GOLDSMITHS, UNIVERSITY OF LONDON

DOCTORAL THESIS

Predicting Student Performance on Virtual Learning Environment

Author:
Fatema Mohammad Alnassar

Supervisor:
Dr.Tim Blackwell
Dr. Elaheh Homayounvala
Dr. Matthew Yee-King

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy in the

Department of Computing
Goldsmiths, University of London

September 2023

Declaration of Authorship

I, Fatema Mohammad Alnassar, declare that this thesis titled, “Predicting Student Performance on Virtual Learning Environment" and the work presented in it are my own. I confirm that this thesis has not been previously submitted for a degree or other qualifications at any other university.

Signature

Date

Abstract

Virtual learning has gained increased importance because of the recent pandemic situation. A mass shift to virtual means of education delivery has been observed over the past couple of years, forcing the community to develop efficient performance assessment tools. Prediction of students performance using different relevant information has emerged as an efficient tool in educational institutes towards improving the curriculum and teaching methodologies. Automated analysis of educational data using state of the art Machine Learning (ML) and Artificial Intelligence (AI) algorithms is an active area of research.

The research presented in this thesis addresses the problem of students performance prediction comprehensively by applying multiple machine learning models (i.e., Multilayer Perceptron (MLP), Decision Tree (DT), Random Forest (RF), Extreme Gradient Boosting (XGBoost), CATBoost, K-Nearest Neighbour (KNN) and Support Vector Classifier (SVC)) on the two benchmark VLE datasets (i.e., Open University Learning Analytics Dataset (OULAD), Coursera). In this context, a series of experiments are performed and important insights are reported. First, the classification performance of machine learning models has been investigated on both OULAD and Coursera datasets. In the second experiment, performance of machine learning models is studied for each course of Coursera dataset and comparative analysis are performed. From the Experiment 1 and Experiment 2, the class imbalance is reported as the highlighted factor responsible for degraded performance of machine learning models. In this context, Experiment 3 is designed to address the class imbalance problem by making use of multiple Synthetic Minority Oversampling Technique (SMOTE) and generative models (i.e., Generative Adversial Networks (GANs)). From the results, SMOTE NN approach was able to achieve best classification performance among the implemented SMOTE techniques. Further, when mixed with generative models, the SMOTENN-GAN generated Coursera dataset was the best on which machine learning models were able to achieve the classification accuracy around 90%. Overall, MLP, XGBoost and CATBoost machine learning models were emerged as the best performing in context to different experiments performed in this thesis.

Keywords: Virtual Learning Environment, Machine Learning, Classification, Students Performance Prediction, Generative Adversial Networks (GANs), SMOTE, Artificial Intelligence.

Acknowledgements

First and foremost, thanks to Allah almighty for enabling me to achieve this huge milestone in my academic career.

I would like to thank my principal supervisor Professor Tim Blackwell and my co-supervisors Dr. Elaheh Homayounvala and Dr. Matthew Yee-King for their unprecedented advice throughout my PhD. Without their right guidance, support and leadership, I would have not achieved this milestone.

I would like to express my gratitude to my family, my parents, my husband and my children who stood beside me in these challenging years. Especially, my parents and husband who were directly influenced and always supported me during this journey.

Contents

List of Figures

List of Tables

List of Acronyms

k-NN k-Nearest Neighbour

AI Artificial Intelligence

AIRS Artificial Immune Recognition System

ANN Artificial Neural Networks

ARM Academic Resistance Model

BLR Bayesian Linear Regression

BP Back propagation

CART Classification And Regression Tree

CF Collaborative Filtering

CHAID Chi-square Automatic Interaction Detection

DE Differential Evolution

DNN Deep Neural Network

DT Decision Tree

EDM Educational Data Mining

GAN Generative Adversarial Networks

GP Gaussian Process

ICT Information and Communication Technologies

ID3 Iterative Dichotomiser 3

iLMS intelligence Learner Management System

IMD Index of Multiple Deprivation

ITU International Technical University

LAK Learning Analytics Knowledge

LD Learning Design

LMS Learning Management System

LogR Logistic Regression

LR Linear Regression

LSTM Long-Short Term Memory

MARS Multivariate Adaptive Regression Splines

MF Matrix Factorization

ML Machine Learning

MLP Multilayer Perceptron

NB Naive Bayes

OULAD Open University Learning Analytics Dataset

PCA Principal Component Analysis

PLA Predictive Learning Analytics

RBM Restricted Boltzman Machines

RF random forest

RLR Regularized Linear Regression

RMSE Root Mean Square Error

SAPD Students Academic Performance Dataset

SMOTE Synthetic Minority Oversampling Technique

SPD Students' Performance Dataset

SPM Student Probabilities Model

SSTs Student Support Teams

SVM Support Vector Machine

SVR Support Vector Regressor

TAM Technology Acceptance Model

TDIDT Top Down Induction of Decision Trees

UI User Interface

VAE Variational Auto Encoder

VLE Virtual Learning Environment

XGBoost Extreme Gradient Boosting

Chapter 1

Introduction

Technological advances over the years, have influenced the educational system by introducing mobile devices, software systems, internet and range of electronic gadgets. Over the period of last fifty years, technology has paved its way into the education sector in context to online learning. Online learning has a rich history that spans several decades mainly starting from 1960s when the concept of Computer-Based Training (CBT) emerged, utilizing mainframe computers to deliver educational content [2, 3]. The 1970s saw the development of Programmed Logic for Automatic Teaching Operations (PLATO), one of the first online learning platforms [4, 5]. The 1980s witnessed the Western Behavioral Sciences Institute offering the first fully online degree program, while the rise of telecommunications networks and personal computers brought new opportunities for computer-based learning [6, 7]. The 1990s marked a significant milestone as the internet became more accessible, leading to the development of online learning systems. The University of Phoenix launched its online division, and web-based Learning Management Systems (LMS) like Blackboard and Moodle emerged as centralized platforms for online course delivery [8].

The 2000s brought about the popularity of Massive Open Online Courses (MOOCs), which offered free online courses to a global audience [9]. Platforms such as Coursera, edX, and Udacity contributed to the MOOC movement. Virtual classrooms and web conferencing tools like Adobe Connect and Zoom improved synchronous online learning experiences. The proliferation of smartphones and tablets made mobile learning (m-learning) feasible, allowing learners to access educational content on the go. In the 2010s, online learning became increasingly mainstream, with traditional universities and colleges integrating online courses into their curriculum.

Blended learning, combining online and face-to-face instruction, gained recognition as an effective educational approach. Adaptive learning technologies emerged, utilizing data and analytics to personalize learning experiences. Open Educational Resources (OER) and OpenCourseWare initiatives promoted the sharing of educational content freely online [10]. The 2020s brought a seismic shift in online learning due to the COVID-19 pandemic. Educational institutions worldwide rapidly shifted to remote teaching and learning, leading to a significant surge in online learning. Video conferencing tools, learning management systems, and other online platforms became even more prevalent. Augmented Reality (AR) and Virtual Reality (VR) technologies started showing promise in creating immersive learning experiences [8, 11].

In recent years, virtual learning through online platforms (i.e., Virtual Learning Environment (VLE)) has emerged as a potential means of education mainly because of technological advancements in computers and the easy availability of the internet [12]. The significance of VLE was first noticed in higher education and later, its usefulness was observed among the undergraduate students' [13]. Jani et al. [14] argued that blended learning of face-to-face and using the VLE platform increased the students' understanding and performance. There are multiple VLE platforms available nowadays, however, the most commonly used within the education sector is Moodle.

The Coursera platform is also widely used for virtual learning and is among one of the most accessed education material providers. The recent COVID-19 pandemic situation has further highlighted the need and importance of virtual education because of the shift in teaching from on-campus to online in most educational institutes [15, 16]. As of now, most educational institutes are using custom online tools and VLE for teaching. The COVID-19 crisis has led to the rise of unprecedented challenges for education. There has been a shift to fully utilise the virtual learning capacities to provide alternative learning experiences to students [17].

A reliable and effective assessment of students' performance on a virtual learning platform is a very challenging and unconventional task to achieve [18]. The prediction of students' performance will help instructors at the initial stages of the course to care for students who need help by encouraging them to do the classwork [19]. Each institute is following a different means of assessing students [20, 21, 22], however, it is far from the real performance of students mainly because of external help, the inability to replicate exam room isolated environment and the pressure of time.

The inability to evaluate the close-to-real performance of students has raised questions on the quality of education in the past and on-campus education has always been given preference. However, the recent mass shift to virtual learning due to the pandemic has exponentially escalated the issue and forced the community to develop efficient performance assessment mechanisms and tools. It is now important to inquire into the contributions of VLE to students' performance and various factors that might affect it. Adopting of these environments has made it necessary to evaluate students' results in VLE may differ from regular teaching modes (face-to-face) as the most affected factor is location and time difference.

The difference in the teaching mode and results obtained by students through VLE requires verification if they are eligible and how to compare the results obtained from face-to-face learning and VLE. This difference shows the importance of anticipating the students' performance in various courses and programs that stem from the fact that prediction creates real opportunities for improving educational outcomes and finding patterns of students' VLEs usage [23]. Among others, Educational Data Mining (EDM) which is a process used to extract useful information and patterns from a huge educational database [24] has emerged as the most effective approach.

It involves the application of data mining techniques to data obtained from students' use of VLEs, actions and behaviors recorded in it [25]. Artificial Intelligence (AI) learning based approaches (e.g., classification, regression) have been found effective in predicting students' performance based on the data collected through VLE. Effectiveness and accuracy of prediction is highly dependent on the data type of features being used, size of a dataset and diversity in the dataset.

1.1 Problem Statement

To improve the overall quality of education systems, special emphasis has been paid to new strategies of Information and Communication Technologies (ICT) in higher education. There are many examples of employing ICT in higher education, such as the use of distance learning, blended E-learning and VLEs [26]. Although VLE has succeeded in becoming a common practice in current educational systems, however, predicting students' performance is still a complex process as it depends on several variables [27]. Despite efforts being exerted to improve students' performance, high percentages of students' are at risk of academic failure, dropout and low achievements.

The problem lies in the fact that predicting performance itself is a challenging task, as many factors can affect the prediction process and the large volume of data associated with these factors. Cheating during the e-assessment by accessing information from the Internet, written materials and other helpful stuff [28] has been identified as one of the significant factors to identify and prevent from while making the assessment. Although, this is also an important aspect to explore, however is not addressed under the scope of the research presented in this thesis.

For years, distant learning has been criticized for its unreliable grading criteria and assessment approaches and is given less consideration in comparison to in-person face to face education [29, 30]. Conventionally, distant learning through VLE (i.e., method of education where students and instructors are physically separated, and learning takes place through the use of various instructional materials and communication technologies [30, 31]) uses timed quiz, discussions, graded assignments, case studies and timed exams to evaluate the performance of students' during a certain course. However, ensuring that student himself/herself is attempting all the assessments and virtually determining the behaviour of student is still a research gap to be addressed. This may be a potential future research lead by the classification results reported in this thesis. Recent COVID-19 pandemic has forced the whole education system to adapt for VLE and hence the problem of virtual assessment of students' has been scaled up significantly.

As of now, there are no standard procedures around the world, rather different educational institutes are adapting different experimental setups. Given the success of Machine Learning (ML) and AI in different domains, there is a scope of investigating different AI-oriented solutions for e-assessment of students' in VLE. The impact of different features on prediction performance is another domain of research that needs to be explored. The approach adopted in the presented thesis is to perform a comprehensive literature survey to identify the significant features related to predicting students' performance. The most important features in existing dataset, with high correlation, include course design features, teaching style, access pattern and student life circumstances. Listed are brief details about each influential feature:

- **Course Design:** The course design is the resources available on the VLE platform for all courses, including videos, quizzes, assignments and reading materials. The course design is assumed to be taken as a static value, since it exists at the start of the course. Each material number, type and length affects student performance [32]. The number of learning materials taken by the student depends on one's intentions [32]. However, the level of student involvement in discussion forums is an effective way to exchange information. Based on the course content prepared by the instructor, this number is a key issue to encourage student interactions in the learning materials in the course [33].

- **Teaching Style:** The teaching style is taken as a dynamic value that depends on what teacher does after what the exists at the start of the course. The number of scheduled teaching activities determines the teaching style and whether the instructor uses live or offline sessions. The author [34] referred to the teaching style as Learning Design (LD). Each VLE platform contains a different type of data about teaching style. Also, each instructor builds his strategies and techniques under the guidance of the information flow to deliver the most comprehensible information to the students. The course curriculum acts as the foundation to establish the route map by the instructor.

- **Access Pattern:** The access pattern is the data accessed by the student or teacher, such as time spent on learning materials, the number of times students' participate in the course forum and number of hits (mouse clicks) per resource students' do. This feature is taken as a dynamic value. Previous researchers showed a strong relationship between the access pattern and student performance [35]. These students' access patterns are determined as behavioural features, and the number of actions by a student during the course is recognized as the access pattern.

- **Student Life Circumstances:** Student life circumstances may affect his performance and also life circumstances about student living environment. The essential data for every student's circumstances, affects the performance and learning path. Student circumstances such as student employment status (part/full-time), marital status, number of children, student first language and studying language and type of study funding (self-funded or scholarship). Student performance may differ depending on their demographics and life circumstances. Some students' circumstances are dynamic and in some cases keep fixed like student employment status, marital status and the number of children could be changed.

ML and AI techniques have been employed in research studies to forecast student performance by leveraging diverse features. However, the literature lacks uniformity in reporting the accuracy of predictions across publicly available datasets. An acknowledged challenge within these datasets is the existence of class imbalance, a condition where the distribution of classes is significantly skewed, potentially leading to biased predictions and impacting model performance. To address the aforementioned issue, it is imperative to consider the class imbalance problem as an integral part of the problem statement.

1.2 Research Questions and Thesis Objectives

To address the problem of predicting students' performance over VLE, a number of corresponding research questions have been formulated. The scope of thesis is restricted to classification of students' grades using the latest learning-based models. In this context, research questions are defined as four-fold:

1. What are the most influential and significant features related to prediction of students' performance over VLE?
   (a) To what extent does the course design (i.e., number of quizzes, assignments) affect students' performance on VLEs?
   (b) What is the link between student access pattern (i.e., time spend on VLE, number of hits) and students performance on VLEs?

2. To what extent students' performance can be predicted using classification models on VLE?
3. To what extent balancing techniques can be used to improve the class imbalance problem in existing VLE datasets?
4. To what extent available techniques can be used to improve the class imbalance problem in existing VLE datasets?

To address the formulated research questions, following tasks (T) were performed in this thesis:

- T1 : ML classification approaches are implemented using existing datasets (i.e., Open University Learning Analytics Dataset (OULAD), Coursera) to predict the failure or success of students. The impact of influential features on the classification performance is also be evaluated.
- T2 : Multiple Synthetic Minority Oversampling Technique (SMOTE) techniques are implemented to balance the Coursera dataset and their impact on the ML classification performance is studied.
- T3 : The state of the art generative models are implemented as a potential approach to simulate the data and compare the results with VAE. The role of simulated data is evaluated in the performance of classification models.

1.3 Research Limitations

This section states the research limitations faced during this study. These limitations listed are as follows:

- Some of the student information about their life aspects like (demographic information and student life circumstances) could be examined through survey, but survey publishing was restrained by the University of London.
- Limited access to information on the Coursera dataset, which is restricted by University of London. Some information like "final grade" is not available on the Coursera dataset.

1.4 Chapters Outline

This thesis contains seven chapters, which aligned with the research strategy 1.1:

- Chapter 2: It shows the previous studies approaches and results, eliminating the potential research aspects of student performance prediction on VLEs.
- Chapter 3: It describes the datasets used in this research and the research approach followed by the study designed.
- Chapter 4: This chapter applies ML models on the Coursera and OULAD datasets (RQ2).
- Chapter 5: It shows the significant features in predicting students' performance based on each subject of the Coursera dataset (RQ1).
- Chapter 6: It applies the potential techniques of class imbalance, which contains the state of the art method (RQ3, RQ4).
- Chapter 7: This chapter summarizes the main contribution of this thesis and possible future studies.

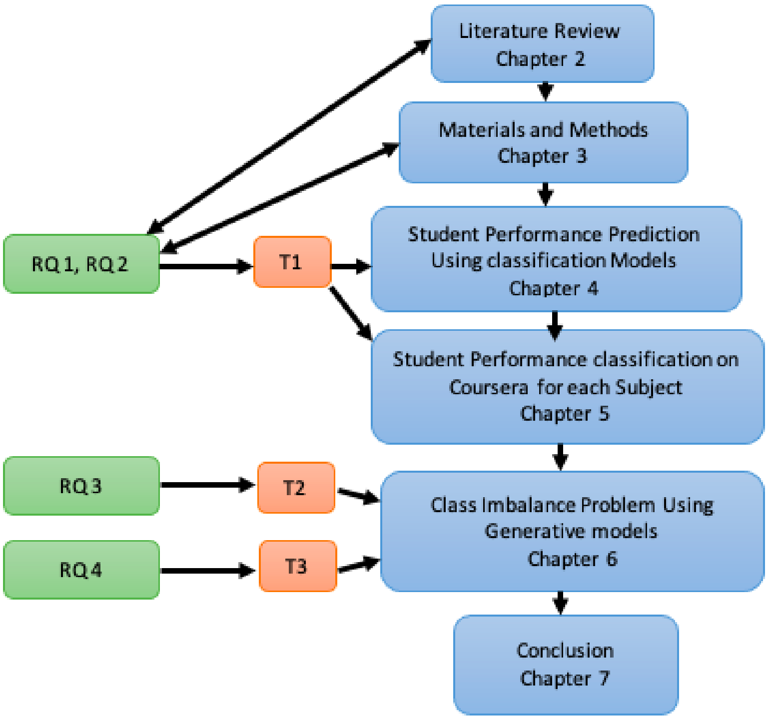


Figure 1.1: Overview of thesis outline.

Chapter 2

Literature Review

Presented systematic literature review is performed using the guidelines reported by [36]. A list of relevant keywords was prepared to explore the literature for formulated research questions. Combination of “Machine Learning", “Artificial Intelligence", “Educational Data Mining", “Virtual Learning Datasets", “Students’ Performance Prediction", “Learning Material", “Teaching Styles",“Student Access Patterns" and “Student social context". Academic databases including IEEE Xplore, Scopus and Science Direct were searched against the combinations of listed keywords. The resulting literature was then filtered using inclusion/exclusion criteria. This study, only included studies (articles) written in English and published in the last three decades. The main filtering criteria was the subjective manual screening of extracted literature to determine the scope of literature entry to be included in the presented review. The relevance of the literature was determined based on a sequential process of title screening, abstract screening and full article study. For this study, only the benchmark state of the art literature entries that were included published after year, 2015. The focus of the presented review is a ML-based investigation of the impact of course design, teaching style, teacher access pattern, student access patterns and student life circumstances on the performance of students. Recently in 2021, the four factors were illustrated by [37]. In total, 13 benchmark literature review, entries were shortlisted to be included in the main review.

This chapter presents a systematic literature review to investigate the impact of course design, teaching style, teacher access pattern, student access patterns and student life circumstances on the performance of student. Focus of the review is to identify the key ML-based technologies and methods used to predict the performance of students. Furthermore, relevant literature is critically analysed to highlight the potential shortcomings reported in the literature regarding the prediction of students' performance. The chapter also outlines the recent research performed using the OULAD dataset and briefly o utlines t he a dopted m ethodology t o p reform the literature review.

The chapter is structured as follows: Section 2.1 provides basic information about the VLE and its importance in the current time, Section 2.2 introduces the basic concepts of EDM as a context to presented review, Section 2.3 presents the fundmental concepts of LMS, Section 2.4 provides the detailed information about the MOOC, and Section 2.5 is the main section and presents the review of most relevant literature in chronological order to highlight the advancements in this domain.

2.1 Virtual Learning Environment (VLE)

VLE is a digital platform or soft-ware application that allows educational courses and resources to be delivered and managed online. It provides tools for course organization, content delivery, com-munication, collaboration, assessment, and tracking of student progress [38, 39]. Interaction of students and teachers through VLE creates an interactive learning en-vironment where students can benefit from the online materials and interactions with teachers. Although, VLE is not an ideal platform to fit in all scenarios [40], however, proposes several creative and flexible options for learning and teaching. Some highlighted advantages of VLE include flexibility in use, cost and time effi-cient, remote access and interactive learning environment. Some limitations of VLE include requirement of high speed internet connection, reliable network, availability of computer and lack of assessment tools. Measuring the performance of students on VLE is one of the most discussed limitations [41] of VLE mainly because of insuf-ficient face-to-face communication and lack of in-class assessments. Despite all of its disadvantages, VLE has played a key role during the COVID-19 pandemic [42, 43] situations and is used as a main learning tool.

The shift towards VLEs and technology-enhanced education has gained significant attention in recent times, replacing traditional classroom methods in many educational institutions [44]. VLEs, also known as e-learning systems, have been embraced by colleges and universities to enhance the learning process. These platforms provide virtual spaces for teaching and learning, supplementing face-to-face interactions with web technologies [45, 46]. The use of VLEs allows for the collection of valuable data on study syllabi and student engagement, which can be analyzed to improve educational experiences and facilitate data-driven strategic planning [47]. VLEs serve as facilitators in the educational process by fostering communication, understanding, and interpretation among learners. Through educational projects and digital provisions of materials, notifications, and chat rooms, VLEs simulate the traditional classroom experience while offering creative teaching approaches [48, 49]. The accessibility of extensive datasets in technology-enhanced learning systems enables the analysis of learners' academic abilities, problem resolution, and performance evaluation. By employing data mining methods, such as classification modules, predictions can be made about student enrollment, academic achievement, and identification of test procedures that may need adjustment [50, 51].

Research indicates that the characteristics of VLEs and the students' interactions with these environments have a significant impact on academic achievement. Factors such as the number of logs, volume of discussion activity, and utilization of tools are indicators of enhanced achievement in virtual learning programs [52, 53, 54, 55]. However, it is important to consider the disparities in technological use among learners and the influence of the learning environment on educational performance [56, 57, 58]. Ignoring the learning environment may lead to over or underestimation of the effects of VLE characteristics on academic achievement. Furthermore, educational data mining can be applied to identify learners' strengths and weaknesses, allowing teachers to enhance instructional strategies accordingly [59]. Personalized teaching and learning techniques based on analyzed datasets provide valuable insights into students' performance, attentiveness, and goals. This information enables teachers to modify courses, themes, and approaches to meet individual student needs. Notably, research suggests that virtual, interactive, real-time, instructor-led (VIRI) online learning environments have a similar impact on student performance as face-to-face courses, highlighting the effectiveness of virtual learning platforms [60].

2.2 Educational Data Mining (EDM)

EDM involves an application of data mining techniques to data obtained from students' use of VLEs, actions and behaviors recorded in it. A general definition of EDM was given by [61]. Data Mining is a multidisciplinary approach that involves the analysis of psychometric, educational psychology, social networks and cognitive psychology to obtain information from large amounts of data [62, 63]. EDM analyses learning systems data and takes into account administrative, demographic and motivational variables that might affect these systems. Some of the challenges of the e-learning were presented by [64].

It analyses the data collected from students involved in the VLE in terms of their facial expressions and conversations as an audio file. EDM in higher education institutions is an emerging research topic; it has succeeded in gained popularity in the area of academic research due to its positive effects on educational outputs. The students performance can be predicted with different modules, like the classification module that forms a link between students grades and factors affecting them [25].

Educational institutes have shown considerable interest in the prediction of students performance in virtual and personal learning environments [65, 66] based on the different types of datasets towards improving their curriculum and teaching methodologies. EDM and learning analytics are the research domains for automated analysis of educational resources. ML, Artificial Neural Networks (ANN), Collaborative Filtering (CF), Decision Tree (DT), Naive Bayes (NB) and k-Nearest Neighbour (k-NN) are commonly used tools for processing the educational data. Extracted information and patterns from such analysis can be used in different utilities towards improving the overall learning process for students. One such application is to predict the performance of students by analysing the dataset recorded through virtual platforms [67].

The identification of students behaviour in terms of their strengths [68] and weaknesses using EDM can significantly facilitate instructors in improving their teaching methodologies. ML has been used as an application to improve the quality of VLE [69, 70, 71]. It is a sub-field of AI that enables computers to learn from experiences and data to carry out complex processes, instead of depending on pre-determined programmed rules [72]. The use of state-of-the-art ML and AI algorithms for predicting students' performance is an active area of research [19].

2.3 Learning Management System (LMS)

LMS emerged as a dominant and successful technology for education in early 2000, providing a centralized platform for e-learning in universities. Before the LMS, e-learning relied on various tools, resulting in inconsistent quality and varied tool sets across institutions [8]. The LMS offered a comprehensive solution by integrating popular tools into a single platform, enabling universities to progress more rapidly in their e-learning initiatives. However, over time, institutions became locked into contracts with LMS vendors, limiting their flexibility and hindering innovation.

LMS adoption grew significantly in the first half of the decade, with most higher education institutions deploying an LMS. Commercial providers like WebCT and Blackboard, as well as open-source solutions like Moodle and Bodington, were available options [8]. Initially, the LMS was used in familiar ways, resembling virtual classrooms and replicating traditional teaching methods rather than embracing more experimental pedagogies. One challenge with enterprise systems like the LMS is the accumulation of institutional practices and structures around them, creating a form of "software sedimentation" [73]. Institutions invest substantial resources in technology, training, and support, leading to the development of administrative structures aligned with the specific LMS. This focus on the LMS as the solution for all educational needs stifles innovation and limits exploration of alternative approaches.

In the mid-2000s, there was interest in service-oriented architectures, where systems could be assembled using discrete services. This approach would allow institutions to curate the best tools for their specific needs instead of relying on a single LMS. However, the convenience and support offered by specific LMS providers discouraged the pursuit of this approach. Lock-in with particular tools has been a driving factor in the sedimentation process. The Open University (OU) adopted a service-oriented architecture approach in 2004, leveraging the open-source Moodle platform to integrate various tools while maintaining customization options [8]. The OU's adoption of Moodle significantly accelerated their e-learning initiatives. However, the service-oriented architecture approach using protocols like SOAP did not gain widespread traction due to practical limitations and the focus on maintaining and developing their version of Moodle. The investment in Moodle has made it challenging for institutions to transition to other solutions.

In 2007, there were discussions about the decline of the LMS and the emergence of loosely coupled third-party tools. However, the LMS remained prevalent, continuously evolving to incorporate new features and shedding its classroom-centric imagery. The robustness and stability of the LMS have contributed to its enduring popularity, although it often goes unappreciated in the educational technology community. Balancing freedom, innovation, and core functionality is crucial, allowing experimentation within the LMS framework while pushing boundaries at the fringes.

In the context of the history of online learning, the LMS emerged as a central technology in the early 2000s, providing universities with a comprehensive solution for e-learning. It replaced disparate tools and allowed for more consistent and efficient implementation of online education. However, the LMS also presented challenges, including vendor lock-in, limited innovation, and the development of institutional structures centered around the specific technology. Despite debates about its decline, the LMS has persisted and evolved, offering integrated features and remaining a vital component of online education.

2.4 Massive Open Online Courses (MOOCs)

MOOCs emerged in 2012 and was instantly picked by media. The aim of this program was to provide students with online free courses in large numbers and it claimed to revolutionize the education section, however, these claims proved overblown and several issues were identified. The low completion rates, with only around 10% of registered students finishing the courses, have been a persistent problem. Additionally, the demographics of successful MOOC learners revealed that most were already well-educated individuals, undermining the notion of MOOCs democratizing education. Another challenge has been the sustainability of MOOCs, as their costs can vary considerably, particularly when factoring in staff time, marketing, and support [8].

As a result of these challenges, there has been a shift in the approach to MOOCs. Providers like Coursera and Udacity started exploring blended learning models and offering paid courses for credit. The focus shifted from being free providers of education to becoming courseware providers within traditional education systems. This transition, while not necessarily negative, deviated from the initial vision of MOOCs as open and accessible platforms.

Despite the limitations and issues, MOOCs have had practical applications and positive outcomes. Millions of people have enrolled in MOOCs and found them valuable for learning. They have been used to expand formal education by offering credit transfer programs and enhancing the curriculum. MOOCs also raised the profile of educational technology and open practices, driving the adoption of open access resources and fostering a culture of openness. The MOOC phenomenon reflects broader attitudes toward educational technology and the desire to "fix" education. It aligns with the narrative of education being broken and the belief that technology-driven solutions can address the rising costs of higher education. However, this perspective is primarily applicable to the United States of America (USA) context, where higher education is primarily financed by students through debt [8]. In many countries, higher education is still considered a public good, and different models for funding higher education are not explored.

The emphasis on disruption, often associated with MOOCs and educational technology, should be critically examined. Disruption, as a complete systemic change that replaces an entire industry, is rare and not necessarily desirable in the case of education. The notion of educators resisting change needs to be challenged, as many educational technologists actively embrace technology and explore its possibilities. While the initial experimental approach of MOOCs was beneficial for exploring new pedagogy, technology, and subject matter, the subsequent commercialization of MOOCs led to a more conventional approach and limited openness [8]. The potential for MOOCs to be fully open, not just in terms of access but also in terms of reusability and openness of content, was not fully realized.

In summary, the rise of MOOCs has had both positive and negative impacts on online learning and open education. The field of educational technology needs to find a balance between the opportunities presented by MOOCs and the potential pitfalls associated with hype and commercialization.

2.5 Artificial Intelligence (AI) and Machine Learning (ML) for Students' Performance Prediction

This section presents the review of literature where AI and ML are used for the students, performance prediction based on learning materials, teaching style, access patterns and students' social context datasets. Literature is presented in chronological order to highlight the shift in trends over the years, in this domain.

The objective of the ML was deeply studied and presented by [74], While using ML in the educational field was initially presented by Korkmaz et al. [75], Chen et al. [76] and Cheng et al. [77]. The first Principles on Machine Learning Virtual Flow were presented by [78]. Then Elbadrawy et al. [79] implemented a class of linear multi-regression models to predict the performance of students' using the educational data. Models used data features including past performance, interaction with Learning Management System (LMS) and course-related activities. Proposed models were validated on a custom-collected dataset from the University Minnesota of 11,556 student entries and 832 courses studying on-campus. From the results, Root Mean Square Error (RMSE) for the multi-regression models was reported as 0.147, improved from single regression model.

The k-NN based module was predicted by Yee-king et al. [80] in the student's grades from collaborative social learning. The authors implemented a k-NN algorithm to classify students with 2, 3 and 10 grade bands. Multivariate classification approach was used to avoid the weak classification. The proposed approach was validated on a custom-generated dataset from virtual programming course at Coursera in 2014. The gathered dataset consisted of a total number of User Interface (UI) clicks and mouseovers generated during the course. Authors were able to achieve a classification accuracy of 88%, 77% and 31% for 2, 3 and 10 grade bands, respectively. Although, the reasonable performance was reported, however, a comparison of performance with literature was not performed to highlight the scope of the study. Furthermore, the potential of other latest algorithms was not explored for standard datasets.

The performance of students in the final exam using by [81] k-NN and Support Vector Machine (SVM) ML approaches. A custom-collected dataset from the University of Minho, Portugal with 395 data samples, was used to validate the performance of ML models. Dataset consisted of students' background and personal information attributes. From the analysis, relatively comparative results were reported for both approached with SVM slightly better (96% accuracy) than k-NN (95% accuracy).

A comparative study was performed and presented by Iqbal et al. [82] of three different ML approaches including CF, Matrix Factorization (MF) and Restricted Boltzman Machines (RBM) towards predicting the grade of students. A custom collected dataset from International Technical University (ITU), Pakistan with 225 students' entries was used for the validation of ML algorithms. The dataset consisted of performance-based features including previous academic performance and interview score. From the results, the RBM approach was reported best among implemented three with an RMSE of 0.3.

In a similar way a comparative study to predict the students' engagement and its impact on performance using several learning-based algorithms were presented by Hussin et al. [33]. Authors implemented DT, Classification And Regression Tree (CART), JRIP Decision Rules, GB trees and NB classifier on OU acquired dataset to predict the students' engagement. The dataset of only the July, 2013 session (384 records) was used with demographic, performance and learning behaviour features. Authors reported that the J48 decision tree algorithm outclassed others with the highest accuracy of 88.52% and recall of 93.4%.

