**Predicting Student Success in Online Learning Environments using Machine Learning**

**INTRODUCTION**

Online education has revolutionized the way people learn and has made education more accessible and affordable to numerous people worldwide. Despite the advantages and increased interest in online and distance learning, educational institutions are becoming increasingly concerned about students’ performance and retention rates, particularly low certification/graduation and dropout/completion rates. Failing or dropping out of an online course or program is often considered a key parameter by institutional authorities for assessing program/course quality and allocating resources. Dropout and low certification rates can also pose a potential risk to an institution’s reputation, profit, and funding. These outcomes also have significant consequences for a student’s self-esteem, well-being, employment, and chances of graduating. As a result, finding more efficient approaches to forecasting students’ performance as early as possible is critical for institutions, students, and educators to take proactive steps toward improving students’ online learning experiences and establishing intervention strategies that target students’ needs. With the increased interest in online education and the large amount of data produced by learners through their interactions with online platforms, researchers have proposed methods to analyse learners’ behavioural data to predict and improve educational outcomes. Learning analytics (LA), more commonly known as educational data mining (EDM), the task of analysing and finding patterns in learners’ data for decision-making purposes, has attracted many researchers in recent years. Learning analytic tools enable institutions to gain an understanding of their students’ status, actions and preferences individually, and in relation to their peers and the targeted educational objective. This allows the tailoring of material for individual students based on the projected outcomes and preferred learning styles. In online education, LA systems assess students’ learning behaviour by utilizing extensive data collection of learners’ data, including student enrolment information, past and current academic records, students’ online behaviour, student surveys via questionnaires concerning courses and teaching techniques, and data from online discussion forums. Scholars have also examined various learning-behaviour attributes to predict learning outcomes, such as learners’ performance and retention. To predict and analyse students’ outcomes in online courses, researchers have examined several machine learning models, including support vector machines (SVMs), linear regression (LR), random forest (RF), and deep learning models such as convolutional neural networks (CNNs) and long short-term memory (LSTM).

**1.1 Objective of the project:**

Recent years have witnessed an increased interest in online education, both massive open online courses (MOOCs) and small private online courses (SPOCs). This significant interest in online education has raised many challenges related to student engagement, performance, and retention assessments. With the increased demands and challenges in online education, several researchers have investigated ways to predict student outcomes, such as performance and dropout in online courses. This paper presents a comprehensive review of state-of-the-art studies that examine online learners’ data to predict their outcomes using machine and deep learning techniques. The contribution of this study is to identify and categorize the features of online courses used for learners’ outcome prediction, determine the prediction outputs, determine the strategies and feature extraction methodologies used to predict the outcomes, describe the metrics used for evaluation, provide a taxonomy to analyse related studies, and provide a summary of the challenges and limitations in the field.

**Literature Survey:**

**“Some studies in machine learning using the game of checkers”**

Samuel A.L. 1959

Two machine-learning procedures have been investigated in some detail usi!Jg the game of checkers. Enough work has been done to verify the fact that a computer can be programmed so that it will learn to play better game of checkers than can be played by the person who wrote the program. Furthermore, it can learn to do this in a remarkably short period of time (8 or 10 hours of machine-playing time) when given only the rules of the game, a sense of direction, and a redundant and incomplete list of parameters which are thought to have something to do with the game, but whose correct signs and relative weights are unknown and unspecified. The principles of machine learning verified by these 'experiments are, of course, applicable to many other situations.

**“Machine learning”**

Mitchell T. 1997.

Today, all institutions and companies are accelerating the use of AI technologies in their businesses to achieve a clear vision and quality results. The education sector is one of the sectors where AI can be used because of big data. In this work we created a machine-based learning model to predict a student's educational performance. The developed model relied on the student's previous data and performance in the last stage of the school. The model showed a very accurate accuracy rate that can be adopted.

**"Use of machine learning techniques for educational proposes: a decision support system for forecasting students’ grades."**

Kersaints, Sotiris B., 2012

The computer programming course has always been considered a difficult course to get started, especially the teaching quality of computer programming courses in higher vocational schools is relatively poor, which has been a problem and challenge for schools and teachers. This paper designs a teaching evaluation method of computer programming course in higher vocational colleges based on big data. Firstly, the paper analyses the application of big data in the computer programming course of higher vocational college, and confirms the promoting effect of big data on the teaching of computer programming course of higher vocational college. Based on the evaluation results, the author puts forward some strategies to improve the teaching quality. The research results provide valuable reference for the teaching of computer programming courses and the application of big data in higher vocational schools.

**"Preventing student dropout in distance learning using machine learning techniques."**

Kersaints, Sotiris B., C. J. Pierrakeas, and Panayiotis E. Pintelas. 2003.

Student dropout occurs quite often in universities providing distance education. The scope of this research is to study whether the usage of machine learning techniques can be useful in dealing with this problem. Subsequently, an attempt was made to identifying the most appropriate learning algorithm for the prediction of students’ dropout. A number of experiments have taken place with data provided by the ‘informatics’ course of the Hellenic Open University and a quite interesting conclusion is that the Naive Bayes algorithm can be successfully used. A prototype web based support tool, which can automatically recognize students with high probability of dropout, has been constructed by implementing this algorithm.

**“An empirical study of three machine learning methods for spam filtering”**

Lai C.-C. 2007.

The increasing volumes of unsolicited bulk e-mail (also known as spam) are bringing more annoyance for most Internet users. Using a classifier based on a specific machine-learning technique to automatically filter out spam e-mail has drawn many researchers’ attention. This paper is a comparative study the performance of three commonly used machine learning methods in spam filtering. On the other hand, we try to integrate two spam filtering methods to obtain better performance. A set of systematic experiments has been conducted with these methods which are applied to different parts of an e-mail. Experiments show that using the header only can achieve satisfactory performance, and the idea of integrating disparate methods is a promising way to fight spam.

**"Mining Moodle data to detect the inactive and low-performance students during the Moodle course"**

M Hussain, S Hussain, W Zhang, W Zhu, P Theodorou, SMR Abidi, 2018

In web-based learning systems such as massive open online course (MOOC) and modular object-oriented developmental learning environment (Moodle), monitoring the student's activities as well as predict the low-performance students is an important task because it enables the instructors to award the students when their activities level drops from normal activities levels as well as having lower grades. We used several machine learning (ML) classification and clustering techniques to extract the pattern from student data during completing the Moodle course; which enables the instructor to detect the low-performance student in advance before the examination. The experimental result shows that the fuzzy unordered rule induction algorithm (FURIA) classification technique achieves high accuracy in detecting inactive students as well as predicts the different categories of the student during the Moodle course. The K-means clustering is also able to group the inactive and active users and poorly performed users. The result demonstrates that our proposed system will be easily integrated to Moodle system to send alert to inactive and low- performance students while completing the course and build efficient education environment for the students.

**Introduction to Machine Learning**

Dietterich T. 2010

The goal of machine learning is to program computers to use example data or past experience to solve a given problem. Many successful applications of machine learning exist already, including systems that analyze past sales data to predict customer behavior, optimize robot behavior so that a task can be completed using minimum resources, and extract knowledge from bioinformatics data. The second edition of Introduction to Machine Learning is a comprehensive textbook on the subject, covering a broad array of topics not usually included in introductory machine learning texts. In order to present a unified treatment of machine learning problems and solutions, it discusses many methods from different fields, including statistics, pattern recognition, neural networks, artificial intelligence, signal processing, control, and data mining. All learning algorithms are explained so that the student can easily move from the equations in the book to a computer program. The text covers such topics as supervised learning, Bayesian decision theory, parametric methods, multivariate methods, multilayer perceptrons, local models, hidden Markov models, assessing and comparing classification algorithms, and reinforcement learning. New to the second edition are chapters on kernel machines, graphical models, and Bayesian estimation; expanded coverage of statistical tests in a chapter on design and analysis of machine learning experiments; case studies available on the Web (with downloadable results for instructors); and many additional exercises. All chapters have been revised and updated. Introduction to Machine Learning can be used by advanced undergraduates and graduate students who have completed courses in computer programming, probability, calculus, and linear algebra. It will also be of interest to engineers in the field who are concerned with the application of machine learning methods.

**“SPOC-MFLP: A multi-feature learning prediction model for SPOC students using machine learning”**

C Yu, 2018

Learning analysis is one of the most important applications of machine learning. Many studies have proposed solutions to learning performance prediction using online learning data. Unlike the previous studies, this paper analyses online learning environment and formalizes the problem of online learning prediction. Based on the formalization, a multi-feature based learning prediction model for SPOC is proposed, called SPOC-MFLP, which generalizes the prediction problem of SPOC learning including objective, constraints, system and algorithms. The proposed SPOC-MFLP could be extended for MOOC and other online learning forms. Principle components analysis is adopted to discover the correlations of students ‘online multi features, and linear regression and deep neural network are used to predict the learning performance. The predicted results include specific scores or segmented grades of the final exam of SPOC, as well as students ‘future specialized courses. Experimental data are collected from a SPOC in Huzhou University for two years and the experimental results reveal that the proposed SPOC-MFLP performs well.

**"Predicting student achievement based on temporal learning behaviour in MOOCs"**

S Qu, K Li, B Wu, S Zhang, Y Wang, 2019

With the development of data mining technology, educational data mining (EDM) has gained increasing amounts of attention. Research on massive open online courses (MOOCs) is an important area of EDM. Previous studies found that assignment-related behaviours in MOOCs (such as the completed number of assignments) can affect student achievement. However, these methods cannot fully reflect students’ learning processes and affect the accuracy of prediction. In the present paper, we consider the temporal learning behaviours of students to propose a student achievement prediction method for MOOCs. First, a multi-layer long short-term memory (LSTM) neural network is employed to reflect students’ learning processes. Second, a discriminative sequential pattern (DSP) mining-based pattern adapter is proposed to obtain the behaviour patterns of students and enhance the significance of critical information. Third, a framework is constructed with an attention mechanism that includes data pre-processing, pattern adaptation, and the LSTM neural network to predict student achievement. In the experiments, we collect data from a C programming course from the year 2012 and extract assignment-related features. The experimental results reveal that this method achieves an accuracy rate of 91% and a recall of 94%.

**“A semi-supervised regression algorithm for grade prediction of students in distance learning courses”**

G Kostopoulos, S Kersaints, N Fazekas, G Katsuhiko’s, C Peiraeus, 2019

Applying data mining methods in the educational field has gained a lot of attention among researchers in recent years. Educational Data Mining has turned into an effective tool for uncovering hidden relationships in educational data and predicting students’ learning outcomes. Several supervised methods have been successfully applied with the purpose of identifying students at risk of failing or of predicting their academic performance. Recently, the implementation of Semi-Supervised Learning (SSL) methods in the educational process indicated their superiority over the supervised ones. SSL is an emerging subfield of machine learning seeking to effectively exploit a small pool of labeled examples together with a large pool of unlabelled ones. On this basis, a small number of students’ data from previous years may be used as the training set of a learning model to predict future outcomes of current students. A number of rewarding studies deal with the implementation of classification methods in the educational field in contrast to regression, which is deemed to be a slightly touched task. In this paper, a novel semi-supervised regression (SSR) algorithm is presented for predicting the final grade of undergraduate students in a distance online course. To the best of our knowledge there is no study dealing with the implementation of SSR methods in the educational field. A plethora of attributes related to students’ characteristics, academic performance and interaction within the course online platform form the training set, while several experiments were carried out confirming the superiority of the proposed algorithm over familiar regression methods. The experiment results show that the predictive performance of the proposed algorithm is increasing significantly over time, achieving a MAE value of less than 1.2358 before the middle of the academic year, which provides the advantage of early warnings and interventions.

**3. SYSTEM ANALYSIS**

**3.1 EXISTING SYSTEM**

In Existing system, using machine learning algorithms like Naïve Bayes, Logistic Regression to predict student success in online learning environments.

**Disadvantages:**

1. Less Accuracy
2. More time taking process

**3.2 PROPOSED SYSTEM**

In this proposed system, we are employing OULAD dataset to predict grade and score using various machine learning algorithms like Random Forest and gradient boosting. Each algorithm performance is evaluated using accuracy, precision, recall and FSCORE.

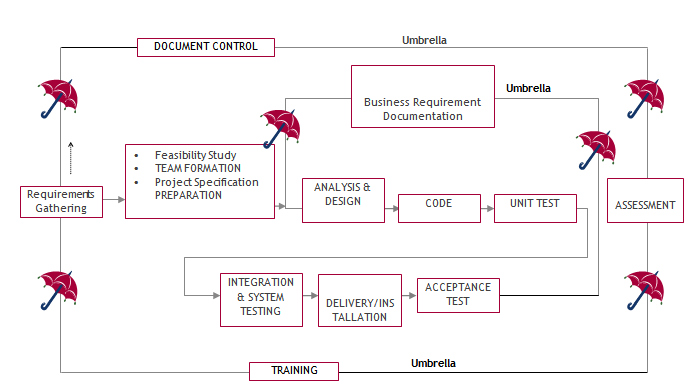
Random Forest classifier is used to predict grade and Random Forest Regressor is used to predict score. Similarly Gradient Boosting classifier to predict Grade and Gradient Boosting Regressor to predict score. Regressor algorithm performance is evaluated using RMSE (root mean square error). RMSE refers to difference between true value and predicted value so the lower the difference the better is the value.

**Advantages:**

1. High Accuracy
2. Takes less time

**3.3. PROCESS MODEL USED WITH JUSTIFICATION**

**SDLC (Umbrella Model):**

****

SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

**Stages in SDLC:**

* Requirement Gathering
* Analysis
* Designing
* Coding
* Testing
* Maintenance

**Requirements Gathering stage:**

The requirements gathering process takes as its input the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports. A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirement title and textual description.



These requirements are fully described in the primary deliverables for this stage: the Requirements Document and the Requirements Traceability Matrix (RTM). The requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are *not* included in the requirements document.

The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product components developed during each stage of the software development lifecycle are formally connected to the components developed in prior stages.

In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term requirements traceability.

The outputs of the requirements definition stage include the requirements document, the RTM, and an updated project plan.

* Feasibility study is all about identification of problems in a project.
* No. of staff required to handle a project is represented as Team Formation, in this case only modules are individual tasks will be assigned to employees who are working for that project.
* Project Specifications are all about representing of various possible inputs submitting to the server and corresponding outputs along with reports maintained by administrator.

