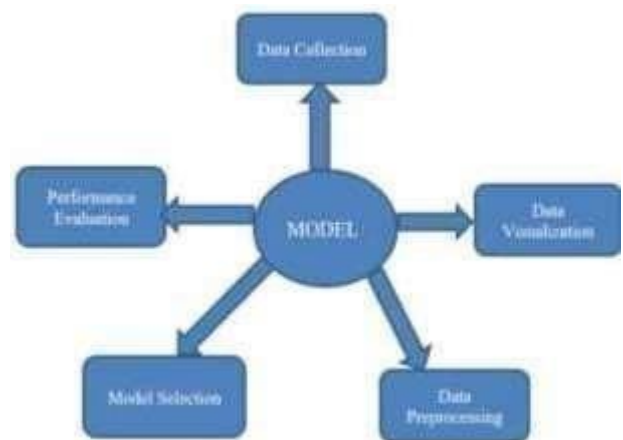


Fig.3(a) 5 Steps involved in the Model

Machine learning can assist students in making judgements about whether or not to continue in the course when it comes to early risk prediction of people. These algorithms had capability of effectively self-preparation and can also deal with the early risk of prediction for students to pass their courses.

4. LITERATURE SURVEY

We must take into account demographic information, a crucial component for early intervention, in order to anticipate children at an earlier stage. Assessment results and Clickstream data are significant time-dependent variables in addition to Demographic data. With an accuracy rate of 85%, it uses a random forest model to identify at-risk students in online courses and provides them with intense early assistance.

5. PROPOSED SYSTEM

5.1 Pre-processing

Pre-processing refers to the modifications done to our data before we give it to the algorithm. Data Pre-processing is a technique used to create clean data collections from raw data. In other words, if data are gathered from various sources, they are gathered in an unprocessed way that prevents analysis. A data collection should be structured to allow for the usage of multiple Machine learning algorithms in order to achieve the best results.

5.1.1 Dataset

We used the datasets from the Open University of learning analytics Dataset, which the University makes available. Seven tables include structured data on students. It includes information on demographics (student information and registration), student involvement learning multiple number of courses in the platforms. The student VLE table stores submitted work of the student.

	code	modcode	presid	student	gender	region	highest	second	band	age	band	num	of	student	or	disability	final	result
1	AAA	20131	11391	M		East Anglia	HE Qualifs	30-39%	55+	0	240	N					Pass	
2	AAA	20131	28400	F		Scotland	HE Qualifs	20-30%	35-55	0	60	N					Pass	
3	AAA	20131	30268	F		North West	A Level or	30-40%	35-55	0	60	Y					Withdrawn	
4	AAA	20131	31604	F		South East	A Level or	50-60%	35-55	0	60	N					Pass	
5	AAA	20131	32085	F		West Midlands	Lower Th	50-60%	0-35	0	60	N					Pass	
6	AAA	20131	38053	M		Wales	A Level or	30-40%	35-55	0	60	N					Pass	
7	AAA	20131	45462	M		Scotland	HE Qualifs	30-40%	0-35	0	60	N					Pass	
8	AAA	20131	45642	F		North West	A Level or	90-100%	0-35	0	120	N					Pass	
9	AAA	20131	52130	F		East Anglia	A Level or	70-80%	0-35	0	90	N					Pass	

Fig 5.1.1 Dataset

5.1.2 Dataset Description

1. Code Presentation term represents the semester of the course for which student joined.

1. The student id identifies the each student by using distinct numbers.
2. Gender specifies the student gender
3. Region specifies location in the region student living.
4. Highest education specifies that student completed before taking the course.
5. The Index of Multiple Deprivation band (IMD band) indicates that the percentile students with an IMD band of below of 20 are from the most poor areas.
6. Age specifies age of the Student.
7. The number of priorities refers to the number of students who have already attempted the chosen course and their credit totals. The total number of credits a student is enrolled in during the course at the Open University.
8. Disability Student alleges a logical disability of any kind.
9. Final result specifies the final result of the student in the course