Heuer and Breiter [83] implemented several machine-learning approaches to identify the risks students from their first assessments. The authors used a standard OULAD Dataset with 32,593 student entries. Activity-based and performance features were used to predict the performance. Authors implemented SVM, NB, random forest (RF), Extreme Gradient Boosting (XGBoost) and Logistic Regression (LogR) ML approaches. From the results, SVM outperformed other implemented algorithms with 87.98% accuracy. Sekeroglu et al. [84] investigated the students performance prediction and classification using a variety of ML algorithms. The authors used Long-Short Term Memory (LSTM), Back propagation (BP) and Support Vector Regressor (SVR) for prediction while BP, SVM and GB classifier for classification. Students' Performance Dataset (SPD) was used for prediction analysis while Students Academic Performance Dataset (SAPD) was used for classification analysis. Datasets mainly included students' demographic, academic background and behavioural features. The authors reported SVR as the best algorithm for prediction while BP for the classification.

Although the authors achieved acceptable results, however, no comparison with existing literature was performed. El fouki et al. [85] proposed an improved classification model based on deep learning and Principal Component Analysis (PCA) for prediction of students performance. The proposed multi-dimensional approach, aimed at reducing the dimensions of data and extracting relevant information from the data to improve the model classification accuracy. A custom-collected dataset with 496 records, consisting of features including students performance, section information and activity participation. The dataset was pre-processed using PCA for dimensional reduction and then analysed using deep learning model, Multilayer Perceptron (MLP) and BayesNet. The authors reported the highest classification accuracy of 92.54% for the deep learning model.

A module model was proposed by Hussain et al. [70] based on internal assessment using deep learning with Adam optimizer to predict students performance. In addition to the deep learning model, two other approaches including Artificial Immune Recognition System (AIRS) v2.0 and AdaBoost were also implemented for comparative investigation. The authors used custom collected dataset of 10,140 records from 3 different colleges in India. The performance of students in multiple tests was the main feature of used dataset towards predicting the final grades. From the results, deep learning model with binary cross entropy loss and sigmoid activation was reported as best with classification accuracy of 95.34%.

Ajibade et al. [71] implemented various classification algorithms on the behavioural learning data of students' to predict the performance. In addition, authors used Differential Evolution (DE) for behavioural feature selection. Proposed approaches were validated against the custom-collected dataset with record of 500 students. Dataset consisted of demographic, academic, learning process and behavioural learning features. DT, k-NN and SVM approaches were applied and DT was reported best among three but not with huge margin. The authors also implemented an ensemble of multiple models using bagging, boosting and random forest approaches towards improving the classification results. Classification accuracy was improved to 91.5% by using an ensemble approach.

Tomasevic et al. [69] performed a comparative study to investigate the effect of different features on students assessment prediction using a variety of ML and statistical approaches. The authors used k-NN, SVM, ANN, DT, Bayesian Linear Regression (BLR) and Regularized Linear Regression (RLR). Authors used a part of the OULAD dataset with demographics, engagement and performance features. F1 score and RMSE were used as performance measures for classification and regression models. Authors reported 96.62% F1 score for ANN using engagement and performance features, while 96.04% SVM (RBF kernel) using demographics, engagement and performance features. Authors predicted the use of deep learning based approaches in near future given the increased data availability.

Hooshyar et al. [86] proposed a novel approach to PPP based on the procrastination behaviour of students to predict their performance. Proposed algorithm focused on students' assignment submission behaviour as main indicator in predicting their performance. The authors validated the proposed approach for a custom-collected dataset of 242 students from the University of Tartu, Estonia. Common ML approaches including Linear SVM (L-SVM), Radial SVM (R-SVM), DT, Gaussian Process (GP), RF, NN, AdaBoost and NB were implemented on the dataset. From the results, NN was reported as best for categorical features with 96% accuracy while L-SVM was reported as overall best with 95% classification accuracy.

Waheed et al. [87] proposed the use of a Deep Neural Network (DNN) for predicting the academic performance of students from the VLE big data. The authors used the OULAD open-source dataset consisted of 32,593 student records. Dataset features included demographics, clickstream behaviour and assessment performance. From the results, the authors reported that the proposed deep learning-based approach outclassed conventional regression and SVM approaches with an accuracy of up to 93%.

From the literature review, the significance of automated analysis on education data has been highlighted. Educational datasets with features of students' access patterns, availability of course design, different teaching styles and students' activities have been used to predict the performance of students. Some standard datasets being used by researchers for performance prediction include OU, OULAD, SPD and SAPD. ML and AI have emerged as a key role in exploiting the educational datasets in comparison to conventional statistical approaches. ML approaches used by researchers include k-NN, DT, CART, JRIB, GBT, NBC, LSTM, BP, SVR, SVM, GBC, DNN, MLP, PCA, BayesNet, ANN, BLR, RLR, CF, MP, RBM, RF, XGBoost, LR, AIRS, AdaBoost, L-SVM, R-SVM, GP, NB and Ensemble. A shift has been observed from literature from conventional ML approaches (i.e., SVR, DT, GBT, PCA, BLR, RLR, NB, CF, MP, XGBoost, LR) towards deep learning approaches (i.e., DNN, MLP, ANN and LSTM). However, the availability of training datasets for deep learning approaches has been one of a major shortcomings till to date. Given the exponential rise in use of VLE due to COVID-19 [88, 40], it is expected to have huge datasets available soon suitable for deep learning approaches in future. F1-Score, accuracy, recall score and RMSE are reported to be commonly used evaluation measures for trained ML models. Table 2.1 presents the comparison of cited literature in terms of proposed approaches, used datasets, types of datasets and results.

Table 2.1: Comparison of Literature Related to use of AI and ML for Students' Performance Prediction

Authors	Proposed Approach	Dataset	Type of Dataset	Results
[79]	Multi-Regression Models	Custom Collected Dataset 11,556 Entries	Past Performance, Interaction with LMS, Course Activities	RMSE of 0.147
[80]	k-NN	Custom Collected Dataset	UI clicks and Mouseovers	88% Classification Accuracy
[81]	k-NN, SVM	Custom Collected Dataset 395 Entries	Students' Background, Personal Information	SVM with 96% Accuracy
[82]	CF, MF, RBM	Custom Collected Dataset 225 Entries	Performance Features	RMSE of 0.3 for RBM
[33]	DT, CART, JRIP, GBT, NBC	OU Dataset 384 Entries	Demographic, Performance and Learning	88.52% for J48 (DT)
[83]	SVM, NB, RF, XGBoost and LR	OULAD Dataset 32,593 Entries	Performance and Activity Features	87.98% for SVM
[84]	LSTM, BP, SVR,SVM, GBC	SPD and SAPD	Demographic, Academic and Behavioural	SVR and BP as best algorithms
[85]	DNN, MLP, PCA,BayesNet	Custom Collected Dataset 496 Entries	Performance and Participation	92.54% for DNN
[70]	DNN, AIRS, AdaBoost	Custom Collected Dataset 10,140 Entries	Performance	95.34% for DNN
[71]	DT, k-NN, SVM, Ensemble	Custom Collected Dataset 500 Entries	Demographic, Academic and Behaviour	91.5% for ensemble approach
[69]	k-NN, SVM, ANN, DT, BLR, RLR	OULAD dataset 32,593 Entries	Demographic, Engagement and Performance	96.04% for SVM (RBF kernel)
[86]	L-SVM, R-SVM, DT, GP, RF, NN, AdaBoost, NB	Custom Collected Dataset 242 Entries	Procrastination Behaviour	95% for L-SVM
[87]	DNN	OULAD Dataset 32,593 Entries	demographics, clickstream behaviour and assessment performance	max 93% for DNN

Table 2.2 presents the comparison type of grouped algorithms in the literature based on proposed approaches, used datasets, types of the dataset, number of entries and results. The algorithms are grouped as ML, DL, and hybrid techniques. There are many researchers who applied ML in the education context with the different number of entries. Based on the results of the ML grouped algorithms, the number of entries is not a comparative indicator in showing improved results in the education domain. As per DL grouped algorithms the past performance is a key indicator to predict student future performance. Some researchers used the hybrid algorithms in predicting students' performance which showed that SVM had the highest results. Input features such as (demographics, engagement and past performance) were the best indicators features over ML and DL algorithms/ with the highest result for SVM.

2.6 Challenges in Student Performance Prediction

Although a significant amount of research has been performed in the context of AI and ML for students' performance prediction, however, a practical solution ~~to the problem~~ in practice is still lacking. Therefore, the research in the field of AI-based prediction of students' performance [89] is still considered an active area of research. Some highlighted limitations of the existing research, extracted from the literature are as follows:

(a) Impact of temporal features on the students' performance prediction is not investigated to its potential despite the significant importance of these features because of the temporal nature of the problem. It is expected that considering this problem as a time-series problem will enhance the prediction performance

(b) Given the technological advancement in data collection approaches and the recent dramatic shift to VLE due to the COVID-19 pandemic [90], a huge number of VLE datasets have emerged. However, it has been observed from the literature that despite the comprehensiveness of the existing different benchmark datasets, models trained on one type of dataset failed to generalize the performance for other similar datasets. This brings to the discussion of cleaning the existing datasets [91, 92] and improving the quality of data towards getting the improved performance (i.e., rubbish in rubbish out)

Table 2.2: Comparison of Literature Related to Algorithms Grouped of ML and DL for Students' Performance Prediction

Author	Proposed Approach	Dataset	Type of dataset	Results
Technique: ML				
[79]	Multi-Regression Models	11,556 Entries	Past Performance, Interaction with LMS, Course Activities	RMSE of 0.147
[80]	k-NN	Custom Dataset	UI clicks and Mouseovers	88% Classification Accuracy
[81]	k-NN, SVM	395 Entries	Students' Background, Personal Information	SVM with 96% Accuracy
[82]	CF, MF, RBM	225 Entries	Performance Features	RMSE of 0.3 for RBM
[33]	DT, CART, JRIP, GBT, NBC	384 Entries	Demographic, Performance and Learning	88.52% for J48 (DT)
[83]	SVM, NB, RF, XGBoost and LR	OULAD Dataset 32,593	Performance and Activity Features	87.98% for SVM
Technique: Deep learning				
[85]	DNN, MLP, PCA,Bayes Net	496 Entries	Performance and Participation	92.54% for DNN
[70]	DNN, AIRS, AdaBoost	10,140 Entries	Performance	95.34% for DNN
[87]	DNN	OULAD Dataset 32,593 Entries	demographics, clickstream behaviour and assessment performance	max 93% for DNN
Technique: Hybrid ML and DL				
[84]	LSTM, BP, SVR,SVM, GBC	SPD and SAPD	Demographic, Academic and Behavioural	SVR and BP as best algorithms
[71]	DT, k-NN, SVM, Ensemble	500 Entries	Demographic, Academic and Behaviour	91.5% for ensemble approach
[86]	L-SVM, R-SVM, DT, GP, RF, NN, AdaBoost, NB	242 Entries	Procrastination Behaviour	95% for L-SVM
[69]	k-NN, SVM, ANN, DT, BLR, RLR	OULAD Dataset 32,593 Entries	Demographic, Engagement and Performance	96.04% for SVM (RBF kernel)

(c) Data pre-processing often plays an important role in improving the performance of ML models, however, from literature, specifically for the students' performance prediction. This aspect has not been addressed in detail, rather most of the times models are trained on the raw datasets [93, 94]. It is an active research gap to deploy multiple state-of-the-art pre-processing approaches to see the impact on the prediction performance

(d) Feature selection is one of the key operations for ML model training since, most of the time, irrelevant features result in degraded performance. This aspect of the research has not been addressed in detail for students' performance prediction. It will be a significant contribution to the knowledge to study different feature selection approaches towards improving the performance of ML models

(e) The problem of students' performance prediction has been mostly addressed in literature as a supervised learning (i.e., classification, regression) problem. However, it is equally of interest to study this problem as unsupervised learning (i.e., clustering) problem

(f) The role of simulated and synthetic datasets on the performance of ML models has not been addressed in detail in the literature. The use of generative models can lead to quality simulated datasets which may result in much-improved prediction results

(g) The class imbalance problem while dealing with the problem as classification problem has not been paid attention in the literature.

Overall, detailed survey of the existing literature was conducted. The review analyzed the impact of various factors, including student characteristics (such as prior knowledge, teaching modes, and motivation), course design (such as the use of multimedia resources and interactive activities), and technological factors (such as the availability and reliability of the VLE), on the prediction performance of ML models. For example, it was found that many modules, such as the classification module, which establishes a connection between students' scores and factors influencing them, can be used to forecast how well students perform in VLE [95]. Models were introduced that incorporated data features like past performance, engagement with LMS and course-related activities [75]; [76]; [77]; [78]; [79]. The dataset consisting of students' backgrounds and personal information attributes was used in another research to assess the students' performances in VLE [81]. Similarly, the dataset consisted of performance-based features including previous academic performance and interview scores were also used as significant indicators of students' performance over VLE [82]. Another study made use of a dataset with demographic, performance and learning behavior features as the most influential and significant features related to the prediction of students' performance over VLE [64]. So, it can be concluded from the research that the most influential and significant features related to the prediction of students' performance over VLE included past performance, engagement with LMS, course-related activities, students' backgrounds, personal information attributes, academic performance, interview scores, demographic, performance, learning behavior features, students' assignment submission behavior, click stream behavior, assessment performance, students' access patterns, availability of course design, and different teaching styles. Among these significant features, demographic, performance, students' backgrounds, personal information attributes, and students' access patterns are observed to be the most significant factors in assessing students' performance in VLE. The review also examined the role of social and emotional factors, such as student engagement and self-regulation, in predicting students' performance in a VLE. Through this process, the review established an understanding of the significant factors that influence students' performance in a VLE and provided a detailed account of the latest research in this area. The literature review also identified a number of research gaps in the field of predicting students' performance in a VLE. One significant gap that was identified is the class imbalance problem, which occurs when there are very few instances of one class (the minority class) compared to the other class (the majority class). This can lead to ML models being biased towards the majority class, resulting in poor performance when predicting the minority class. To address this problem, researchers have proposed a number of techniques, to balance the class distribution and improve the performance of ML models. Other research gaps that have been identified include the need for more personalized approaches to prediction and the need for more research on the interactions between different factors.

2.7 Class Imbalance Problem in Existing VLEs

The class imbalance problem is a significant issue in various domains, including education-related datasets [82]. It arises when there is a significant disparity in the number of instances between different classes, leading to biased training of machine learning models. Most ML models are designed to handle balanced datasets, where classes are evenly represented, and as a result, imbalanced datasets can cause the models to be biased towards the majority class. While the class imbalance problem has gained attention in other contexts, such as spam filtering and fault detection, it has not been extensively addressed in the literature concerning students' performance prediction.

In the context of VLEs, class imbalance can pose challenges for developing trustworthy classifiers. Many existing methods for addressing the class imbalance problem, such as sampling strategies, expense procedures, kernel-based techniques, and active learning methodologies, require a significant number of training instances. However, when the availability of representative training data is limited, constructing a reliable classifier becomes more challenging due to the imbalanced distribution among classes. To overcome this issue, an algorithm can be developed that corrects the label space skewness and compensates for the lack of examples in the training set by incorporating data from a supplementary domain. This approach utilizes knowledge transfer to handle the class imbalance problem and create a trustworthy classifier.

Addressing the class imbalance problem is crucial in VLEs, where data arrives in streaming instances over time. Traditional ML algorithms may overlook or overfit the minority class, leading to learning difficulties. Moreover, the assumption that there are only two classes represented in the data is often untrue in real-world scenarios. Multi-class tasks in online learning settings can present even more challenges due to the dynamic nature of the data and the increased dimensionality. Therefore, developing effective methods to handle class imbalance and adapt to concept drift (a change in the underlying data distribution) is essential in VLEs and other applications impacted by these learning difficulties.

Progressive learning methods store and process data in batches, whereas online learning methods learn provided data one by one while requiring no prior knowledge. Many online tactics have been proposed to address class inequality. One illustration is a Naive Bayes clustering technique based on random under sampling. There are a few articles that have been provided for OOB and UOB [96] that deal with imbalanced data and an ever-changing imbalance rate. However, they cannot effectively and adaptively balance multi-class data. The original OOB and UOB covered broad online cases, however their sampling rates were not set consistently when the class distribution changed. Cost-sensitive approaches, such as cost-sensitive Bagging and Boosting, RLSACP, WOS-ELM and ESOS-ELM assign a different misclassification cost for each class. Only two-class situations have cost-setting processes and those costs are pre-defined. VWOS-ELM was recently introduced in order to address the problems with class imbalance in multi-class data streams. This strategy uses an ensemble of different WOS-ELM base classifiers. WOS-ELM is a kernel extreme learning machine. Before sequential learning can start, initialisation requires a dataset. To combat class imbalance, distinct group weights are maintained based on the models' performance on a validation dataset. However, if the validation data set doesn't precisely reflect the status of the data, the class weights won't be accurate for learning. Initialization information could also not always be available.

In this section, two re-sampling-based ensemble methods are suggested: multi-class re-sampling and bagging in VLE. They use over or under-sampling in the context of online bagging to solve class disparity, as suggested by their titles (OB). Re-sampling is algorithm-independent, therefore the ensemble can be built using any basis classifiers. For instance, multi-class data can be processed directly using neural pathways. Additionally, re-sampling is one of the easiest and most effective imbalance techniques used in online classes. In both fixed and dynamic situations, a time-decayed class size is employed to address the issue of class imbalance through the VLE. It is a real-time indicator that shows the current level of class inequality. It is used in each of the aforementioned systems to automatically select the sample rate [97].

Online class imbalance learning's multi-class problem hasn't gotten much attention. Multi-class education may have an impact on problems with learning and workable solutions, but this is still unknown. More effective and adaptive technique that is suitable for both fixed and dynamic circumstances must be developed to more accurately, analyse the impact of several classes on online class imbalance learning. The primary focus will be on the following research questions:

- What possible method could be develop to apply on dataset to handle a class directly while adapting to class imbalance?
- What impact does a class imbalance with a stationary imbalance condition have on VLE?
- What impact would an unbalanced class with a dynamic imbalance state have on VLE?

2.8 Predictive Learning Analytics (PLA) and Learning Design

PLA is an emerging field that integrates educational data and machine learning techniques to forecast and analyze student outcomes and behavior in educational settings. This research area aims to leverage data from various sources, such as learning management systems, online platforms, and academic records, to develop predictive models that can provide actionable insights for educators, administrators, and policymakers. By applying advanced statistical and machine learning methodologies, PLA enables the identification of patterns, trends, and factors that influence student outcomes. This information can help in identifying at-risk students, personalizing learning experiences, and optimizing educational interventions. The PLA process involves data collection, preprocessing, feature engineering, model selection and training, evaluation, and interpretation of results. Ethical considerations, including privacy, data security, bias, and model transparency, are crucial when employing PLA. This research area holds great promise for improving educational practices, enhancing student engagement and success, and supporting evidence-based decision-making in education. In context to the research presented in this thesis, the trained machine learning model has to be deployed in practice and integrated with existing learning system, therefore, it is important to review the literature where the machine learning models are used in real-world settings to address the education related challenges. Furthermore, the link between the learning design and predictive analysis is also important to explore to demonstrate how different courses may impact the performance of machine learning models, or otherwise. Followings are a few of the benchmark studies where a ML model is trained and deployed to help the educators in better understanding the ongoing performance of class. Further, this brief review also includes the studies related to learning design and its connection with PLA.

Herodotou et al. [98] explored the use of PLA in higher education and its impact on teachers' perceptions, practices, and students' performance. Drawing upon the Technology Acceptance Model (TAM) and Academic Resistance Model (ARM), authors investigated factors influencing teachers' engagement with PLA and their acceptance of technology in educational contexts. The study, conducted at a distance learning higher education institution, involved 59 teachers facilitating nine courses with a total of 1325 students. Using a multi-methods approach, the study measured the impact of teachers' engagement with PLA on students' performance. Additionally, semi-structured interviews were conducted to understand teachers' utilization of PLA data and the underlying reasons explaining their teaching practices. The manuscript emphasized the need for further research in understanding how teachers perceive, use, and interpret PLA data and highlights the significance of longitudinal learner and learning data in identifying at-risk students. The OU Analyse system, an in-house PLA system developed by the OU UK, is introduced as the platform for data collection and analysis.

Herodotou et al. [99] aimed to investigate the effectiveness of PLA in informing the design of motivational interventions and their impact on student retention in higher education. The Student Probabilities Model (SPM) was utilized to predict students' likelihood of completing their courses. A randomized control trial was conducted involving 630 undergraduate students identified as at risk of not completing their studies. These students were randomly assigned to either the control group ($n = 312$) or the intervention group ($n = 318$). The intervention group received motivational interventions delivered by the university's Student Support Teams (SSTs) through text messages, phone calls, and emails. The results demonstrated statistically significant improvements in student retention outcomes for the intervention group, indicating the effectiveness of the proposed intervention in facilitating course completion. The study showcased how PLAs can inform SSTs in identifying appropriate motivational interventions to enhance student engagement and improve course completion rates. Overall, this study highlighted the potential of PLAs and the benefits of implementing targeted interventions based on predictive models, providing evidence of improved student support at scale and low cost.

Calvert [100] performed a case study which focused on distance learning students enrolled in open access courses and highlighted the application of predictive analytics to create a model that predicts the probabilities of success and retention at different milestones in the student journey. The study identified a set of explanatory variables and determines their varying importance at different milestones. While the specific variables and milestones may differ across institutions, the approach can be generalized to distance learning and higher education institutions in general. The study emphasized the need for institutions, especially those offering distance education, to utilize recorded student information to identify students who may be at risk of leaving. By tailoring student support based on the identified factors, institutions can improve retention rates in open access distance education. Logistic regression is employed to generate probabilities of success at module-related milestones, module completion, and student return in subsequent academic years. These probabilities serve as indicators of student outcomes and can inform interventions to support student success. The identified variables serve as proxies for factors such as motivation, opportunity, realism, and ability, and are readily available within existing data sources.

Cechinel et al. [101] introduced MAD2, a Learning Analytics Dashboard developed for Moodle, which is a widely used Learning Management System. MAD2 offered various visualizations to track and comprehend students' interactions within the Moodle environment. It also incorporated machine learning techniques to predict students at risk of failure at an early stage. The paper emphasized the importance of providing such a tool within Moodle to assist professors and managers in better supporting students throughout their courses. By offering a panoramic overview of student interactions and utilizing predictive analytics, MAD2 enhanced the understanding of student behavior and enables proactive interventions. Xin and Singh [102] focused on the development of a learning analytics dashboard to enhance learning outcomes for educators and students in the context of digitalization in education. With the proliferation of e-learning applications and learning management systems, educators face challenges in monitoring student progress. By analyzing data generated from user usage patterns, analytics provide insights into student performance, enabling educators to apply early interventions and modify their teaching methods to better meet students' needs. The study presented the development approach of the analytics dashboard, including the design and development of the back-end system to ensure accurate and relevant data display, as well as the creation of a user-friendly interface for easy data interpretation. The analytics dashboard aims to provide educators with meaningful and relevant data, empowering them to gain a better understanding of their students' performance.

Islam and Mahmud [103] described the development of an intelligence Learner Management System (iLMS) that integrated learning analytics into the traditional learner management system. The iLMS utilized machine learning techniques for descriptive, predictive, and prescriptive analytics of learner data, aiming to enhance the learning experience and improve teaching support. The system was developed as part of a Knowledge Transfer Partnership project between the University of East London and Mediprospects, an independent training provider in the UK. The iLMS offered advanced learning analytics capabilities, going beyond traditional descriptive analytics, by reasoning and predicting from learner data. The paper presents the key features of the iLMS, including user interfaces, reports, and learning analytics. ML classifiers such as LogR, k-NN, and DT were employed for the learning analytics tasks. The proposed system provided insights based on various indicators such as gender, age, highest education, assessment results, and online activity logs. Susnjak et al. [104] examined the existing approaches to learning analytics dashboards and highlighted the challenges faced by education providers in implementing them effectively. The analysis revealed that most dashboards primarily rely on descriptive analytics, with only a few incorporating predictive analytics. To address these limitations, the study proposed a state-of-the-art dashboard that combines descriptive analytics with ML to enable both predictive and prescriptive analytics. The researchers demonstrated how emerging analytics tools can enhance learners' understanding of predictive models and comply with regulatory requirements. They also emphasized the deployment of data-driven prescriptive analytics to provide actionable advice to learners and encourage behavioral changes. The proposed dashboard, currently being trialed at a higher education institution, is unique in its comprehensive integration of analytics components.

In context to learning design and PLA, Rienties et al. [105] reviewed 10 years of learning design research at the OU UK and shown that the OU's learning design approach, particularly the OULDI taxonomy, has been widely adopted by academics and instructional designers. There has been a recent increase in large empirical studies testing the effectiveness of learning design activities in relation to students' behavior and learning outcomes. The research has revealed that learning design decisions made by OU teachers directly and indirectly impact students' online and offline engagement, satisfaction, and learning outcomes. Visualizing initial learning design decisions to teachers has been found to influence their final mix of learning design activities, leading to a shift towards more student-centered learning activities. Furthermore, advanced statistical models and analyses have demonstrated that the balance of learning design activities on a week-by-week basis significantly affects students' engagement with the VLE, with 40-69% of the variance in VLE engagement predicted by learning design and module characteristics. However, there are notable misalignments in learning design practices within disciplines or qualifications, which may pose challenges for students' progression and require adjustments in their learning strategies. The findings emphasized the importance of providing support, training, and opportunities for sharing good practices to optimize learning design and maximize students' potential.Rienties and Toetenel [106] linked 151 modules and 111,256 students at the OU UK to examine the impact of learning design on students' behavior, satisfaction, and performance in blended and online learning environments. Through multiple regression models, the findings demonstrated a strong relationship between learning design and students' VLE behavior and performance. The primary predictor for academic retention was the time learners spent on communication activities, highlighting the importance of well-designed communication tasks aligned with course objectives. This study is innovative in empirically testing the impact of learning design on behavior and outcomes, while controlling for institutional and disciplinary factors. The researchers aim to expand the sample size and incorporate additional data, such as student and teacher comments, to gain a deeper understanding of the complex relationships between learning design and learning processes. Integrating demographic, individual, and socio-cultural data can also help analyze subgroups and predict the impact of specific learning designs on satisfaction and outcomes. The practical implications emphasize the need for collaboration between researchers, teachers, and policymakers to explore how context, learner characteristics, and institutional learning design activities influence students' learning journeys over time.

Toetenel and Rienties [107] emphasized the need for educators to adapt their practices in response to changing educational contexts. By employing learning analytics methods and visualizing learning design decisions, the study explored the tacit knowledge of educators regarding course material, activity types, and workload. Analyzing the learning designs of over 60,000 students across 157 courses, common pedagogical patterns were identified. The majority of educators widely used assimilative activities (such as reading, watching videos, and listening to audio) and assessment activities. Surprisingly, educators did not choose different activity types based on their function, but combinations of assimilative, productive, and assessment activities or assimilative, finding and handling information, and communication tasks can be observed. However, there was no positive correlation found between the seven learning design activity types and student outcomes. Initial findings suggested a negative correlation between a high proportion of assimilative activities and student outcomes. Further research is needed to explore the relationship between specific learning design decisions and student outcomes in different settings. The study was the first to compare learning design decisions across a large number of modules, contributing to the understanding of pedagogical implications. The authors advocated for more institutions to make their learning design decisions explicit and share data, enabling large-scale studies to validate and generalize the findings.

By explicitly selecting variables and analyzing their impact, course success can be predicted and student outcomes improved.

Rienties et al. [108] highlighted the importance of considering learning design in predictive modeling within the Learning Analytics Knowledge (LAK) community. Despite progress in predictive modeling, the role of learning design in relation to LMS usage and learning performance has often been overlooked. The study compared the design of 87 modules and examines its impact on LMS behavior and learning performance through cluster and correlation analyses. Four distinct learning design patterns are identified: constructivist, assessment-driven, balanced-variety, and social constructivist modules. The findings indicated that learning design activities strongly influence student engagement online and have an impact on learning performance, particularly when modules rely on assimilative activities. However, the study acknowledged the limitation of a relatively small sample size, which hindered more advanced statistical analyses. Future research aims to expand the sample size and integrate demographic, individual, and socio-cultural data to enhance the understanding of the complex relationships between learning design, learning processes, and outcomes. The practical implications emphasizd the need for researchers, teachers, and policymakers to consider the influence of learning design choices on students' learning journeys and performance, emphasizing the importance of combining research data and institutional data for comprehensive analysis. Rienties et al. [109] focused on the design and implementation of blended and online transitional courses offered by higher educational institutes. Data was collected through an online questionnaire, and 118 course descriptions were analyzed using multiple correspondence analysis and two-step clustering analysis. The results revealed five dimensions that explain the courses: ICT, Mathematics versus language, Lower versus higher Bloom levels, Gamma sciences versus others, and Very small group size versus others. The courses were then categorized into six distinct clusters. One significant finding is that teachers tend to design and implement similar course designs when given the same content, context, and pedagogical approach. However, the study also highlighted that teachers' choices regarding ICT use are not consistently linked to content and pedagogical decisions.

In summary, several benchmark studies have demonstrated the effectiveness of machine learning models in education, showcasing their potential to enhance student engagement, improve retention rates, and support evidence-based decision-making. The integration of machine learning and learning analytics holds great promise for advancing educational practices and empowering educators and learners with meaningful insights for better outcomes.

Chapter 3

Materials and Methods

This chapter presents the detailed information about the datasets and methods used for the research performed under the scope of the presented thesis. A detailed information about different education related data sets used to predict the performance of students' is provided. Furthermore, background to different ML and deep learning approaches used to predict the performance of students' in VLE is presented. Finally, experimental protocols and methodology adopted to carryout the corresponding simulations are detailed.

3.1 Datasets

For the presented research, two benchmark datasets (i.e., OULAD, Coursera) were used for the prediction of students' performance in VLE. Both the datasets are commonly used in literature in context to predicting performance of students [refs]. The use of two datasets serves multiple purposes in the analysis of student performance prediction. Firstly, each dataset represents different char-acteristics and contexts, making it important to analyze them individually to un-cover dataset-specific insights. By examining each dataset separately, we can iden-tify unique patterns and factors that contribute to student performance within their respective contexts. This approach allows for a more nuanced understanding of the educational settings and helps tailor recommendations accordingly. Additionally, by comparing the performance of ML models trained on both datasets, we gain valuable insights into the features that are crucial for training effective models. This comparative analysis enables us to identify common predictive factors that transcend dataset boundaries and contribute to accurate student performance prediction, highlighting the generalizable patterns across diverse educational contexts. Details about each dataset are presented in the following subsections.

3.1.1 Open University Learning Analytics Dataset (OULAD)

OULAD dataset proposed by [110, 111] is one of the most used and benchmark data set to developed to facilitate the development of data driven solutions for the students' learning predictions within the VLE. This data combines the demographic information of the students' along with the activity (i.e., click stream) data to make it a unique combination towards identifying the behaviour of students. Generally, the OULAD dataset consists of demographic (e.g., region, age, gender), performance (e.g., results, achievements) and behavioural features (e.g., click stream, activity logs) collected from the online courses and students' interaction with VLE platform. The reported dataset was collected from the selected modules taught during the period of 2013 and 2014. The collected dataset was then processed to remove or modify the personal information (e.g., name, unique identification, date of birth) to maintain the privacy and following the ethical guidelines. The dataset consisted of total of 32,593 students' entries from 15 different countries and is available as open source on the Kaggle website.

Data Collection Process

The data collection process at the Open University involves multiple information systems that support student and module management. To consolidate the information from these disparate systems, the Open University implemented a data warehouse using SAS technology. This data warehouse serves as a central repository, aggregating data from various sources and providing researchers with a unified dataset for analysis.

The dataset comprises three main types of data: demographic data, performance data, and learning behavior data.