**Analysis Stage:**

The planning stage establishes a bird's eye view of the intended software product, and uses this to establish the basic project structure, evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches.



The most critical section of the project plan is a listing of high-level product requirements, also referred to as goals. All of the software product requirements to be developed during the requirements definition stage flow from one or more of these goals. The minimum information for each goal consists of a title and textual description, although additional information and references to external documents may be included. The outputs of the project planning stage are the configuration management plan, the quality assurance plan, and the project plan and schedule, with a detailed listing of scheduled activities for the upcoming Requirements stage, and high level estimates of effort for the out stages.

**Designing Stage:**

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.

  
When the design document is finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement. The outputs of the design stage are the design document, an updated RTM, and an updated project plan.

**Development (Coding) Stage:**

The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artifacts will be produced. Software artifacts include but are not limited to menus, dialogs, and data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artifacts, and an online help system will be developed to guide users in their interactions with the software.



The RTM will be updated to show that each developed artifact is linked to a specific design element, and that each developed artifact has one or more corresponding test case items. At this point, the RTM is in its final configuration. The outputs of the development stage include a fully functional set of software that satisfies the requirements and design elements previously documented, an online help system that describes the operation of the software, an implementation map that identifies the primary code entry points for all major system functions, a test plan that describes the test cases to be used to validate the correctness and completeness of the software, an updated RTM, and an updated project plan.

**Integration & Test Stage:**

During the integration and test stage, the software artifacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability. During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Production Initiation Plan.



The outputs of the integration and test stage include an integrated set of software, an online help system, an implementation map, a production initiation plan that describes reference data and production users, an acceptance plan which contains the final suite of test cases, and an updated project plan.

* **Installation & Acceptance Test:**

During the installation and acceptance stage, the software artifacts, online help, and initial production data are loaded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer.

After customer personnel have verified that the initial production data load is correct and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software.



The primary outputs of the installation and acceptance stage include a production application, a completed acceptance test suite, and a memorandum of customer acceptance of the software. Finally, the PDR enters the last of the actual labor data into the project schedule and locks the project as a permanent project record. At this point the PDR "locks" the project by archiving all software items, the implementation map, the source code, and the documentation for future reference.

**Maintenance:**

Outer rectangle represents maintenance of a project, Maintenance team will start with requirement study, understanding of documentation later employees will be assigned work and they will undergo training on that particular assigned category. For this life cycle there is no end, it will be continued so on like an umbrella (no ending point to umbrella sticks).

**3.4. Software Requirement Specification**

**3.4.1. Overall Description**

A Software Requirements Specification (SRS) – a requirements specification for a software system is a complete description of the behaviour of a system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. Non-functional requirements are requirements which impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

System requirements specification: A structured collection of information that embodies the requirements of a system. A business analyst, sometimes titled system analyst, is responsible for analysing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Within the systems development lifecycle domain, the BA typically performs a liaison function between the business side of an enterprise and the information technology department or external service providers. Projects are subject to three sorts of requirements:

* Business requirements describe in business terms what must be delivered or accomplished to provide value.
* Product requirements describe properties of a system or product (which could be one of several ways to accomplish a set of business requirements.)
* Process requirements describe activities performed by the developing organization. For instance, process requirements could specify .Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:
* **ECONOMIC FEASIBILITY**

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economic feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs. The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, there is nominal expenditure and economical feasibility for certain.

* **Operational Feasibility**

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits. The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

* **TECHNICAL FEASIBILITY**

Earlier no system existed to cater to the needs of ‘Secure Infrastructure Implementation System’. The current system developed is technically feasible. It is a web-based user interface for audit workflow at NIC-CSD. Thus, it provides an easy access to. the users. The database’s purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security.

**3.4.2. External Interface Requirements**

**User Interface**

The user interface of this system is a user friendly python Graphical User Interface.

**Hardware Interfaces**

The interaction between the user and the console is achieved through python capabilities.

**Software Interfaces**

The required software is python.

**SYSTEM REQUIREMENT:**

**HARDWARE REQUIREMENTS:**

# Processor - Intel i3(min)

* Speed - 1.1 GHz
* RAM - 4GB(min)
* Hard Disk - 500 GB

**SOFTWARE REQUIREMENTS:**

* Operating System - Windows10(min)
* Programming Language - Python 3.7.0

**4. SYSTEM DESIGN**

**UML Diagram:**

The Unified Modelling Language allows the software engineer to express an analysis model using the modelling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

* + **User Model View**
    1. This view represents the system from the users perspective.
    2. The analysis representation describes a usage scenario from the end-users perspective.
  + **Structural Model view**
    1. In this model the data and functionality are arrived from inside the system.
    2. This model view models the static structures.
* **Behavioural Model View**

It represents the dynamic of behavioural as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

* **Implementation Model View**

In this the structural and behavioural as parts of the system are represented as they are to be built.

* **Environmental Model View**

In this the structural and behavioural aspects of the environment in which the system is to be implemented are represented.

**Class Diagram:**

The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed. In the diagram, classes are represented with boxes which contain three parts:

* The upper part holds the name of the class
* The middle part contains the attributes of the class
* The bottom part gives the methods or operations the class can take or undertake.



**Use case Diagram:**

A **use case diagram** at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.

****

**Sequence diagram:**

A sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



**Collaboration diagram:**

A collaboration diagram describes interactions among objects in terms of sequenced messages. Collaboration diagrams represent a combination of information taken from class, sequence, and use case diagrams describing both the static structure and dynamic behaviour of a system.



**Component Diagram:**

In the Unified Modelling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems.

Components are wired together by using an assembly connector to connect the required interface of one component with the provided interface of another component. This illustrates the service consumer - service provider relationship between the two components.

****

**Deployment Diagram:**

A **deployment diagram** in the Unified Modeling Language models the *physical* deployment of artifacts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software components ("artifacts") run on each node (e.g., web application, database), and how the different pieces are connected (e.g. JDBC, REST, RMI).

The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have sub nodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers.

****

**Activity Diagram:**

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. It is basically a flow chart to represent the flow form one activity to another

activity. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent

**Upload OULAD Online Student Dataset**

**Preprocess Dataset**

**Run Random Forest**

**Run Gradient Boosting**

**Comparison Graph**

**Predict Grade & Score**

**Data Flow Diagram:**

Data flow diagrams illustrate how data is processed by a system in terms of inputs and outputs. Data flow diagrams can be used to provide a clear representation of any business function. The technique starts with an overall picture of the business and continues by analysing each of the functional areas of interest. This analysis can be carried out in precisely the level of detail required. The technique exploits a method called top-down expansion to conduct the analysis in a targeted way.

As the name suggests, Data Flow Diagram (DFD) is an illustration that explicates the passage of information in a process. A DFD can be easily drawn using simple symbols. Additionally, complicated processes can be easily automated by creating DFDs using easy-to-use, free downloadable diagramming tools. A DFD is a model for constructing and analysing information processes. DFD illustrates the flow of information in a process depending upon the inputs and outputs. A DFD can also be referred to as a Process Model. A DFD demonstrates business or technical process with the support of the outside data saved, plus the data flowing from the process to another and the end results.

User

1.Upload OULAD Online Student Dataset 2.Upload OULAD Online Student Dataset Successfully

3. Preprocess Dataset 4. Dataset Preprocesed

5. Run Random Forest 6. Run Random Forest Successfully

7. Run Gradient Boosting 8. Run Gradient Boosting Successfully

9. Comparison Graph 10. Comparison Graph Displayed

11. Predict Grade & Score 12. Predict Grade & Score Successfully

**5. IMPLEMETATION**

**5.1 PYTHON**

\* One of the most popular languages is Python. Guido van Rossum released this language in 1991. Python is available on the Mac, Windows, and Raspberry Pi operating systems. The syntax of Python is simple and identical to that of English. When compared to Python, it was seen that the other language requires a few extra lines.

\*It is an interpreter-based language because code may be run line by line after it has been written. This implies that rapid prototyping is possible across all platforms. Python is a big language with a free, binary-distributed interpreter standard library.

\* It is inferior to maintenance that is conducted and is straightforward to learn. It is an object-oriented, interpreted programming language. It supports several different programming paradigms in addition to object-oriented programming, including functional and procedural programming.

\* It supports several different programming paradigms in addition to object-oriented programming, including practical and procedural programming. Python is mighty while maintaining a relatively straightforward syntax. Classes, highly dynamic data types, modules, and exceptions are covered. Python can also be utilised by programmes that require programmable interfaces as an external language.

Here are some key features and characteristics of Python:

* Readability: Python emphasizes code readability with its clean and intuitive syntax. It uses indentation and whitespace to structure code blocks, making it easy to understand and maintain.
* Easy to Learn: Python's simplicity and readability make it an excellent choice for beginners. Its straightforward syntax and extensive documentation make it accessible for newcomers to programming.
* Interpreted Language: Python is an interpreted language, meaning that it doesn't need to be compiled before running. The Python interpreter reads and executes the code directly, making the development process faster and more interactive.
* Cross-platform Compatibility: Python is available for major operating systems like Windows, macOS, and Linux. This cross-platform compatibility allows developers to write code once and run it on different platforms without modifications.
* Large Standard Library: Python comes with a vast standard library that provides ready-to-use modules and functions for various tasks. It covers areas such as file I/O, networking, regular expressions, databases, and more, saving developers time and effort.
* Extensible and Modular: Python supports modular programming, enabling developers to organize code into reusable modules and packages. Additionally, Python allows integrating modules written in other languages, such as C or C++, providing flexibility and performance optimizations.
* Wide Range of Libraries and Frameworks: Python has a vibrant ecosystem with numerous third-party libraries and frameworks. These libraries, such as NumPy, pandas, TensorFlow, and Django, extend Python's capabilities for specific domains, making it a powerful tool for diverse applications.
* Object-Oriented: Python supports object-oriented programming (OOP) principles, allowing developers to create and work with classes and objects. OOP provides a structured approach to code organization, promoting code reuse and modularity.
* Dynamic Typing: Python is dynamically typed, meaning variable types are determined at runtime. Developers do not need to declare variable types explicitly, which enhances flexibility and simplifies code writing.

**5.2 Installation**

To install Python on your computer, follow these basic steps:

* Step 1: Visit the Python website Go to the official Python website at <https://www.python.org/>.
* Step 2: Select the operating system Choose the appropriate installer for your operating system. Python supports Windows, macOS, and various Linux distributions. Make sure to select the correct version that matches your operating system.
* Step 3: Check which version of Python is installed; if the 3.7.0 version is not there, uninstall it through the control panel and
* Step 4: Install Python 3.7.0 using Cmd.
* Step 5: Install the all libraries that required to run the project
* Step 6: Run

**5.3 Python Features:**

1. **Easy:** Because Python is a more accessible and straightforward language, Python programming is easier to learn.
2. **Interpreted language:** Python is an interpreted language; therefore it can be used to examine the code line by line and provide results.
3. **Open Source:** Python is a free online programming language since it is open-source.
4. **Portable:** Python is portable because the same code may be used on several computer standard
5. **libraries:** Python offers a sizable library that we may utilize to create applications quickly.
6. **GUI:** It stands for GUI (Graphical User Interface)
7. **Dynamical typed:** Python is a dynamically typed language; therefore the type of the value will be determined at runtime.

**5.4 Python GUI (Tkinter)**

* Python provides a wide range of options for GUI development (Graphical User Interfaces).
* Tkinter, the most widely used GUI technique, is used for all of them.
* The Tk GUI toolkit offered by Python is used with the conventional Python interface.
* Tkinter is the easiest and quickest way to write Python GUI programs.
* Using Tkinter, creating a GUI is simple.
* A part of Python's built-in library is Tkinter. The GUI programs were created.
* Python and Tkinter together give a straightforward and quick way. The Tk GUI toolkit's object-oriented user interface is called Tkinter.

Making a GUI application is easy using Tkinter. Following are the steps:

1) Install the Tkinter module in place.

2) The GUI application Makes the primary window

3) Include one or more of the widgets mentioned above in the GUI application.

4) Set up the main event loop such that it reacts to each user-initiated event.

Although Tkinter is the only GUI framework included in the Python standard library, Python includes a GUI framework. The default library for Python is called Tkinter. Tk is a scripting language often used in designing, testing, and developing GUIs. Tk is a free, open-source widget toolkit that may be used to build GUI applications in a wide range of computer languages.

**5.5 Python IDLE**

* Python IDLE offers a full-fledged file editor, which gives you the ability to write and execute Python programs from within this program. The built-in file editor also includes several features, like code completion and automatic indentation, that will speed up your coding workflow.
* Guido Van Rossum named Python after the British comedy group Monty Python while the name IDLE was chosen to pay tribute to Eric Idle, who was one of the Monty Python's founding members. IDLE comes bundled with the default implementation of the Python language since the 01.5. 2b1 release
* IDLE is used to execute statements similar to Python Shell. IDLE is used to create, modify, and execute Python code. IDLE provides a fully-featured text editor to write Python scripts and provides features like syntax highlighting, auto-completion, and smart indent.
* IDLE has two modes: interactive and script. We wrote our first program, “Hello, World!” in interactive mode. Interactive mode immediately returns the results of commands you enter into the shell. In script mode, you will write a script and then run it.
* The IDE Python IDLE is a good place to start as it helps you become familiar with the way Python works and understand its syntax. This IDE is good to start programming in Python due to its great debugger, but once you are fluent and start developing projects it is necessary to jump to another, more complete IDE.
* Python IDLE (Integrated Development and Learning Environment) is an interactive development environment included with the Python programming language. It provides a convenient way to write, execute, and debug Python code.

When you install Python, IDLE is typically installed along with it. To open IDLE, you can follow these steps:

* Open the command prompt (Windows) or terminal (macOS/Linux).
* Type "idle" and press Enter. Alternatively, you can specify the version with "idle3" or "idle2" for Python 3 or Python 2, respectively.
* Once IDLE is launched, you will see the Python shell, which is an interactive environment where you can type and execute Python code directly.