5.1.3 Dataset Preprocessing and Merging

Effective preprocessing and the choice of the best classifier have a significant impact on the outcome and precision of the machine learning-based approach. A data mining approach called data preprocessing purifies raw data and produces more pertinent data. Raw data is full of outliers, unneeded features, and many missing values. The required data for the prediction are merged and performed preprocessing on the required dataset.

```
studentInfo_df.isnull().sum()
code_modcode 0
course_presentation 0
id_student 0
gender 0
region 0
highest_education 0
ind_band 0
age_band 0
num_of_prior_attempts 0
studied_credits 0
disability 0
final_result 0
dtype: object

[ ] studentInfo_df["ind_band"] = studentInfo_df["ind_band"].fillna(studentInfo_df["ind_band"].mode()[0])
studentInfo_df["ind_band"] = studentInfo_df["ind_band"].fillna(studentInfo_df["ind_band"].mode()[0])

studentInfo_df.isnull().sum()
code_modcode 0
course_presentation 0
id_student 0
gender 0
region 0
highest_education 0
ind_band 0
age_band 0
num_of_prior_attempts 0
studied_credits 0
disability 0
final_result 0
dtype: object
```

Fig 5.1.3 Replacing the Missing Values in the dataset

5.1.4 Outlier analysis

Since there are many noisy data, cleaning the data is crucial for improved model performance. In this instance, the Turkey's method is employed as a

traditional method of eliminating outliers. We use box plots to show the outliers. Many insightful conclusions may be drawn from these plots, including the fact that each of the data sets contains a significant number of outliers and extreme values.

	id_student	num_of_prev_attempts	studied_credits
count	3.258300e+04	32583.000000	32583.000000
mean	7.066877e+05	0.163225	79.758691
std	5.491673e+05	0.479758	41.071900
min	3.733000e+03	0.000000	30.000000
25%	5.085730e+05	0.000000	80.000000
50%	5.903100e+05	0.000000	80.000000
75%	6.444530e+05	0.000000	120.000000
max	2.716795e+06	6.000000	655.000000

Fig 5.1.4 Outlier Analysis

5.1.5 Splitting the Set of data



Fig 5.1.5 Spitting of data

The data set is split at a ratio of 20:80. In this scenario, 20% of the data is used for testing, and 80% is used for training.

5.2 Correlation

The correlation matrix is helpful in showing the variables' strong correlation. The matrix shows that because they all have an impact on the prediction variable, each independent variable is significant for it.



Fig 5.2 Correlation Heat Map

5.3 Feature Engineering

It is crucial to draw some conclusions from the available data and create data sets that contain more insightful information. As the dataset already contains the course id and student information, additional calculations are made to the registration, click sums, before click, after click, and score that may affect the student's ultimate result.