- The demographic data includes information about students' age, gender, region, highest education level, and disability status. This data provides insights into the student population and allows researchers to explore how these factors may influence academic performance and outcomes.
- The performance data captures students' results and achievements throughout their studies at the Open University. It includes information such as module identification codes, presentation identification codes, student identification numbers, the number of previous attempts for a module, the total number of credits studied, and the final result in each module presentation. This performance data allows researchers to investigate patterns in student performance, identify factors that contribute to success or failure, and evaluate the effectiveness of interventions or support strategies.
- The learning behavior data focuses on students' interactions with the VLE. It includes information about the modules studied, presentation codes, student identification numbers, VLE material identification numbers, dates of interaction, and the number of interactions. This data provides insights into how students engage with online learning resources, their usage patterns, and their level of participation in online discussions and activities. Researchers can analyze this data to understand the relationship between VLE usage and learning outcomes, identify effective learning strategies, and explore the impact of different types of interactions on student performance.

Data Selection and Anonymization

The dataset selection process involved choosing representative modules taught at the Open University during 2013 and 2014. Several criteria were considered, including the number of students enrolled in the module presentation (with a minimum threshold of 500 students), the availability of VLE data for the module presentation, and the presence of a significant number of failing students. This selection process aimed to ensure that the dataset represents a diverse range of modules and captures a variety of student experiences.

To address privacy concerns, the dataset underwent a comprehensive anonymization process. Personally identifiable information such as social security numbers, dates of birth, and unique identifiers were removed to protect student privacy. Module names were replaced with semantic-free symbols, and temporal information was expressed relative to the start of the module presentation. Numeric identifiers, including student IDs and module codes, were randomized and reassigned to ensure anonymity.

Anonymization was further reinforced through the use of the ARX anonymization tool, which applies additional anonymization methods to preserve privacy. Quasi-identifying attributes, including gender, Index of Multiple Deprivation (IMD) band, highest education level, age, region, and disability, were identified as potential identifiers that could be used to re-identify individuals when combined with publicly available information. An anonymization hierarchy was constructed for each quasi-identifying attribute, and several anonymization rules were applied to the dataset using the ARX tool. The primary measure of anonymity used was k-anonymity, with a threshold of 5, ensuring that each combination of quasi-identifiers is indistinguishable from at least four other individuals in the dataset.

Dataset Structure and Variables

The final OULAD dataset consists of several tables listed as follows:

- The "studentInfo" table contains student demographic information, such as module and presentation codes, student IDs, gender, region, highest education level, IMD band, age band, number of previous attempts, studied credits, disability status, and final module presentation results. The "courses" table lists all available modules and their presentations, providing information about module codes, presentation codes, and module presentation lengths.
- The "studentRegistration" table contains information about student module registration and unregistration. It includes module codes, presentation codes, student IDs, dates of registration, and dates of unregistration (if applicable). This table allows researchers to analyze student enrollment patterns and study the factors that influence student persistence and attrition.
- The "assessments" table provides details about the assessments within module presentations. It includes module codes, presentation codes, assessment

IDs, assessment types (such as Tutor Marked Assessment, Computer Marked Assessment, and Final Exam), assessment dates, and assessment weights. This table enables researchers to examine the assessment structure within modules, evaluate the distribution of assessment types, and investigate the relationship between assessment characteristics and student performance.

- The "studentAssessment" table contains the results of students' assessments. It includes assessment IDs, student IDs, dates of assessment submission, information on whether the assessment result has been transferred from a previous presentation, and the scores achieved by students. This table allows researchers to explore patterns in students' assessment submissions, analyze assessment scores, and investigate the relationship between assessment performance and other variables.
- The "studentVle" table contains information about students' interactions with the VLE. It includes module codes, presentation codes, student IDs, VLE material identification numbers, interaction dates, and the number of interactions. This table provides researchers with a comprehensive view of student engagement with online learning materials and allows for the analysis of VLE usage patterns, the identification of effective learning resources, and the exploration of the relationship between VLE interactions and student performance.
- The "vle" table provides details about the materials available in the VLE. It includes VLE material identification numbers, module codes, presentation codes, activity types, and the planned usage duration of the materials. This table enables researchers to understand the types of materials offered in the VLE, track their usage across different modules and presentations, and analyze the relationship between specific activities and student engagement.

3.1.2 Coursera Dataset

The Coursera dataset is online courses taught at Stanford University's computer science department. The Coursera itself is an online platform developed by Andrew Ng and Daphne Koller at Stanford in collaboration with over 160 universities around the world. The scale of the Coursera platform can be determined by its offered courses (i.e., 5100) and registered students' (i.e., 77 million).

The Coursera data, used in this research went throughout the University of London ethics committee to access the Coursera data. The accessed data was from 2018 to 2019 of eight courses run by the Computer department. The data contains courses and students' information.

Coursera dataset consists of nine key groups of information, including course information, course contents, course progress, assessments, course grades, discussions, feedback, learner and demographics. The followings are the details of each data key element.

- **Course information** includes the basic information of course, including the name of the course, session in which it is taught, etc. Dataset contains the information about eight courses including Algorithm and Data Structure, Computational Mathematics, Discrete Mathematics, Fundamental of Computer Science, How Computer Works, Introduction to Programming I, Introduction to Programming II and Web Development.
- **Course progress** describes learners' interaction with course contents.
- **Assessments** provide in-depth details of interactions with assessments.
- **Course grades** details the learners' grades and passing states within a course. The distribution of students' passing or failing each course is presented in Figure 1.1. In addition, the distribution of final grades for six courses is illustrated in Figure 1.2.
- **Course contents** refer to the materials of the course, including modules, lessons, items and mapping to specializations.
- **Discussions** contain forums, forum posts and vote information.
- **Feedback** contains information regarding user ratings of course content and courses.
- **Learner** describes learners' info, like when/where the user joined Coursera.
- **Demographics** contain demographic data based on user surveys.

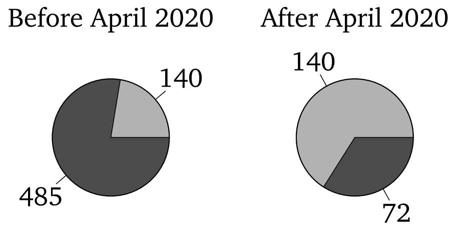
Data Pre-Processing

For the experiments, reported in this thesis, the Coursera dataset was prepared into five main indicators: (a) course design (b) teaching style (c) teacher access pattern (d) student access pattern and (e) students' life circumstances. Each of the indicators contains number of features briefly summarised as follows:

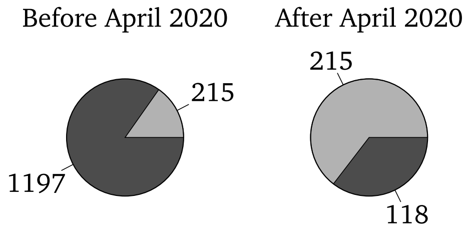
- **Course design** includes the number of learning materials (quizzes, exams, videos, etc.). Course design could be illustrated as per student, per course or combined (see Figure 3.4 and Figure 3.3).
- **Teaching style** includes presentations, methods (e.g., in-line-video, off-line-video), number of the instructor,s feedback to students, instructor's and students' first language and type of communication method with students.
- **Teacher's access pattern** is referred to as the data accessed by the teachers such as number of times teacher viewed the grade book, the number of times the teacher participated in the course forum and the number of announcements sent to the students' on the Coursera.
- **Student's access pattern** includes student activities performed on online platform including the number of hits per resource, duration spent per resource of learning materials, participation in the related discussion forums and feedback ratings to courses. Figure 3.5 shows the distribution of number of hits by students' per course.
- **Student's life circumstances** data includes the information about student employment status (part/full time), marital status, number of children, first language of the student and type of study funding.

Figure 3.1 showed that the number of total students' who passed or failed their course work before, in April and after taking the final exam which was after April, for all eight courses.

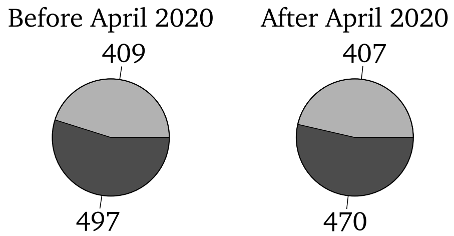
The data collection and visualisation stage showed a mix of categorical and numerical types of datasets. Some data with missing values in the Coursera dataset observed, especially with students' demographics information. Looking at the number of learning materials in each course, around 2000 items of learning materials are there in Introduction to programming 1 course, while the courses (Algorithms and data structure, How computers work and Introduction to programming 2) contains around 650 learning materials items. While the highest withdrawl rate was noted for the Computational Mathematics course, as well as the highest student mouse hits with 1585656 hits. In contrast the lowest withdrawl rate for the Discrete Mathematics course with the consistency pass number of registered students. The lowest number of student hits, counted for Introduction to programming 2 course with 71089 mouse hits. The core feature of the dataset such as course design, teaching style, teacher access pattern, student access pattern and student life circumstances. These features will be used to predict students' performance using various ML algorithms.



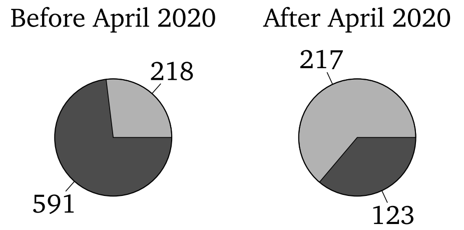
(a) Algorithms and Data Structures 1



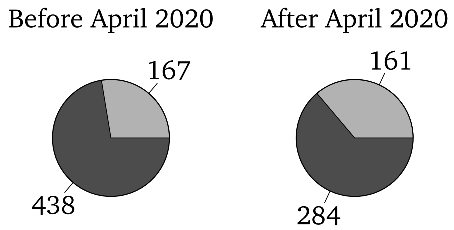
(b) Computational Mathematics



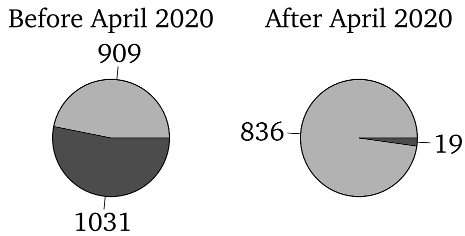
(c) Discrete Mathematics



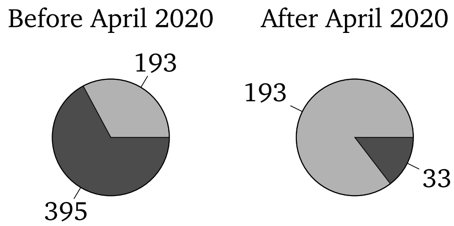
(d) Fundamentals of Computer Science



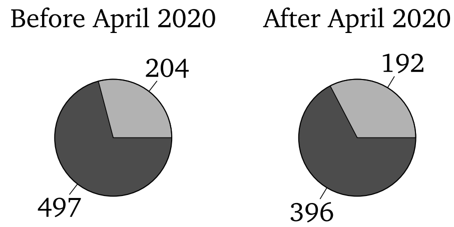
(e) How Computers Work



(f) Introduction to Programming 1



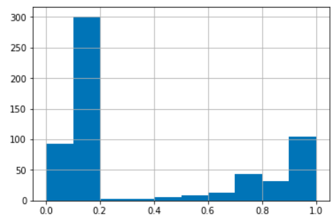
(g) Introduction to Programming 2



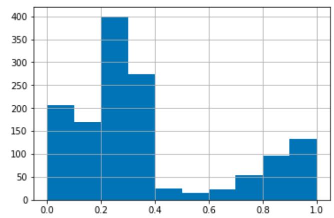
(h) Web Development

● Fail ● Pass

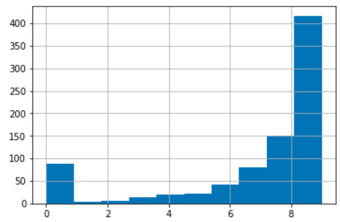
Figure 3.1: Student pass/fail results of all courses



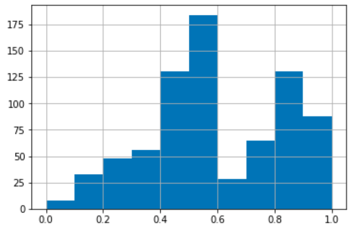
(a) Algorithm and Data Structure



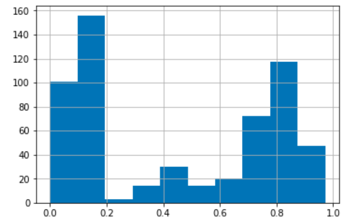
(b) Computational Mathematics



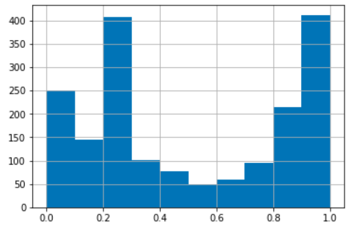
(c) Discrete Mathematics



(d) Fundamentals of Computer Science



(e) How Computers Work



(f) Introduction to Programming 1

Figure 3.2: Final course grade distribution per course

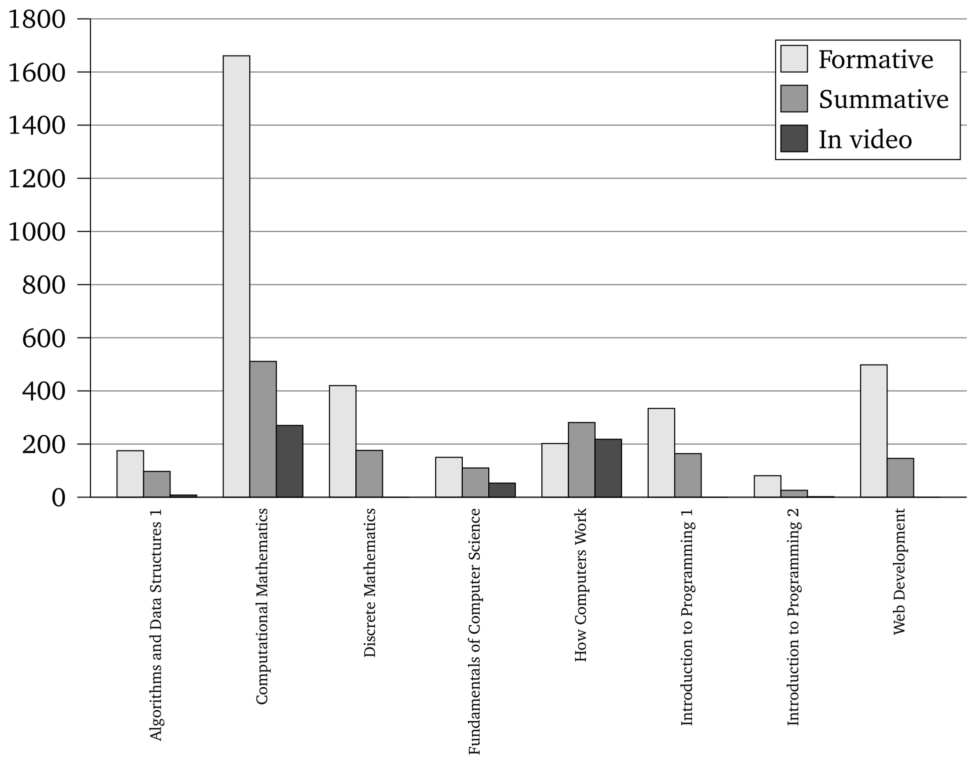


Figure 3.3: Number of learning materials, type per course

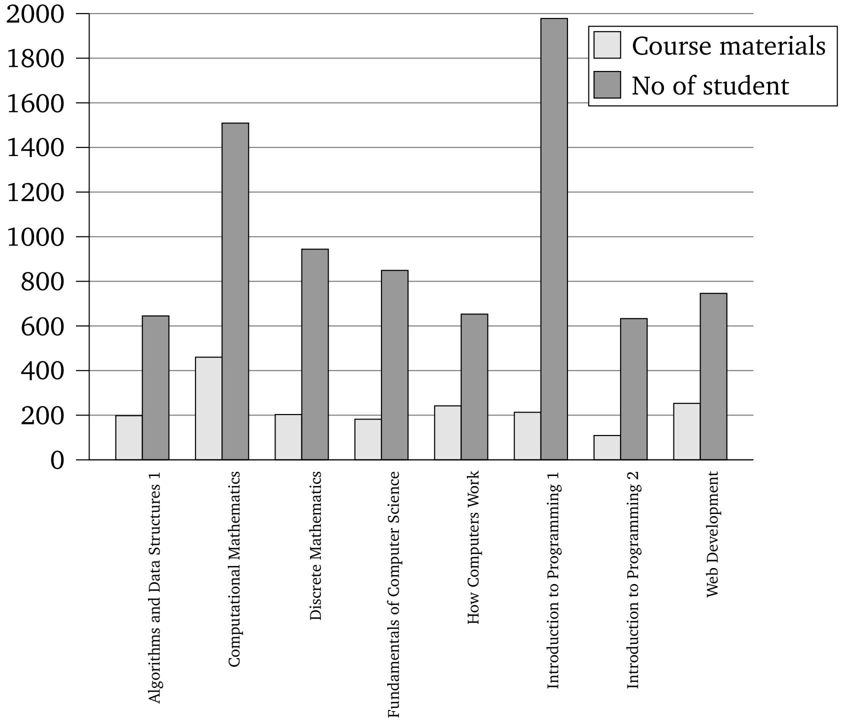


Figure 3.4: Number of learning materials and student enrolled per course

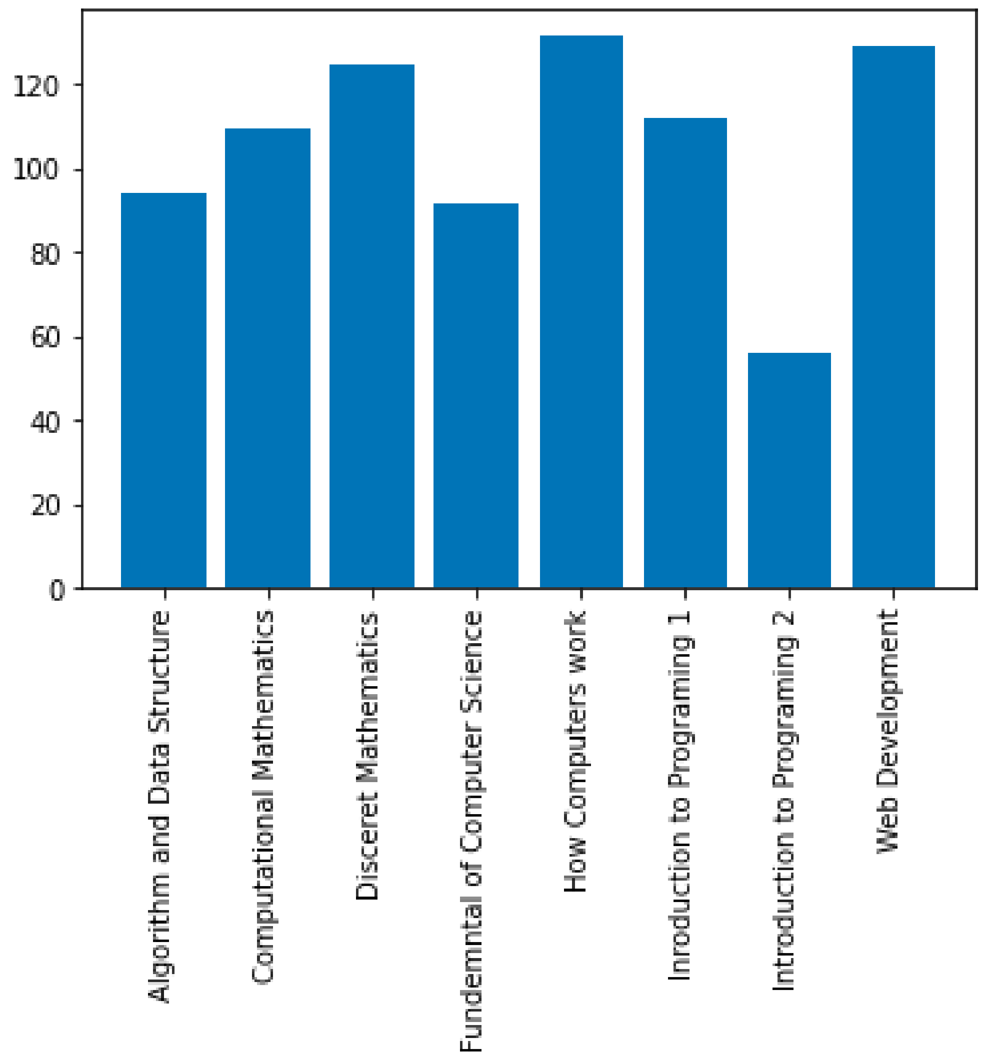


Figure 3.5: Total number of hits done by students' per course

3.2 Difference between OULAD and Coursera

The OULAD dataset and the Coursera dataset are both widely used for developing data-driven solutions in the field of education, particularly for predicting students' learning outcomes and understanding their behavior within VLEs. However, they differ in terms of their composition and scope, which can impact the implementation of ML models.

The OULAD dataset combines demographic information, performance data, and behavioral features such as clickstream and activity logs. The dataset is available as open-source on Kaggle and includes several tables that provide information on student demographics, course registration, assessments, assessment results, and student interactions with the VLE. This rich dataset allows for the exploration of various features and their relationships with student performance and behavior. When implementing ML models on the OULAD dataset, one can leverage its comprehensive nature to consider a wide range of factors influencing students' learning outcomes. In contrast, the Coursera dataset focuses on online courses offered by Stanford University's computer science department and accessed through the Coursera platform. It includes key groups of information such as course information, course progress, assessments, course grades, discussions, feedback, learner information, and demographics. While the Coursera dataset offers a diverse range of course materials and learner information, it may have a more limited scope compared to the OULAD dataset in terms of the number of courses and students included.

In summary, the OULAD dataset provides a more extensive and varied dataset with detailed demographic, performance, and behavioral features, enabling a comprehensive analysis of student behavior and learning outcomes. On the other hand, the Coursera dataset offers specific information on course progress, assessments, grades, discussions, and learner demographics from a narrower set of courses.

3.3 Machine Learning (ML) Approaches

ML is a field of AI that studies the algorithms used to teach machines towards automate certain processes. Now a days, ML is used for problem solving in all fields In general, a model is trained over number of training samples and is used for unseen data based on its training. There are four main types of ML approaches including supervised, unsupervised, semi-supervised, and reinforcement learning [112, 113].

- **Supervised Learning** involves the development of a model based on the labelled training data. The trained model is then used to make decision for unseen samples based on learning.
- **Unsupervised Learning** involves the raw training data without labels and model learns the patterns and useful information independently. Later, this trained model is used to make decisions on the unseen test samples based on its learning.

- **Semi-Supervised** approach as reflected by its name, involves both unlabelled and labelled training data in proportion to model development.
- **Reinforcement Learning** involves dealing with software agents within an environment taking actions towards maximising the output reward.

ML model development involves a number of steps including dataset preparation, dataset pre-processing model, hyper-parameters section, training and validation. As the first step in the process, the raw dataset is explored and visualized to get it prepared for the ML analysis. Once the dataset is arranged and a number of features are selected, then dataset cleaning is performed to remove the missing entries and sometimes the irrelevant features. The next step in the pipeline involves the selected ML model hyper-parameters settings for the training process. The dataset is split into train and test portions for training and validating the ML model, respectively. Once the training is done, the ML model is validated on the unseen samples to evaluate the performance using some standard evaluation measures [113].

3.3.1 Classical Approaches

K Nearest Neighbours (k-NN)

k-NN is a non-parametric ML approach first introduced in 1951 by [114, 115]. Algorithm for classification works on the principle of classifying an input into one of the target classes based on popularity among its neighbours (i.e., classes of nearest neighbours). In ML domain, k-NN classification is the most commonly used approach for the case when there is no knowledge about the data distribution. Original algorithm has been extended over the years, in terms of definition of formal properties [116], introduction of new rejection approaches [117], Bayes error rate refinements [118], distance weighted technique [119], soft computing approaches [120] and fuzzy approaches [121].

Algorithm basically works on computing the Euclidean distance between test and training samples. The $y_i = (y_{i1}, y_{i2}, \dots y_{im})$ be the input sample with m features where $(i = 1, 2, \dots n)$. The Euclidean distance formula [122] between y_i (training sample) and y_t (test sample) can be determined using the expression in Equation 3.1.

$$d(y_i, y_t) = \sqrt{(y_{i1} - y_{t1})^2 + (y_{i2} - y_{t2})^2 + \cdots + (y_{im} - y_{tm})^2} \tag{3.1}$$

A major shortcoming of the majority, based voting occurs for the unbalanced class dataset which is a common scenario in the real-world. Since in this case, each new test example will be biased to be classified to the class with a greater number of samples. One approach to address this problem is to assign the weight to neighbours (i.e., weighted k-NN). Most important part of this algorithm is to select the appropriate value of k which is highly dependent on the dataset. In general, the greater value of k reduces the noise effect in the dataset, however, it makes classes boundary less distinct [123]. Since, the k-NN is primarily based on the Euclidean distance, normalizing the training data can significantly improve the classification performance.

Figure 3.6 shows a typical example to understand the working of k-NN classification algorithm. There are two target classes, red triangle and blue square while solid circle represents $k = 3$ and dotted circle represents $k = 5$. Green dot is the test sample that has to be classified among two target classes. For the first case when k is 3, test sample will be classified as red triangles based on the voting. While, for the second case when k is 5, test sample will be classified as the blue square.

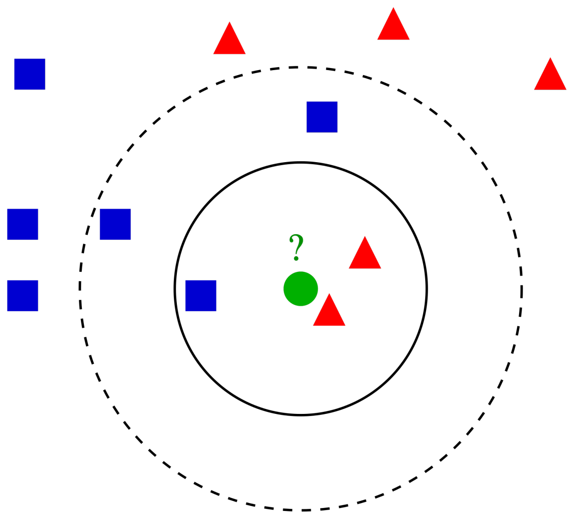


Figure 3.6: Example of k-NN classification.

Some advantages of k-NN include simple implementation, instance-based learning and the addition of new training data without affecting the accuracy. However, there are some limitations of k-NN including sensitivity to noise, poor performance for larger datasets, the requirement of feature scaling, the poor performance of imbalanced dataset and the inability to work with high dimensional data.

Support Vector Classifiers (SVC)

SVC are one of the most robust supervised ML algorithms, introduced originally by Vapnik in 1963 and extended by [124]. Algorithm is based on the construction of multiple hyperplanes in the high-dimensional space with the aim to achieve good separation between them. A high margin between hyperplanes indicates the reduced generalization loss. Given the test samples to be classified into one of two classes, each test sample is represented as an m-dimensional vector and separated by an $(m-1)$ dimensional hyperplane. The test sample may be separated by multiple hyperplanes; however, the best one is selected based on the maximum separation in linear classification case [125, 126, 127, 1]. Figure 3.7 shows a typical example of data points classified by hyperplanes.

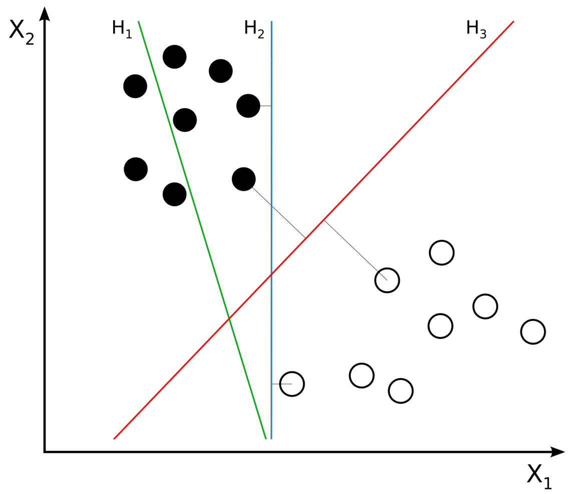


Figure 3.7: Example of SVC Classification by Hyperplanes.

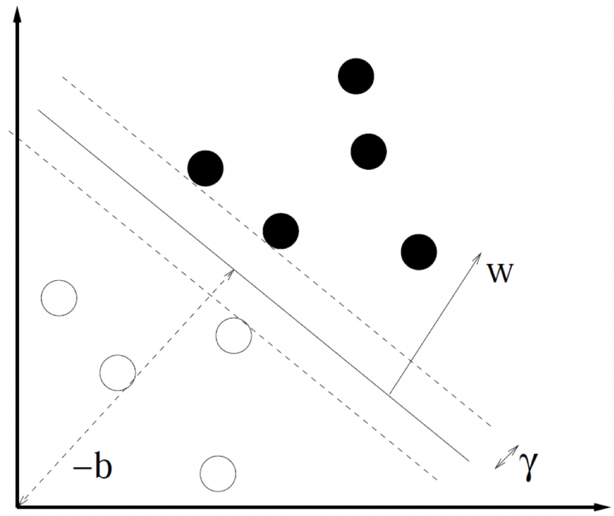


Figure 3.8: Representation of a Linear Binary SVC [1].

For a binary classification case, consider (x_i, y_i) represent the labelled input samples, where x_i is feature vector [128] and y_i is their respective class labels i.e., -1 or 1. SVM aims to construct a model to assign each input feature vector to one of the target classes. Equation 3.2 mathematically expresses the functionality of the binary classifier [1].

$$f(x, w, b) =< w, x > +b \tag{3.2}$$

where w denotes the decision hyperplane and b is the intercept. Figure 3.8 shows a typical feature space for a binary classifier where $(w - b)$ denotes the decision hyperplane while γ denotes the margin.

Some highlighted advantages of SVC include ability to dealing with high dimensional data, memory efficiency and kernel trick functionality. However, there are a few limitations of SVC which includes poor performance for larger datasets and the inability to handle noise.

Decision Tree (DT)

DT is a non-parametric supervised learning approach in ML and uses a tree structure to predict the target value from the observations. Usually, branches in the tree represent the observations and connections between features, while the leaves of the trees represent the target values.Starting from the root node, a feature is evaluated and one of the two nodes is selected; each node in the tree is basically a decision rule. This procedure is repeated until a final leaf is reached, which normally represents the target [129, 130].

The root node of the tree (source set) is split into children nodes based on the defined set of rules. This splitting process is repeated recursively for each child node to split further, referred to as recursive portioning. Recursion is completed when the children nodes have the same value for the target variable. This learning approaches for DT is referred to as Top Down Induction of Decision Trees Top Down Induction of Decision Trees (TDIDT) and is one of the common approach used for DT learning [131]. Iterative Dichotomiser 3 Iterative Dichotomiser 3 (ID3) [132], Chi-square Automatic Interaction Detection Chi-square Automatic Interaction Detection (CHAID) [133], CART [134] and Multivariate Adaptive Regression Splines Multivariate Adaptive Regression Splines (MARS) [135] are few commonly used DT algorithms.

Listed are the main components of DT:

- **Root Node** is the first node in decision trees and also referred as the source set.
- **Splitting** is a process of dividing a node into two or more sub-nodes, starting from the root node. Usually, this process is repeated until all the nodes have the same target value.
- **Leaf or terminal node** is the end of a decision tree and represents either a target variable (for regression) or a target class (for classification).
- **Branch** is the connection between two nodes/features of the decision tree.
- **Parent and Child Node:** A node, which is divided into sub-nodes is called parent node of sub-nodes whereas sub-nodes are the children of parent node.
- **Pruning** is a technique to reduce the the decision tree's by removing sub-nodes. The aim is to reduce the complexity for improved predictive accuracy and to avoid over-fitting.

Some advantages of DT include their simple structure, ability to handle both numerical and categorical data, robustness against co-linearity, ability to approximate any

Boolean function, embedded feature selection, the resemblance with human decision making and easy data preparation. However, there are certain limitations of DT as well including non-robustness, biasness issues, selection of DT depth and the problem of learning an optimal decision tree.