Here are some features and functionalities provided by Python IDLE:

* Editor: IDLE includes a text editor where you can write your Python code. It offers syntax highlighting, automatic indentation, and code completion to enhance your coding experience.
* Interactive Shell: The Python shell in IDLE allows you to execute Python code interactively. You can type commands, statements, or function calls directly in the shell, and Python will execute them immediately.
* Debugging: IDLE provides basic debugging capabilities to help you find and fix errors in your code. You can set breakpoints, step through code, inspect variables, and track the program's execution.
* Python Help: IDLE provides access to the Python documentation and built-in help. You can access the help menu to find information about Python modules, functions, classes, and more.
* Script Execution: In addition to the interactive shell, IDLE allows you to run Python scripts stored in files. You can write your code in the editor and execute it as a script to see the output or interact with the program.
* Customization: IDLE can be customized to suit your preferences. You can modify settings related to syntax highlighting, indentation, fonts, and more.
* Python IDLE serves as a beginner-friendly development environment and learning tool. It is suitable for writing small scripts, testing code snippets, experimenting with Python features, and learning the language's basics. However, for more advanced development projects, you may consider using other code editors or integrated development environments (IDEs) that provide additional features and better project management capabilities.

**5.6 Libraries**

In Python, libraries (also referred to as modules or packages) are collections of pre-written code that provide additional functionality and tools to extend the capabilities of the Python language. Libraries contain reusable code that developers can leverage to perform specific tasks without having to write everything from scratch.

Python libraries are designed to solve common problems, such as handling data, performing mathematical operations, interacting with databases, working with files, implementing networking protocols, creating graphical user interfaces (GUIs), and much more. They provide ready-to-use functions, classes, and methods that simplify complex operations and save development time.

**Libraries in Python offer various advantages:**

* Code Reusability:
* Efficiency:
* Collaboration
* Domain-Specific Functionality
* To use a Python library, you need to install it first.

There are some libraries following:

* **Pandas:**

Pandas are a Python computer language library for data analysis and manipulation. It offers a specific operation and data format for handling time series and numerical tables. It differs significantly from the release3-clause of the BSD license. It is a well-liked open-source of opinion that is utilized in machine learning and data analysis.

Pandas are a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real-world data analysis in Python. Pandas are a Python library used for working with data sets.

* It has functions for analysing, cleaning, exploring, and manipulating data.
* The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008.
* Pandas allow us to analyse big data and make conclusions based on statistical theories.
* Pandas can clean messy data sets, and make them readable and relevant.

Relevant data is very important in data science. Pandas are a Python library for data analysis. Started by Wes McKinney in 2008 out of a need for a powerful and flexible quantitative analysis tool, pandas have grown into one of the most popular Python libraries. It has an extremely active community of contributors. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals. Its name is a play on the phrase "Python data analysis" itself.

* **NumPy:**

The NumPy Python library for multi-dimensional, big-scale matrices adds a huge number of high-level mathematical functions. It is possible to modify NumPy by utilizing a Python library. Along with line, algebra, and the Fourier transform operations, it also contains several matrices-related functions.

NumPy can be used to perform a wide variety of mathematical operations on arrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices.

* NumPy is a Python library used for working with arrays.
* It also has functions for working in domain of linear algebra, Fourier transform, and matrices.
* NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely.
* NumPy stands for Numerical Python.
* In Python we have lists that serve the purpose of arrays, but they are slow to process.
* NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.
* The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy.
* Arrays are very frequently used in data science, where speed and resources are very important.
* **Matplotlib:**

It is a multi-platform, array-based data visualization framework built to interact with the whole SciPy stack. MATLAB is proposed as an open-source alternative. Matplotlib is a Python extension and a cross-platform toolkit for graphical plotting and visualization.

Matplotlib is a popular Python library for creating static, animated, and interactive visualizations. It provides a flexible and comprehensive set of tools for generating plots, charts, histograms, scatter plots, and more. Matplotlib is widely used in various fields, including data analysis, scientific research, and data visualization.

Here are some key features and functionalities of the Matplotlib library:

* Plotting Functions
* Customization Options
* Multiple Interfaces
* Integration with NumPy and pandas
* Subplots and Figures:
* Saving and Exporting
* **Scikit-learn:**

The most stable and practical machine learning library for Python is scikit-learn. Regression, dimensionality reduction, classification, and clustering are just a few of the helpful tools it provides through the Python interface for statistical modeling and machine learning. It is an essential part of the Python machine learning toolbox used by JP Morgan. It is frequently used in various machine learning applications, including classification and predictive analysis.

Scikit-learn (also referred to as sklearn) is a widely used open-source machine learning library for Python. It provides a comprehensive set of tools and algorithms for various machine learning tasks, including classification, regression, clustering, dimensionality reduction, model selection, and pre-processing.

Here are some key features and functionalities of the Scikit-learn library:

* Easy-to-Use Interface:
* Broad Range of Algorithms:
* Data Pre-processing and Feature Engineering:
* Model Evaluation and Validation:
* Integration with NumPy and pandas:
* Robust Documentation and Community Support:
* **Keras:**

\* Google's Keras is a cutting-edge deep learning API for creating neural networks. It is created in Python and is designed to simplify the development of neural networks. Additionally, it enables the use of various neural networks for computation. Deep learning models are developed and tested using the free and open-source Python software known as Keras.

Keras is a high-level deep learning library for Python. It is designed to provide a user-friendly and intuitive interface for building and training deep learning models. Keras acts as a front-end API, allowing developers to define and configure neural networks while leveraging the computational backend engines, such as Tensor Flow or Theano.

Here are some key features and functionalities of the Keras library:

* User-Friendly API
* Multi-backend Support
* Wide Range of Neural Network Architectures
* Pre-trained Models and Transfer Learning:
* Easy Model Training and Evaluation:
* GPU Support:
* **h5py:**

\* The h5py Python module offers an interface for the binary HDF5 data format. Thanks to p5py, the top can quickly halt the vast amount of numerical data and alter it using the NumPy library. It employs common syntax for Python, NumPy, and dictionary arrays.

h5py is a Python library that provides a simple and efficient interface for working with datasets and files in the Hierarchical Data Format 5 (HDF5) format. HDF5 is a versatile data format commonly used for storing and managing large volumes of numerical data.

Here are some key features and functionalities of the h5py library:

* + HDF5 File Access
  + Dataset Handling:
  + Group Organization:
  + Attributes:
  + Compatibility with NumPy
  + Performance
* **Tensor flow**

TensorFlow is a Python library for fast numerical computing created and released by Google. It is a foundation library that can be used to create Deep Learning models directly or by using wrapper libraries that simplify the process built on top of TensorFlow. TensorFlow is an end-to-end open source platform for machine learning. TensorFlow is a rich system for managing all aspects of a machine learning system; however, this class focuses on using a particular TensorFlow API to develop and train machine learning models.

TensorFlow is a popular open-source library for machine learning and deep learning. It provides a comprehensive set of tools, APIs, and computational resources for building and training various types of machine learning models, especially neural networks.

Here are some key features and functionalities of TensorFlow:

* Neural Network Framework:
* Computational Graphs
* Automatic Differentiation
* GPU and TPU Support
* Distributed Computing
* Deployment Capabilities
* **Tkinter**

Tkinter is an acronym for "Tk interface". Tk was developed as a GUI extension for the Tcl scripting language by John Ousterhout. The first release was in 1991. Tkinter is the de facto way in Python to create Graphical User interfaces (GUIs) and is included in all standard Python Distributions. In fact, it's the only framework built into the Python standard library.

Tkinter is a standard Python library used for creating graphical user interfaces (GUIs). It provides a set of modules and classes that allow you to develop interactive and visually appealing desktop applications.

Here are some key features and functionalities of Tkinter:

* Cross-Platform Compatibility
* Simple and Easy-to-Use
* Widgets and Layout Management
* Event-Driven Programming
* Customization and Styling
* Integration with Other Libraries
* **NLTK**

NLTK is a toolkit build for working with NLP in Python. It provides us various text processing libraries with a lot of test datasets. A variety of tasks can be performed using NLTK such as tokenizing, parse tree visualization, etc NLTK (Natural Language Toolkit) is the go-to API for NLP (Natural Language Processing) with Python. It is a really powerful tool to pre-process text data for further analysis like with ML models for instance. It helps convert text into numbers, which the model can then easily work with.

NLTK (Natural Language Toolkit) is a Python library widely used for working with human language data and implementing natural language processing (NLP) tasks. It provides a set of tools, corpora, and resources for tasks such as tokenization, stemming, tagging, parsing, sentiment analysis, and more.

Here are some key features and functionalities of NLTK:

* Text Processing
* Part-of-Speech Tagging
* Named Entity Recognition
* Chunking and Parsing
* Sentiment Analysis:
* WordNet Integration:
* **Scipy**

SciPy is a collection of mathematical algorithms and convenience functions built on the NumPy extension of Python. It adds significant power to the interactive Python session by providing the user with high-level commands and classes for manipulating and visualizing data.

SciPy is a powerful scientific computing library for Python that provides a wide range of mathematical algorithms and functions. It builds upon NumPy, another fundamental library for numerical computing, and extends its capabilities by adding additional tools for scientific and technical computing tasks.

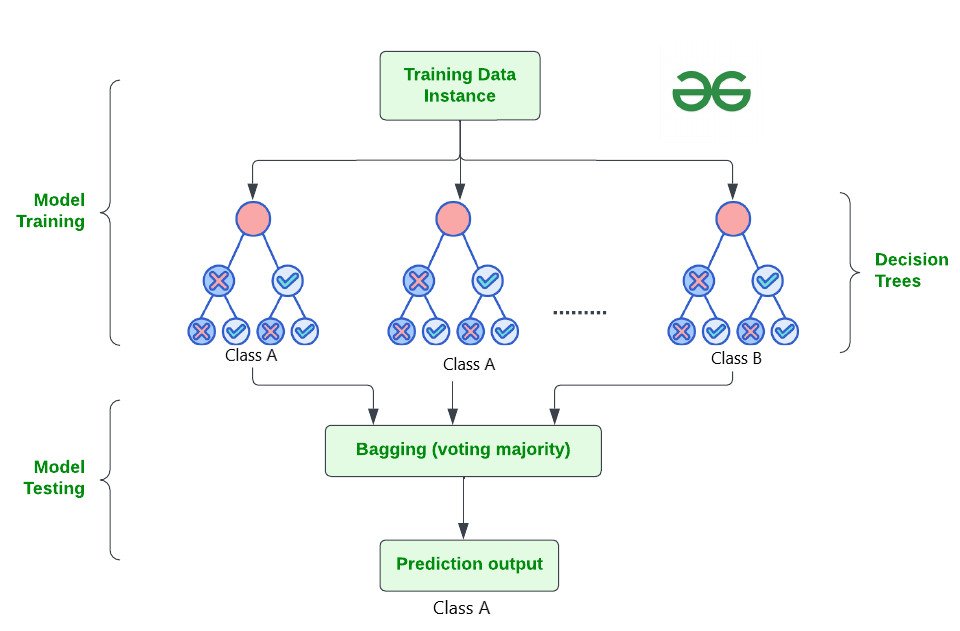
Here are some key features and functionalities of SciPy:

* Numerical Integration:
* Optimization and Root Finding
* Linear Algebra
* Signal and Image Processing
* Statistics

**Random Forest:**

Machine learning, a fascinating blend of computer science and statistics, has witnessed incredible progress, with one standout algorithm being the **Random Forest**. **Random forests or Random Decision Trees** is a collaborative team of **decision trees** that work together to provide a single output. Originating in 2001 through Leo Breiman, Random Forest has become a cornerstone for machine learning enthusiasts. In this article, we will explore the fundamentals and implementation of **Random Forest Algorithm.**

Random Forest algorithm is a powerful tree learning technique in Machine Learning. It works by creating a number of Decision Trees during the training phase. Each tree is constructed using a random subset of the data set to measure a random subset of features in each partition. This randomness introduces variability among individual trees, reducing the risk of overfitting and improving overall prediction performance. In prediction, the algorithm aggregates the results of all trees, either by voting (for classification tasks) or by averaging (for regression tasks) This collaborative decision-making process, supported by multiple trees with their insights, provides an example stable and precise results. Random forests are widely used for classification and regression functions, which are known for their ability to handle complex data, reduce overfitting, and provide reliable forecasts in different environments.



**How Does Random Forest Work?**

The random Forest algorithm works in several steps which are discussed below–>

* **Ensemble of Decision Trees:** Random Forest leverages the power of ensemble learning by constructing an army of Decision Trees. These trees are like individual experts, each specializing in a particular aspect of the data. Importantly, they operate independently, minimizing the risk of the model being overly influenced by the nuances of a single tree.
* **Random Feature Selection:** To ensure that each decision tree in the ensemble brings a unique perspective, Random Forest employs random feature selection. During the training of each tree, a random subset of features is chosen. This randomness ensures that each tree focuses on different aspects of the data, fostering a diverse set of predictors within the ensemble.
* **Bootstrap Aggregating or Bagging:** The technique of bagging is a cornerstone of Random Forest’s training strategy which involves creating multiple bootstrap samples from the original dataset, allowing instances to be sampled with replacement. This results in different subsets of data for each decision tree, introducing variability in the training process and making the model more robust.
* **Decision Making and Voting:** When it comes to making predictions, each decision tree in the Random Forest casts its vote. For classification tasks, the final prediction is determined by the [mode](https://www.geeksforgeeks.org/mode/) (most frequent prediction) across all the trees. In regression tasks, the average of the individual tree predictions is taken. This internal voting mechanism ensures a balanced and collective decision-making process.

**Key Features of Random Forest**

Some of the Key Features of Random Forest are discussed below–>

1. **High Predictive Accuracy:** Imagine Random Forest as a team of decision-making wizards. Each wizard (decision tree) looks at a part of the problem, and together, they weave their insights into a powerful prediction tapestry. This teamwork often results in a more accurate model than what a single wizard could achieve.
2. **Resistance to Overfitting:** Random Forest is like a cool-headed mentor guiding its apprentices (decision trees). Instead of letting each apprentice memorize every detail of their training, it encourages a more well-rounded understanding. This approach helps prevent getting too caught up with the training data which makes the model less prone to overfitting.
3. **Large Datasets Handling:** Dealing with a mountain of data? Random Forest tackles it like a seasoned explorer with a team of helpers (decision trees). Each helper takes on a part of the dataset, ensuring that the expedition is not only thorough but also surprisingly quick.
4. **Variable Importance Assessment:** Think of Random Forest as a detective at a crime scene, figuring out which clues (features) matter the most. It assesses the importance of each clue in solving the case, helping you focus on the key elements that drive predictions.
5. **Built-in Cross-Validation:** Random Forest is like having a personal coach that keeps you in check. As it trains each decision tree, it also sets aside a secret group of cases (out-of-bag) for testing. This built-in validation ensures your model doesn’t just ace the training but also performs well on new challenges.
6. **Handling Missing Values:** Life is full of uncertainties, just like datasets with missing values. Random Forest is the friend who adapts to the situation, making predictions using the information available. It doesn’t get flustered by missing pieces; instead, it focuses on what it can confidently tell us.
7. **Parallelization for Speed:** Random Forest is your time-saving buddy. Picture each decision tree as a worker tackling a piece of a puzzle simultaneously. This parallel approach taps into the power of modern tech, making the whole process faster and more efficient for handling large-scale projects.