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32593 entries, 0 to 32592
Data columns (total 12 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   code_module                           32593 non-null  object
1   code_presentation                     32593 non-null  object
2   id_student                           32593 non-null  int64
3   gender                               32583 non-null  object
4   region                               32593 non-null  object
5   highest_education                     32593 non-null  object
6   imd_band                             31482 non-null  object
7   age_band                             32593 non-null  object
8   num_of_prev_attempts                  32593 non-null  int64
9   studied_credits                       32593 non-null  int64
10  disability                            32593 non-null  object
11  final_result                          32593 non-null  object
dtypes: int64(3), object(9)
memory usage: 3.0+ MB

```

Fig 5.3 Feature engineering

5.4 Model Selection

The ultimate accuracy of the model is estimated once it has been tested using test data. Machine learning models were trained and evaluated using various dataset splits to produce accurate findings. There are four steps that training and testing might take

PHASE-1:DEMOGRAPHIC DATA

PHASE-2:DEMOGRAPHIC and CLICKSTREAM DATA

PHASE-3:DEMOGRAPHIC, CLICKS TRACK,and EVALUATION

FEATURE ENGINEERING:

Merging the various classes, such as Distinction- Pass being combined into the Pass class and Withdrawn- Fail being combined into the Fail class, can help the performance outcomes even more. beat all other models in performance.

5.4.1 Random Forest

A random forest model is created using a large number of decision trees. The programmes are essentially creates a forest out of the forecasted results of trees. The algorithm also employs three random ideas: randomly choosing training data when creating trees, randomly selecting specific subsets of variables when dividing nodes, and using only a small fraction of all variables to split each node in each basic decision tree.

5.5 Data Visualisation

Using various graphs, charts, plots, and other visual aids, data visualization makes it easier to understand the data.

The below graph specifies about the comparison of code modules. Based on this graph we can predict the highly popular module. Based on the graph we predict the students are very much interested towards the BBB ,FFFmodules.

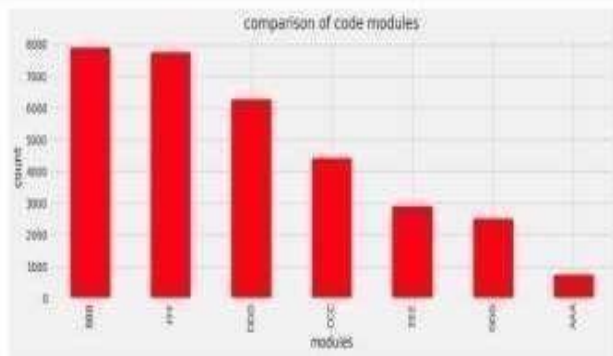


Fig 5.5.1(a) Comparison of Code modules

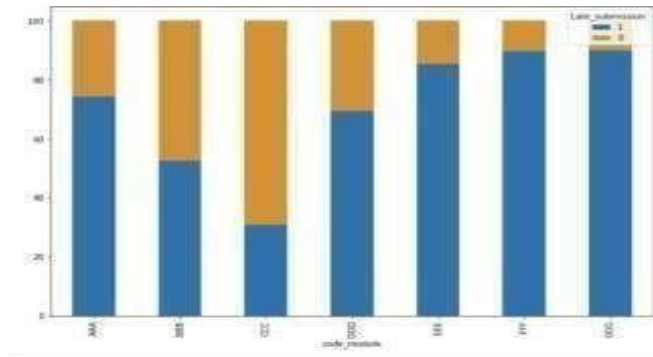


Fig 5.5.1(b) Late submission of assessments

The above graph specifies about the late submissions in the courses. In this graph “0” representing the late submission of assessment and “1” representing the submission of courses by students. Most of the students in FFF andGGG are submitted their assessment

5.6 Proposed Model

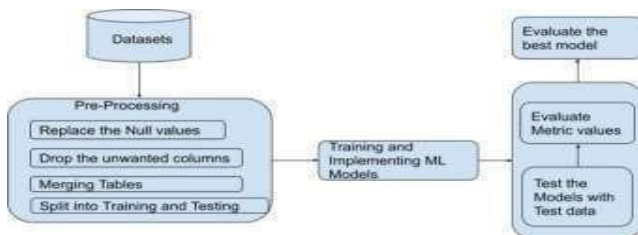


Fig 5.6 Proposed Model Architecture

For the earliest possible prediction of student’s performance, divide the course length into 20%,40%,60%,80% and 100% of course completed.

6. RESULT ANALYSIS

The primary objective of this project is the earliest possible identifying students who may be at risk dropouts by leveraging Machine Learning techniqueto understand variables associated with the learning behavior of students and how they

Models	Accuracy
Random Forest	85.15%
Support Vector Machine	79%
Extra Tree Classifier	78%
Adaboost Classifier	75%
Gradient Boost	85%

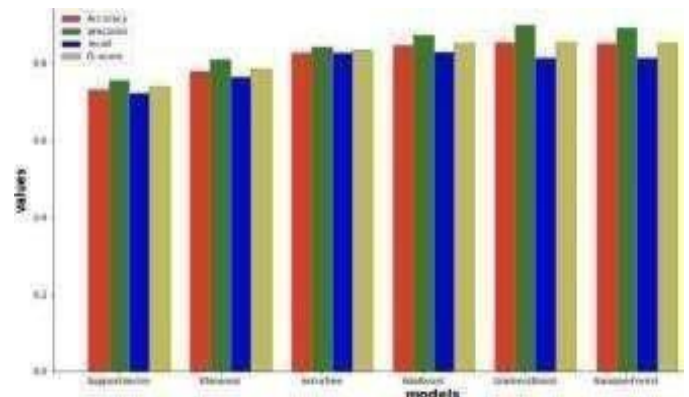


Fig 6(a) Comparison of Models

Virtual Learning Environment. so ,we consider Random Forest as our model with highest accuracy 85.15%.

The above graph specifies the comparison between modules and their accuracy.After the performance of 6 models. Accuracy, Precision,Recall, F1score are predicted.Based on the prediction graphs are plotted.

7. OUTPUT SCREENS



Fig 7(a) Data Uploading Screen



Fig 7(b) Output Screen

8. CONCLUSION