Classification And Regression Tree (CART)

Leo Breiman proposed the traditional non-parametric DT approach for classification or regression predictive modelling issues and it is now known as Classification And Regression Tree CART. The dataset is divided into a DT by CART, using the Gini Impurity. Gini impurity [136] is an indicator of how frequently a randomly selected element from the set would be mislabelled if it were randomly classified in accordance with the distribution of labels in the subset. Gini impurity can be calculated by adding the probabilities of an item having the right label and the probabilities of an error in accurately categorising that item [134]. Mathematically, the following expression can be used to calculate the Gini impurity [137] for a collection of L classes, where p_i is the probability of items being labelled with class i and $i \in (1, 2, 3, \ldots, L)$.

$$\text{Impurity} = \sum_{i=1}^{L} \left(p_i \sum_{k \neq i} p_k \right) = 1 - \sum_{i=1}^{L} p_i^2$$

The continuous output variable's variance is reduced when CART is used for regression [138]. Mathematically, the variance reduction at node N can be written as:

$$\begin{aligned}\text{Variance Reduction} = &\frac{1}{|\Gamma|^2} \sum_{k \in \Gamma} \sum_{l \in \Gamma} \frac{1}{2}(y_k - y_l)^2 \\ &- \left(\frac{1}{|\Gamma_t|^2} \sum_{k \in \Gamma_t} \sum_{l \in \Gamma_t} \frac{1}{2}(y_k - y_l)^2 + \frac{1}{|\Gamma_f|^2} \sum_{k \in \Gamma_f} \sum_{l \in \Gamma_f} \frac{1}{2}(y_k - y_l)^2 \right)\end{aligned}$$

where Γ stands for a group of pre-split indices, Γ_t for the split test true sample indices, and Γ_f for the split test false sample indices.

Linear Regression (LR)

A supervised learning method is called linear regression, Linear Regression (LR) predicts the association between two or more explanatory variables (i.e., dependent and independent variables). Simple linear regression refers to the scenario when just one explanatory variable is used, whereas multi-variable linear regression refers to the scenario where numerous explanatory variables are used [139, 140].

In an instance of basic linear regression, the goal is to fit the data to a straight line with the equation $y = \beta_0 + \beta_1 x$. where y is the response variable and x denotes the predictor variable. The straight line's y-intercept is represented by β_0, while its slope is represented by β_1. Being able to define the strength of a link between variables, β_1 is a crucial quantity (i.e., if slope is near to zero, it indicates no or very low relation). In linear regression, the best line is fitted to the data after determining the goodness of fit. Usually, the goodness of fit is assessed using probabilistic models. Mathematically, the paired points [141] are represented as $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, where each y_i is assumed to be generated using a function of xi and the true line [142] $y = \beta_0 + \beta_1 x$. Furthermore, the new equation can be written as follows if the extra noise is denoted by the variable ϵ_i:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

The next step is to solve for fit by treating it like an optimisation problem [143, 144], which entails identifying the line for which the probability of data is highest. In mathematics, it can be written as

$$\min_{\beta_0,\beta_1} : \sum_{i=1}^{n} [y_i - (\beta_0 + \beta_1 x_i)]^2$$

This is known as the least-squares linear regression problem [145] and the following equations can be used to get the solution:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} x_i y_i - \frac{1}{n} \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sum_{i=1}^{n} x_i^2 - \frac{1}{n} \left(\sum_{i=1}^{n} x_i\right)^2} = r \frac{s_y}{s_x}$$
$$\hat{\beta}_0 = \overline{y} - \hat{\beta}_1 \overline{x}$$

where s_x, s_y signify the standard deviation [146] of x and y samples, r denotes the correlation coefficient, and $\overline{x}, \overline{y}$ denote the sample means of x and y. The expression that can be used to calculate r is as follows:

$$r = \frac{1}{n-1} \sum_{i=1}^{n} \left(\frac{x_i - \overline{x}}{s_x} \right) \left(\frac{y_i - \overline{y}}{s_y} \right)$$

Many assumptions about the variables and their relationships are made by standard linear regression methods. Although these assumptions are either diminished or eliminated totally in the extended forms of linear regression. They are still present in the majority of linear regression approaches and should therefore be discussed. Here is a list of some highlighted presumptions:

- Treating the predictor variables as fixed rather than random is referred to as **weak exogeneity**. Although this presumption is incorrect in real-world situations, keeping it in place creates more complicated issues.
- The term "**linearity**" refers to the assumption that the goal variable and predictor variables have a linear connection, which is not necessarily the case.
- The idea that there is a correlation between the errors of the response variable is known as **independence of errors**.

High performance of linearly separable datasets, simplicity of implementation and capacity to prevent over-fitting through the use of regularisation techniques are only a few advantages of linear regression. The assumption of a linear relationship between the variables, sensitivity to noise, propensity for over-fitting and propensity for multicollinearity are some of the constraints of linear regression.

3.3.2 Ensemble Methods

A ML technique called the ensemble techniques combines multiple base models to create a single, ideal predictive model. The "intelligence of numbers" hypothesis, which contends that decisions made by a larger group of people are typically smarter than those made by a single expert, is supported by ensemble learning. In line with the foregoing, ensemble learning describes a collection (or ensemble) of fundamental learners, or models, who collaborate to produce a more precise prediction at the end. Due to excessive variation or significant bias, a single model, often referred to as a basic or weak learner, may not perform effectively. However, when weak learners are combined, they could develop into strong learners due to the improved model performance brought about by the decrease in bias or variance that result from their union. Take a step back and consider the ultimate purpose of ML and model development to better comprehend this term. As we explore specific examples and the rationale behind the usage of Ensemble methods. This will make more sense as explore specific examples and the rationale behind using Ensemble methods. Large describes the meaning and usefulness of ensemble methods using decision trees (however it is important to note that Ensemble Methods do not only pertain to DTs). In this case, classification models can be helpful.

Ensemble learning approaches fall into two main categories: boosting and bagging. The main difference between different learning tactics is how they are taught.

Bagging

Bootstrap Aggregating, or Bagging, gets its name from the fact that it combines Bootstrapping and Aggregation to produce a single ensemble model. From a sample of data, several bootstrapped subsamples are taken. A few highlighted advantages of bagging include reducing the overfitting, improving the model's accuracy, ability with high dimensional data, ease in implementation and reduced variance. On the other hand, some limitations of bagging include loss of interpretability, increased computational cost and reduced adaptability. Random forests is one of the common bagging type used in this research.

Random Forest (RF):

The models that use random forests may incorporate the idea of bagging with a minor adjustment. Bagged result from any feature that is offered to trees allows them to decide where to divide and how to make decisions. Because of this, even though the bootstrapped samples can differ significantly, the data often separate off at the same features for each model. In contrast, RF models choose which characteristics to divide into groups at random. Since each tree will divide based on various attributes. Random Forest models employ a level of difference rather than splitting at comparable characteristics at each node throughout. A larger ensemble is employed to aggregate over this level of variation, leading to a more precise predictor. To understand better, please use the example.

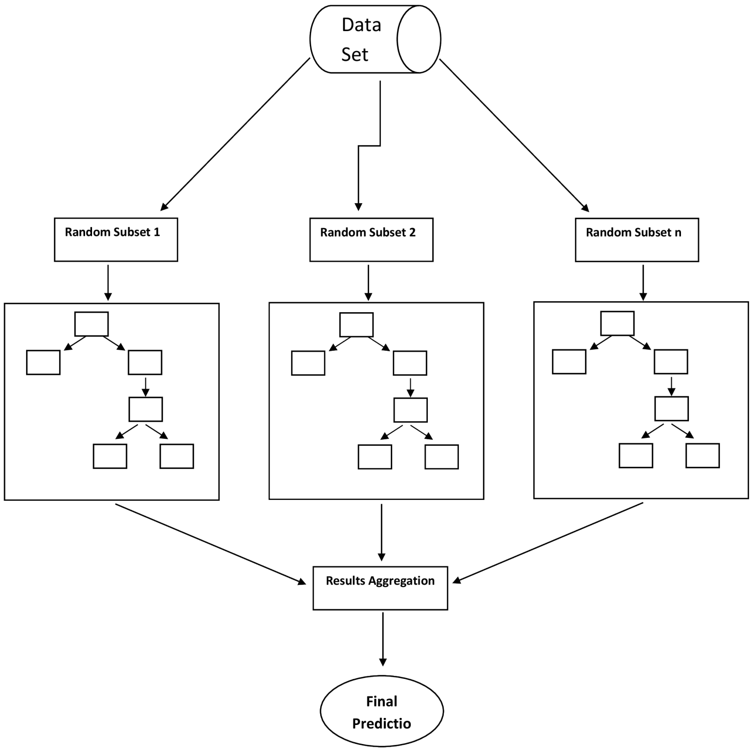


Figure 3.9: A Forest Diagram.

Bootstrapped sub-samples are also drawn from a bigger dataset, much like bagging. For each sub-sample, an analytical DT is constructed. The DT is divided using a number of characteristics (in this diagram the features are represented by shapes).

The random forest-based models may also include the idea of bagging with a minor adjustment. Any feature that is available may be used by bagged decision trees to decide where to divide and how to make decisions. The data often divides off at the same features for each model as a result, despite the fact that the bootstrapped samples may differ slightly.

Followings are the steps involved in the implementation of random forest algorithms:

- **Step 1:** Assume the training data set has M features and N observations. From the training data set, a random sample with replacement is first taken.

- **Step 2:** The node is split repeatedly using the feature that produces the best split using a subset of M characteristics that is randomly selected.
- **Step 3:** The tree is at its maximum size.
- **Step 4:** A forecast is created using the average of the predictions from n trees after repeating steps 1 through 3.

A few highlighted advantages of random forests include their ability to deal with higher dimensionality data and ensured correctness for missing data handling. On the other hand, the major limitation is their inability to provide exact predictions because of the involvement of means.

Boosting

Boosting increases the expected accuracy of ML models by combining numerous weak learners into a single strong learner. Algorithms for ML can learn well or poorly. Boosting improves the performance and projected accuracy of ML models by combining numerous weak learners into a single strong learner. A few highlighted advantages of the boosting include ease of implementation, reduction of bias, and improved computational efficiency. Strong or weak learners can be used to describe ML algorithms:

- **Weak Learners:** Weak learner refers to a learning algorithm that performs slightly better than random guessing on a given task. For in-stance, to recognise the cat image, it combines a weak learner who guesses for pointed ears with another learner who guesses for cat-shaped eyes. The technology looks for pointed ears on the animal image before looking again for cat-like eyes. The system's overall accuracy is improved as a result.
- **Strong Learners:** Strong learners have higher prediction efficiency. A group of weak learners is transformed into a single group of strong learners by boosting. For example, it combines a weak learner who guesses cat-shaped eyes with a strong learner who guesses pointed ears to identify the cat image. The technology initially looks for pointed ears in the animal's image before looking again for cat-like eyes. This improves the system's overall accuracy.

Adaptive Boosting:

Adaptive boosting was one of the earliest boosting models developed (AdaBoost). The boosting procedure adjusts and tries to self-correct with each repetition. AdaBoost initially gives each dataset the same weight. The weights of the data points are then automatically adjusted following each decision tree. It provides them with greater weight in order to make up for misclassified products in the subsequent round. The procedure is continued until the residual error, or the difference between actual and projected values, is below a desired level. AdaBoost can be used with a wide variety of predictors and is frequently less sensitive than other boosting techniques. This strategy doesn't work well when there is a correlation between qualities or a lot of data dimensionality. AdaBoost is an effective boosting technique for classification problems overall.

Extreme Gradient Boosting (XGBoost):

Similar to AdaBoost is the sequential training technique known as GB. The difference between GB and AdaBoost is that GB does not give items that were incorrectly classified more weight. Instead, GB software optimises the loss function by building base learners in a sequential order, making sure that each base learner is always more efficient than the previous one. This method, in contrast to AdaBoost, strives to deliver accurate results up front rather than correcting errors as they happen. As a result, GB software might produce conclusions that are more accurate. Gradient boosting is useful for both classification- and regression-based problems.

XGBoost improves gradient boosting in several ways for processing performance and scalability. To enable simultaneous learning during training, XGBoost makes use of the CPU's multiple cores. Due to its ability to handle enormous datasets, it is a boosting technique that is interesting for big data applications. The primary features of XGBoost are parallelization, distributed computing, cache optimisation and core processing. The open-source Extreme Gradient Boosting software provides an excellent and efficient implementation of the XGBoost. In ML competitions, XGBoost quickly became the go-to method and frequently the winning formula for classification and regression problems. Before XGBoost, there were previous open-source implementations of the technique, but its debut seemed to unleash its full potential and raise gradient boosting profile more extensively in the field of applied ML. Some highlighted advantages include high flexibility, ability to use parallel processing, speed and support for regularization. Furthermore, sparsity and customization are also referred to as the positive aspects of using XGBoost algorithms.

CATBoosting

Gradient boosting is a method for decision trees called CATBoost. It was developed by Yandex and is employed in many other fields and companies, including CERN, Cloud Flare and Careem Taxi, as well as for search, recommendation systems, personal assistants, self-driving cars, weather forecasting and many other things. The words "Category" and "Boosting" are combined to form the phrase "CATBoost". Gradient boosting is a method for decision trees called CATBoost. It was developed by Yandex and is employed in many other fields and companies, including CERN, Cloud Flare Careem Taxi, as well as for search, recommendation systems, personal assistants, self-driving cars, weather forecasting and many other things. It is highly effective in two ways:

- Without the extensive data training that conventional ML algorithms often requires, it offers cutting-edge results.
- It provides effective out-of-the-box support for the more descriptive data formats that go along with many business issues and commercial obstacles.

An algorithm follows the general steps outlined below to train the boosting model:

- **Step 1:** The boosting technique assigns equal weight to each data sample. It gives the starting machine model, the fundamental algorithm, the data. The fundamental algorithm produces predictions for each data sample.
- **Step 2:** With the boosting strategy, samples with larger errors are given more weight when evaluating model predictions. In addition, a weight based on the model's performance is given. A model that delivers excellent forecasts will have a significant impact on the final decision.
- **Step 3:** The weighted data is sent to the following decision tree in the algorithm.
- **Step 4:** The algorithm iterative repeats steps 2 and 3 until the total number of training errors are below a predetermined threshold.

Few highlighted advantages of the CATBoost algorithm are listed as follows.

- **Performance:** CATBoost gives cutting-edge results and can compete with any top ML algorithm in terms of performance.
- **Handling Categorical Features Automatically:** To convert categories into numbers without performing any explicit pre-processing, one may use CATBoost. For the purpose of converting categorical values into numerical values, CATBoost employs various statistics on categorical feature combinations as well as categorical and numerical feature combinations. Continue reading to learn more about it.
- **Robust:** Models are able to grow more general without having to make significant hyper-parameter adjustments or over fit them. The number of trees, learning rate, regularisation, tree depth, fold size, bagging temperature and other settings are among those that may be altered in CATBoost. This page describes each of these traits in detail.
- **Easy to Use:** You can use CATBoost directly from the command line by using an accessible Python, R and API.

3.3.3 Deep Learning Approaches

Artificial Neural Network (ANN)

ANN are the ML algorithms inspired by the biological functionality of the animal brain and have proven effective. A network consists of nodes, connections and layers. Nodes are the representations of artificial neurons and are capable of processing the input signal and transmitting it to other neurons. Each neuron transforms the input by some non-linear function and transmits the output. These neurons are connected with each other over number of layers where each layer is responsible for certain transformations. Each layer is assigned the weights which represent the strength of a signal at a given neuron and improves over the training iterations. Typically, ANN consists of an input layer, an output layer and a number of hidden layers with artificial neurons at each layer connected with each other [147, 148, 149, 150]. Figure 3.10 shows a typical representation of ANN.

Listed are the main components of ANN:

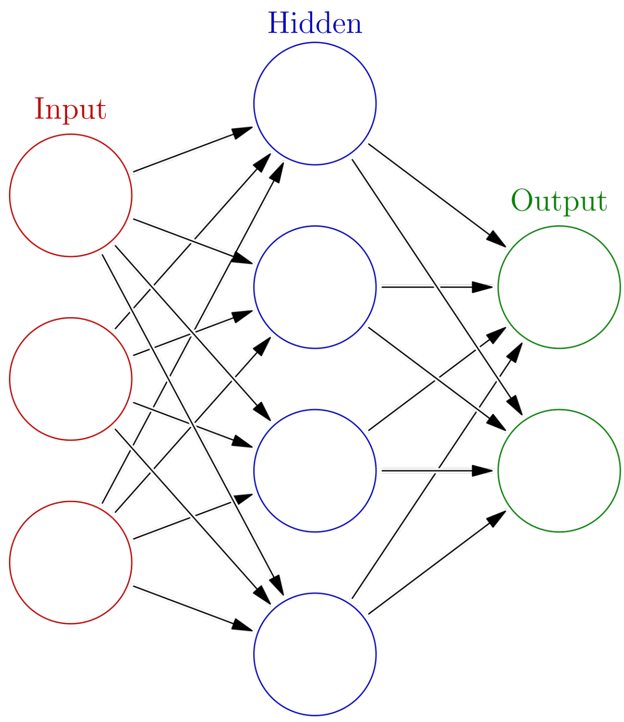


Figure 3.10: A Typical Representation of a Neural Network.

- **Artificial Neurons** are conceptually inspired by the biological neurons and process the given input using a non-linear activation function to generate output.

- **Connections** Neurons in the network are connected with each other by connections which are responsible for transmitting the output of one neuron as an input to other neurons. One neuron is usually connected to multiple neurons.

- **Weights:** Each connection in the network has a weight which determines its significance. These weights are updated during the training process to optimize the performance of the network.

Performance of an ANN is dependent on number of hyper parameters. Listed are some important hyper parameters adjusted, prior to training process:

- **Learning Rate:** is the value that represents the step size that the model takes to correct the error and adjust network weights accordingly. In general, higher learning rate results in quicker training but with degraded performance, however, lower learning rate increases the training time but results in higher accuracy.

- **Hidden Layers:** is the number of hidden layers in the network. It defines the depth of network and is dependent on the size of the dataset and the number of features. Usually, for huge datasets, hidden layers are increased while for small datasets, hidden layers are reduced.
- **Batch Size:** is the value which determines the number of input samples being processed for training to the model. This number depends on the computational resources available for the training process.

ANN has many advantages including its ability to store the information on the entire network, fault tolerance, ability to work with incomplete data, gradual degradation of performance and distributed memory. However, there are certain limitations of ANN as well including their hardware dependence, training duration, selection of network nodes and depth and lack of theoretical support.

Synthetic Minority Oversampling Technique (SMOTE)

SMOTE is a pre-processing method for addressing a datasets class imbalance. In the real world, we frequently find ourselves attempting to train a model on a dataset with extremely few examples of a specific class, which leads to subpar performance (for example, rare disease diagnosis, manufacturing failures and fraudulent transactions). It's often impractical to go out and gather more data because of the nature of the data (occurrences are so uncommon). The majority class should be undersampled as one approach to resolving this problem. To make the number of rows for the majority and minority classes nearly equal, could eliminate rows that correspond to the majority class. But in doing so, we miss out on a significant amount of data that could be utilised to train our model and increase its accuracy (e.g., higher bias). The minority class can also be over-sampled as a further option. In other words, we duplicate minority class observations at random. This method has the drawback of over fitting because the model is trained using the same instances over and over again. SMOTE can help in this situation [151]. The SMOTE algorithm can be summed up in the following way:

- Consider the variation between a sample and its closest neighbour.
- Divide the difference by a number at random between 0 and 1.

- To create a new synthetic example in feature space, add this difference to the sample.
- Continue the next closest neighbour till a user-defined number.

Generative Adversial Networks (GANs)

In June 2014, Ian Goodfellow and his colleagues created a family of ML frameworks known as generative adversarial networks (Generative Adversarial Networks (GAN)s). In a zero-sum game, where one agent's gain equals another agent's loss, two neural networks compete against one another. This method learns to produce fresh data with the same statistics as the training set given a training set. For instance, a GAN trained on images can produce new images with numerous realistic features that, at least on the surface, appear to be created by humans. GANs have proven helpful for semi-supervised learning, fully supervised learning and reinforcement learning, while initially presented as a type of generative model for unsupervised learning [152, 153].

The fundamental principle of a GAN is built on "indirect" training via a discriminator, a different neural network that can assess how "realistic" the input seems and that is also dynamically updated. This indicates that the generator is taught to trick the discriminator rather than to reduce the distance to a particular image. This makes it possible for the model to learn without supervision [153, 154].

Candidates are created by the generative network, who then evaluates them with the discriminate network. Data distributions are used to operate the competition. The discriminate network separates candidates, generated by the generator from the actual data distribution, while the generative network often learns to map from a latent space to an interest data distribution. The goal of the generative network's training is to trick the discriminate network into making more mistakes by providing unique candidates that it believes are not synthesised but instead are a part of the real data distribution [154].

The discriminator's initial training data is taken from a well-known dataset. It is trained by repeatedly exposing it to samples from the training dataset until it reaches a satisfactory level of accuracy. Based on whether the generator can trick the discriminator, it is trained. Typically, randomised input sampled from a predetermined latent space is used to seed the generator (e.g., a multivariate normal distribution).

The discriminator then assesses the candidates created by the generator. Both networks undergo independent BP techniques, resulting in the generator producing better samples and the discriminator improving its ability to identify fake samples. A convolutional neural network serves as the discriminator and a deconvolutional neural network serves as the generator when utilised for image production [152, 154].

3.4 Research Approach

A three-stage research approach (see Figure 3.11) has been adopted in the development of ML oriented solutions for students' performance prediction problem. These research activities were carried out in a hybrid approach that combined iterative and waterfall elements. Initially, s linear progression using three steps of data preparation, model development and performance evaluation was followed, however, each stage involved iterative processes and feedback loops to refine and improve the approach.

Details for processes and activities involved under each stage are provided as follows:

- **Stage I - Data Preparation** At the first stage, the raw data has been processed and prepared to make it ready for the ML training. The data preparation involved number of steps, including the cleaning of data, selection of appropriate features and annotation of the data. First, the raw data from multiple sources was sorted together and cleaned for the missing values. Second, the important features based on the research questions and correlation analysis were selected to prepared subsets of datasets. Finally, the data was annotated where needed to facilitate the training of ML algorithms. This stage also involved the generation of simulated data from multiple techniques to address the class imbalance problem. The feedback from model development and performance evaluation stages also prompted to revisit data preparation steps to improve the quality of the dataset.

- **Stage II - Model Development** In the second stage, once the data has been prepared, ML and deep learning models from the classification and regression classes were selected to classify and predict the final score of students, respectively. The selection of models was based on the literature where similar problems were addressed, on the type of data and on the ease of implementation (an iterative process). The selected models were then tuned for the hyperparamters, using the optimization approaches. Finally, the models were trained using the dataset for the prediction and classification analysis. For the training, Google Colab platform with Python and SciKit packages were used. The feedback from the model evaluation stage was used to fine-tune the performance.

- **Stage III - Performance Evaluation** At the third and final stage, the trained models were subjected to unseen test data towards assessing the performance. Multiple standard evaluation measures including accuracy, precision, recall and F1 score were used. Further, the performance was also visualized using the curves, bar graphs and confusion matrices.

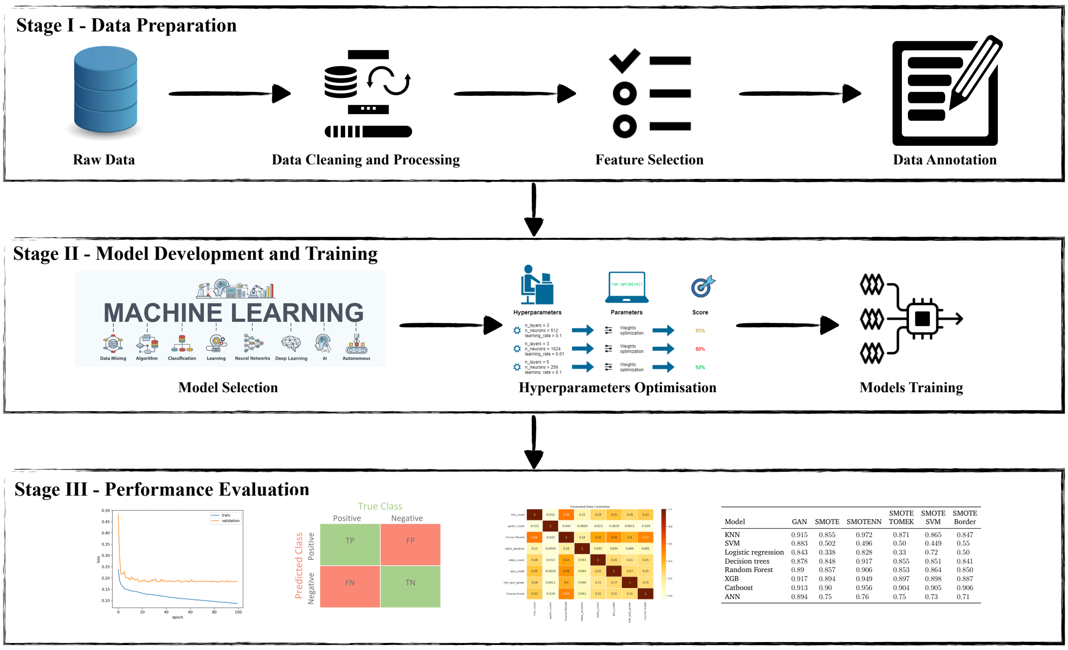


Figure 3.11: Three-Stage Research Approach Adopted for Development of ML Oriented Solutions for the Students' Performance Prediction.

3.5 Study Design

To comprehensively address the problem of students' performance prediction from multiple aspects, we conducted three main experiments. Each experiment aimed to explore different aspects of the prediction problem and evaluate the performance of various machine learning models. The experimental design for each of the experiments is provided in the following sections:

3.5.1 Experiment 1: Prediction of Students' Performance using Classification Models

In first experiment, performance of students is predicted by considering students grades as pass and fail (a classification problem). In this context, performance of multiple state of the art classification models including MLP, DT, RF, XGBoost, CATBoost, k-NN and SVC were evaluated and compared using two benchmark VLE datasets (i.e., OULAD, Coursera). This experiment was performed in four different settings:

- Experiment 1A: Classification of students' performance as "Pass" or "Fail" using multiple ML models over the OULAD dataset. Best ten features from the OULAD dataset based on the correlation were selected. The OULAD dataset collected and pre-processed through different steps (i.e., data cleaning, replace categorical values and data selection). The choice of Experiment 1A was driven by the desire to predict students' performance in the OULAD dataset, which covers a diverse range of subjects and academic domains.
- Experiment 1B: Classification of students' performance as "Pass" or "Fail" using multiple ML models over the Coursera dataset combined for all the subjects. This experiment allowed us to evaluate the performance of the same ML models on a dataset that is more subject-specific, potentially offering insights into the effectiveness of models on different academic disciplines.
- Experiment 1C: Classification of students' performance as "High", "Medium" and "Low" using multiple ML models over the feature engineered Coursera dataset. This experiment provided a finer-grained analysis of performance prediction and was relevant given the variable accuracy observed in the literature depending on the number of grading bands.
- Experiment 1D: Classification of students' performance as "Pass" or "Fail" using ensemble of three ML models (i.e., DT, k-NN, XGBoost) over the combined Coursera dataset. This ensemble method allowed us to leverage the strengths of individual models and potentially improve the overall prediction accuracy.

3.5.2 Experiment 2: Classification of Students' Performance for Each Course of Coursera Dataset

In the second experiment, performance of ML models (i.e., MLP, DT, RF, XGBoost, CATBoost, k-NN and SVC) were evaluated for classification of students' performance as "Pass" or "Fail" using the each course of Coursera Dataset. Given that Coursera is more limited to specific courses, therefore, it was anticipated to study the prediction performance for each course. Studying course-level predictions in the Coursera dataset offers insights, personalization, curriculum improvement, and advancements in EDA. In the end, a comparison was performed to demonstrate for which course ML performed well. Given the diversity of courses offered on the Coursera platform, we evaluated the performance of various ML models on each course individually. This approach allowed us to identify which courses had higher prediction accuracy and potentially discover course-specific factors that influenced students' performance.

3.5.3 Experiment 3: Class Imbalance Problem using Generative Models

In the third experiment, the problem of class imbalance was investigated in detail by making using of SMOTE and GAN approaches. SMOTE approaches were used to balance the class samples for classification problem, while the GAN models were implemented to generate the simulated dataset. These experiments were performed using the Coursera dataset. A series of investigations under this experiment included:

- Experiment 3A: Multiple SMOTE techniques were used to balance the Coursera dataset and ML models were implemented to compare the performance. This allowed us to assess the impact of class balancing on the performance of ML models.
- Experiment 3B: Generative models were used to enhance the existing Coursera dataset with simulated samples and impact of simulated data was assessed by implementing ML models. The goal was to investigate the effectiveness of simulated data in improving the performance of ML models.

3.6 Performance Evaluation Measures

To evaluate the performance of ML algorithms for student' performance prediction, number of measures were used including classification model accuracy. Details of each of the classification measures, used in the presented research are provided as follows.

A variety of metrics were employed to evaluate the effectiveness of ML algorithms for predicting students' achievement, including classification accuracy, F1 score and J-Index [155]. For the evaluation of classification performance of any learning model, the evaluation metrics stated above are frequently reported in the literature. The following is a brief summary of each measure:

- The percentage of total data inferences that were correctly categorised is known as classification accuracy. A high classification accuracy value is a sign of improved model performance.
- The measurement known as the F1 Score employs precision and recall in concert with harmonic methods. The F1 score is calculated using the following expression in Equation 3.3. A higher number (max 1) showed that the ML model was performing better.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.3}$$

- Recall score measures the ability of a ML model to accurately estimate the positives from the actual positive values. It is determined from the confusion matrix using the following expression:

$$\text{Recall Score} = \frac{\text{TP}}{\text{FN} + \text{TP}} \tag{3.4}$$

- Precision is referred to as the ratio of relevant samples among the retrieved. In other words, it measures the proposition of positively predicted samples that are actually correct. Mathematically, it can be determined using the following expression:

$$\text{Precision Score} = \frac{\text{TP}}{\text{FP} + \text{TP}} \tag{3.5}$$

- Specificity also referred to as the true negative rate is the probability of a negative test given that the condition of actually being negative.
- Sensitivity also referred to as the true positive rate is the probability of a positive test given that the condition of actually being positive.

3.7 Summary

As a summary, this chapter presented a detailed information about the datasets used in this research, theoretical knowledge about the methods and the datasets (OULAD and Coursera) used in this research and background to different ML and deep learning approaches were used to predict the performance of the students. In the domain of VLE, both the datasets mentioned above were considered to be the most comprehensive datasets by the research community. Different theoretical knowledge about ML models, research approaches and experimental designs were also used in this research.

Finally, this chapter also listed standard evaluation measures used for the assessment of ML classification models in context to students performance prediction.

Chapter 4

Students' Performance Prediction using Classification Models

This chapter outlines the detailed information about the Experiment 1 performed under this research in the context to the prediction of students' performance using different ML classification models. A detailed analyses on the experimental investigations performed, using the OULAD and Coursera datasets has been provided. Results are presented in both qualitative explanation and quantitative forms to highlight the important insights from the experiments. The chapter has been structured based on the sub-experiments performed under the Experiment 1.

4.1 Experiment 1A: Classification using the OULAD Dataset

This section presents the information about the ML classification of students' performance using OULAD dataset. A range of ML models including MLP, DT, RF, XG-Boost, CATBoost, k-NN and SVC have been implemented on the selected features of the OULAD dataset.

4.1.1 Dataset

For the investigations performed under Experiment 1A, a part of OULAD dataset was used. Features from the raw dataset were selected using the correlation analysis. The dataset consisted of 11 input features. The dataset was cleaned to remove any missing values and was encoded to perform the ML analysis (i.e., accepts numerical values). Figure 4.1 shows the correlation map between the selected features and the target variable. A correlation heatmap is generated by computing the correlation matrix, which represents the pairwise correlation coefficients between variables in a dataset. The correlation coefficients quantify the strength and direction of linear or monotonic relationships. From the correlation map, it can be observed that *imd_band* and *num_of_prev_attempts* were the two most correlated features while the *region* and *code_module* were the least correlated features. Overall, the plot demonstrates that there is no strong correlation between the input features and target variable, suggesting a challenging problem from ML perspective. Another challenge posed by the dataset is the class imbalance (see Figure 4.2 for target class distribution), which is anticipated to be a significant factor in the ML analysis. The 'Fail' class contains approximately half of the samples of the 'Pass' class.