## Implement Random Forest for Classification :

# Import necessary libraries

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score, classification\_report

import warnings

warnings.filterwarnings('ignore')

# Load the Titanic dataset

url = "https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic.csv"

titanic\_data = pd.read\_csv(url)

# Drop rows with missing target values

titanic\_data = titanic\_data.dropna(subset=['Survived'])

# Select relevant features and target variable

X = titanic\_data[['Pclass', 'Sex', 'Age', 'SibSp', 'Parch', 'Fare']]

y = titanic\_data['Survived']

# Convert categorical variable 'Sex' to numerical using .loc

X.loc[:, 'Sex'] = X['Sex'].map({'female': 0, 'male': 1})

# Handle missing values in the 'Age' column using .loc

X.loc[:, 'Age'].fillna(X['Age'].median(), inplace=True)

# Split the dataset into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Create a Random Forest Classifier

rf\_classifier = RandomForestClassifier(n\_estimators=100, random\_state=42)

# Train the classifier

rf\_classifier.fit(X\_train, y\_train)

# Make predictions on the test set

y\_pred = rf\_classifier.predict(X\_test)

# Evaluate the model

accuracy = accuracy\_score(y\_test, y\_pred)

classification\_rep = classification\_report(y\_test, y\_pred)

# Print the results

print(f"Accuracy: {accuracy:.2f}")

print("\nClassification Report:\n", classification\_rep)

**Output:**

Accuracy: 0.80

Classification Report:

precision recall f1-score support

0 0.82 0.85 0.83 105

1 0.77 0.73 0.75 74

accuracy 0.80 179

macro age 0.79 0.79 0.79 179

weighted age 0.80 0.80 0.80 179

In the above code, we’re using a Random Forest Classifier to make sense of the Titanic dataset. First, we gather our tools – importing libraries to handle data and evaluate our model. Next, we dive into the Titanic dataset, fixing missing information and choosing important details like a detective solving a mystery. We even teach the computer to understand ‘male’ and ‘female’ by turning them into numbers. Then, we split our dataset into pieces – one part to train our model, and the other to test its newfound skills. Our Random Forest Classifier is like a student, learning from the training set. Once trained, it faces a test – making predictions on the test set. We’re like judges, using a classification report to grade how well our model did.

### **Gradient Boosting**

Gradient Boosting is a popular boosting algorithm. In gradient boosting, each predictor corrects its predecessor’s error. In contrast to Ad boost, the weights of the training instances are not tweaked, instead, each predictor is trained using the residual errors of predecessor as labels.

There is a technique called the Gradient Boosted Trees whose base learner is CART (Classification and Regression Trees).

**5.2 Sample Code:**

**Main.py**

from Tkinter import messagebox

from tkinter import \*

from tkinter.filedialog import askopenfilename

from tkinter import simpledialog

import tkinter

import numpy as np

from tkinter import filedialog

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score

import matplotlib.pyplot as plt

import pandas as pd

from sklearn.preprocessing import LabelEncoder

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.metrics import accuracy\_score

from sklearn.metrics import f1\_score

from sklearn.metrics import precision\_score

from sklearn.metrics import recall\_score

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import mean\_squared\_error

from sklearn.ensemble import RandomForestRegressor

from sklearn.ensemble import GradientBoostingClassifier

from sklearn.ensemble import GradientBoostingRegressor

from math import sqrt

main = tkinter.Tk()

main.title("Predicting Student Success in Online Learning Environments using Machine Learning")

main.geometry("1300x1200")

global filename

global accuracy, precision, recall, fscore, rmse

global dataset

global X, Y, score

global score\_X\_train, score\_X\_test, score\_y\_train, score\_y\_test

global label\_encoder, rf\_rg, rf\_cls, labels

global grade\_X\_train, grade\_X\_test, grade\_y\_train, grade\_y\_test

def uploadDataset():

global filename, dataset

filename = filedialog.askopenfilename(initialdir = "Dataset")

pathlabel.config(text=filename)

text.delete('1.0', END)

text.insert(END,'Dataset loaded\n\n')

dataset = pd.read\_csv(filename,nrows=5000)

dataset.drop(['code\_module'], axis = 1,inplace=True)

text.insert(END,str(dataset))

label = dataset.groupby('final\_result').size()

label.plot(kind="bar")

plt.xlabel('Grades')

plt.ylabel('Number of Records')

plt.xticks(rotation=90)

plt.title("Grades Graph")

plt.show()

def processDataset():

global dataset, labels, X, Y

global score\_X\_train, score\_X\_test, score\_y\_train, score\_y\_test

global grade\_X\_train, grade\_X\_test, grade\_y\_train, grade\_y\_test

global label\_encoder, score

text.delete('1.0', END)

labels = np.unique(dataset['final\_result'])

label\_encoder = []

columns = dataset.columns

types = dataset.dtypes.values

for i in range(len(types)):

name = types[i]

if name == 'object': #finding column with object type

le = LabelEncoder()

dataset[columns[i]] = pd.Series(le.fit\_transform(dataset[columns[i]].astype(str)))#encode all str columns to numeric

label\_encoder.append([columns[i], le])

dataset.fillna(0, inplace = True)

Y = dataset['final\_result'].ravel()

score = dataset['score'].ravel()

dataset.drop(['final\_result', 'score'], axis = 1,inplace=True)

X = dataset.values

text.insert(END,str(dataset)+"\n\n")

grade\_X\_train, grade\_X\_test, grade\_y\_train, grade\_y\_test = train\_test\_split(X, Y, test\_size=0.2) #split dataset into train and test

score\_X\_train, score\_X\_test, score\_y\_train, score\_y\_test = train\_test\_split(X, score, test\_size=0.2) #split dataset into train and test

text.insert(END,"Dataset Train & Test Split\n\n")

text.insert(END,"80% dataset size used for training : "+str(grade\_X\_train.shape)+"\n")

text.insert(END,"20% dataset size used for testing : "+str(grade\_X\_test.shape)+"\n")

def calculateMetrics(algorithm, predict, y\_test, rmse\_score):

global accuracy, precision, recall, fscore, rmse

a = accuracy\_score(y\_test,predict)\*100

p = precision\_score(y\_test, predict,average='macro') \* 100

r = recall\_score(y\_test, predict,average='macro') \* 100

f = f1\_score(y\_test, predict,average='macro') \* 100

accuracy.append(a)

precision.append(p)

recall.append(r)

fscore.append(f)

rmse.append(rmse\_score)

text.insert(END,algorithm+" Accuracy : "+str(a)+"\n")

text.insert(END,algorithm+" Precision : "+str(p)+"\n")

text.insert(END,algorithm+" Recall : "+str(r)+"\n")

text.insert(END,algorithm+" FSCORE : "+str(f)+"\n")

text.insert(END,algorithm+" RMSE : "+str(rmse\_score)+"\n\n")

def runRandomForest():

text.delete('1.0', END)

global accuracy, precision, recall, fscore, rmse

global rf\_rg, rf\_cls

global score\_X\_train, score\_X\_test, score\_y\_train, score\_y\_test

global grade\_X\_train, grade\_X\_test, grade\_y\_train, grade\_y\_test

accuracy = []

precision = []

recall = []

fscore = []

rmse = []

rf\_cls = RandomForestClassifier() #create Random Forest object

rf\_cls.fit(grade\_X\_train, grade\_y\_train)

predict = rf\_cls.predict(grade\_X\_test)

rf\_rg = RandomForestRegressor() #create Random Forest object

rf\_rg.fit(score\_X\_train, score\_y\_train)

predicts = rf\_rg.predict(score\_X\_test)

mse\_value = mean\_squared\_error(score\_y\_test, predicts)

rmse\_value = sqrt(mse\_value)

calculateMetrics("Random Forest", predict, grade\_y\_test, rmse\_value)

def runGradient():

global accuracy, precision, recall, fscore, rmse

global score\_X\_train, score\_X\_test, score\_y\_train, score\_y\_test

global grade\_X\_train, grade\_X\_test, grade\_y\_train, grade\_y\_test

gb\_cls = GradientBoostingClassifier() #create GradientBoostingClassifier object

gb\_cls.fit(grade\_X\_train, grade\_y\_train)

predict = gb\_cls.predict(grade\_X\_test)

gb\_rg = GradientBoostingRegressor() #create Random Forest object

gb\_rg.fit(score\_X\_train, score\_y\_train)

predicts = gb\_rg.predict(score\_X\_test)

mse\_value = mean\_squared\_error(score\_y\_test, predicts)

rmse\_value = sqrt(mse\_value)

calculateMetrics("Gradient Boosting", predict, grade\_y\_test, rmse\_value)

def graph():

global accuracy, precision, recall, fscore, rmse

df = pd.DataFrame([['Random Forest','Precision',precision[0]],['Random Forest','Recall',recall[0]],['Random Forest','F1 Score',fscore[0]],['Random Forest','Accuracy',accuracy[0]],

['Gradient Boosting','Precision',precision[1]],['Gradient Boosting','Recall',recall[1]],['Gradient Boosting','F1 Score',fscore[1]],['Gradient Boosting','Accuracy',accuracy[1]],

],columns=['Algorithms','Performance Output','Value'])

df.pivot("Algorithms", "Performance Output", "Value").plot(kind='bar')

plt.show()

def predict():

text.delete('1.0', END)

global rf\_rg, rf\_cls, label\_encoder, labels

filename = filedialog.askopenfilename(initialdir = "Dataset")

test = pd.read\_csv(filename)

test.drop(['code\_module'], axis = 1,inplace=True)

testData = pd.read\_csv(filename)

testData.drop(['code\_module','final\_result', 'score'], axis = 1,inplace=True)

testData = testData.values

for i in range(len(label\_encoder)):

temp = label\_encoder[i]

name = temp[0]

le = temp[1]

test[name] = pd.Series(le.transform(test[name].astype(str)))#encode all str columns to numeric

test.fillna(0, inplace = True)

test.drop(['final\_result', 'score'], axis = 1,inplace=True)

test = test.values

predict\_score = rf\_rg.predict(test)

predict\_grade = rf\_cls.predict(test)

for i in range(len(predict\_score)):

text.insert(END,"Test Data = "+str(testData[i])+"\n")

if labels[predict\_grade[i]] == 'Fail':

predict\_score[i] = predict\_score[i] / 2

text.insert(END,"Predicted Score = "+str(predict\_score[i])+"\n")

text.insert(END,"Predicted Grade = "+str(labels[predict\_grade[i]])+"\n\n")

def close():

main.destroy()

font = ('times', 16, 'bold')

title = Label(main, text='Predicting Student Success in Online Learning Environments using Machine Learning')

title.config(bg='chocolate', fg='white')

title.config(font=font)

title.config(height=3, width=120)

title.place(x=0,y=5)

font1 = ('times', 13, 'bold')

upload = Button(main, text="Upload OULAD Online Student Dataset", command=uploadDataset)

upload.place(x=700,y=100)

upload.config(font=font1)

pathlabel = Label(main)

pathlabel.config(bg='lawn green', fg='dodger blue')

pathlabel.config(font=font1)

pathlabel.place(x=700,y=150)

processButton = Button(main, text="Preprocess Dataset", command=processDataset)

processButton.place(x=700,y=200)

processButton.config(font=font1)

rfButton = Button(main, text="Run Random Forest", command=runRandomForest)

rfButton.place(x=700,y=250)

rfButton.config(font=font1)

gbButton = Button(main, text="Run Gradient Boosting", command=runGradient)

gbButton.place(x=700,y=300)

gbButton.config(font=font1)

graphButton = Button(main, text="Comparison Graph", command=graph)

graphButton.place(x=700,y=350)

graphButton.config(font=font1)

predictButton = Button(main, text="Predict Grade & Score", command=predict)

predictButton.place(x=700,y=400)

predictButton.config(font=font1)

closeButton = Button(main, text="Exit", command=close)

closeButton.place(x=700,y=450)

closeButton.config(font=font1)

font1 = ('times', 12, 'bold')

text=Text(main,height=30,width=80)

scroll=Scrollbar(text)

text.configure(yscrollcommand=scroll.set)

text.place(x=10,y=100)

text.config(font=font1)

main.config(bg='light salmon')

main.mainloop()

**6. TESTING**

**Implementation and Testing:**

Implementation is one of the most important tasks in project is the phase in which one has to be cautions because all the efforts undertaken during the project will be very interactive. Implementation is the most crucial stage in achieving successful system and giving the users confidence that the new system is workable and effective. Each program is tested individually at the time of development using the sample data and has verified that these programs link together in the way specified in the program specification. The computer system and its environment are tested to the satisfaction of the user.

## **Implementation**

## The implementation phase is less creative than system design. It is primarily concerned with user training, and file conversion. The system may be requiring extensive user training. The initial parameters of the system should be modifies as a result of a programming. A simple operating procedure is provided so that the user can understand the different functions clearly and quickly. The different reports can be obtained either on the inkjet or dot matrix printer, which is available at the disposal of the user. The proposed system is very easy to implement. In general implementation is used to mean the process of converting a new or revised system design into an operational one.

## **Testing**

Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property functions as a unit. The test data should be chosen such that it passed through all possible condition. Actually testing is the state of implementation which aimed at ensuring that the system works accurately and efficiently before the actual operation commence. The following is the description of the testing strategies, which were carried out during the testing period.

### **System Testing**

Testing has become an integral part of any system or project especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it to be check if one is capable to with stand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to use the software must be tested whether it is solving the purpose for which it is developed. This testing involves various types through which one can ensure the software is reliable. The program was tested logically and pattern of execution of the program for a set of data are repeated. Thus the code was exhaustively checked for all possible correct data and the outcomes were also checked.