The main goal of the project to use the Machine learning algorithm to estimate student risk over a range of course lengths. Four classification metrics were used in the study and for evaluations. The research revealed that using clickstream data and assignment ratings significantly improved the models performance. The best performing Random Forest predictive model (85%) was chosen to forecast student performance. Clickstream data and assessment scores have the biggest influence on the outcome of all the variables.

9. REFERENCES

- [1] A. Oritgosa , R. M. Carro, J. Bravo-Agapito, D. Lizcano, J. J. Alcolea, and O. Blanco, "From lab to production: Lessons learnt and real-life challenges of an early student-dropout prevention system," *IEEE Trans. Learn. Technol.*, vol. 12, no. 2, pp. 264–277, Apr. 2019.
- [2] A. S. Imran, F. Dalipi, and Z. Kastrati, "Predicting student dropout in a MOOC: An evaluation of a deep neural network model," in *Proc. 5th Int. Conf. Comput. Artif. Intell. (ICCAI)*, 2019, pp. 190–195.
- [3] A. A. Mubarak, H. Cao, and S. A. M. Ahmed, "Predictive learning analytics using deep learning model in MOOCs' courses videos, " *Edu. Inf. Technol.*, vol. 6, pp. 1–22, Jul. 2020.
- [3] B. Sekeroglu, K. Dimililer, and K. Tuncal, "Student performance prediction and classification using machine learning algorithms," in *Proc. 8th Int. Conf. Educ. Inf. Technol.*, Mar. 2019, pp. 7–11.
- [4] A. Behr, M. Giese, and K. Theune, "Early Prediction of University dropouts-A random forest approach," *J. Nat. stat.*, vol. 1, pp. 743–789, Feb. 2020.
- [5] J. Figueroa-Cañas and T. Sancho-Vinuesa, "Predicting early dropout student is a matter of checking completed quizzes: The case of an online statistics module," in *Proc. LASI- SPAIN*, 2019, pp. 100–111.
- [6] J. Xu, K. H. Moon, and M. van der Schaar, "A machine learning approach for tracking and predicting student performance in degree programs," *IEEE J. Sel. Topics Signal Process.*
- [7] S. Lee and J. Y. Chung, "The machine learning-based dropout early warning system for improving the performance of dropout prediction," *Appl. Sci.*, vol. 9, no. 15, p. 3093, Jul. 2019.

6%

SIMILARITY INDEX

1%

INTERNET SOURCES

3%

PUBLICATIONS

4%

STUDENT PAPERS

PRIMARY SOURCES

1

Muhammad Adnan, Asad Habib, Jawad Ashraf, Shafaq Mussadiq, Arsalan Ali Raza, Muhammad Abid, Maryam Bashir, Sana Ullah Khan. "Predicting at-risk students at different percentages of course length for early intervention using machine learning models", IEEE Access, 2021

Publication

2%

2

sciencescholar.us

Internet Source

1%

3

Submitted to The University of Memphis

Student Paper

1%

4

Submitted to University of Wales Institute, Cardiff

Student Paper

1%

5

Submitted to Coventry University

Student Paper

1%

6

Submitted to University of Finance and Economics

Student Paper

1%



Mojtaba Nabipour, Pooyan Nayyeri, Hamed Jabani, Shahab S., Amir Mosavi. "Predicting Stock Market Trends Using Machine Learning and Deep Learning Algorithms Via Continuous and Binary Data; a Comparative Analysis", IEEE Access, 2020

Publication

<1 %

Exclude quotes On

Exclude matches Off

Exclude bibliography On



NARASARAOPETA
ENGINEERING COLLEGE
(AUTONOMOUS)



Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website: www.nrtec.in

International Conference on

PAPER ID
NECICAIEA2K23050

Artificial Intelligence and Its Emerging Areas

NEC-ICAIEA-2K23

17th & 18th March, 2023

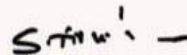
Organized by Department of Computer Science and Engineering in Association with CSI

Certificate of Presentation

This is to Certify that **Danda Vamsika**, **Narasaraopeta Engineering College** has presented the paper title **Prediction of Student Performance on Virtual Platform Using Machine Learning Algorithm** in the International Conference on Artificial Intelligence and Its Emerging Areas-2K23 [NEC-ICAIEA-2K23], Organized by Department of **Computer Science and Engineering** in Association with **CSI** on 17th and 18th March 2023 at **Narasaraopeta Engineering College, Narasaraopet, A.P., India.**