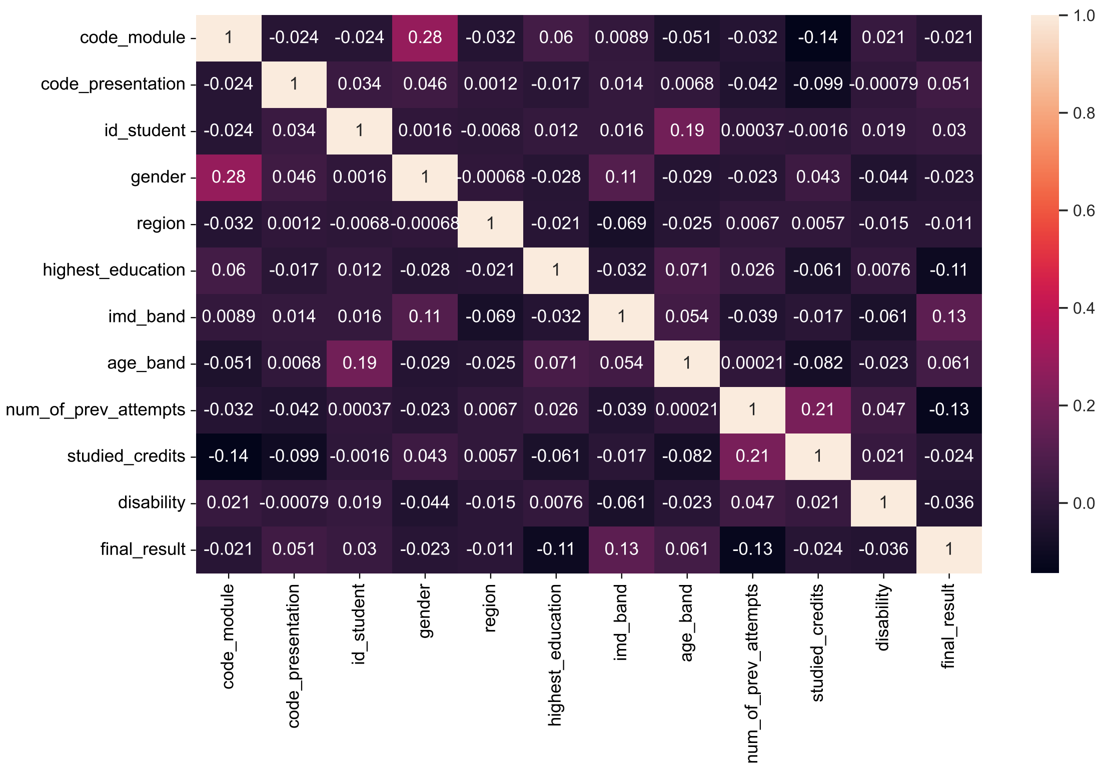


Figure 4.1: Feature Correlation Map for OULAD Dataset.

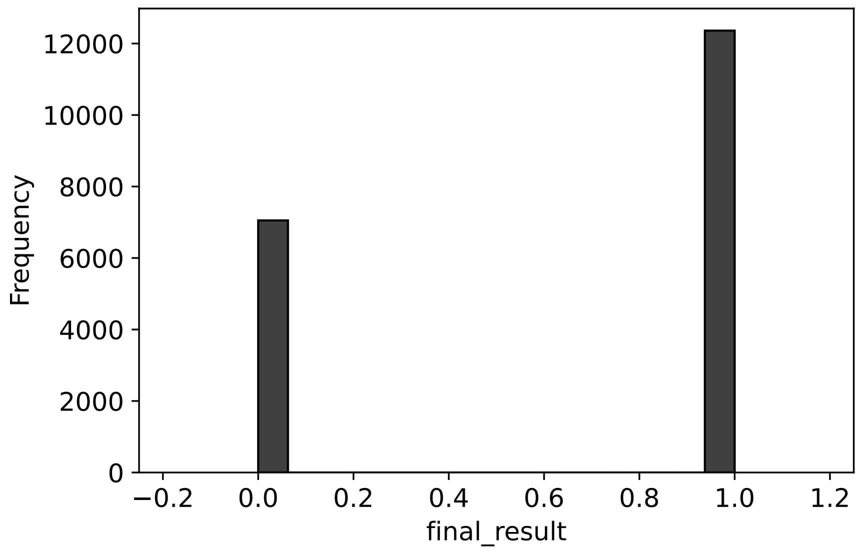


Figure 4.2: Class Distribution for OULAD Dataset.

4.1.2 Experimental Settings

The experiments were performed using the Python programming language, and TensorFlow along with Scikit package were used. The dataset was split into 80:20 ratios for train and test, respectively. For input features, the Standard Scalar transformation approach was used for better ML performance. A 5-Fold cross validation approach was used for the training and Grid Search was adopted to find the optimal hyper parameters. Table 4.1 shows the best hyperparamters resulted by Grid search for the implemented ML models. The performance of models was assessed, using the standard evaluation measures (see Section 3.6 for detailed description) including accuracy, F1 score, precision, recall, sensitivity, and specificity. In addition, confusion metrics and ROC curves were plotted to further establish the understanding of the implemented models.

Table 4.1: Hyperparameters for Machine Learning (ML) Models Implemented under Experiment 1A.

Model	**Hyperparameters**
MLP	activation: logistic, learning_rate_init: 0.001, solver: sgd, iter=500
DT	criterion: entropy, min_samples_leaf: 100, min_samples_split: 2, splitter: random
k-NN	algorithm: auto, leaf_size: 10, n_neighbors: 100, weights: distance
RF	criterion: entropy, min_samples_leaf: 50, min_samples_split: 2, n_estimators: 30
XGBoost	booster: gblinear, learning_rate: 0.1, max_depth: 3, n_estimators: 100
CATBoost	depth: 4, iterations: 150, l2_leaf_reg: 1, learning_rate: 0.1
SVC	degree: 1, gamma: auto

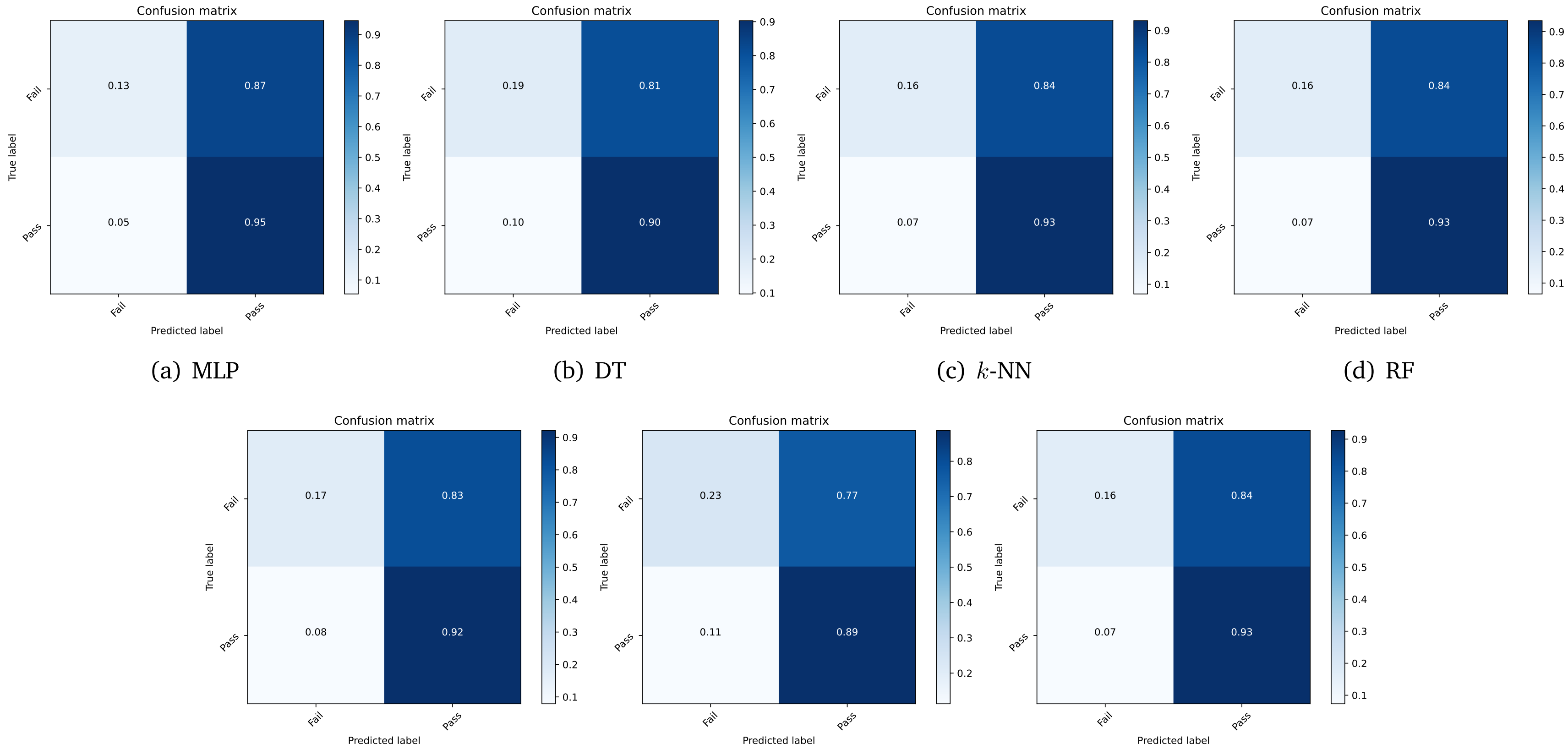
4.1.3 Results

This section presents the results of Experiment 1A in both quantitative and qualitative ways. Table 4.2 presents the detailed quantitative test results for the implemented ML classification models in the context of students' performance prediction, using the OULAD dataset. From the Table 4.2, it can be observed that almost all models, comparatively preformed similar with accuracy around 65%. The performance of all models in the same range indicates the challenging nature of the dataset and the inability of implemented models to learn significant features from the dataset.

Table 4.2: Quantitative Test Results for Machine Learning (ML) Classification of Students' Performance using OULAD Dataset.

Models	Accuracy	F1 Score	Precision	Recall	Sensitivity	Specificity
MLP	0.65	0.50	0.61	0.54	0.13	0.95
DT	0.64	0.53	0.59	0.55	0.19	0.90
k-NN	0.65	0.51	0.61	0.55	0.16	0.94
RF	0.65	0.50	0.61	0.54	0.16	0.94
XGBoost	0.65	0.51	0.60	0.54	0.17	0.92
CATBoost	0.65	0.54	0.62	0.56	0.23	0.88
SVC	0.65	0.51	0.60	0.54	0.16	0.93

To further explore the performance of implemented models in terms of Type I and Type II errors, confusion metrices were also plotted for each model (see Figure 4.3). Aim of the ML model is to minimize the Type II error, which in this case is "Pass student predicted as Fail". From the Figure 4.3, it can be observed that CATBoost achieved the Type I error of 77%, a very high values, however, the least among the implemented. On the other hand, MLP was able to achieve the least Type II error of only 5%. Overall, based on the distribution of Type I and Type II errors, the CATBoost model preformed best among all. Figure 4.4 shows the ROC curves for the implemented models and confirms that all models performed in a similar capacity, with CATBoost slightly on the better end with an AUC of 0.63.



(a) MLP (b) DT (c) k-NN (d) RF

(e) XGBoost (f) CATBoost (g) SVC

Figure 4.3: Confusion Metrices of Implemented Machine Learning (ML) Models for Classification of Students' Performance using OULAD Dataset.

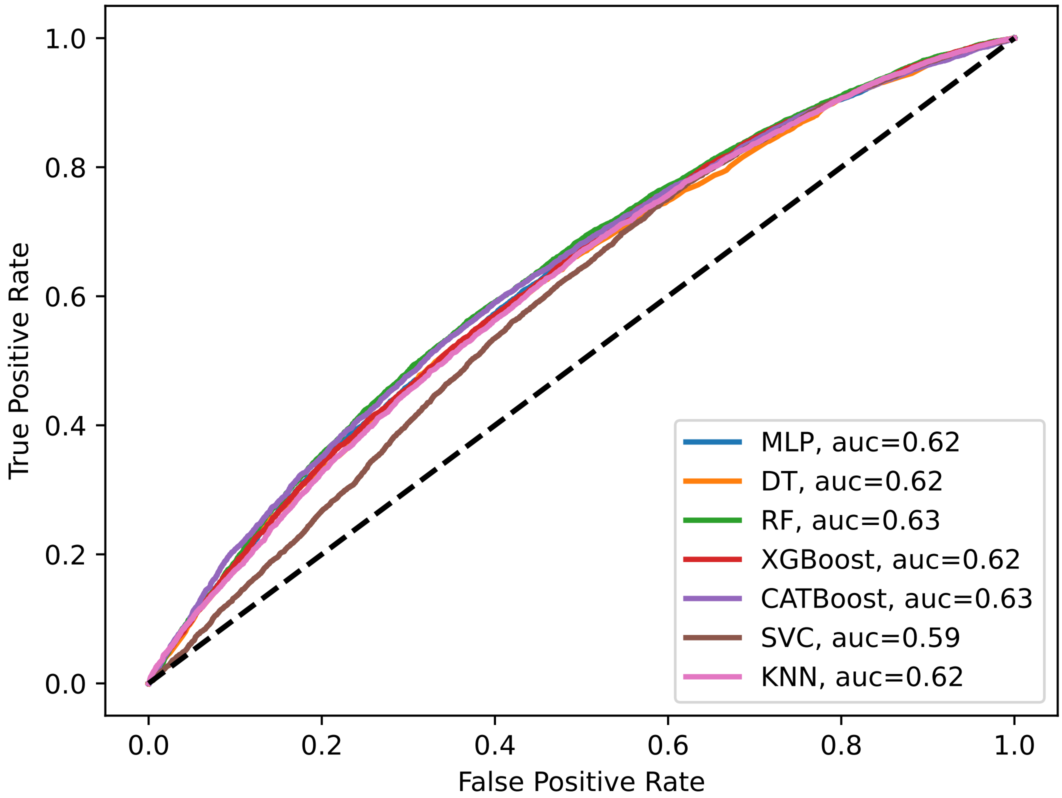


Figure 4.4: ROC Curves of the Implemented Machine Learning (ML) Models for the Classification of Students Performance using OULAD Dataset.

4.2 Experiment 1B: Classification using the Combined Coursera Dataset

This section outlines the implementation of ML classification models to predict the performance of students as Pass or Fail using the Coursera dataset. The implemented ML models for comparative analysis included MLP, DT, RF, XGBoost, CATBoost, k-NN and SVC. Following sections present the information about the dataset, experimental protocols and highlighted results from the investigation.

4.2.1 Dataset

The Coursera dataset combined using all the courses were used for the investigations in Experiment 1B. From the raw dataset, based on the correlation matrix, top ten correlated features were selected. In total, there were 9 input features and one target variable. The nan values were removed from the dataset as part of the pre-processing and Standard Scalar transformation was applied to input features towards achieving optimized ML performance. From the correlation map (see in Figure 4.5), it can be observed that *hits_count* and *quiz _count* are most correlated to the target variable, while *partic_count* and *video_duration* were the least correlated. Overall, most of the features show good positive correlation with the target variable except for the *assessment_type_id_7* where negative correlation was observed. Figure 4.6 shows the class distribution for the target variable and clearly demonstrates the data imbalance problem (i.e., samples from Fail class are almost double in comparison to Pass class).

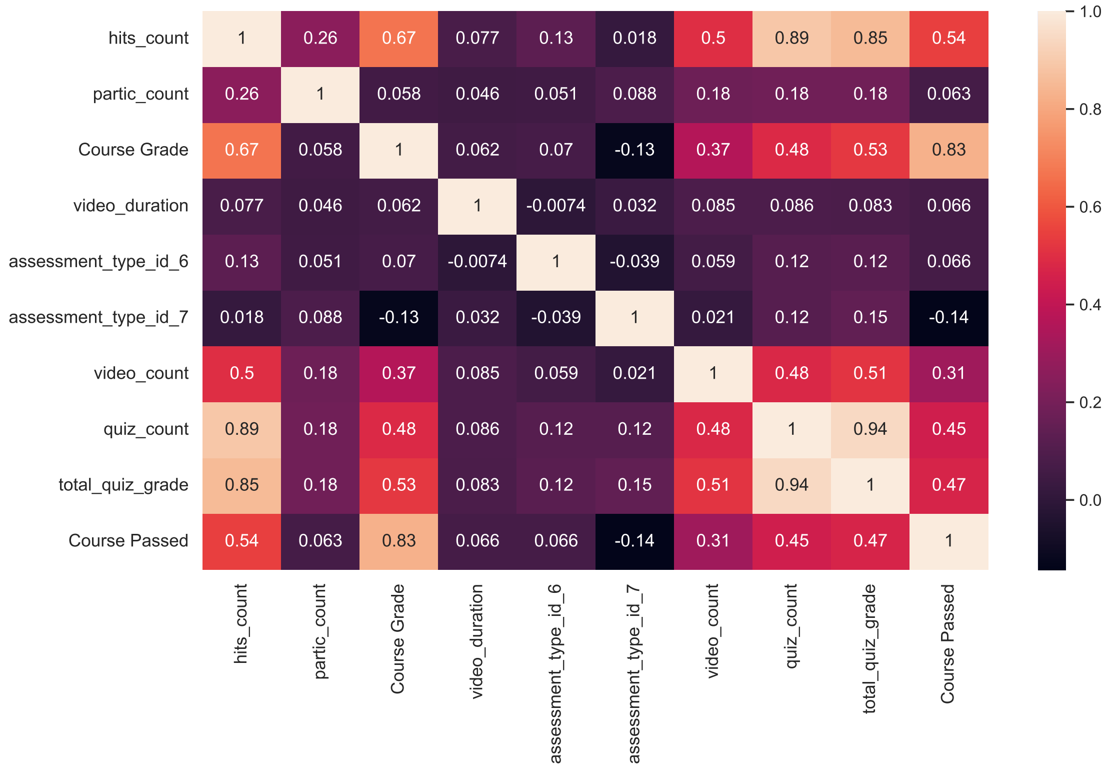


Figure 4.5: Feature Correlation Map for Coursera Dataset.

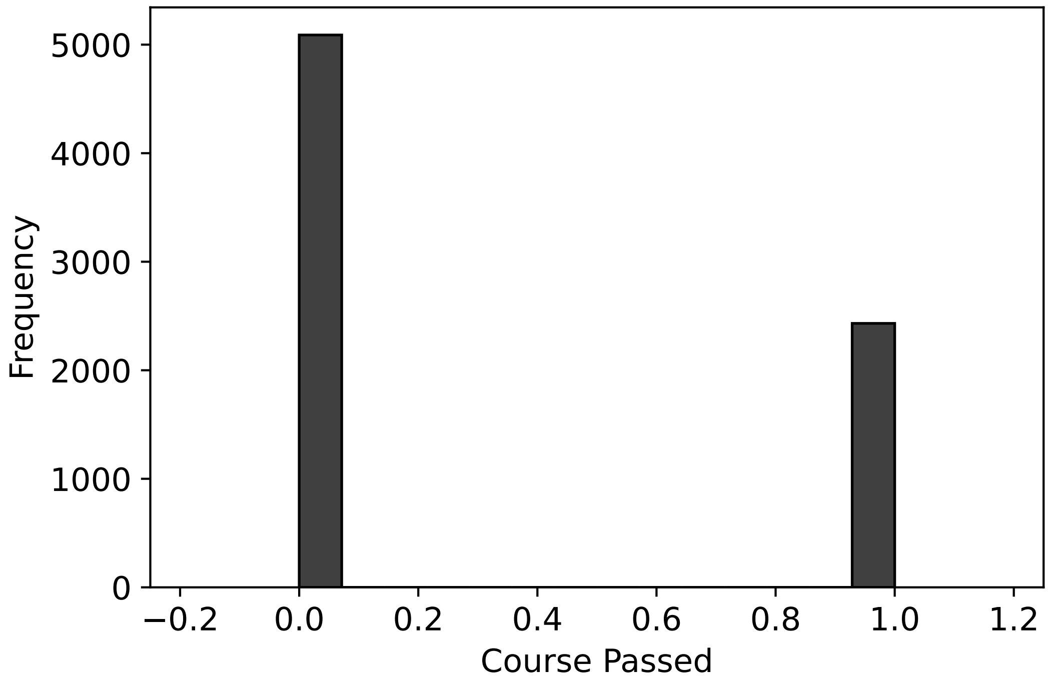


Figure 4.6: Class Distribution for Coursera Dataset.

4.2.2 Experimental Settings

For the Experiment 1B, the same experimental settings were used for Experiment 1A in terms of programming language, python packages, dataset split and data transformation. However, the results for the Grid Search hyperparamters were different for some models, therefore, are provided in Table 4.3. For detailed experimental settings, visit Section 4.1.2.

Table 4.3: Hyperparamters for Machine Learning (ML) Models Implemented under Experiment 1B.

Model	Hyperparamters
MLP	activation: tanh, learning_rate_init: 0.001, solver: sgd, iter=500
DT	criterion: entropy, min_samples_leaf: 100, min_samples_split: 2, splitter: random
k-NN	algorithm: auto, leaf_size: 10, n_neighbors: 100, weights: uniform
RF	criterion: gini, min_samples_leaf: 100, min_samples_split: 100, n_estimators: 50
XGBoost	booster: gblinear, learning_rate: 0.1, max_depth: 50, n_estimators: 100
CATBoost	depth: 5, iterations: 150, l2_leaf_reg: 1, learning_rate: 0.001
SVC	degree: 1, gamma: auto

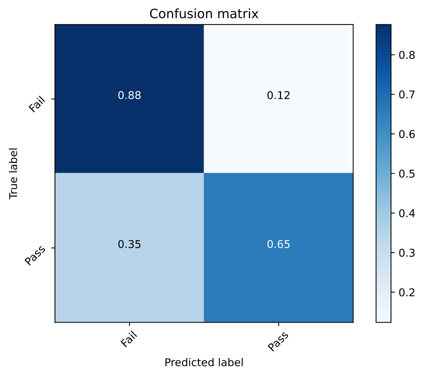
4.2.3 Results

Results for the Experiment 1B are presented qualitatively in tabular format and quantitatively in terms of plots. Table 4.4 presents the detailed quantitative test results for the implemented ML classification models in the context to students' performance prediction using the Coursera dataset. From the Table 4.4, it can be observed that, similar to Experiment 1A, the performance of ML models was comparable to each other with accuracy towards 80%. Although a higher accuracy was observed for the Coursera dataset classification in comparison to OULAD, however, the accuracy still needs to be improved towards practical implementation of such systems. The degraded performance may be attributed to the class imbalance and the inability of models to learn the important features from the dataset. In addition to class imbalance problem, there could be other reasons for degraded performance including the dataset complexity, dataset quality, lack of relevant features and lack of dataset samples.

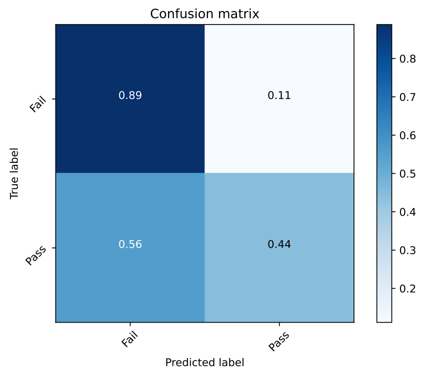
The performance of models was further explored for Type I and Type II errors, using the confusion matrix plots for each implemented model. From the Figure 4.7, it can be observed that a high Type II error was observed overall for all the models. The least Type II error was recorded for the MLP model (i.e., 35%). On the other hand, the Type I error was observed to be the least for the XGBoost (i.e, 10%). Overall, in terms of the balanced distribution of errors, from confusion matrices, MLP model can be nominated as the performing among all the implemented models. Figure 4.8 shows the ROC curves for the implemented models demonstrates that XGBoost, MLP and RF were among the performing models with an AUC around 0.85.

Table 4.4: Quantitative Test Results for Machine Learning (ML) Classification of Students' Performance using Coursera Dataset.

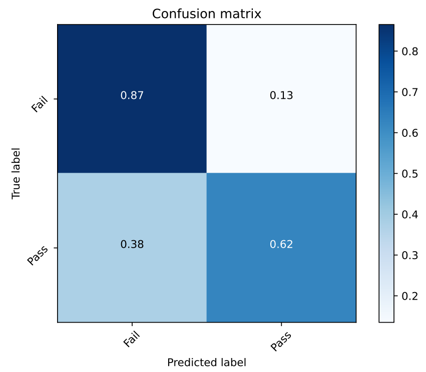
Models	Accuracy	F1 Score	Precision	Recall	Sensitivity	Specificity
MLP	0.80	0.75	0.79	0.76	0.88	0.65
DT	0.79	0.74	0.77	0.74	0.88	0.44
k-NN	0.79	0.74	0.77	0.74	0.87	0.62
RF	0.80	0.76	0.79	0.76	0.87	0.64
XGBoost	0.80	0.74	0.79	0.74	0.90	0.57
CATBoost	0.79	0.74	0.79	0.75	0.86	0.64
SVC	0.71	0.64	0.74	0.68	0.75	0.62



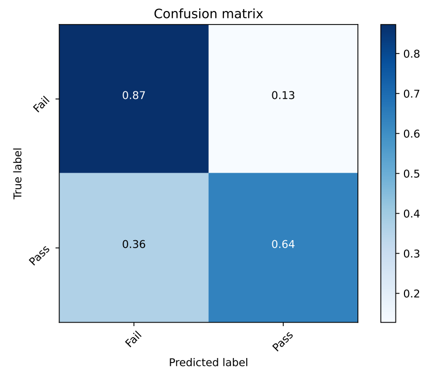
(a) MLP



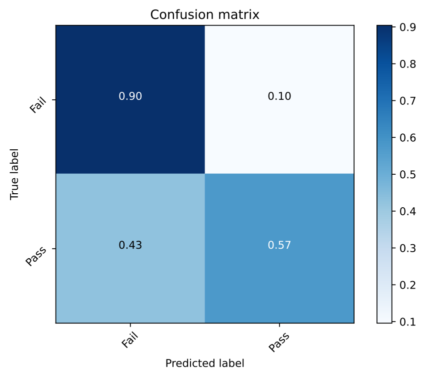
(b) DT



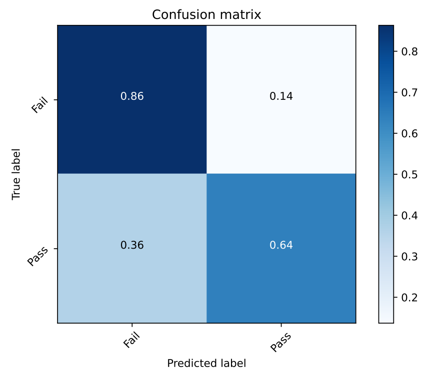
(c) k-NN



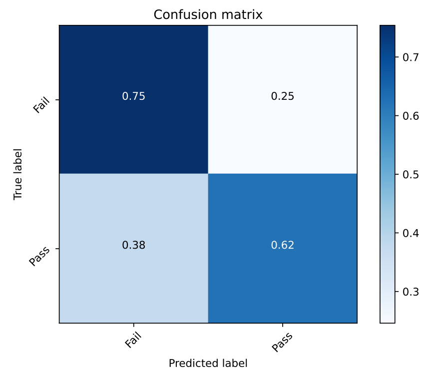
(d) RF



(e) XGBoost



(f) CATBoost



(g) SVC

Figure 4.7: Confusion Metrices of Implemented Machine Learning (ML) Models for Classification of Students' Performance using Coursera Dataset.

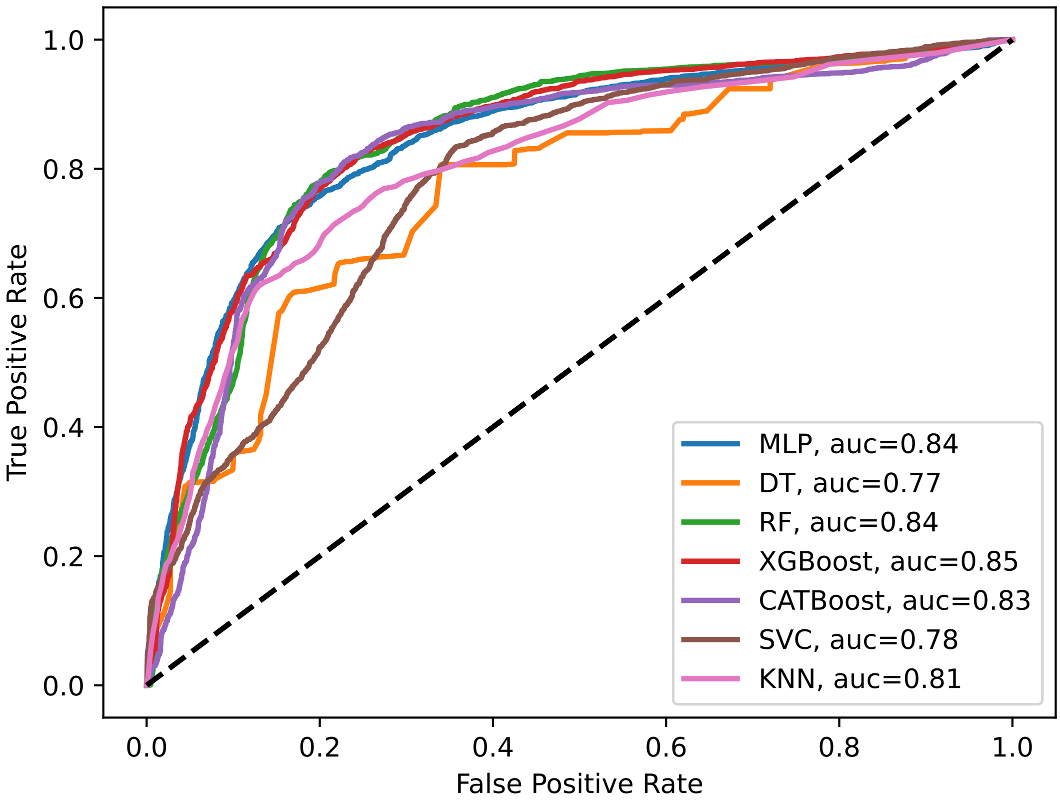


Figure 4.8: ROC Curves of the Implemented Machine Learning (ML) Models for the Classification of Students, Performance using Coursera Dataset.

4.3 Experiment 1C: Classification using the Coursera Dataset with Feature Engineering

In the Experiment 1C, the Coursera dataset was feature engineered and a new target variable was defined based on the grades scores to have three classes (High, Low, Medium) instead of conventional binary classes (Pass, Fail). Feature engineering is a crucial step in optimising the ML classification models and involves transforming raw data. Feature engineering enhances predictive performance, enables effective learning, handles data quality issues, captures non-linear relationships, reduces dimensionality, and aids in result interpretation. By extracting relevant information, creating new features, and addressing missing or noisy data, feature engineering improves the accuracy of models, facilitates pattern recognition, and ensures robustness. It also helps in understanding how different aspects of the data contribute to predictions or classifications. This section outlines the details of the implementation of multiple ML models for the three-class classification of the feature-engineered Coursera dataset towards achieving better performance. In this context, seven ML models including MLP, DT, RF, XGBoost, CATBoost, k-NN and SVC were implemented.

4.3.1 Dataset

For the investigations, performed under Experiment 1C, the Coursera dataset was feature engineered and a new target variable "performance" was added based on the range of course grades. The idea was to have three classes for High, Medium and Low performance measurement of students towards better prediction of ML models. The nan values from the dataset were removed as part of the data pre-processing and input features were normalized using Standard Scalar to get better ML performance. Figure 4.9 shows the correlation map of top 9 correlated features to the newly added "performance" target variable. From the correlation map, it can be observed that *hits_count* and *quiz_count* are the two most correlated features with negative correlation while *partic_count* and *video_duration* are the least correlated. Overall, the correlation trend remained the same for the original Coursera dataset, except for shift correlation sign for some features. Figure 4.10 shows the class distribution for the new target variable and clearly demonstrates the class imbalance.