**Module Testing**

To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. Each module in the system is tested separately. For example the job classification module is tested separately. This module is tested with different job and its approximate execution time and the result of the test is compared with the results that are prepared manually. The comparison shows that the results proposed system works efficiently than the existing system. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time.

**Integration Testing**

After the module testing, the integration testing is applied. When linking the modules there may be chance for errors to occur, these errors are corrected by using this testing. In this system all modules are connected and tested. The testing results are very correct. Thus the mapping of jobs with resources is done correctly by the system.

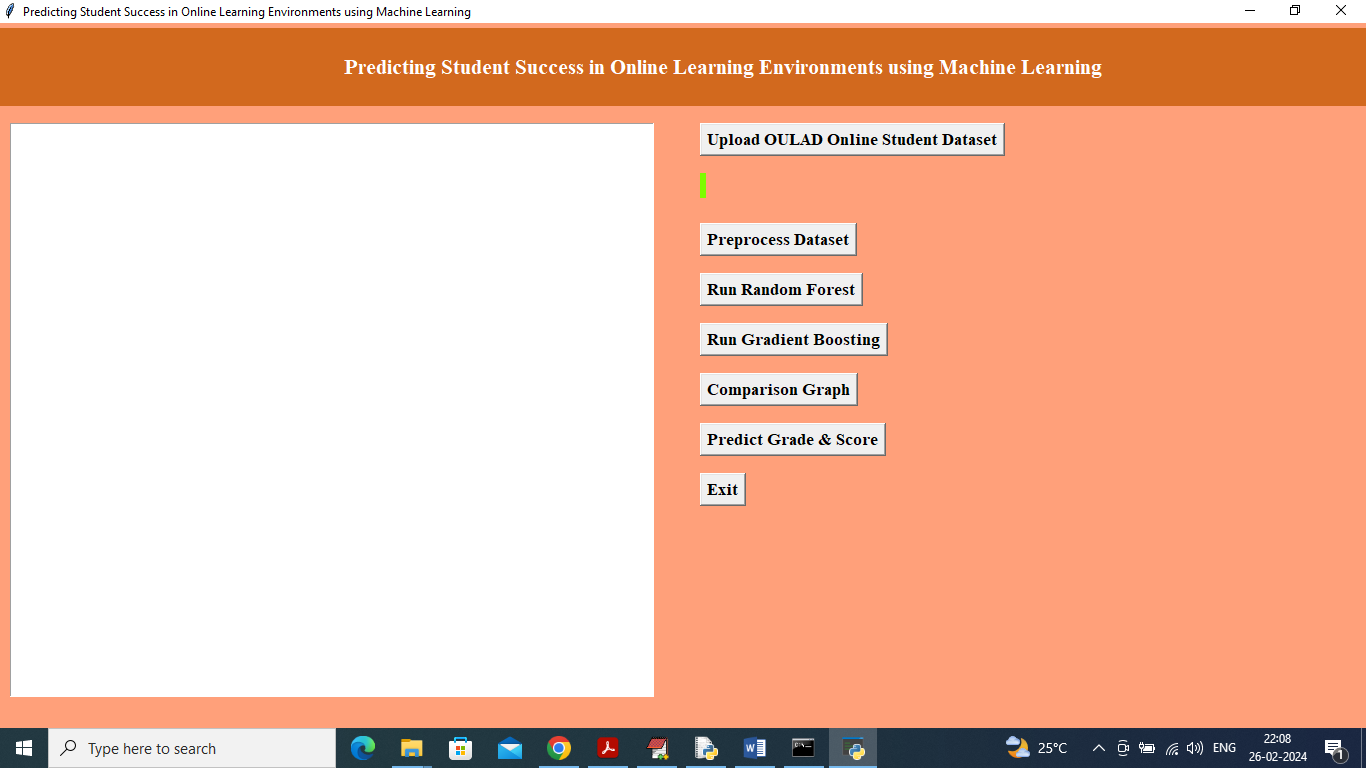
**Acceptance Testing**

When that user fined no major problems with its accuracy, the system passers through a final acceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which elimination wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation

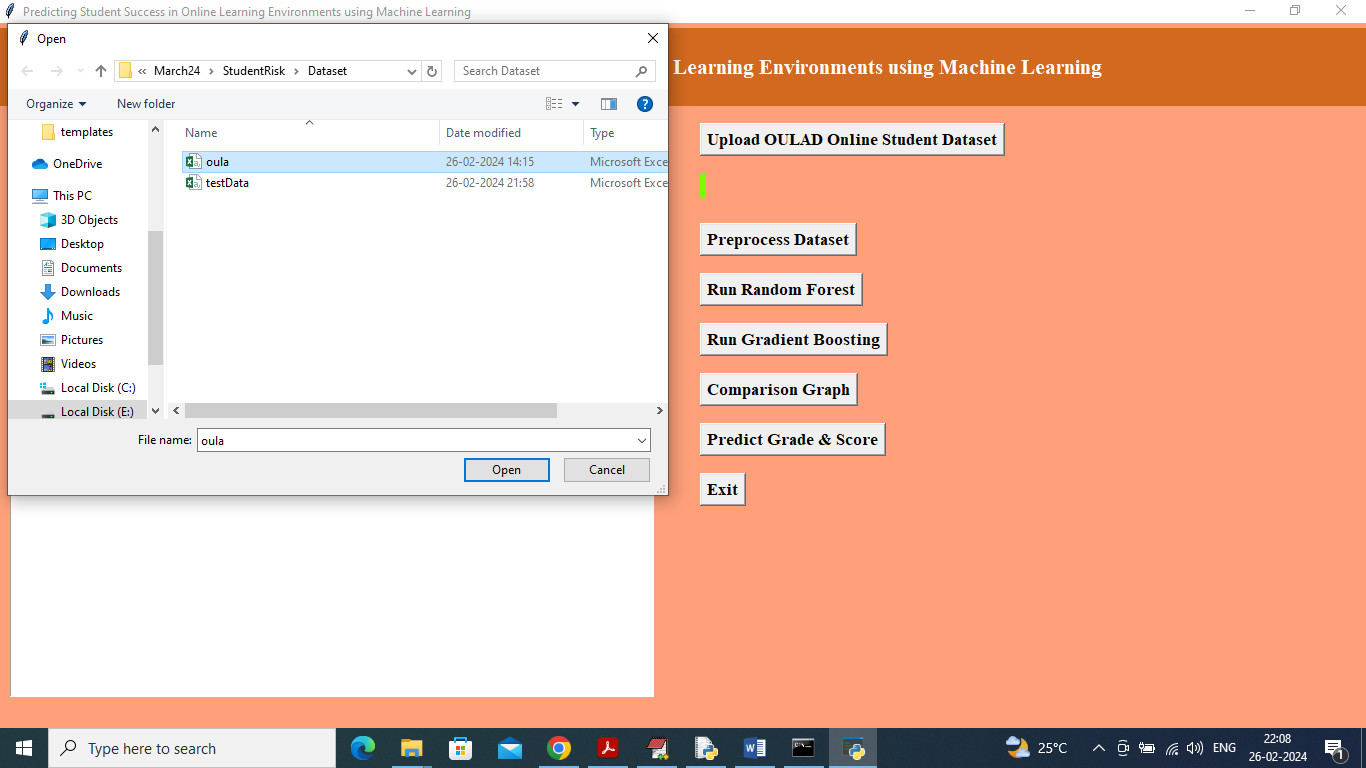
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Id** | **Test Case Name** | **Test Case Desc.** | **Test Steps** | | | **Test Case Status** | **Test Priority** |
| **Step** | **Expected** | **Actual** |
| 01 | Upload OULAD Online Student Dataset | Test whether Dataset is uploaded or not into the system | If the Dataset may not uploaded | We cannot do further operations | Dataset uploaded we will do further operations | High | High |
| 02 | Preprocess Dataset | Test whether the Pre-process Dataset Successfully or not | If the Pre-process Dataset may not Run Successfully | We cannot do further operations | we will do further  operations | High | High |
| 03 | Run Random Forest | Test whether Random Forest Algorithm Run Successfully or not | If the Random Forest Algorithm may not Run Successfully | We cannot do further operations | we will do further  operations | High | High |
| 04 | Run Gradient Boosting | Test whether Gradient Boosting Algorithm Run Successfully or not | If the Gradient Boosting Algorithm may not Run Successfully | We cannot do further operations | we will do further  operations | High | High |
| 05 | Comparison Graph | Test whether Comparison Graph Display Successfully or not | If the Comparison Graph may not Display Successfully | We cannot do further operations | we will do further  operations | High | High |
| 06 | Predict Grade & Score | Test whether Predict Grade & Score Successfully or not | If the Predict Grade & Score may not Successfully | We cannot do further operations | we will do further  operations | High | High |

**7.SCREENSHOTS**:

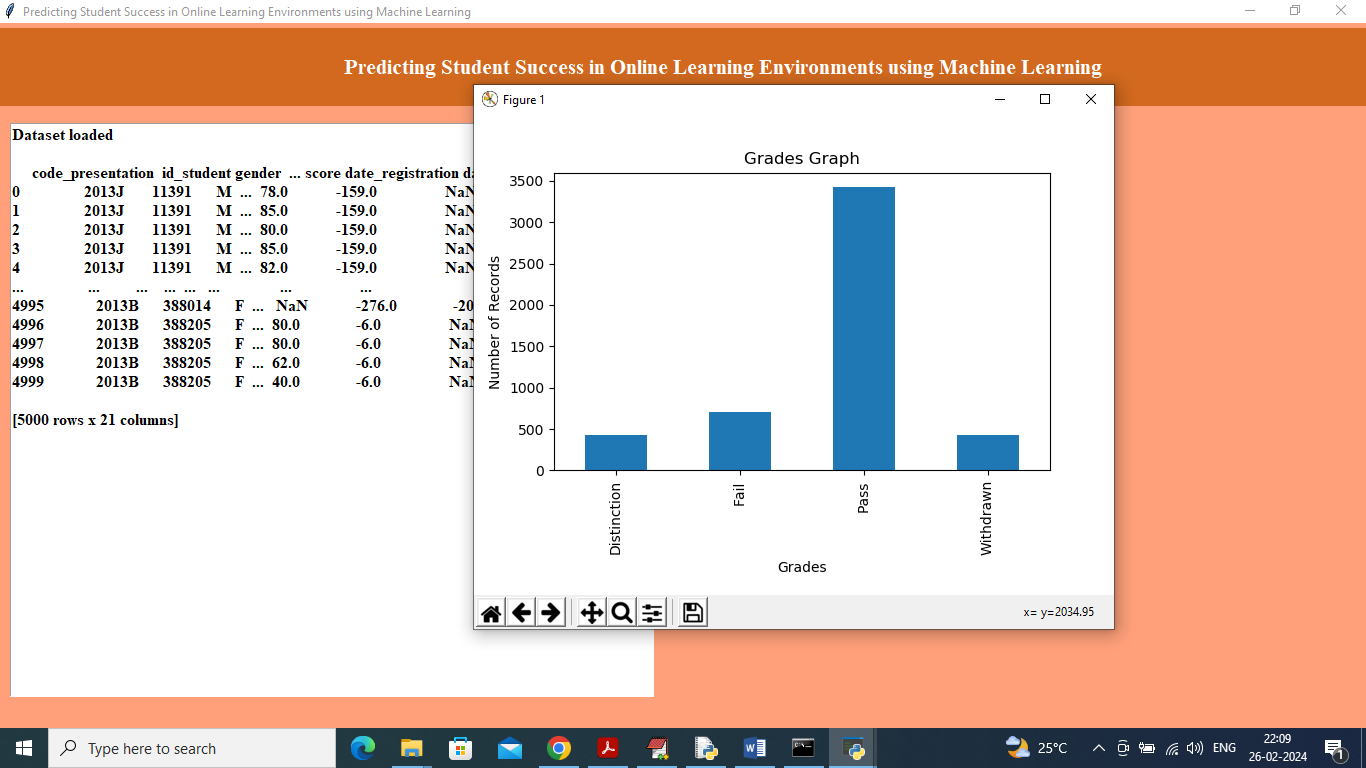
To run project double click on run.bat file to get below screen



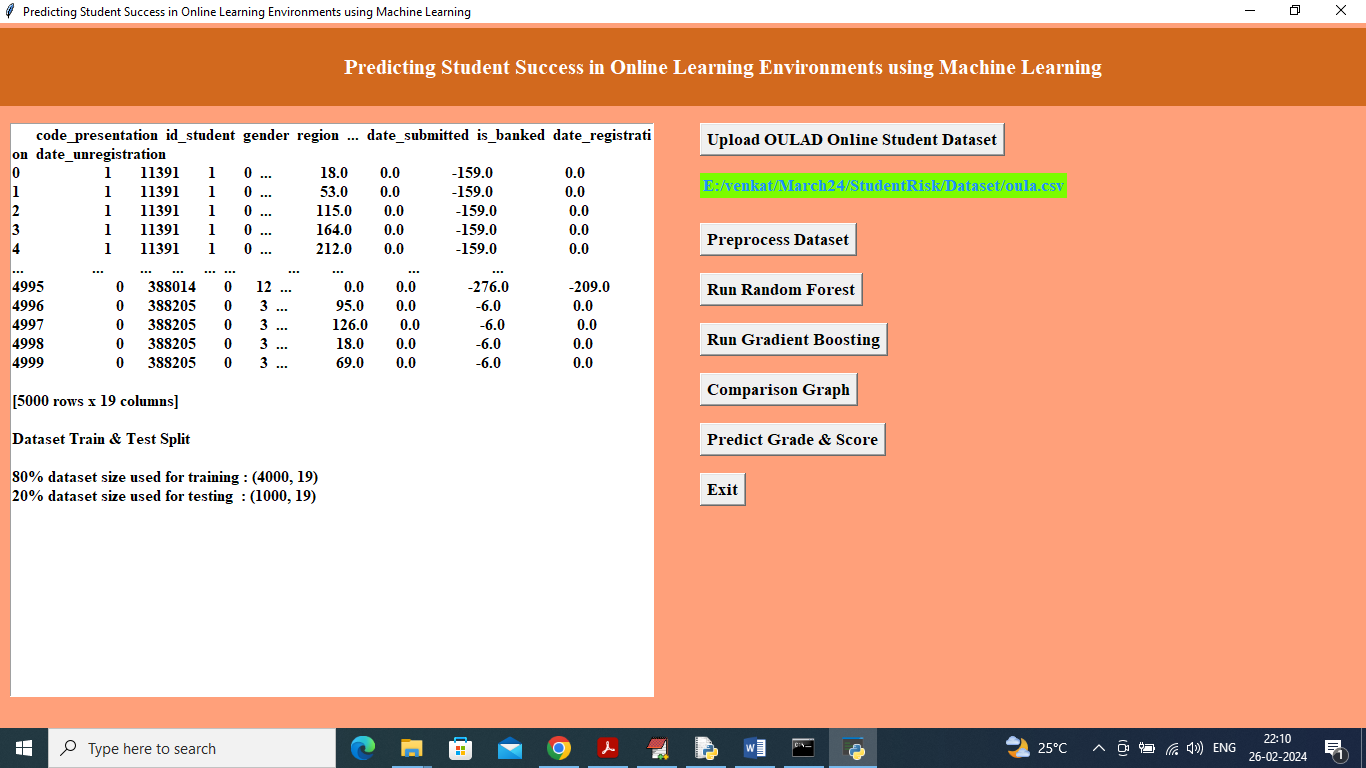
In above screen click on ‘Upload OULAD Online Student Dataset’ button to upload dataset and get below page