Convenor
Dr. S. V. N. Srinivasu



Chief-Convenor
Dr. S. N. Tirumala Rao



Principal, Patron
Dr. M. Sreenivasa Kumar





**NARASARAOPETA
ENGINEERING COLLEGE**
(AUTONOMOUS)



Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website: www.nrtec.in

International Conference on

PAPER ID
NECICAIEA2K23050

Artificial Intelligence and Its Emerging Areas

NEC-ICAIEA-2K23

17th & 18th March, 2023

Organized by Department of Computer Science and Engineering in Association with CSI

Certificate of Presentation

This is to Certify that **Yadlapalli Anjani Priya**, **Narasaraopeta Engineering College** has presented the paper title **Prediction of Student Performance on Virtual Platform Using Machine Learning Algorithm** in the International Conference on Artificial Intelligence and Its Emerging Areas-2K23 [NEC-ICAIEA-2K23], Organized by Department of **Computer Science and Engineering in Association with CSI** on 17th and 18th March 2023 at **Narasaraopeta Engineering College, Narasaraopet, A.P., India.**

Convenor
Dr. S. V. N. Srinivasu

Chief-Convenor
Dr. S. N. Tirumala Rao

Principal, Patron
Dr. M. Sreenivasa Kumar





NARASARAOPETA
ENGINEERING COLLEGE
(AUTONOMOUS)



Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website: www.nrtec.in

International Conference on

Artificial Intelligence and Its Emerging Areas

NEC-ICAIEA-2K23

17th & 18th March, 2023

Organized by Department of Computer Science and Engineering in Association with CSI

Certificate of Presentation

This is to Certify that **Shaik Sameena**, **Narasaraopeta Engineering College** has presented the paper title **Prediction of Student Performance on Virtual Platform Using Machine Learning Algorithm** in the International Conference on Artificial Intelligence and Its Emerging Areas-2K23 [NEC-ICAIEA-2K23], Organized by Department of **Computer Science and Engineering** in Association with **CSI** on 17th and 18th March 2023 at **Narasaraopeta Engineering College, Narasaraopet, A.P., India.**

Convenor
Dr. S. V. N. Srinivasu

Chief-Convenor
Dr. S. N. Tirumala Rao

Principal, Patron
Dr. M. Sreenivasa Kumar





NARASARAOPETA
ENGINEERING COLLEGE
(AUTONOMOUS)



Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website: www.nrtec.in

International Conference on

Artificial Intelligence and Its Emerging Areas

NEC-ICAIEA-2K23

17th & 18th March, 2023

Organized by Department of Computer Science and Engineering in Association with CSI

Certificate of Presentation

This is to Certify that **Regalagadda Chaitra**, **Narasaraopeta Engineering College** has presented the paper title **Prediction of Student Performance on Virtual Platform Using Machine Learning Algorithm** in the International Conference on Artificial Intelligence and Its Emerging Areas-2K23 [NEC-ICAIEA-2K23], Organized by Department of **Computer Science and Engineering** in Association with **CSI** on 17th and 18th March 2023 at **Narasaraopeta Engineering College, Narasaraopet, A.P., India.**

Convenor
Dr.S.V.N.Srinivasu

Chief-Convenor
Dr.S.N.Tirumala Rao

Principal, Patron
Dr. M. Sreenivasa Kumar



Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website: www.nrtec.in

PAPER ID
NECICAIEA2K23050

Artificial Intelligence and Its Emerging Areas

NEC-ICAIEA-2K23

17th & 18th March, 2023

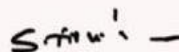
Organized by Department of Computer Science and Engineering in Association with CSI

Certificate of Presentation

This is to Certify that **Dr. S V N Sreenivasu**, **Narasaraopeta Engineering College** has presented the paper title **Prediction of Student Performance on Virtual Platform Using Machine Learning Algorithm** in the International Conference on Artificial Intelligence and Its Emerging Areas-2K23 [NEC-ICAIEA-2K23], Organized by Department of **Computer Science and Engineering in Association with CSI** on 17th and 18th March 2023 at **Narasaraopeta Engineering College, Narasaraopet, A.P., India.**



Convenor
Dr. S.V.N. Srinivasu



Chief-Convenor
Dr. S.N. Tirumala Rao



Principal, Patron
Dr. M. Sreenivasa Kumar