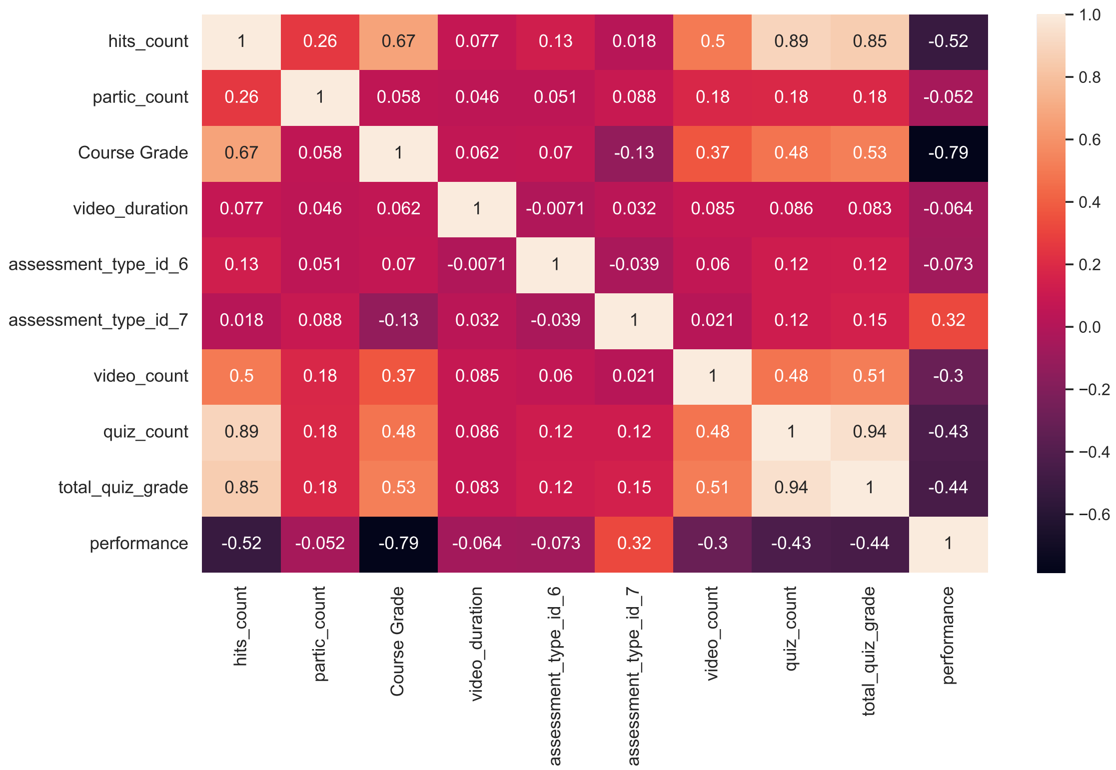


Figure 4.9: Feature Correlation Map for Coursera Dataset with Feature Engineering.

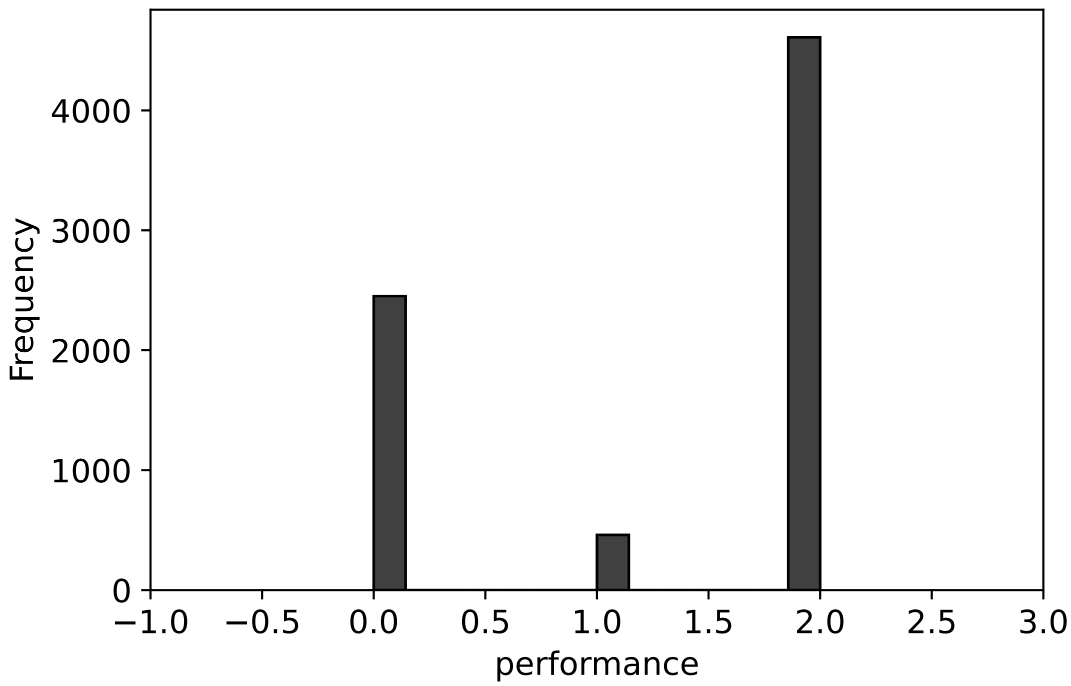


Figure 4.10: Class Distribution for Coursera Dataset with Feature Engineering.

4.3.2 Experimental Settings

For the investigations, performed under the Experiment 1C, similar experimental settings as reported in Section 4.1.2 for Experiment 1A were adopted. However, the values of hyperparamters resulted from Grid Search were reported different and are tabulated in Table 4.5.

Table 4.5: Hyperparamters for Machine Learning (ML) Models Implemented under Experiment 1A.

Model	Hyperparamters
MLP	activation: logistic, learning_rate_init: 0.001, solver: sgd, iter=500
DT	criterion: gini, min_samples_leaf: 100, min_samples_split: 2, splitter: random
k-NN	algorithm: auto, leaf_size: 10, n_neighbors: 100, weights: uniform
RF	criterion: gini, min_samples_leaf: 100, min_samples_split: 50, n_estimators: 30
XGBoost	booster: gblinear, learning_rate: 0.1, max_depth: 50, n_estimators: 100
CATBoost	depth: 5, iterations: 150, l2_leaf_reg: 0.5, learning_rate: 0.001
SVC	degree: 1, gamma: scale

4.3.3 Results

Results were presented quantitatively and graphically to better understand the performance of implemented ML models. Table 4.6 outlines the detailed quantitative results for the implemented models and demonstrate that XGBoost and CATBoost were the top performing models with accuracy of 81%. While MLP and RF were reported as the second best with accuracy of 80%. However, the overall performance of all the models was found comparable to each other with not much difference. The performance for three-class classification was slightly better in comparison to original binary classification, however, in terms of accuracy, there was no huge difference.

To further analyse the models in terms of Type I, Type II and Type III errors, the confusion matrices were plotted for each implemented models. In this case, Type II error is defined as the "instances from High class predicted as Low class". While the Type I error is defined as the "instances from Low class predicted as H class". From the Figure 4.11, it can be observed that the RF model was able to get the minimum value of Type II error (i.e., 2%), while the DT model got the best value of Type I error (i.e., 1%). Overall, it can be observed that Medium class was most often miss-classified as the High class and same is the case where High class was miss-classified as the Medium class. Overall, in terms of class distribution, DT model can be nominated as the best given that it offers the least miss-classifications between the distant classes. For example, there are the least miss-classifications between the High and Low classes for DT model. Most of the false predictions are between Medium and High classes which are acceptable to some extent and not as critical in comparison to the Low class predicted as Medium or High.

Table 4.6: Quantitative Test Results for Machine Learning (ML) Classification of Student,s Performance using Coursera Dataset with Feature Engineering.

Models	Accuracy	F1 Score	Precision	Recall	Sensitivity	Specificity
MLP	0.80	0.78	0.81	0.79	0.94	0.87
DT	0.77	0.72	0.79	0.74	0.87	0.87
k-NN	0.79	0.77	0.79	0.78	0.93	0.82
RF	0.80	0.76	0.83	0.76	0.97	0.85
XGBoost	0.81	0.79	0.83	0.80	0.96	0.88
CATBoost	0.81	0.81	0.84	0.81	0.96	0.88
SVC	0.78	0.75	0.79	0.77	0.93	0.81

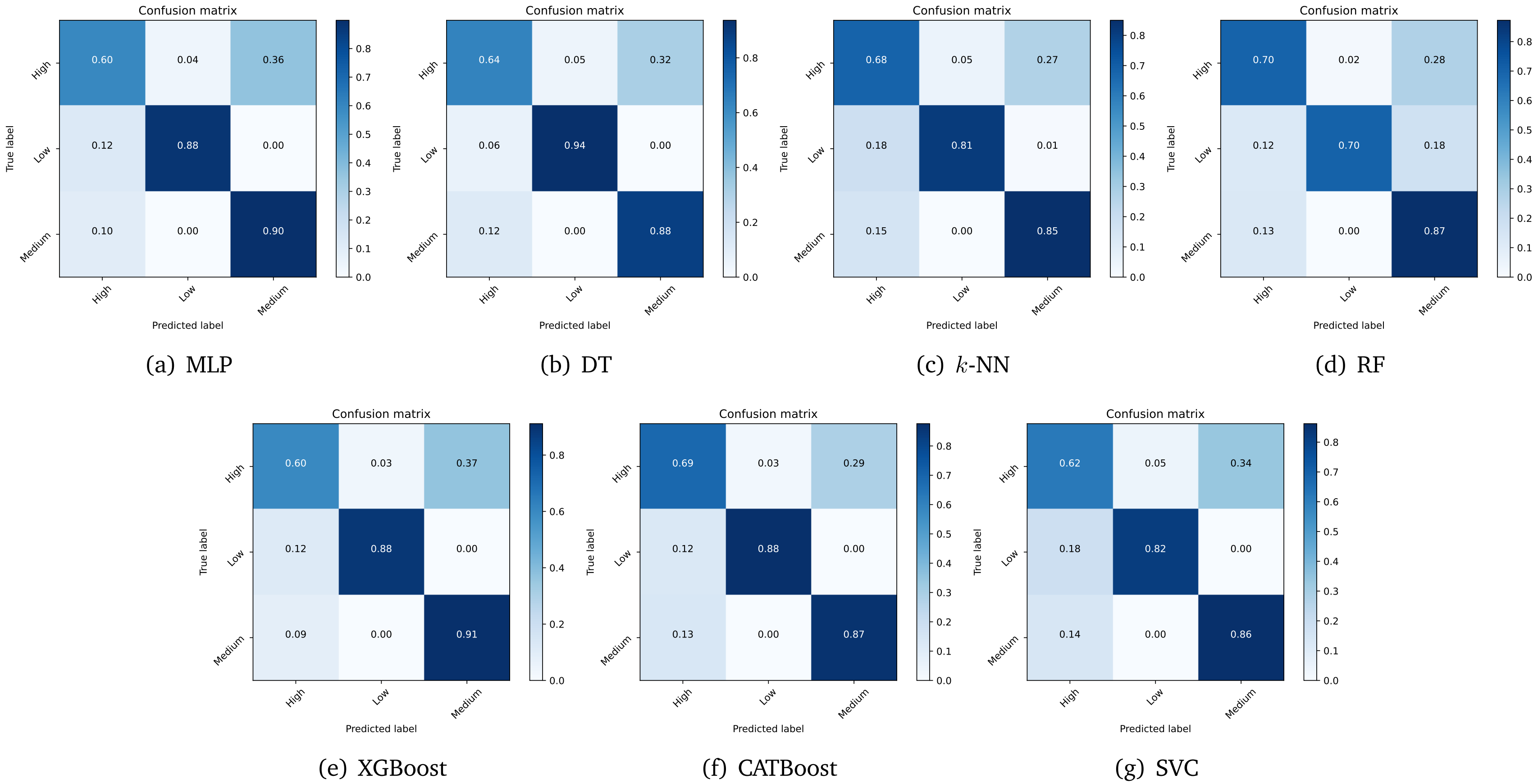


Figure 4.11: Confusion Metrics of Implemented Machine Learning (ML) Models for Classification of Students' Performance using Coursera Dataset with Feature Engineering.

4.4 Experiment 1D: Ensemble Classification using the Combined Coursera Database

Finally, an ensemble of three models (i.e., DT, k-NN, XGBoost) was implemented on the Coursera datasets. In this experiment two methods of Ensemble methods were used including stacking and voting. Each method has a specific approach of combining the results from the ensemble models. This experiment showed the implementation of multiple ML models for the Corusera dataset towards achieving better performance. Although, its a known fact that ensemble approaches require more computational resources, where available, they can be used towards improved classification accuracy.

4.4.1 Experimental Settings

For this experiment, similar experimental settings were used as reported for previous experiments in this chapter including the same programming language, same data split and same models. The performance of ensemble model was assessed in terms of classification accuracy.

4.4.2 Results

Quantitative results are presented in Table 4.7 for the classification performance of ensemble models in terms of the classification accuracy. From the results, it can be observed that an ensemble of DT, k-NN and XGBoost by the stacking method achieved the classification accuracy of 97% while by the voting method achieved 85%. This indicates a significant improvement in compared individual models implemented previously in Experiment 1B. The effectiveness of stacking can be attributed to its ability to capture the strengths and compensate for the weaknesses of individual base models. By combining the predictions of multiple models, stacking leverages their diverse perspectives and expertise, resulting in improved overall performance. Each base model may excel at capturing certain patterns or relationships in the data, and the meta-learner learns to weigh their predictions effectively, potentially leading to more accurate and robust predictions. Typically, a meta learner is trained on a separate dataset or using cross-validation techniques. The base models' predictions serve as the input features for the meta-learner, while the true class labels are used as the target variable for training. The meta-learner is a supervised learning algorithm, a decision tree in this case. The training process involves optimizing the meta-learner's parameters to minimize prediction errors and improve overall performance. From the results, it can be reported that the combination of multiple models resulted in better performance, however, may have impacted the inference speed and use of computational resources. Stacking can be more demanding compared to individual models. The process involves training multiple base models and a meta-learner, which requires additional computation time and memory resources. The increased computational resources required may pose challenges, especially when dealing with large datasets or limited computational capabilities. However, the impact of increased computational resources depends on the specific context and available resources. It's essential to consider the trade-off between improved performance and the practical constraints of computation time and resources when implementing stacking or any ensemble technique.

Table 4.7: Quantitative Test Results for Ensemble Classification of Student,s Performance using Coursera Dataset.

Ensemble Approach	Stacking	Voting
Classification Accuracy	0.97	0.85

4.5 Summary

This chapter presented the detailed information about the different experimental settings, performed under this research, in the connection to prediction of students' performance using different ML classification models (i.e., MLP, DT, RF, XGBOOST, CATBOOST, k-NN, SVC) for OULAD and Coursera datasets. All these models were implemented on the selected feature of the OULAD and Coursera datasets.

Results were presented in both qualitative and quantitative forms to highlight the importance of different experiments done. From the results, the importance of ML models, can easily be observed that almost all models comparatively performed similar with slight change of accuracy, score, perception, recall, sensitivity and specificity.

Overall, for the OULAD dataset, Coursera dataset, feature engineered Coursera dataset, and ensemble models Coursera dataset achieved high accuracy around 69%, 80%, 81%, and 97% respectively. The results suggested that the performance of models can be improved further. In this context the class imbalance problem was highlighted in all the investigations as the leading factor for the degraded performance. Degraded performance can be caused by various factors beyond class imbalance. Dataset complexity, including intricate patterns, noise, and outliers, can make it difficult for models to generalize effectively. Poor dataset quality, such as missing values or incorrect labels, hampers the model's ability to learn meaningful patterns. Insufficient or biased features may not capture relevant information for accurate predictions. Limited dataset samples can lead to overfitting and inadequate generalization. Data distribution shifts, where training and deployment data differ significantly, can adversely impact performance. Model complexity, hyperparameter tuning, and inadequate feature engineering also play crucial roles.

Chapter 5

Students' Performance Classification on Coursera for Each Subject

This chapter presents the details of Experiment 2, performed under this research where the performance of students was classified for each course under the Coursera dataset using ML models. The aim of this experiment is to investigate the impact of each course on the ML performance in comparison to the combined dataset consisting of eight courses. Results are presented in both qualitative and quantitative forms to highlight the important insights from the experiments.

5.1 Experiment 2A: Algorithms and Data Structures

Details about the ML investigations, performed using Algorithms and Data Structures course dataset from the Coursera are provided in this section. Performance of MLP, DT, RF, XGBoost, CATBoost, k-NN and SVC ML models has been compared important insights are reported.

5.1.1 Dataset

The dataset used for this investigation consisted of samples related to "Algorithms and Data Structures" course within the Coursera dataset. Number of input features and the target variables were the same as for the Coursera dataset. Figure 5.1 shows the correlation map between the input features and target variable. From the map, it can be observed that *hits_count* and *Course Grade* were the two most correlated features while the *video_duration* and *assessment_type_id_6* were the least correlated features. In terms of dataset target variable distribution (see Figure 5.2), there was a clear imbalance between "Pass" and "Fail" class was observed.

5.1.2 Experimental Settings

For the performed investigations, same experimental protocols were followed as reported in Section 4.1.2 in terms of programming language, package, data pre-processing, dataset split and evaluation measures. Only difference was the values of hyper paramters resulted from the Grid search reported in Table 5.2.

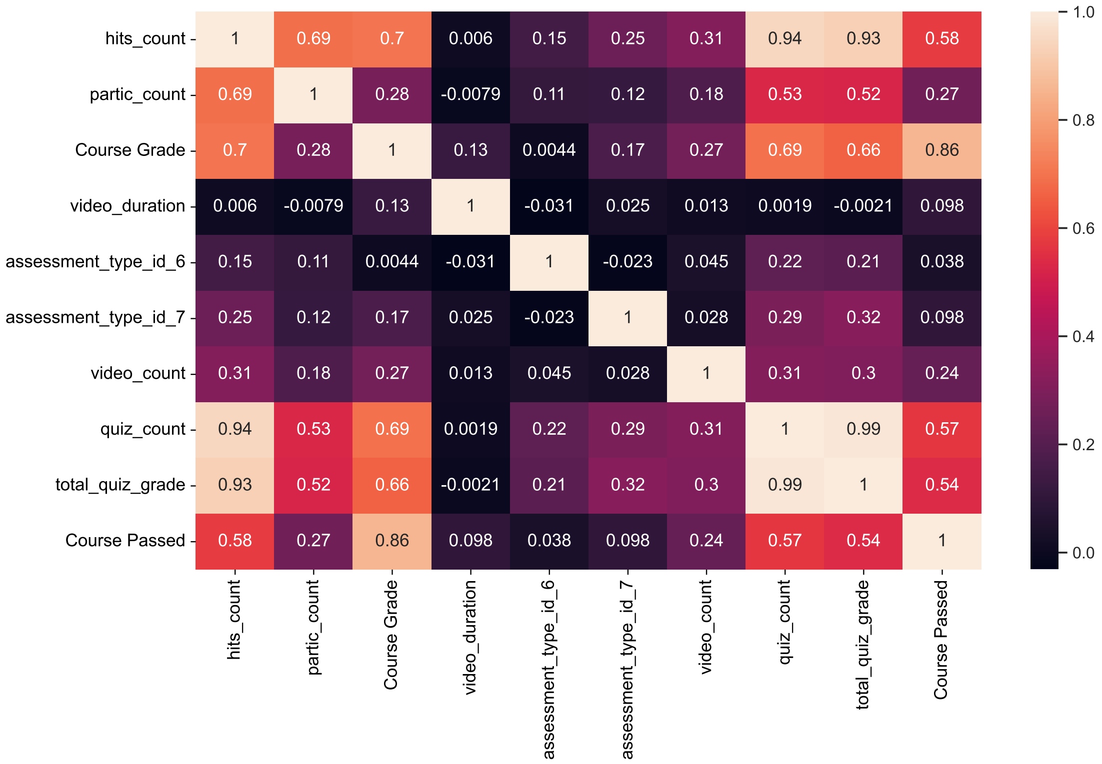


Figure 5.1: Feature Correlation Map for Coursera Algorithms and Data Structures Dataset.

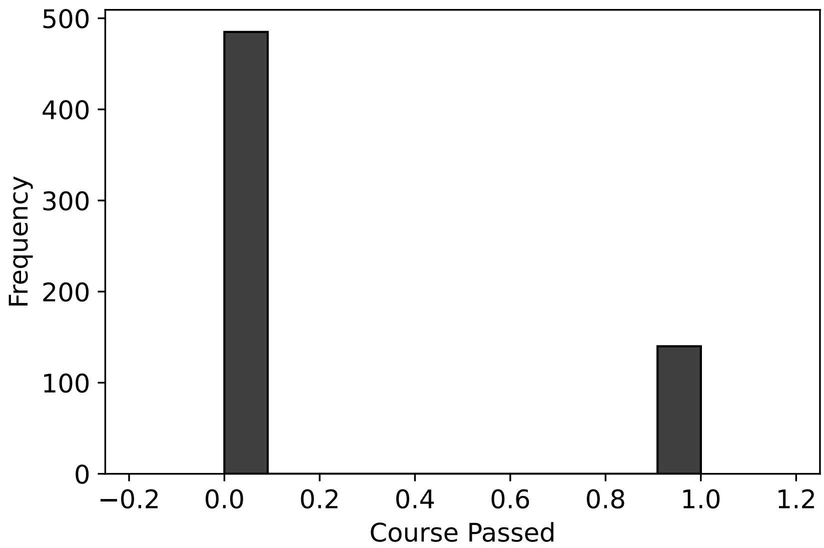


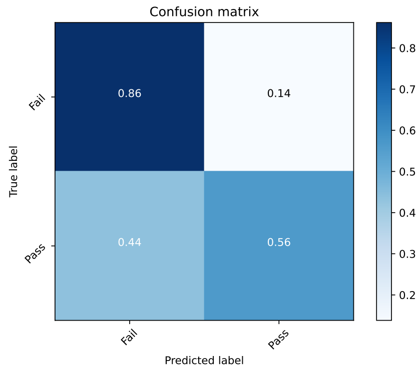
Figure 5.2: Class Distribution for Coursera Algorithms and Data Structures Dataset.

Table 5.1: Hyper parameters for Machine Learning (ML) Models Implemented under Experiment 2A.

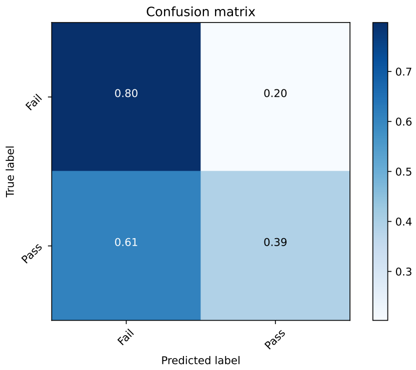
Model	Hyper parameters
MLP	activation: identity, learning_rate_init: 0.001, solver: adam, iter=500
DT	criterion: gini, min_samples_leaf: 100, min_samples_split: 2, splitter: random
k-NN	algorithm: auto, leaf_size: 10, n_neighbors: 50, weights: distance
RF	criterion: gini, min_samples_leaf: 100, min_samples_split: 50, n_estimators: 10
XGBoost	booster: gblinear, learning_rate: 0.1, max_depth: 3, n_estimators: 100
CATBoost	depth: 4, iterations: 150, l2_leaf_reg: 1, learning_rate: 0.001
SVC	degree: 1, gamma: scale

5.1.3 Results

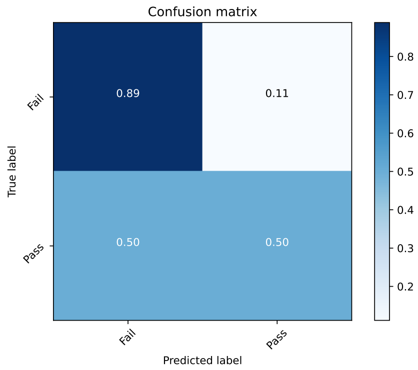
Results are presented quantitatively in the Table above and graphically in Figure 5.3 and Figure 5.4. Table in above presents the detailed quantitative test results for the implemented ML models in context to the students' performance prediction using Algorithms and Data Structure course dataset within the Coursera. From the table, XGB and k-NN models were the top performers with the classification accuracy of 0.80. CATBoost was reported as the worst model with an accuracy of 0.77. Overall, there was not much difference reported between the performance of models indicating the challenging nature of the dataset, mainly because of class imbalance.



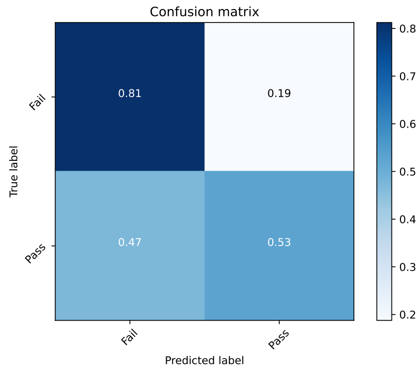
(a) MLP



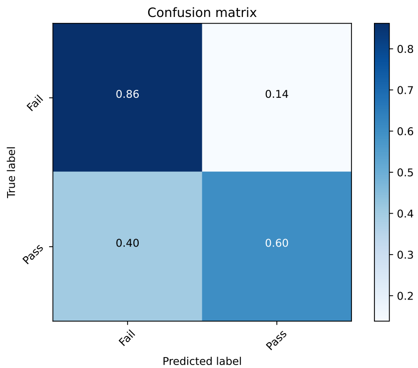
(b) DT



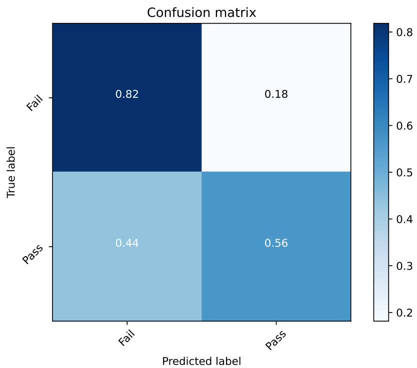
(c) k-NN



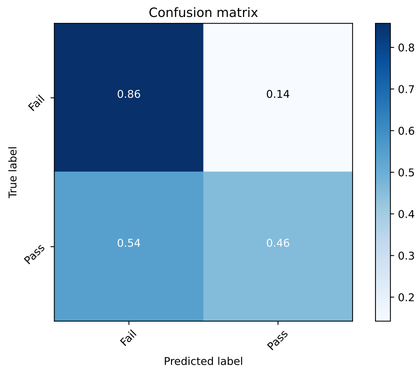
(d) RF



(e) XGBoost



(f) CATBoost



(g) SVC

Figure 5.3: Confusion Metrices of Implemented Machine Learning (ML) Models for Classification of Students, Performance using Coursera Algorithms and Data Structures Dataset.

Table 5.2: Quantitative Test Results for Machine Learning (ML) Classification of Students' Performance using Coursera Algorithms and Data Structures Dataset.

Models	Accuracy	F1 Score	Precision	Recall	Sensitivity	Specificity
MLP	0.79	0.69	0.77	0.72	0.85	0.56
DT	0.78	0.68	0.87	0.74	0.80	0.39
k-NN	0.80	0.65	0.77	0.69	0.89	0.5
RF	0.79	0.71	0.79	0.76	0.81	0.53
XGBoost	0.80	0.70	0.77	0.73	0.86	0.60
CATBoost	0.76	0.53	0.77	0.69	0.82	0.56
SVC	0.77	0.60	0.75	0.65	0.85	0.45

The performance of models was also assessed in terms of Type I and Type II errors from the confusion matrices plotted in Figure 5.3. It can be observed that XGBoost was able to achieve the least Type II error of 40% which on very higher end. On the other hand, k-NN model was able to achieve the best Type I error of only 11%. Overall, in terms of class distribution, XGBoost was reported as the best classifier for the Algorithms and Data Structures course. This can also be confirmed from the ROC curves reported in Figure 5.4 where XGBoost is among the smoothest curves with AUC of 0.78.

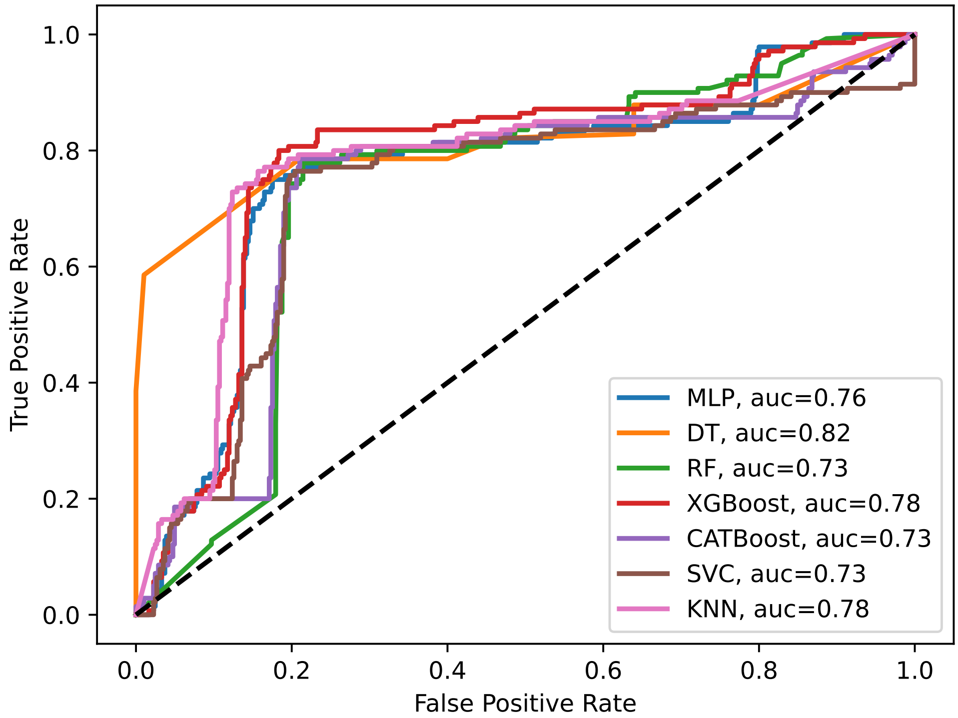


Figure 5.4: ROC Curves of the Implemented Machine Learning (ML) Models for the Classification of Students' Performance using Coursera Algorithms and Data Structures Dataset.

5.2 Experiment 2B: Computational Mathematics

The information in this section describes the ML analyses that were carried out using the Coursera dataset for the Computational Mathematics course. A comparison of the performance of the ML models MLP, DT, RF, XGBoost, CATBoost, k-NN, and SVC has shown some key findings.

5.2.1 Dataset

Samples from the "Computational Mathematics" course in the Coursera dataset made up the dataset used for this experiment. The number of input attributes and the target variables were the same as for the Coursera dataset. The correlation map between the input features and the target variable is shown in Figure 5.5. From the map, it can be seen that *hits_count* and *Course Grade* were the two most correlated features, whereas *assessment_type_id_6* and *assessment_type_id_7* were the least correlated features. There was a pronounced imbalance between the "Pass" and "Fail" classes observed in terms of the dataset target variable distribution (see Figure 5.6).