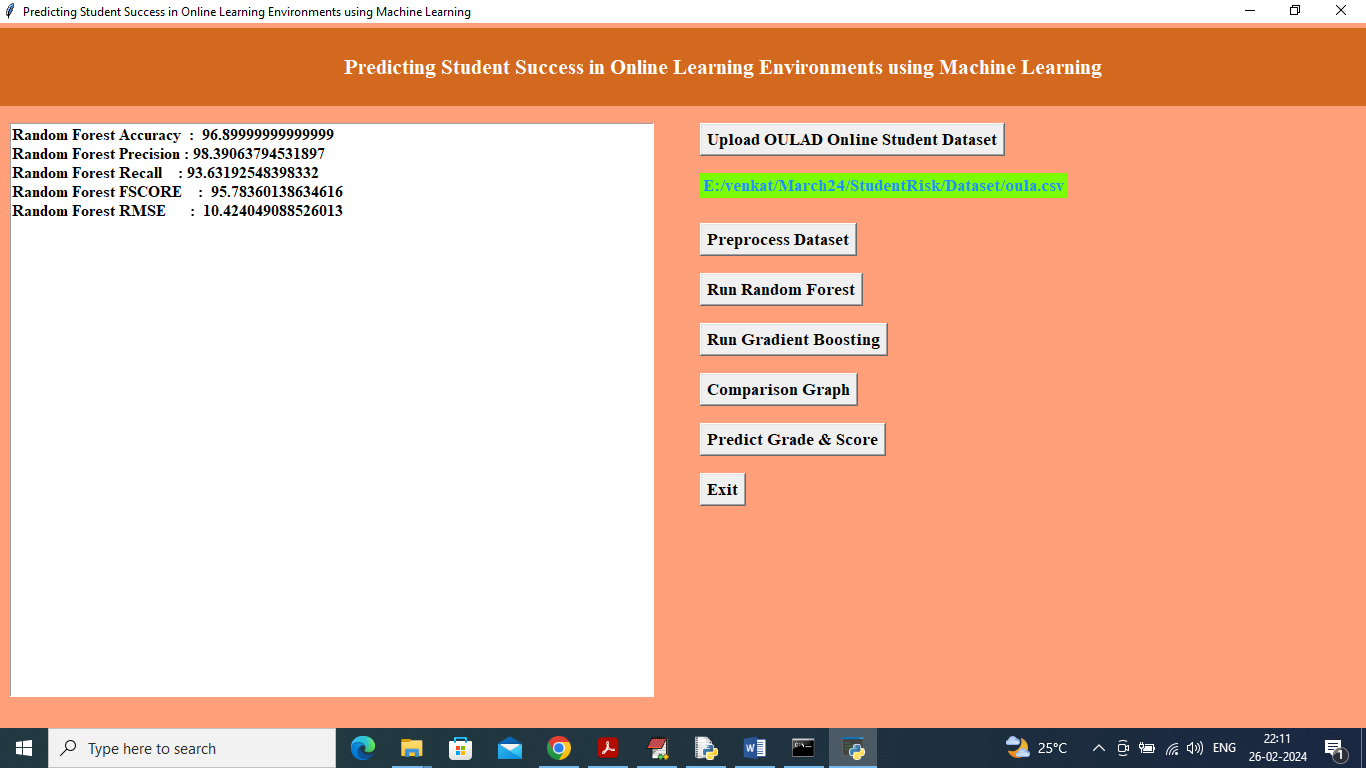
In above screen selecting and uploading OULAD dataset and then click on ‘Open’ button to load dataset and get below page



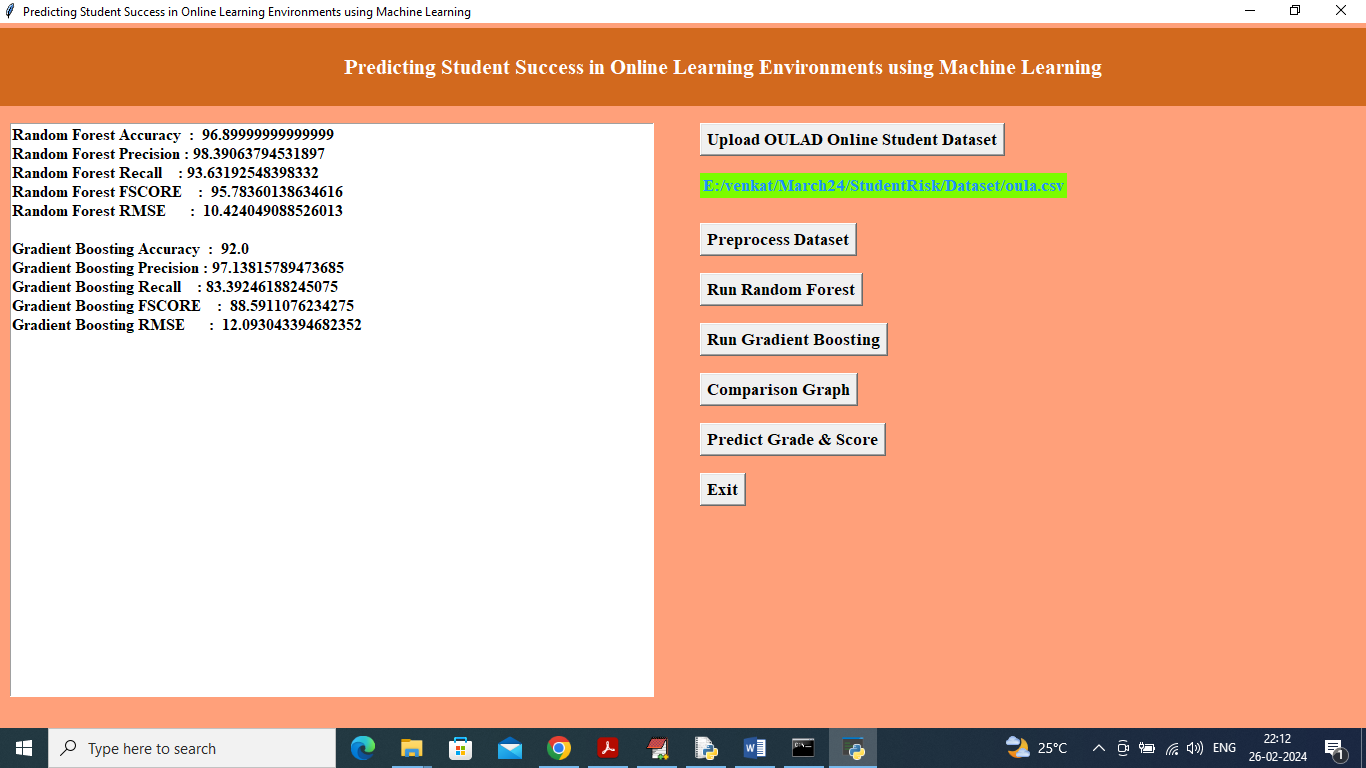
In above screen dataset loaded and can see dataset contains both non-numeric and numeric values and by using processing technique will convert non-numeric to numeric values and in graph x-axis represents Grade and y-axis represents number of records found in that grade and now close above graph and then click on ‘Pre-process Dataset’ button to get below output



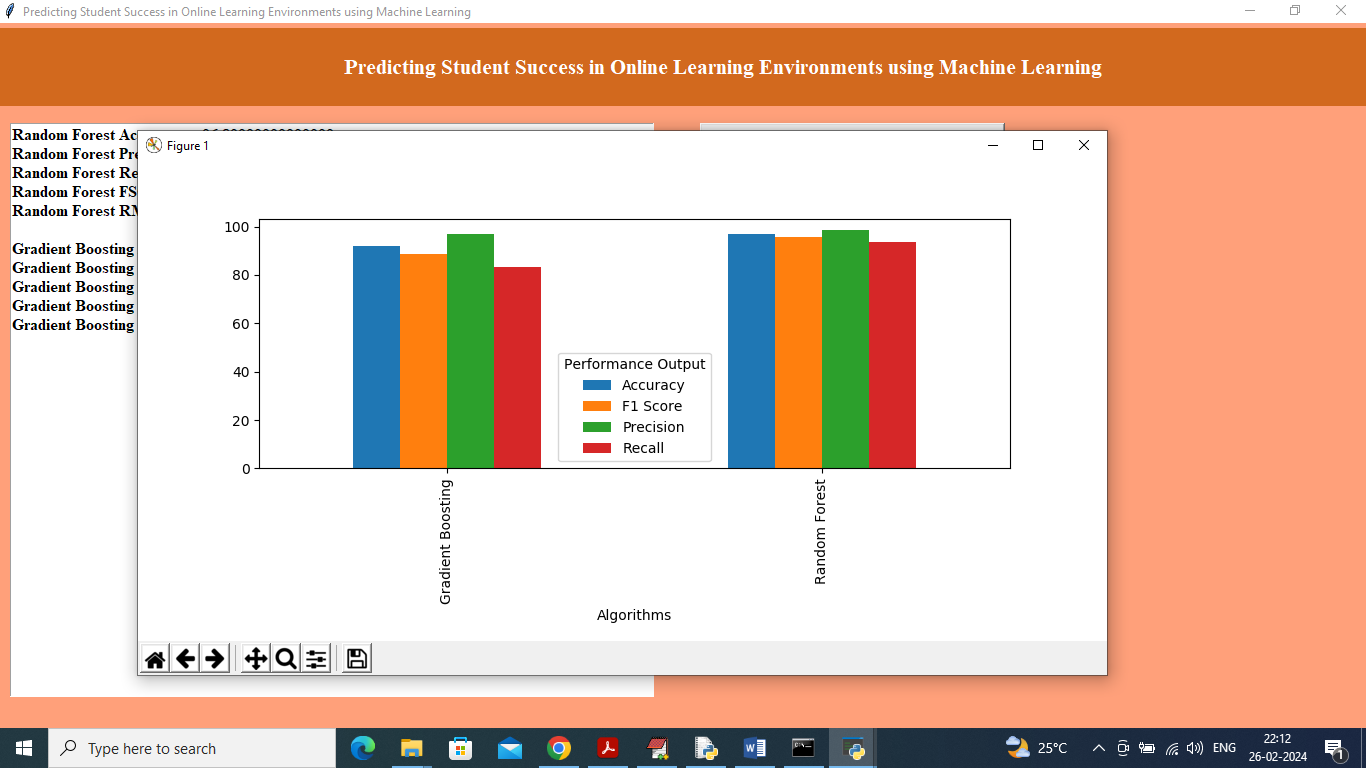
In above screen can see all values are converted to numeric format and then can see train and test data size and now click on ‘Run Random Forest’ algorithm button to get below output



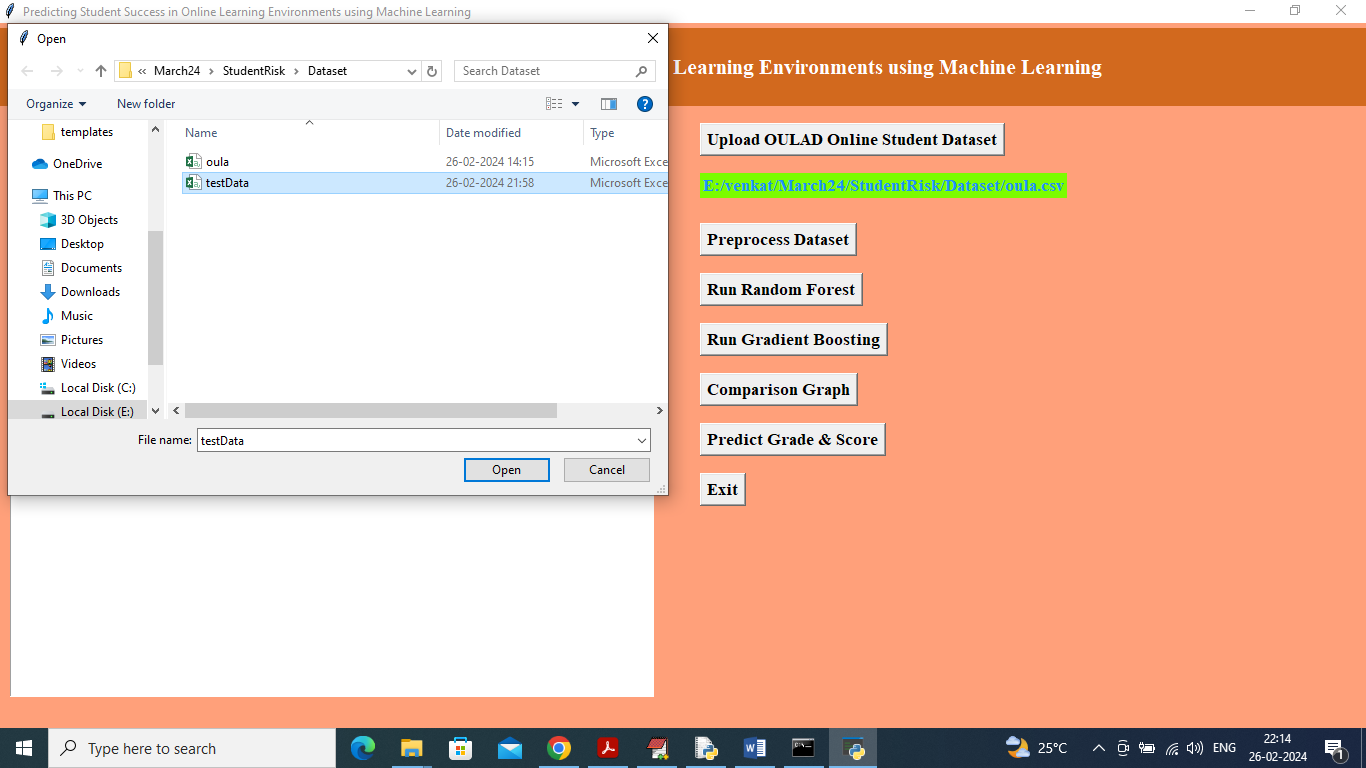
In above screen can see Random Forest accuracy as 96% and RMSE as 10% and can see other metrics like precision, recall and FCSORE. Now click on ‘Run Gradient Boosting’ button to train boosting and get below output



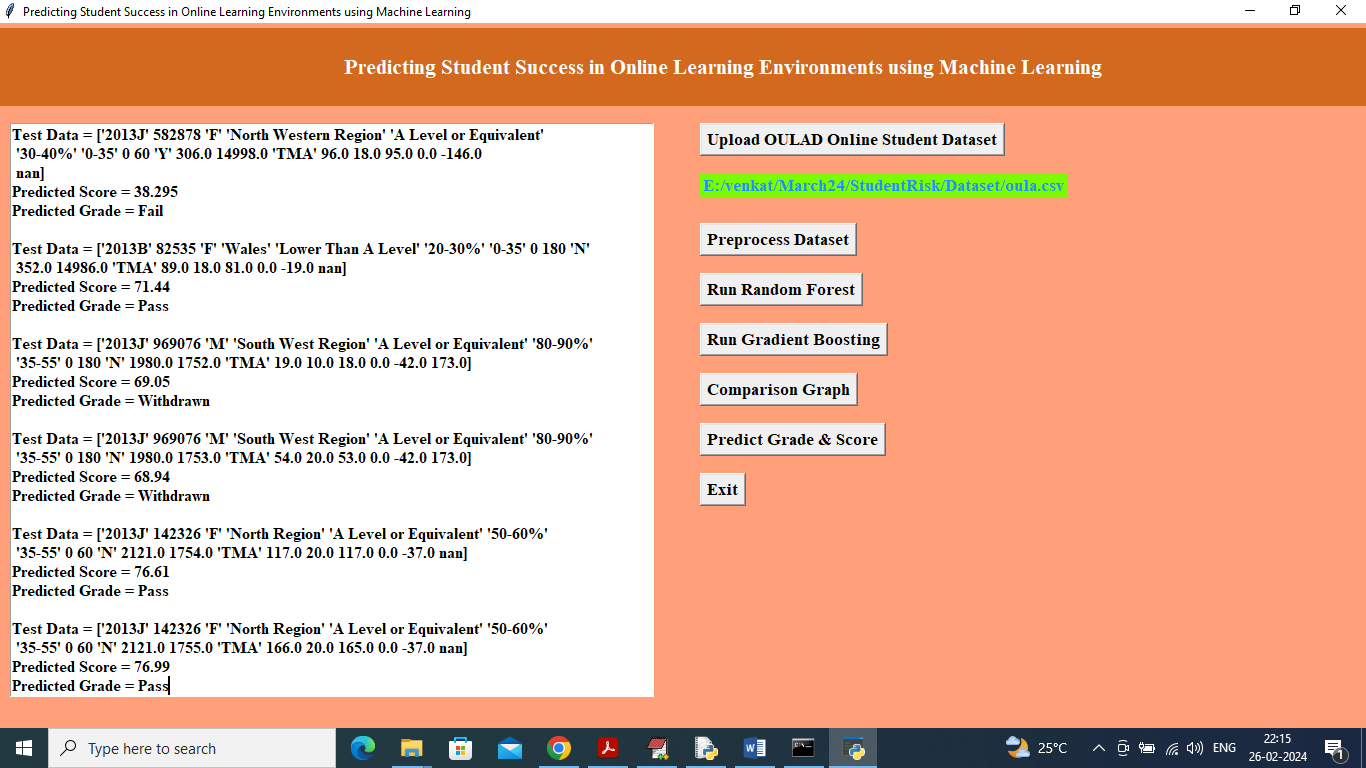
In above screen gradient boosting accuracy is 92% and RMSE is 12% and now click on ‘Comparison Graph’ button to get below graph



In above graph x-axis represents algorithm names and y-axis represents accuracy and other metrics in different colour bars and in both algorithm Random Forest got high accuracy and less RMSE. Now click on ‘Predict Grade & Score’ button to upload test data and get below output



In above screen selecting and uploading test data and then click on ‘Open’ button to get below prediction output



In above screen in square bracket we can see Test data values and then in next two lines can see Predicted score and predicted Grade.

**8. CONCLUSION:**

With the emergence of MOOCs and the expansion of online education in recent years, the prediction of learners’ outcomes in online environments has attracted considerable attention. The spread of COVID-19 has also aided the growth of SPOCs, blended education, and an interest in monitoring student engagement and performance. Therefore, this study reviewed current strategies for predicting online-student outcomes in MOOCs and SPOCs. It summarized the predictive variables, online learning platforms, feature extraction, selection techniques, evaluation metrics, and the predictive models employed in this area. It also provided a thorough analysis and taxonomy for related research. Throughout our analysis, we found that most studies in the field utilized statistical features such as the number of downloaded materials and duration of video watching in a given time period. A small number of studies examined statistical temporal and raw temporal features in predicting learner outcomes. Studies conducted on benchmark datasets showed that statistical temporal features provide better results than raw features. Thus, further investigation of temporal features will provide a valuable understanding of users’ learning progress and, eventually, their learner outcomes. Most temporally based LSTM or GRU models learners’ time-series features. Further investigation of other recent sequence-based models, such as the attention-based model, is required. Studies using one-hot encoding to represent raw features and different representation techniques for raw features are worth investigating. Different machine-learning and deep-learning models have been used to predict learners’ outcomes. RF and ANN are among the most effective machine learning models’ performance and dropout prediction, whereas the sequence-based model provides the best performance on the publicly available dropout dataset. Further investigation of deep-learning models is recommended to predict student performance.