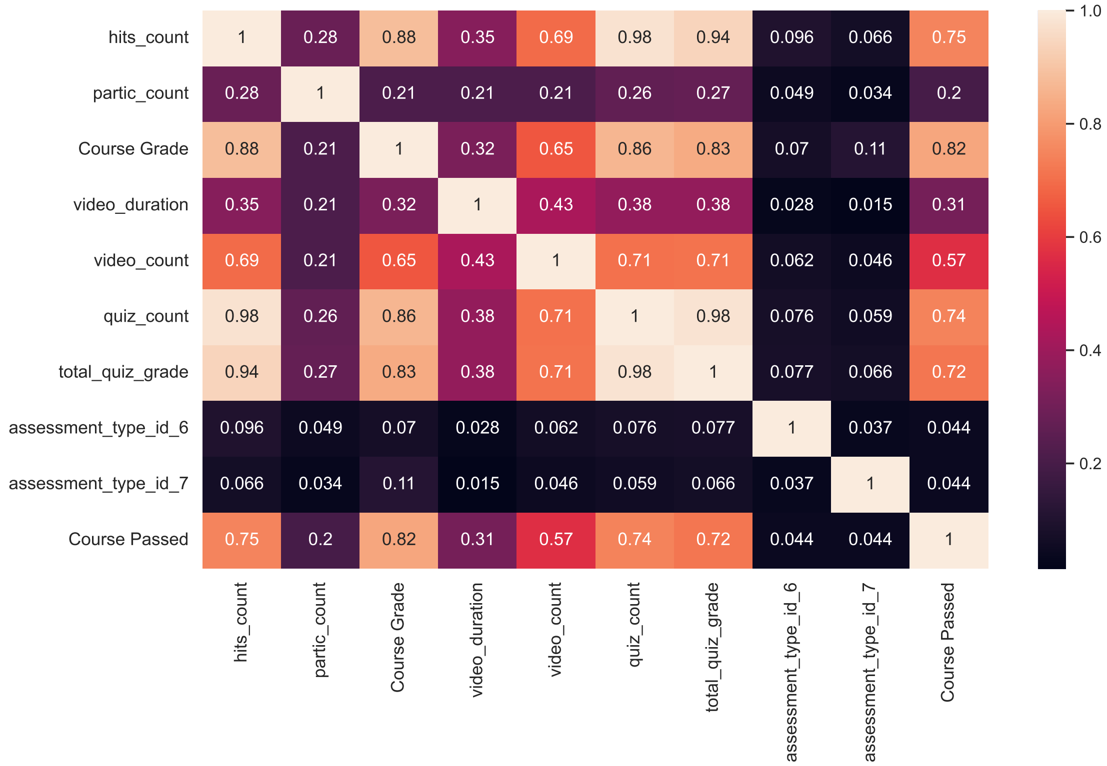


Figure 5.5: Feature Correlation Map for Coursera Computational Mathematics Dataset.

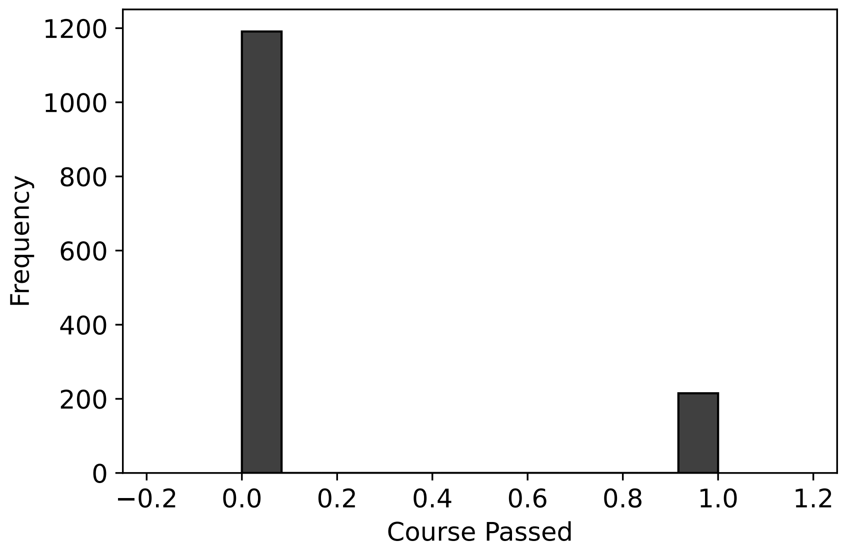


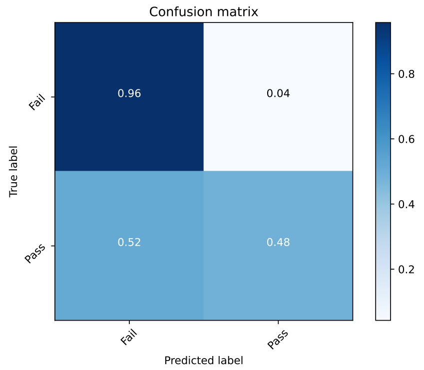
Figure 5.6: Class Distribution for Coursera Computational Mathematics Dataset.

5.2.2 Experimental Settings

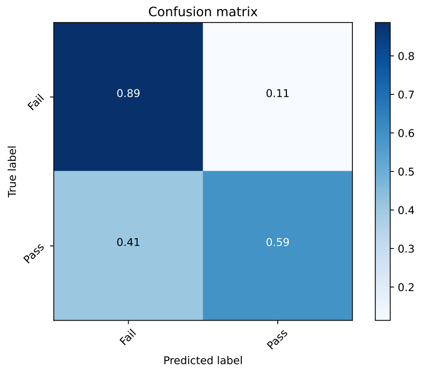
The programming language, package, data pre-processing, dataset splitting and evaluation metrics used for the research were the same as those presented in Section 4.1.2. The values of the hyperparameters obtained from the GridSearch and given in Table 5.3 were the only change.

Table 5.3: Hyperparameters for Machine Learning (ML) Models Implemented under Experiment 2B.

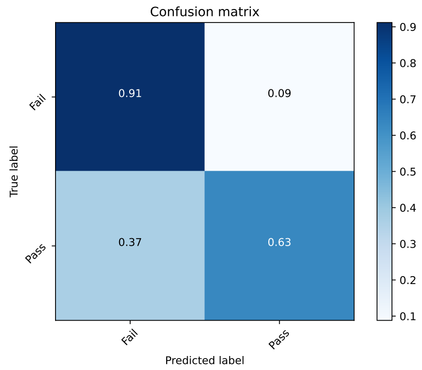
Model	**Hyperparameters**
MLP	activation: logistic, learning_rate_init: 0.001, solver: sgd, iter=500
DT	criterion: gini, min_samples_leaf: 10, min_samples_split: 30, splitter: random
k-NN	algorithm: auto, leaf_size: 10, n_neighbors: 100, weights: uniform
RF	criterion: entropy, min_samples_leaf: 100, min_samples_split: 100, n_estimators: 10
XGBoost	booster: gblinear, learning_rate: 0.001, max_depth: 3, n_estimators: 2
CATBoost	depth: 5, iterations: 150, l2_leaf_reg: 0.5, learning_rate: 0.001
SVC	degree: 1, gamma: auto



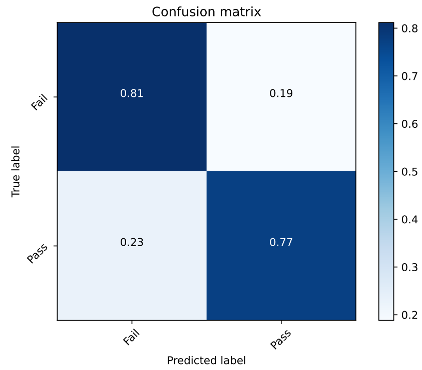
(a) MLP



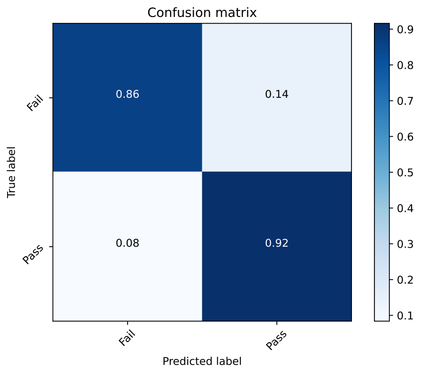
(b) DT



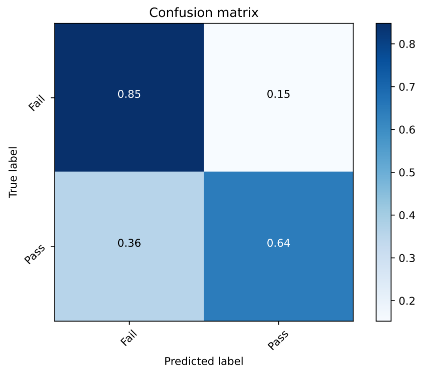
(c) k-NN



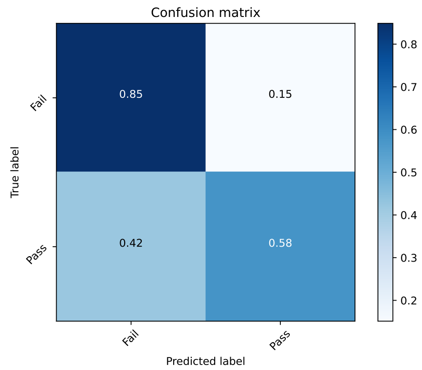
(d) RF



(e) XGBoost



(f) CATBoost



(g) SVC

Figure 5.7: Confusion Metrics of Implemented Machine Learning (ML) Models for Classification of Students' Performance using Coursera Computational Mathematics Dataset.

5.2.3 Results

Results are shown graphically in Figures 5.7 and 5.8 and quantitatively in Table 5.4. The comprehensive quantitative test results for the ML models are presented in Table 5.4 in relation to the Coursera dataset used to forecast students' success using Computational Mathematics. The highest performers in the table were the MLP and XGBoost models, with classification accuracy of 0.88. The worst model, according to reports, had an accuracy of 0.78 for DT. Overall, the models' performance revealed that the dataset was difficult, primarily due to class imbalance.

Table 5.4: Quantitative Test Results for Machine Learning (ML) Classification of Students' Performance using Coursera Computational Mathematics Dataset.

Models	Accuracy	F1 Score	Precision	Recall	Sensitivity	Specificity
MLP	0.88	0.70	0.90	0.71	0.95	0.48
DT	0.78	0.53	0.87	0.58	0.89	0.59
k-NN	0.87	0.75	0.90	0.77	0.91	0.63
RF	0.81	0.73	0.90	0.80	0.81	0.77
XGBoost	0.87	0.84	0.87	0.89	0.86	0.92
CATBoost	0.81	0.69	0.79	0.74	0.85	0.64
SVC	0.81	0.67	0.88	0.72	0.85	0.58

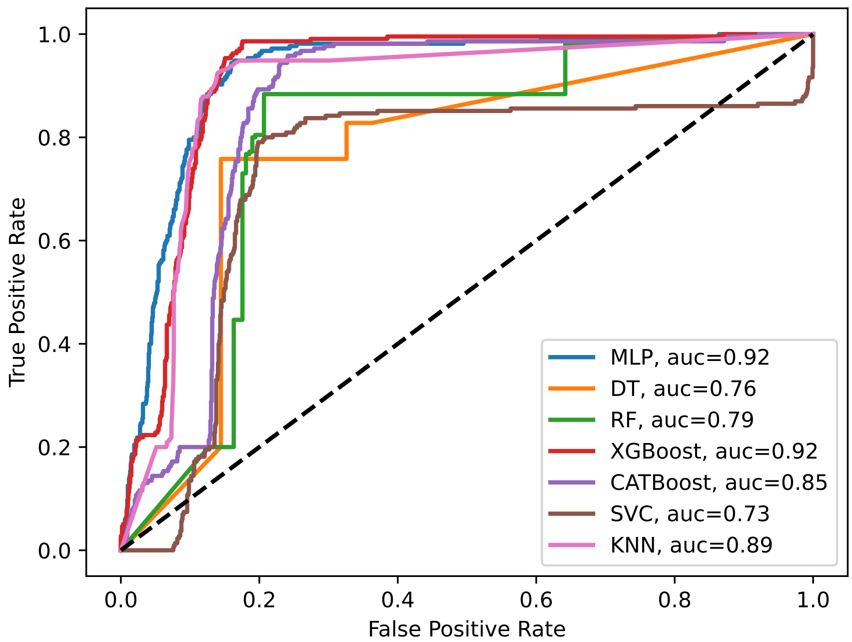


Figure 5.8: ROC Curves of the Implemented Machine Learning (ML) Models for the Classification of Students' Performance using Coursera Computational Mathematics Dataset.

In addition, Type I and Type II errors from the confusion matrices presented in Figure 5.7 were used to evaluate the performance of the models. It is clear that XGBoost was successful in achieving the lowest Type II error of 8%. The best Type I error, however, was only 4% for the MLP model. Overall, XGBoost was rated as the best classifier for the computational mathematics course in terms of class distribution. This is further supported by the ROC curves shown in Figure 5.8, where XGBoost has one of the best AUC values of 0.92.

5.3 Experiment 2C: Discrete Mathematics

In this part, specifics of the ML analyses conducted using the Coursera's Discrete Mathematics course dataset are provided. Important insights are presented after performance of the MLP, DT, RF, XGBoost, CATBoost, k-NN, and SVC ML models were compared.

5.3.1 Dataset

Samples from the Coursera dataset that were relevant to the course "Discrete Mathematics" made up the dataset used for this investigation. The target variables and the number of input parameters were the same as for the Coursera dataset. The correlation map between the input features and the target variable is displayed in Figure 5.9. The *total_quiz_grade* and *Course Grade* were the two features with the highest correlation, while *assessment_type_id_6* and *partic_count* had the lowest correlation, as seen on the map. The distribution of the dataset's target variables (see Figure 5.10) showed a balance between the "Pass" and "Fail" classes.

5.3.2 Experimental Settings

In terms of programming language, package, data pre-processing, dataset splitting, and evaluation metrics, the investigations were carried out in accordance with the experimental methods described in Section 4.1.2. The values of the hyperparameters from the GridSearch that were provided in Table 5.5 were the only things that differed.

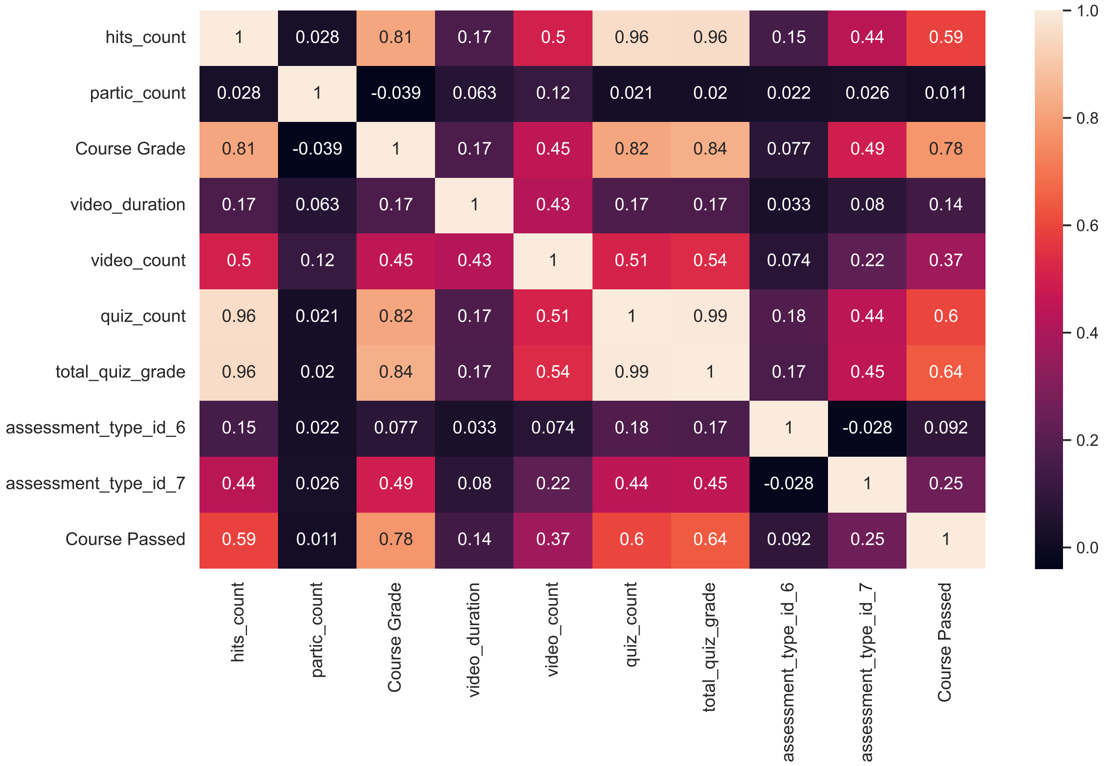


Figure 5.9: Feature Correlation Map for Coursera Discrete Mathematics Dataset.

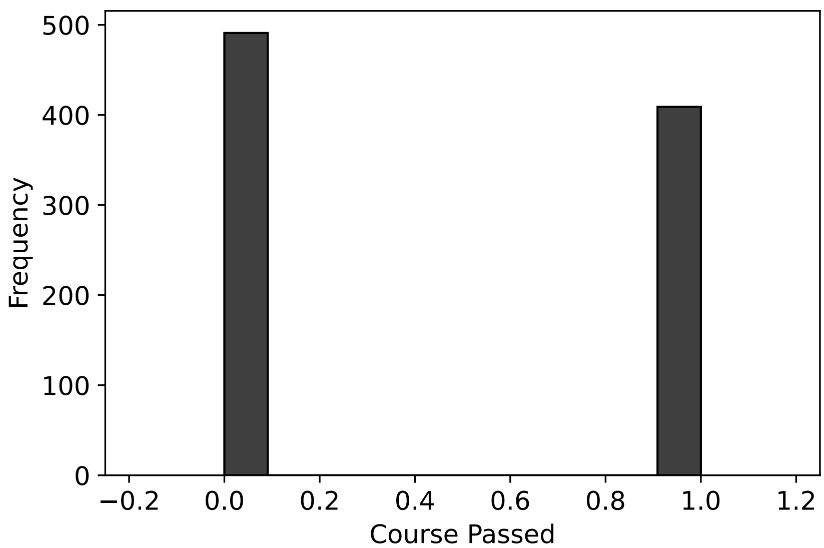


Figure 5.10: Class Distribution for Coursera Discrete Mathematics Dataset.

Table 5.5: Hyperparameters for Machine Learning (ML) Models Implemented under Experiment 2C.

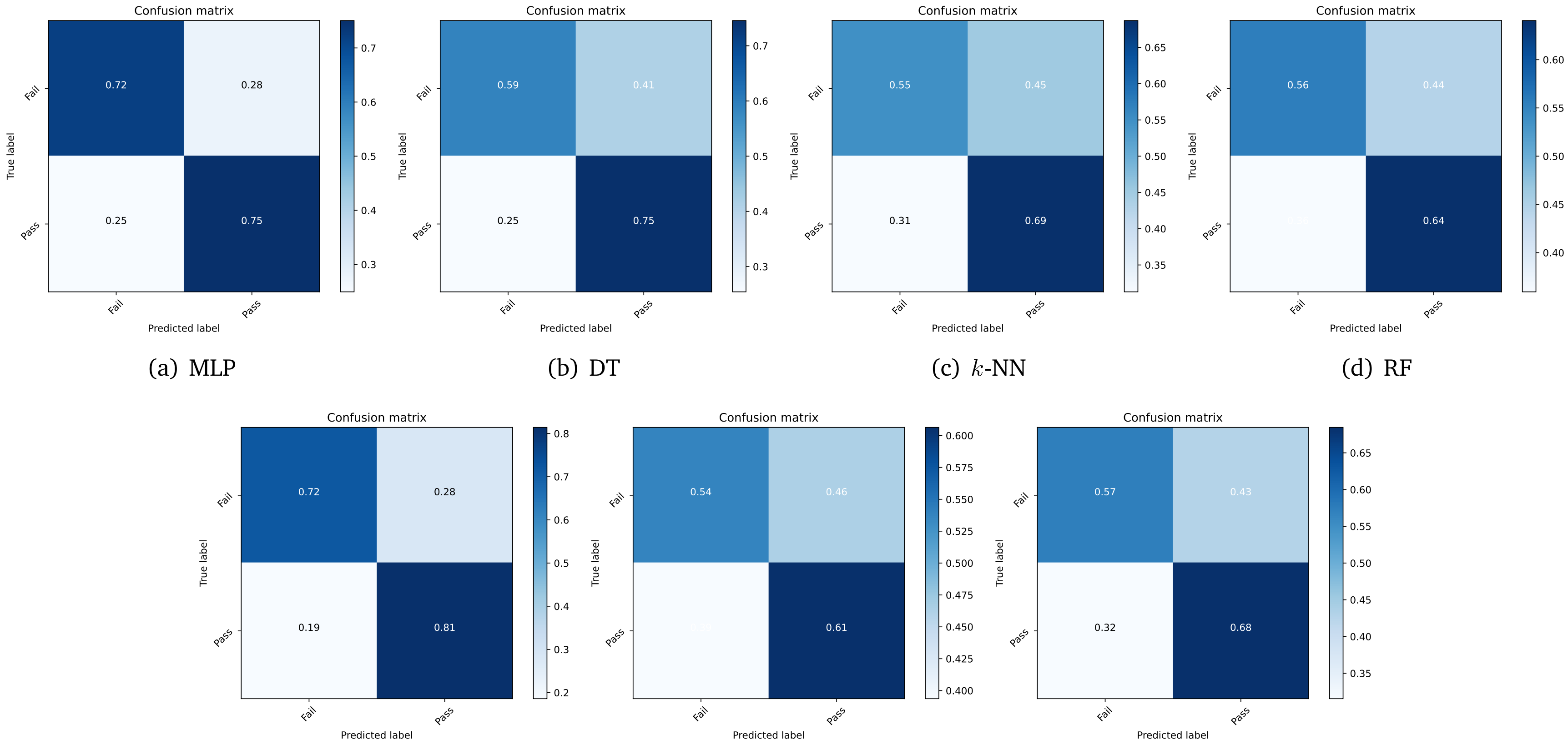
Model	Hyperparameters
MLP	activation: identity, learning_rate_init: 0.001, solver: sgd, iter=500
DT	criterion: gini, min_samples_leaf: 100, min_samples_split: 2, splitter: random
k-NN	algorithm: auto, leaf_size: 10, n_neighbors: 100, weights: uniform
RF	criterion: gini, min_samples_leaf: 100, min_samples_split: 10, n_estimators: 30
XGBoost	booster: gblinear, learning_rate: 0.001, max_depth: 3, n_estimators: 2
CATBoost	depth: 4, iterations: 150, l2_leaf_reg: 0.5, learning_rate: 0.001
SVC	degree: 1, gamma: scale

5.3.3 Results

Here the given results shown graphically in Figures 5.11 and 5.12 and quantitatively in Table 5.6. The comprehensive quantitative test results for the built ML models are shown in Table 5.6 in relation to the prediction of the students' performance using the Coursera dataset for the Discrete Mathematics course. From the table, XGBoost model was the top performer with the classification accuracy of 0.76. The worst model, CATBoost, was noted to have an accuracy of 0.57. Overall, the models' performance showed how difficult the dataset was to work with and how difficult it was for the models to train on the important attributes. Given that the dataset was balanced in this case, however, the degraded performance may be attributed to other factors including dataset complexity and dataset quality.

Table 5.6: Quantitative Test Results for Machine Learning (ML) Classification of Students' Performance using Coursera Discrete Mathematics Dataset.

Models	Accuracy	F1 Score	Precision	Recall	Sensitivity	Specificity
MLP	0.72	0.67	0.75	0.73	0.72	0.72
DT	0.62	0.54	0.79	0.63	0.59	0.75
k-NN	0.61	0.52	0.77	0.61	0.55	0.69
RF	0.61	0.51	0.68	0.61	0.56	0.64
XGBoost	0.76	0.71	0.77	0.77	0.72	0.81
CATBoost	0.57	0.46	0.54	0.57	0.54	0.61
SVC	0.62	0.54	0.77	0.62	0.57	0.68



(a) MLP (b) DT (c) k-NN (d) RF

(e) XGBoost (f) CATBoost (g) SVC

Figure 5.11: Confusion Metrices of Implemented Machine Learning (ML) Models for Classification of Students, Performance using Coursera Discrete Mathematics Dataset.

The effectiveness of the models was also evaluated in terms of Type I and Type II errors using the depicted confusion matrices in Figure 5.11. It is evident that XGBoost was successful in achieving Type II and Type I error rates of 19% and 28%, respectively. Overall, the best classifier for the Discrete Mathematics course was found to be XGBoost in terms of class distribution. This is further supported by the ROC curves shown in Figure 5.12, where XGBoost has one of the smoothest AUC values (0.82), ranking among the top curves.

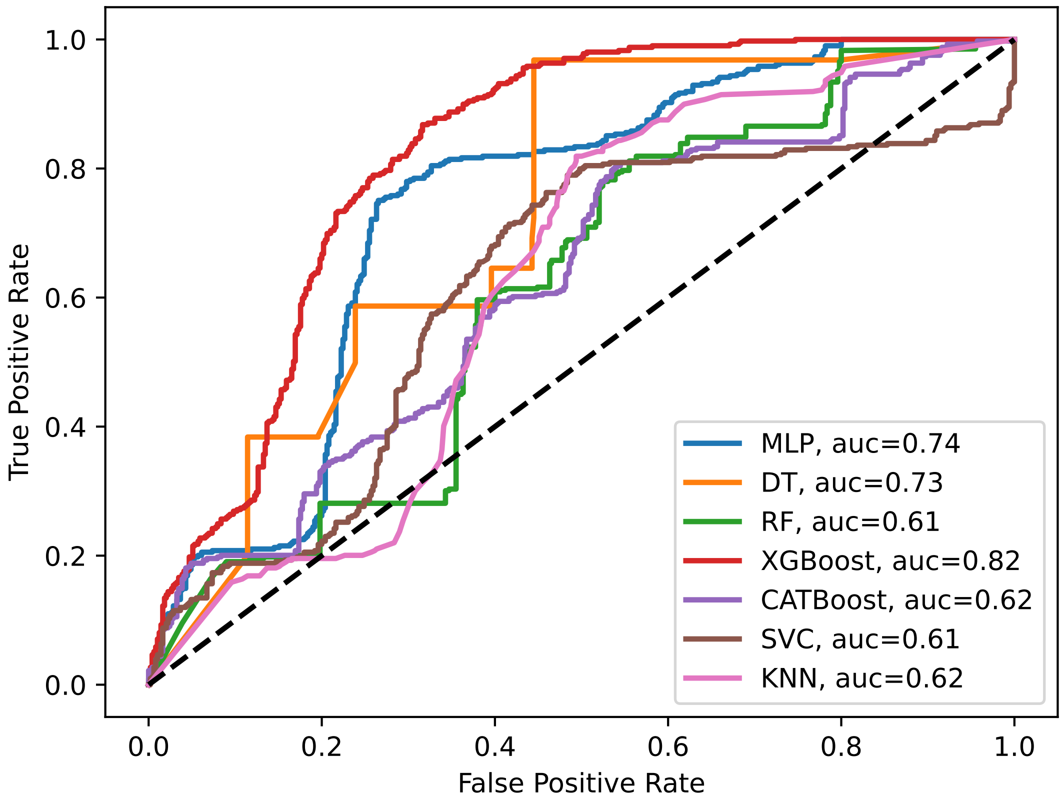


Figure 5.12: ROC Curves of the Implemented Machine Learning (ML) Models for the Classification of Students' Performance using Coursera Discrete Mathematics Dataset.

5.4 Experiment 2D: Fundamentals of Computer Science

This section has information about the ML tests that were done using the Coursera dataset from the Fundamentals of Computer Science course. MLP, DT, RF, XGBoost, CATBoost, k-NN, and SVC ML models' performance has been compared and important insights have been found.

5.4.1 Dataset

The Coursera dataset samples relevant to the "Fundamentals of Computer Science" course were utilised for this enquiry. The number of input features and output variables were identical to those of the Coursera dataset. The correlation map between the input features and the target variable is depicted in Figure 5.13. The map reveals that *quiz_count* and *Course Grade* are the two most correlated parameters, but *assessment_type_id_6* and *video_duration* are the least correlated. Regarding the target variable distribution of the dataset (see Figure 5.14), an imbalance between the "Pass" and "Fail" classes was identified.

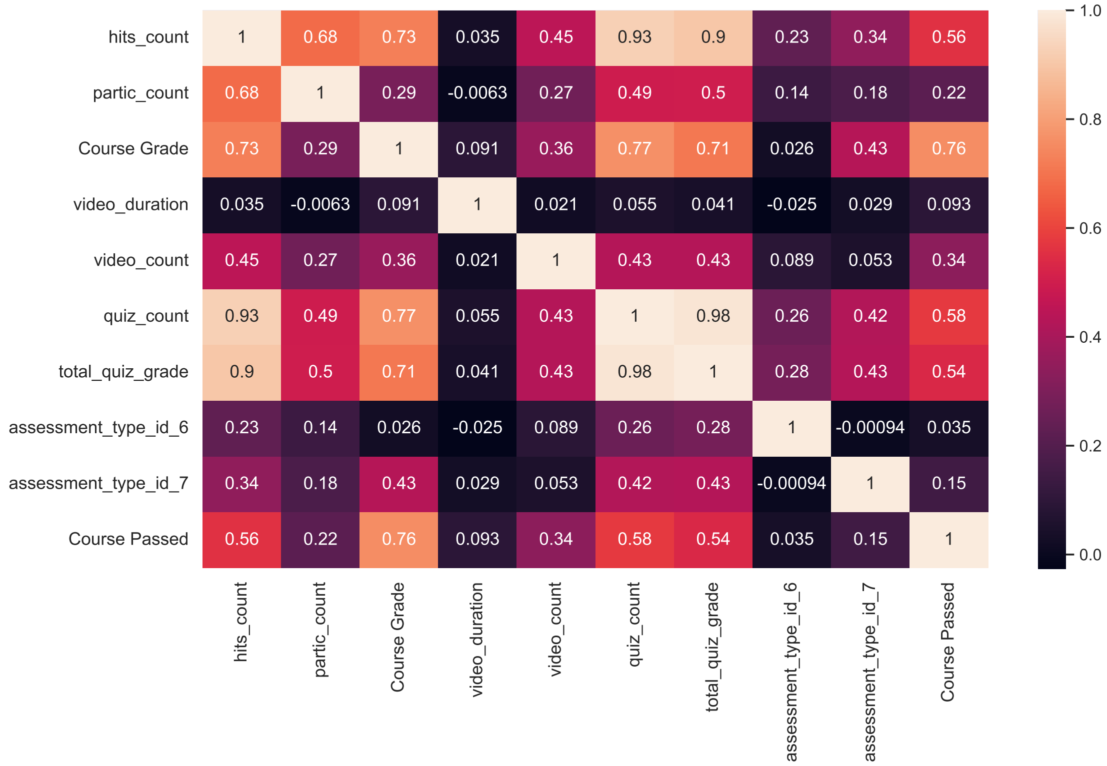


Figure 5.13: Feature Correlation Map for Coursera Fundamentals of Computer Science Dataset.

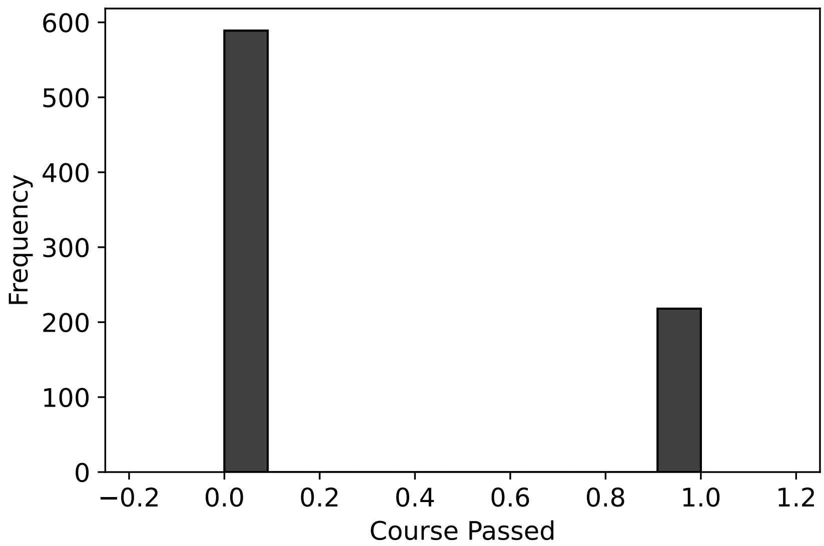


Figure 5.14: Class Distribution for Coursera Fundamentals of Computer Science Dataset.

5.4.2 Experimental Settings

When experimental procedures that were mentioned in Section 4.1.2 were adhered to. These protocols included the programming language, the package, the data pre-processing, the dataset split and the evaluation measures. The only thing that was different was the values of the hyperparameters that were reported in Table 5.7 after the Grid Search had been run.

Table 5.7: Hyperparameters for Machine Learning (ML) Models Implemented under Experiment 2D.