**9. REFERENCES:**

1. Arce, M.E.; Crespo, B.; Miguel-Álvarez, C. Higher Education Drop-out in Spain–Particular Case of Universities in Galicia. *Int. Educ. Stud.* **2015**, *8*, 247–264. [[**Google Scholar**](https://scholar.google.com/scholar_lookup?title=Higher+Education+Drop-out+in+Spain%E2%80%93Particular+Case+of+Universities+in+Galicia&author=Arce,+M.E.&author=Crespo,+B.&author=M%C3%ADguez-%C3%81lvarez,+C.&publication_year=2015&journal=Int.+Educ.+Stud.&volume=8&pages=247%E2%80%93264&doi=10.5539/ies.v8n5p247)] [[**Crossruff**](https://doi.org/10.5539/ies.v8n5p247)] [[**Green Version**](http://www.ccsenet.org/journal/index.php/ies/article/download/43580/25971)]
2. Xavier, M.; Meneses, J. *Dropout in Online Higher Education: A Scoping Review from 2014 to 2018*; eLearn Centre, Universität Oberta de Catalunya: Barcelona, Spain, 2020. [[**Google Scholar**](https://scholar.google.com/scholar_lookup?title=Dropout+in+Online+Higher+Education:+A+Scoping+Review+from+2014+to+2018&author=Xavier,+M.&author=Meneses,+J.&publication_year=2020)]
3. Baker, R.S.; Invent ado, P.S. Educational Data Mining and Learning Analytics. In *Learning Analytics: From Research to Practice*; Larusson, J.A., White, B., Eds.; Springer: New York, NY, USA, 2014; pp. 61–75. [[**Google Scholar**](https://scholar.google.com/scholar_lookup?title=Educational+Data+Mining+and+Learning+Analytics&author=Baker,+R.S.&author=Inventado,+P.S.&publication_year=2014&pages=61%E2%80%9375)] [[**Crossruff**](https://doi.org/10.1007/978-1-4614-3305-7_4)] [[**Green Version**](http://circl.sri.com/archive/primers/CIRCL-Primer-LearningAnalytics.pdf)]
4. Moreno-Marcos, P.M.; Alario-Hoyos, C.; Maoz-Merino, P.J.; Kloos, C.D. Prediction in MOOCs: A Review and Future Research Directions. *IEEE Trans. Learn. Technol.* **2019**, *12*, 384–401. [[**Google Scholar**](https://scholar.google.com/scholar_lookup?title=Prediction+in+MOOCs:+A+Review+and+Future+Research+Directions&author=Moreno-Marcos,+P.M.&author=Alario-Hoyos,+C.&author=Mu%C3%83%C5%9Boz-Merino,+P.J.&author=Kloos,+C.D.&publication_year=2019&journal=IEEE+Trans.+Learn.+Technol.&volume=12&pages=384%E2%80%93401&doi=10.1109/TLT.2018.2856808)] [[**Crossruff**](https://doi.org/10.1109/TLT.2018.2856808)]
5. Ranjeeth, S.; Patchouli, T.; Paul, P.V. A survey on predictive models of learning analytics. *Procedia Comput. Sci.* **2020**, *167*, 37–46. [[**Google Scholar**](https://scholar.google.com/scholar_lookup?title=A+survey+on+predictive+models+of+learning+analytics&author=Ranjeeth,+S.&author=Latchoumi,+T.&author=Paul,+P.V.&publication_year=2020&journal=Procedia+Comput.+Sci.&volume=167&pages=37%E2%80%9346&doi=10.1016/j.procs.2020.03.180)] [[**Crossruff**](https://doi.org/10.1016/j.procs.2020.03.180)]
6. Hamim, T.; Benab Bou, F.; Sael, N. Survey of Machine Learning Techniques for Student Profile Modelling. *Int. J. Emerg. Technol. Learn.* **2021**, *16*, 136–151. [[**Google Scholar**](https://scholar.google.com/scholar_lookup?title=Survey+of+Machine+Learning+Techniques+for+Student+Profile+Modelling&author=Hamim,+T.&author=Benabbou,+F.&author=Sael,+N.&publication_year=2021&journal=Int.+J.+Emerg.+Technol.+Learn.&volume=16&pages=136%E2%80%93151&doi=10.3991/ijet.v16i04.18643)] [[**Crossruff**](https://doi.org/10.3991/ijet.v16i04.18643)]
7. Prianka, B.; Velardi, P.; Stilo, G.; Distant, D.; Ferally, S. A survey of machine learning approaches for student dropout prediction in online courses. *ACM Comput. Surd. (CSUR)* **2020**, *53*, 1–34. [[**Google Scholar**](https://scholar.google.com/scholar_lookup?title=A+survey+of+machine+learning+approaches+for+student+dropout+prediction+in+online+courses&author=Prenkaj,+B.&author=Velardi,+P.&author=Stilo,+G.&author=Distante,+D.&author=Faralli,+S.&publication_year=2020&journal=ACM+Comput.+Surv.+(CSUR)&volume=53&pages=1%E2%80%9334&doi=10.1145/3388792)] [[**Crossruff**](https://doi.org/10.1145/3388792)]
8. Gardner, J.; Brooks, C. Student success prediction in MOOCs. *User Model. User-Adapt. Interact.* **2018**, *28*, 127–203. [[**Google Scholar**](https://scholar.google.com/scholar_lookup?title=Student+success+prediction+in+MOOCs&author=Gardner,+J.&author=Brooks,+C.&publication_year=2018&journal=User+Model.+User-Adapt.+Interact.&volume=28&pages=127%E2%80%93203&doi=10.1007/s11257-018-9203-z)] [[**Crossruff**](https://doi.org/10.1007/s11257-018-9203-z)] [[**Green Version**](http://arxiv.org/pdf/1711.06349)]
9. Katara, R.; Gaba, J.; Garg, A.; Verma, V. A review on machine learning based student’s academic performance prediction systems. In Proceedings of the 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), Coimbatore, India, 25–27 March 2021; pp. 254–259. [[**Google Scholar**](https://scholar.google.com/scholar_lookup?title=A+review+on+machine+learning+based+student%E2%80%99s+academic+performance+prediction+systems&conference=Proceedings+of+the+2021+International+Conference+on+Artificial+Intelligence+and+Smart+Systems+(ICAIS)&author=Katarya,+R.&author=Gaba,+J.&author=Garg,+A.&author=Verma,+V.&publication_year=2021&pages=254%E2%80%93259&doi=10.1109/ICAIS50930.2021.9395767)] [[**Crossruff**](https://doi.org/10.1109/ICAIS50930.2021.9395767)]
10. Filius, R.M.; Bijl, S.G. Teaching Methodologies for Scalable Online Education. In *Handbook for Online Learning Contexts: Digital, Mobile and Open*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 55–65. [[**Google Scholar**](https://scholar.google.com/scholar_lookup?title=Teaching+Methodologies+for+Scalable+Online+Education&author=Filius,+R.M.&author=Uijl,+S.G.&publication_year=2021&pages=55%E2%80%9365)]
11. Amirah, E.A.; Hamsini, T.M.; Ajara, I. Mining Educational Data to Predict Student’s academic Performance using Ensemble Methods. *Int. J. Database Theory Appl.* **2016**, *9*, 119–136. [[**Google Scholar**](https://scholar.google.com/scholar_lookup?title=Mining+Educational+Data+to+Predict+Student%E2%80%99s+academic+Performance+using+Ensemble+Methods&author=Amrieh,+E.A.&author=Hamtini,+T.M.&author=Aljarah,+I.&publication_year=2016&journal=Int.+J.+Database+Theory+Appl.&volume=9&pages=119%E2%80%93136&doi=10.14257/ijdta.2016.9.8.13)] [[**Crossruff**](https://doi.org/10.14257/ijdta.2016.9.8.13)]
12. Rahman, M.H.; Islam, M.R. Predict Student’s Academic Performance and Evaluate the Impact of Different Attributes on the Performance Using Data Mining Techniques. In Proceedings of the 2017 2nd International Conference on Electrical Electronic Engineering (ICEEE), Rajdhani, Bangladesh, 27–29 December 2017; pp. 1–4. [[**Google Scholar**](https://scholar.google.com/scholar_lookup?title=Predict+Student%E2%80%99s+Academic+Performance+and+Evaluate+the+Impact+of+Different+Attributes+on+the+Performance+Using+Data+Mining+Techniques&conference=Proceedings+of+the+2017+2nd+International+Conference+on+Electrical+Electronic+Engineering+(ICEEE)&author=Rahman,+M.H.&author=Islam,+M.R.&publication_year=2017&pages=1%E2%80%934&doi=10.1109/CEEE.2017.8412892)] [[**Crossruff**](https://doi.org/10.1109/CEEE.2017.8412892)]
13. Kozelek, J.; Hosta, M.; Rahal, Z. Open university learning analytics dataset. *Sci. Data* **2017**, *4*, 170171. [[**Google Scholar**](https://scholar.google.com/scholar_lookup?title=Open+university+learning+analytics+dataset&author=Kuzilek,+J.&author=Hlosta,+M.&author=Zdrahal,+Z.&publication_year=2017&journal=Sci.+Data&volume=4&pages=170171&doi=10.1038/sdata.2017.171)] [[**Crossruff**](https://doi.org/10.1038/sdata.2017.171)] [[**Green Version**](https://www.nature.com/articles/sdata2017171.pdf)]
14. Adnan, M.; Habib, A.; Ashraf, J.; Mus Sadiq, S.; Raza, A.A.; Abid, M.; Bashir, M.; Khan, S.U. Predicting at-Risk Students at Different Percentages of Course Length for Early Intervention Using Machine Learning Models. *IEEE Access* **2021**, *9*, 7519–7539. [[**Google Scholar**](https://scholar.google.com/scholar_lookup?title=Predicting+at-Risk+Students+at+Different+Percentages+of+Course+Length+for+Early+Intervention+Using+Machine+Learning+Models&author=Adnan,+M.&author=Habib,+A.&author=Ashraf,+J.&author=Mussadiq,+S.&author=Raza,+A.A.&author=Abid,+M.&author=Bashir,+M.&author=Khan,+S.U.&publication_year=2021&journal=IEEE+Access&volume=9&pages=7519%E2%80%937539&doi=10.1109/ACCESS.2021.3049446)] [[**Crossruff**](https://doi.org/10.1109/ACCESS.2021.3049446)]
15. Jha, N.; Georgescu, I.; Moldovan, A. OULAD MOOC Dropout and Result Prediction using Ensemble, Deep Learning and Regression Techniques. In Proceedings of the 11th International Conference on Computer Supported Education, Heraklion, Greece, 2–4 May 2019; Scoters: Setubal, Portugal, 2019; Volume 2, pp. 154–164. [[**Google Scholar**](https://scholar.google.com/scholar_lookup?title=OULAD+MOOC+Dropout+and+Result+Prediction+using+Ensemble,+Deep+Learning+and+Regression+Techniques&conference=Proceedings+of+the+11th+International+Conference+on+Computer+Supported+Education&author=Jha,+N.&author=Ghergulescu,+I.&author=Moldovan,+A.&publication_year=2019&pages=154%E2%80%93164&doi=10.5220/0007767901540164)] [[**Crossruff**](https://doi.org/10.5220/0007767901540164)]
16. Song, X.; Li, J.; Sun, S.; Yin, H.; Dawson, P.; Doss, R.R.M. SEPN: A Sequential Engagement Based Academic Performance Prediction Model. *IEEE Intel. Syst.* **2021**, *36*, 46–53. [[**Google Scholar**](https://scholar.google.com/scholar_lookup?title=SEPN:+A+Sequential+Engagement+Based+Academic+Performance+Prediction+Model&author=Song,+X.&author=Li,+J.&author=Sun,+S.&author=Yin,+H.&author=Dawson,+P.&author=Doss,+R.R.M.&publication_year=2021&journal=IEEE+Intell.+Syst.&volume=36&pages=46%E2%80%9353&doi=10.1109/MIS.2020.3006961)] [[**Crossruff**](https://doi.org/10.1109/MIS.2020.3006961)]
17. Hussain, M.; Zhu, W.; Zhang, W.; Abidi, R. Student Engagement Predictions in an e-Learning System and Their Impact on Student Course Assessment Scores. *Comput. Intel. Neurosis.* **2018**, *2018*, 6347186. [[**Google Scholar**](https://scholar.google.com/scholar_lookup?title=Student+Engagement+Predictions+in+an+e-Learning+System+and+Their+Impact+on+Student+Course+Assessment+Scores&author=Hussain,+M.&author=Zhu,+W.&author=Zhang,+W.&author=Abidi,+R.&publication_year=2018&journal=Comput.+Intell.+Neurosci.&volume=2018&pages=6347186&doi=10.1155/2018/6347186&pmid=30369946)] [[**Crossruff**](https://doi.org/10.1155/2018/6347186)] [[**PubMed**](http://www.ncbi.nlm.nih.gov/pubmed/30369946)] [[**Green Version**](http://downloads.hindawi.com/journals/cin/2018/6347186.pdf)]
18. Stanford, U. Centre for Advanced Research through Online Learning (CAROL). Available online: [**https://carol.stanford.edu**](https://carol.stanford.edu/) (accessed on 24 August 2021).
19. Mubarak, A.A.; Cao, H.; Hizam, I.M. Deep analytic model for student dropout prediction in massive open online courses. *Comput. Electra. Eng.* **2021**, *93*, 107271. [[**Google Scholar**](https://scholar.google.com/scholar_lookup?title=Deep+analytic+model+for+student+dropout+prediction+in+massive+open+online+courses&author=Mubarak,+A.A.&author=Cao,+H.&author=Hezam,+I.M.&publication_year=2021&journal=Comput.+Electr.+Eng.&volume=93&pages=107271&doi=10.1016/j.compeleceng.2021.107271)] [[**Crossruff**](https://doi.org/10.1016/j.compeleceng.2021.107271)]
20. Mouridi, Y.; Saigal, M.; El Katine, H.; Berrada Fathi, W. A machine learning-based methodology to predict learners’ dropout, success or failure in MOOCs. *Int. J. Web Inf. Syst.* **2019**, *15*, 489–509. [[**Google Scholar**](https://scholar.google.com/scholar_lookup?title=A+machine+learning-based+methodology+to+predict+learners%E2%80%99+dropout,+success+or+failure+in+MOOCs&author=Mourdi,+Y.&author=Sadgal,+M.&author=El+Kabtane,+H.&author=Berrada+Fathi,+W.&publication_year=2019&journal=Int.+J.+Web+Inf.+Syst.&volume=15&pages=489%E2%80%93509&doi=10.1108/IJWIS-11-2018-0080)] [[**Crossruff**](https://doi.org/10.1108/IJWIS-11-2018-0080)]
21. Mouridi, Y.; Saigal, M.; Berrada Fathi, W.; El Katine, H. A Machine Learning Based Approach to Enhance Mooch Users’ Classification. *Turk. Online J. Distance Educ.* **2020**, *21*, 47–68. [[**Google Scholar**](https://scholar.google.com/scholar_lookup?title=A+Machine+Learning+Based+Approach+to+Enhance+Mooc+Users%E2%80%99+Classification&author=Mourdi,+Y.&author=Sadgal,+M.&author=Berrada+Fathi,+W.&author=El+Kabtane,+H.&publication_year=2020&journal=Turk.+Online+J.+Distance+Educ.&volume=21&pages=47%E2%80%9368&doi=10.17718/tojde.727976)] [[**Crossruff**](https://doi.org/10.17718/tojde.727976)]
22. KDD. KDD Cup 2015. Available online: [**https://kdd.org/kdd-cup**](https://kdd.org/kdd-cup) (accessed on 24 August 2021).
23. Lai, S.; Zhao, Y.; Yang, Y. Broad Learning System for Predicting Student Dropout in Massive Open Online Courses. In Proceedings of the 2020 8th International Conference on Information and Education Technology, Okayama, Japan, 28–30 March 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 12–17. [[**Google Scholar**](https://scholar.google.com/scholar_lookup?title=Broad+Learning+System+for+Predicting+Student+Dropout+in+Massive+Open+Online+Courses&conference=Proceedings+of+the+2020+8th+International+Conference+on+Information+and+Education+Technology&author=Lai,+S.&author=Zhao,+Y.&author=Yang,+Y.&publication_year=2020&pages=12%E2%80%9317&doi=10.1145/3395245.3395252)] [[**Crossruff**](https://doi.org/10.1145/3395245.3395252)]
24. Zhou, Y.; Xu, Z. Multi-Model Stacking Ensemble Learning for Dropout Prediction in MOOCs. *J. Phys. Conf. Ser.* **2020**, *1607*, 012004. [[**Google Scholar**](https://scholar.google.com/scholar_lookup?title=Multi-Model+Stacking+Ensemble+Learning+for+Dropout+Prediction+in+MOOCs&author=Zhou,+Y.&author=Xu,+Z.&publication_year=2020&journal=J.+Phys.+Conf.+Ser.&volume=1607&pages=012004&doi=10.1088/1742-6596/1607/1/012004)] [[**Crossruff**](https://doi.org/10.1088/1742-6596/1607/1/012004)]
25. Ho, A.D.; Reich, J.; Nesterov, S.; Seaton, D.T.; Mullaney, T.; Waldo, J.; Chuang, I. *Harvard and MITx: The first year of Open Online Courses*; Harvard and MITx Working Paper No. 1; Harvard University: Cambridge, MA, USA, 2014. [[**Google Scholar**](https://scholar.google.com/scholar_lookup?title=HarvardX+and+MITx:+The+first+year+of+Open+Online+Courses&author=Ho,+A.D.&author=Reich,+J.&author=Nesterko,+S.&author=Seaton,+D.T.&author=Mullaney,+T.&author=Waldo,+J.&author=Chuang,+I.&publication_year=2014)]
26. Imran, A.; Dalipi, F.; Kastrati, Z. Predicting Student Dropout in a MOOC: An Evaluation of a Deep Neural Network Model. In Proceedings of the 2019 5th International Conference on Computing and Artificial Intelligence, Bali, Indonesia, 19–22 April 2019; pp. 190–195. [[**Google Scholar**](https://scholar.google.com/scholar_lookup?title=Predicting+Student+Dropout+in+a+MOOC:+An+Evaluation+of+a+Deep+Neural+Network+Model&conference=Proceedings+of+the+2019+5th+International+Conference+on+Computing+and+Artificial+Intelligence&author=Imran,+A.&author=Dalipi,+F.&author=Kastrati,+Z.&publication_year=2019&pages=190%E2%80%93195&doi=10.1145/3330482.3330514)] [[**Crossruff**](https://doi.org/10.1145/3330482.3330514)]
27. Al-Sha bandar, R.; Hussain, A.; Laws, A.; Keight, R.; Lunn, J.; Radi, N. Machine learning approaches to predict learning outcomes in Massive open online courses. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; pp. 713–720. [[**Google Scholar**](https://scholar.google.com/scholar_lookup?title=Machine+learning+approaches+to+predict+learning+outcomes+in+Massive+open+online+courses&conference=Proceedings+of+the+2017+International+Joint+Conference+on+Neural+Networks+(IJCNN)&author=Al-Shabandar,+R.&author=Hussain,+A.&author=Laws,+A.&author=Keight,+R.&author=Lunn,+J.&author=Radi,+N.&publication_year=2017&pages=713%E2%80%93720&doi=10.1109/IJCNN.2017.7965922)] [[**Crossruff**](https://doi.org/10.1109/IJCNN.2017.7965922)]
28. Liu, K.F.-R.; Chen, J.-S. Prediction and assessment of student learning outcomes in calculus a decision support of integrating data mining and Bayesian belief networks. In Proceedings of the 2011 3rd International Conference on Computer Research and Development, Shanghai, China, 11–13 March 2011; Volume 1, pp. 299–303. [[**Google Scholar**](https://scholar.google.com/scholar_lookup?title=Prediction+and+assessment+of+student+learning+outcomes+in+calculus+a+decision+support+of+integrating+data+mining+and+Bayesian+belief+networks&conference=Proceedings+of+the+2011+3rd+International+Conference+on+Computer+Research+and+Development&author=Liu,+K.F.-R.&author=Chen,+J.-S.&publication_year=2011&pages=299%E2%80%93303&doi=10.1109/ICCRD.2011.5764024)] [[**Crossruff**](https://doi.org/10.1109/ICCRD.2011.5764024)]
29. Wu, W.H.; Jim Wu, Y.C.; Chen, C.Y.; Kao, H.Y.; Lin, C.H.; Huang, S.H. Review of Trends from Mobile Learning Studies: A Meta-Analysis. *Comput. Educ.* **2012**, *59*, 817–827. [[**Google Scholar**](https://scholar.google.com/scholar_lookup?title=Review+of+Trends+from+Mobile+Learning+Studies:+A+Meta-Analysis&author=Wu,+W.H.&author=Jim+Wu,+Y.C.&author=Chen,+C.Y.&author=Kao,+H.Y.&author=Lin,+C.H.&author=Huang,+S.H.&publication_year=2012&journal=Comput.+Educ.&volume=59&pages=817%E2%80%93827&doi=10.1016/j.compedu.2012.03.016)] [[**Crossruff**](https://doi.org/10.1016/j.compedu.2012.03.016)]
30. Chen, W.; Brinton, C.G.; Cao, D.; Mason-Singh, A.; Lu, C.; Chiang, M. Early Detection Prediction of Learning Outcomes in Online Short-Courses via Learning Behaviours. *IEEE Trans. Learn. Technol.* **2019**, *12*, 44–58. [[**Google Scholar**](https://scholar.google.com/scholar_lookup?title=Early+Detection+Prediction+of+Learning+Outcomes+in+Online+Short-Courses+via+Learning+Behaviors&author=Chen,+W.&author=Brinton,+C.G.&author=Cao,+D.&author=Mason-Singh,+A.&author=Lu,+C.&author=Chiang,+M.&publication_year=2019&journal=IEEE+Trans.+Learn.+Technol.&volume=12&pages=44%E2%80%9358&doi=10.1109/TLT.2018.2793193)] [[**Crossruff**](https://doi.org/10.1109/TLT.2018.2793193)]