Model	Hyperparamters
MLP	activation: identity, learning_rate_init: 0.1, solver: lbfgs, iter=500
DT	criterion: entropy, min_samples_leaf: 10, min_samples_split: 100, splitter: random
k-NN	algorithm: auto, leaf_size: 10, n_neighbors: 50, weights: uniform
RF	criterion: entropy, min_samples_leaf: 100, min_samples_split: 10, n_estimators: 10
XGBoost	booster: gblinear, learning_rate: 0.1, max_depth: 10, n_estimators: 2
CATBoost	depth: 4, iterations: 150, l2_leaf_reg: 0.5, learning_rate: 0.001
SVC	degree: 1, gamma: scale

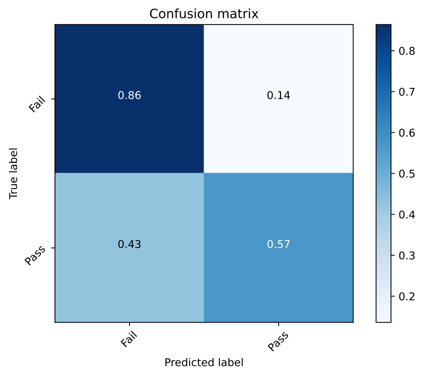
5.4.3 Results

Table 5.8 presents the detailed quantitative test results for the implemented ML models in context to the students' performance prediction using Fundamentals of Computer Science course dataset within the Coursera. Results are presented quantitatively in Table 5.8 and graphically in Figure 5.15 and Figure 5.16. According to the table, the k-NN model performed best, with a classification accuracy of 0.81. CATBoost and DT were reported as the worst models with an accuracy of 0.74. Overall, the models' performance showed that the dataset was difficult, the target class was unbalanced and the models were unable to capture the important features during training.

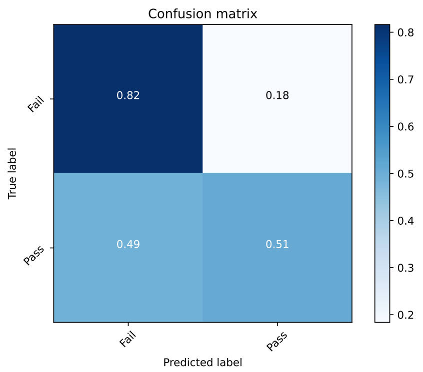
Table 5.8: Quantitative Test Results for Machine Learning (ML) Classification of Students' Performance using Coursera Fundamentals of Computer Science Dataset.

Models	Accuracy	F1 Score	Precision	Recall	Sensitivity	Specificity
MLP	0.79	0.69	0.76	0.71	0.86	0.57
DT	0.74	0.60	0.63	0.66	0.82	0.51
k-NN	0.81	0.74	0.88	0.75	0.88	0.63
RF	0.76	0.67	0.76	0.72	0.79	0.57
XGBoost	0.79	0.75	0.83	0.80	0.78	0.82
CATBoost	0.74	0.63	0.75	0.67	0.81	0.54
SVC	0.77	0.69	0.86	0.71	0.84	0.59

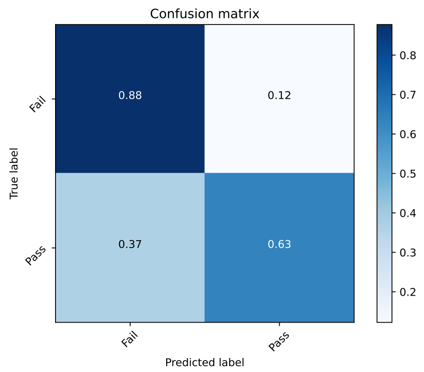
In addition, Type I and Type II errors from the confusion matrices presented in Figure 5.15 were used to evaluate the performance of the models. It is clear that XGBoost was able to obtain Type II error of 18%. The k-NN model, on the other hand, managed to attain the best Type I error of 12%. Overall, XGBoost was rated as the best classifier for the Fundamentals of Computer Science course in terms of class distribution. This can also be validated from the ROC curves given in Figure 5.16 where XGBoost is among the smoothest curves with AUC of 0.82.



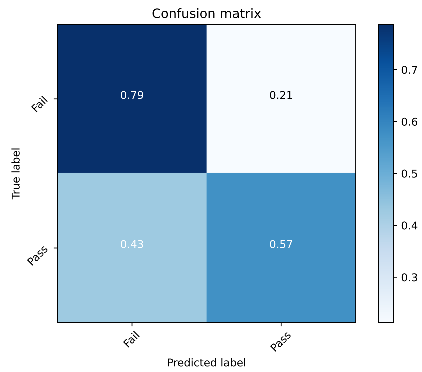
(a) MLP



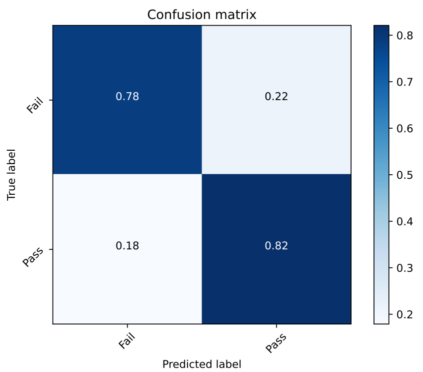
(b) DT



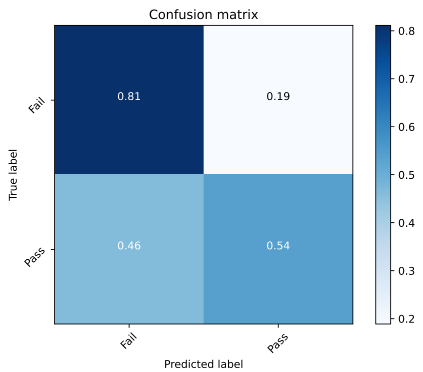
(c) k-NN



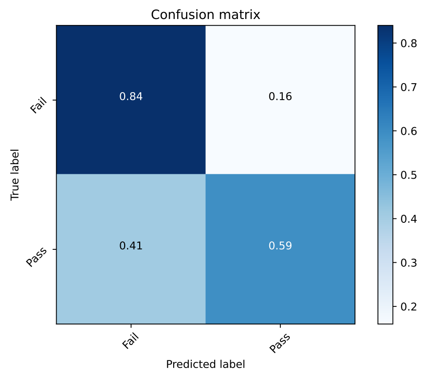
(d) RF



(e) XGBoost



(f) CATBoost



(g) SVC

Figure 5.15: Confusion Metrics of Implemented Machine Learning (ML) Models for Classification of Students' Performance using Coursera Fundamentals of Computer Science Dataset.

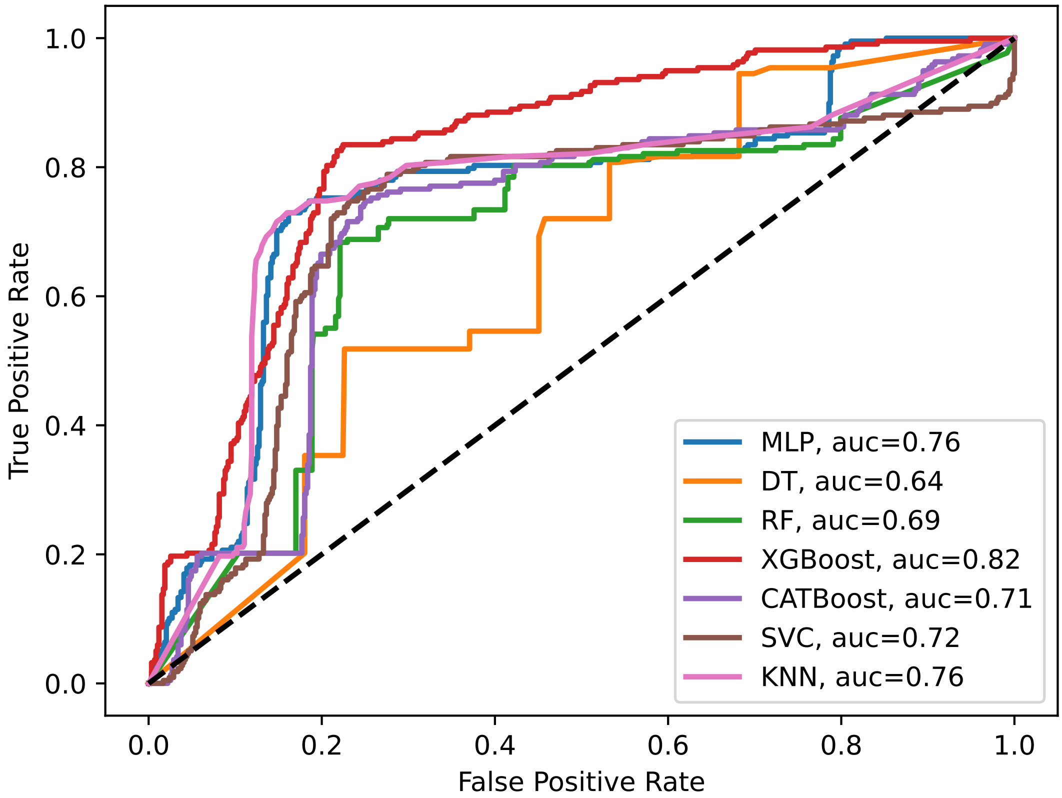


Figure 5.16: ROC Curves of the Implemented Machine Learning (ML) Models for the Classification of Students' Performance using Coursera Fundamentals of Computer Science Dataset.

5.5 Experiment 2E: How Computers Work?

In this part of the experiment, we detail our ML experiments with the How Computers Work dataset from Coursera. Results comparing the effectiveness of various ML models, including MLP, DT, RF, XGBoost, CATBoost, k-NN, and SVC are provided.

5.5.1 Dataset

To conduct this study, we used data from the Coursera database, specifically the "How Computers Work" course. Both the number of input features and the number of variables used in the training were consistent with those found in the Coursera dataset. The input feature to output variable correlation map is depicted in Figure 5.17. The map shows that the features with the highest correlation are *hits_count* and *Course Grade*, whereas the features with the lowest correlation are *assessment_type_id_6* and *assessment_type_id_7*. There was a discrepancy between the distributions of the Pass and Fail classes for the dataset's target variables (Figure 5.18).

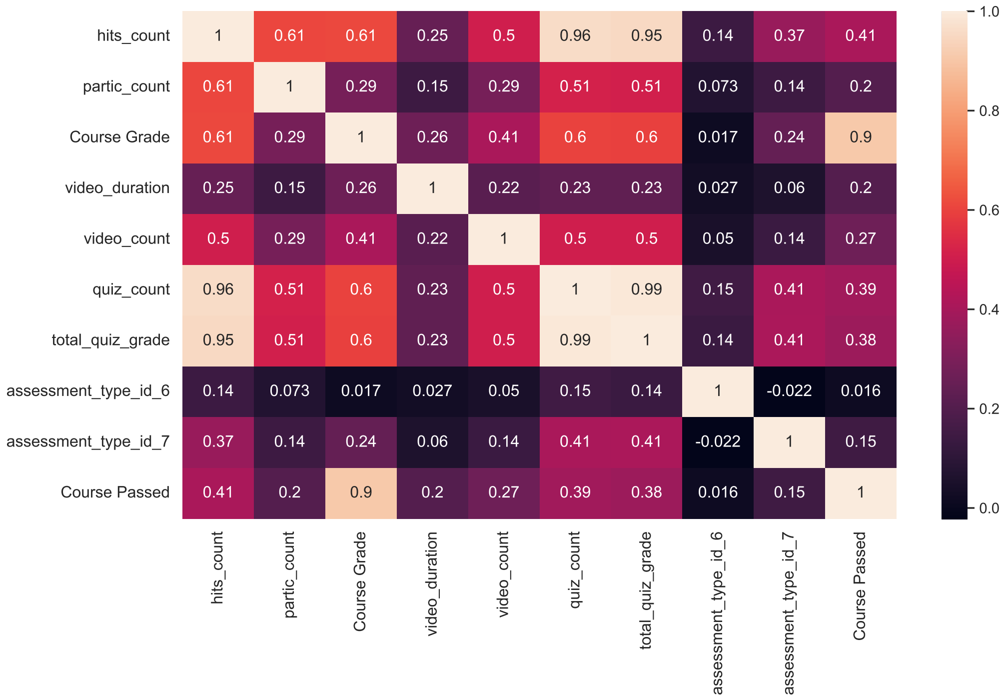


Figure 5.17: Feature Correlation Map for Coursera How Computers Work, Dataset?

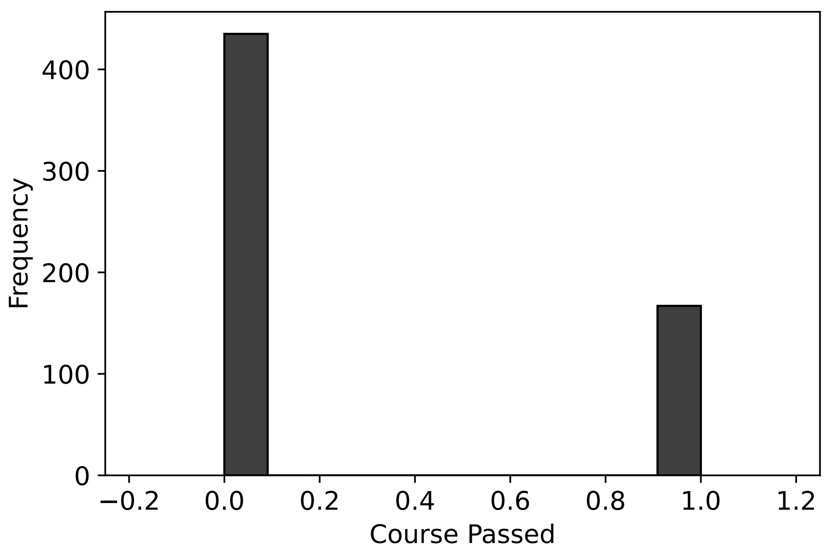


Figure 5.18: Class Distribution for Coursera How Computers Work, Dataset?

5.5.2 Experimental Settings

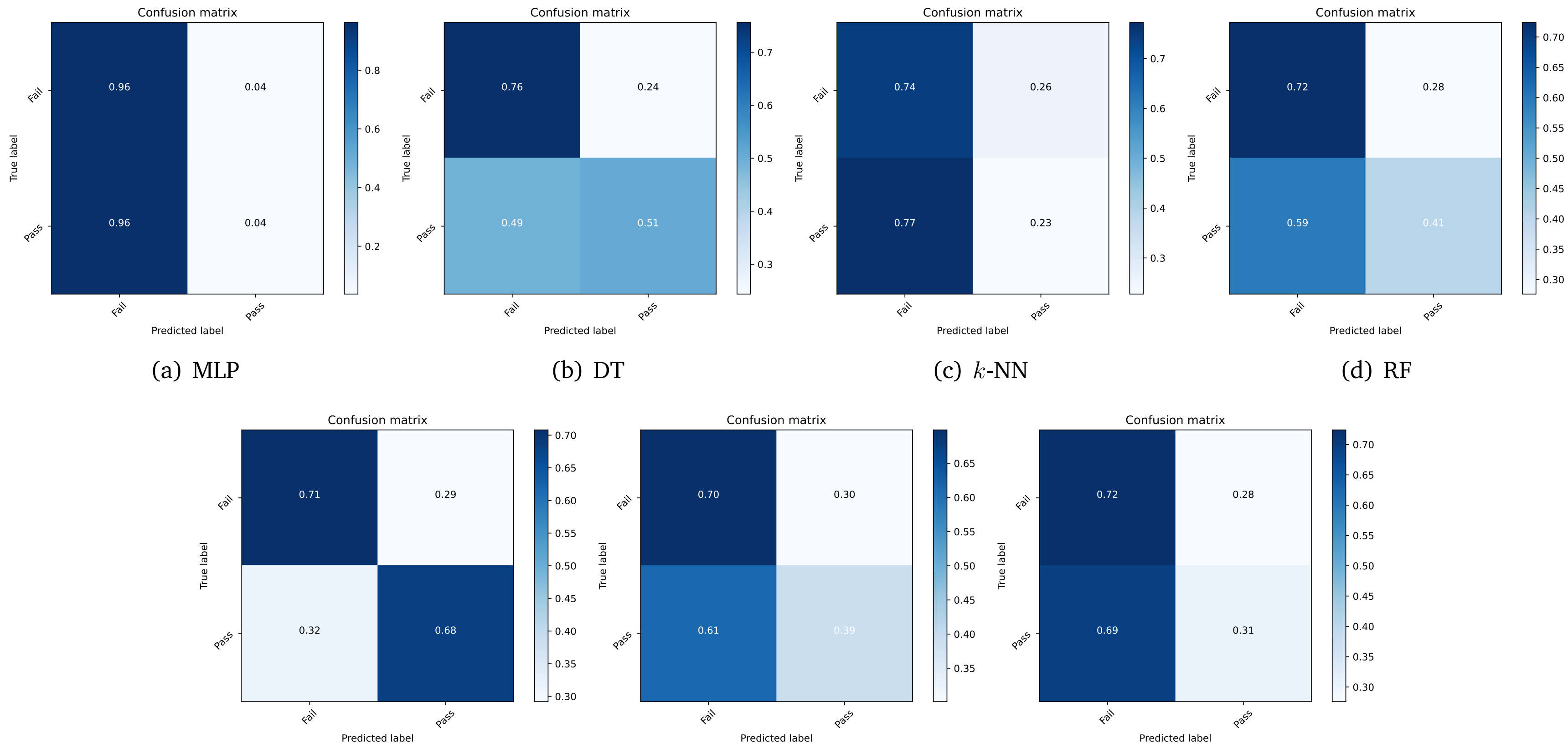
Experiments were conducted using the same experimental techniques as those described in Section 4.1.2, including the same programming language, package, data pre-processing, dataset splitting, and evaluation measures. The only variation was in the GridSearch-reported hypeparameter values (Table 5.9).

Table 5.9: Hyperparameters for Machine Learning (ML) Models Implemented under Experiment 2E.

Model	Hyperparameters
MLP	activation: logistic, learning_rate_init: 0.001, solver: sgd, iter=500
DT	criterion: entropy, min_samples_leaf: 100, min_samples_split: 2, splitter: best
k-NN	algorithm: auto, leaf_size: 10, n_neighbors: 100, weights: uniform
RF	criterion: entropy, min_samples_leaf: 100, min_samples_split: 2, n_estimators: 10
XGBoost	booster: gblinear, learning_rate: 0.001, max_depth: 3, n_estimators: 2
CATBoost	depth: 4, iterations: 150, l2_leaf_reg: 0.5, learning_rate: 0.001
SVC	degree: 1, gamma: auto

5.5.3 Results

Table 5.10 and Figures 5.19 and 5.20 provide numerical and graphical representations of the results, respectively. Using data from a How Computers Work course on Coursera, Table 5.10 displays the quantitative results of extensive testing conducted on the several ML models that were deployed for the purpose of predicting students' grades in that course. Classification accuracy of 0.71 indicates that MLP model is the best performer in the table. It was found that k-NN was the least accurate model, with a reported accuracy of 0.60. The overall performance of the models showed that the dataset was more difficult than expected, that the target class was unbalanced, and that the models failed to adequately capture the most important properties during training.



(a) MLP (b) DT (c) k-NN (d) RF

(e) XGBoost (f) CATBoost (g) SVC

Figure 5.19: Confusion Metrics of Implemented Machine Learning (ML) Models for Classification of Students? Performance using Coursera How Computers Work Dataset?

Table 5.10: Quantitative Test Results for Machine Learning (ML) Classification of Students' Performance using Coursera, How Computers Work Dataset.

Models	Accuracy	F1 Score	Precision	Recall	Sensitivity	Specificity
MLP	0.71	0.42	0.38	0.50	0.96	0.04
DT	0.69	0.57	0.67	0.63	0.76	0.51
k-NN	0.60	0.39	0.52	0.48	0.74	0.23
RF	0.62	0.43	0.49	0.55	0.72	0.41
XGBoost	0.70	0.62	0.64	0.69	0.71	0.68
CATBoost	0.61	0.45	0.46	0.54	0.70	0.39
SVC	0.61	0.44	0.70	0.51	0.72	0.31

The confusion matrices displayed in Figure 5.19 were also used to evaluate the models' performance in terms of Type I and Type II errors. The lowest Type II error (32%), obtained by XGBoost, stands out. However, the MLP model had the lowest Type I error, at 4%. The How Computers Work course's class distribution was shown to be optimal with XGBoost. Figure 5.20's ROC curves corroborate this, showing that XGBoost is one of the smoothest curves with an AUC of 0.68.

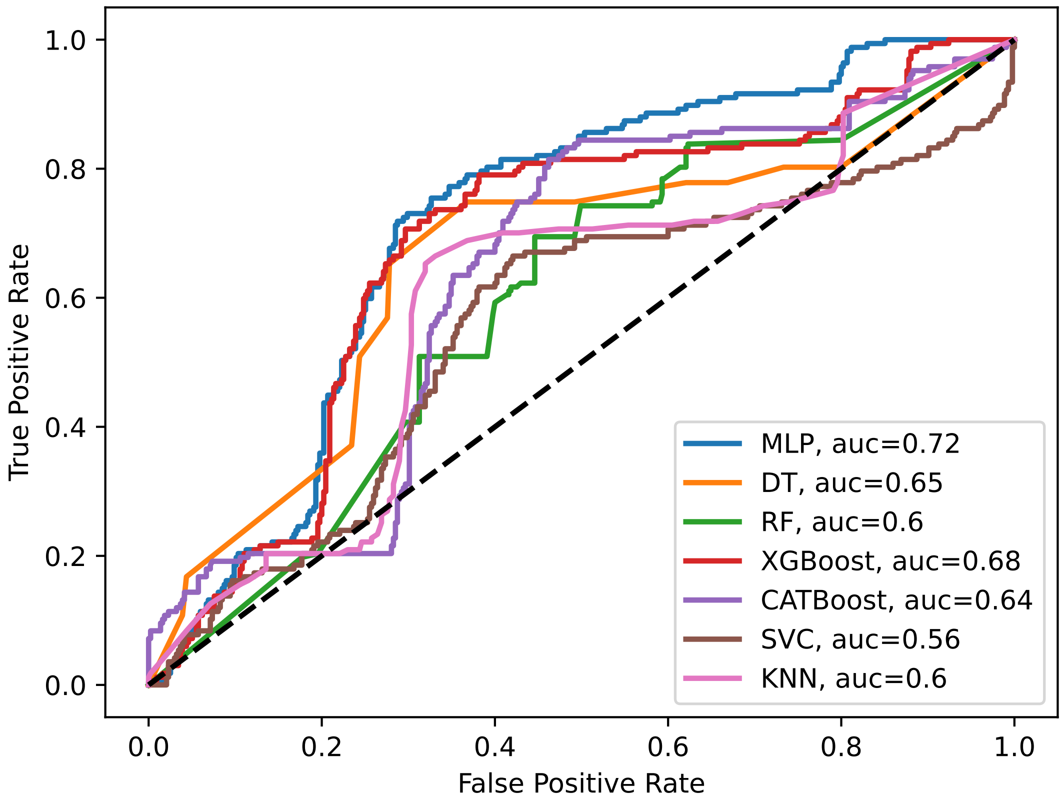


Figure 5.20: ROC Curves of the Implemented Machine Learning (ML) Models for the Classification of Students' Performance using Coursera, How Computers Work Dataset?

5.6 Experiment 2F: Introduction to Programming 1

Details about the ML investigations performed using Introduction to Programming 1 course dataset from the Coursera are provided in this section. Performance of MLP, DT, RF, XGBoost, CATBoost, k-NN and SVC ML models has been compared important insights are reported.

5.6.1 Dataset

Samples from the Coursera dataset that were relevant to the course "Introduction to Programming 1" made up the dataset, used for this enquiry. The target variables and the quantity of input features were the same as for the Coursera dataset. The correlation map between the input features and the target variable is displayed in Figure 5.21. The *hits_count* and *Course Grade* were the two features with the highest correlation and the *partic_count* and *video_duration* had the lowest correlation, as seen on the map. The distribution of the dataset's target variables (see Figure 5.22) showed a balance between the "Pass" and "Fail" classes.

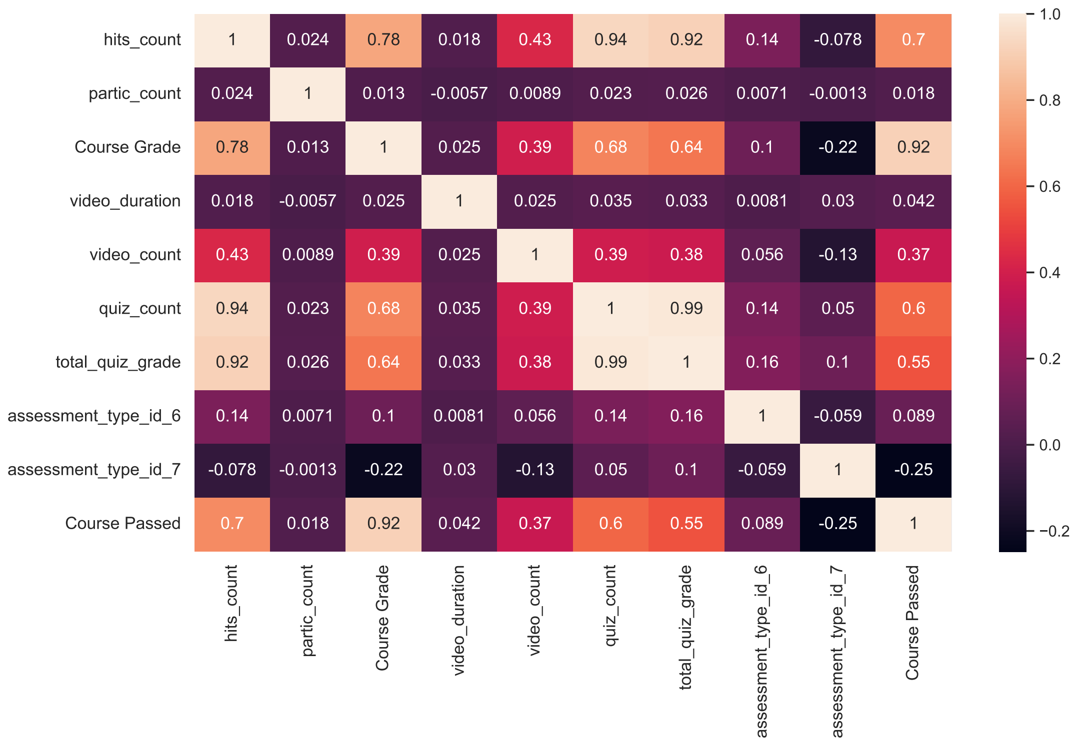


Figure 5.21: Feature Correlation Map for Coursera Introduction to Programming 1 Dataset.

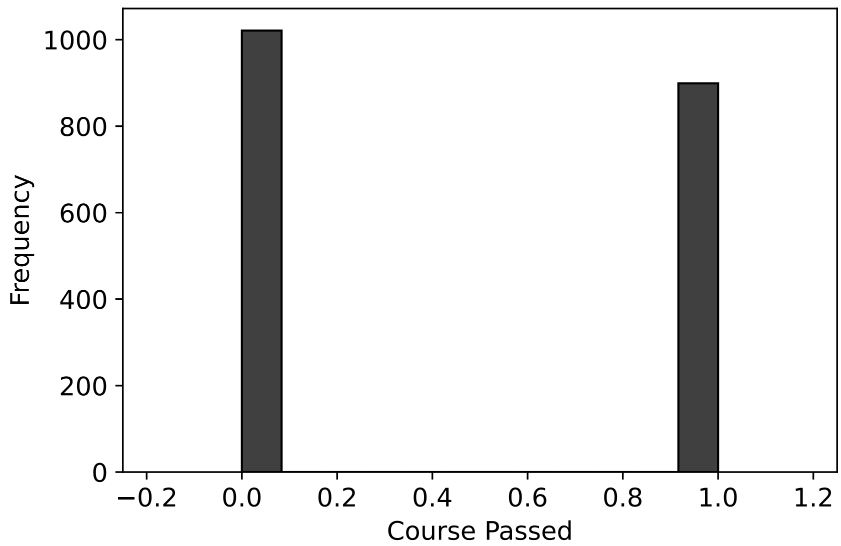


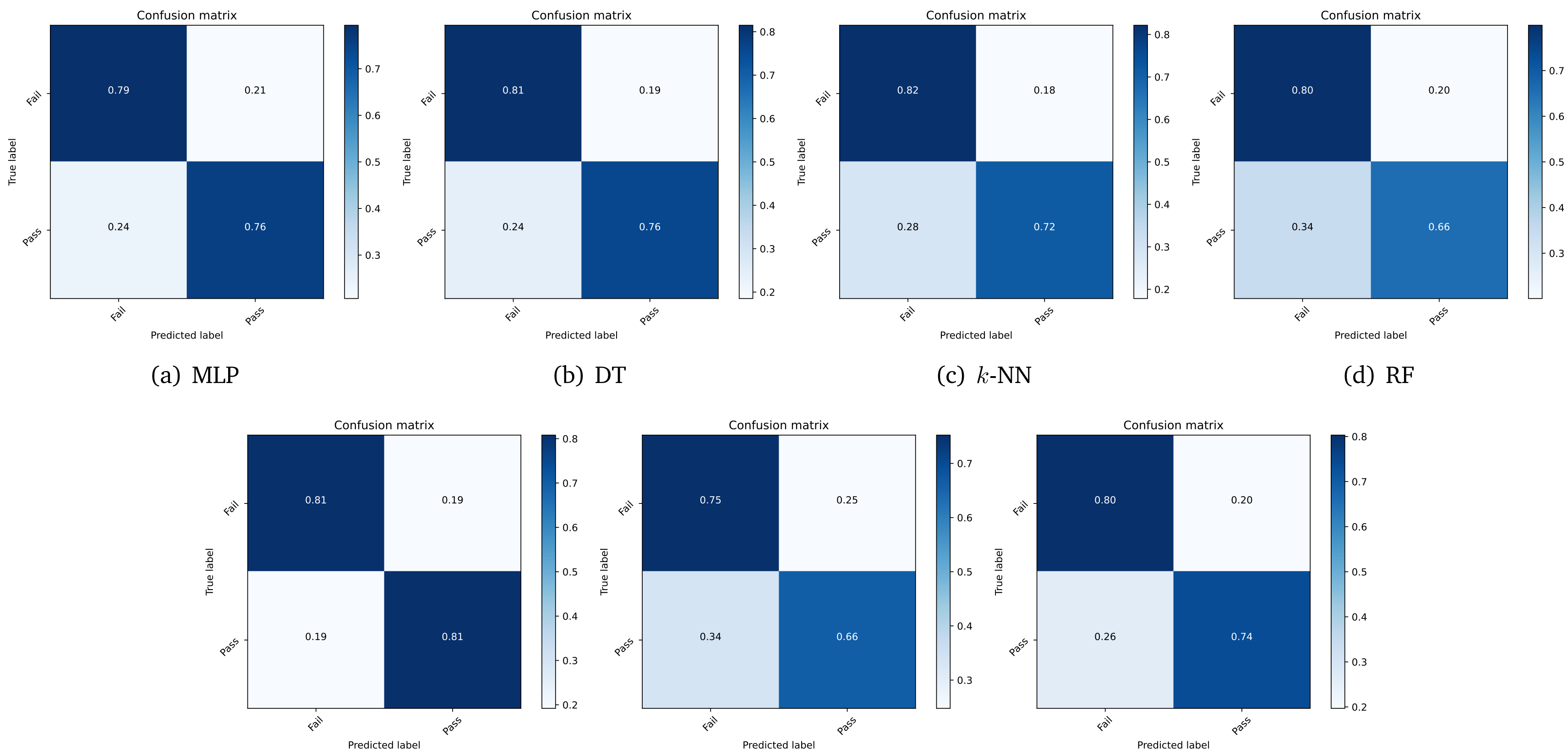
Figure 5.22: Class Distribution for Coursera Introduction to Programming 1 Dataset.

5.6.2 Experimental Settings

Experiments were conducted using the same experimental techniques as those described in Section 4.1.2, including the same programming language, package, data pre-processing, dataset splitting and evaluation measures. The only variation was in the Grid Search-reported hyperparameter values (Table 5.11).

Table 5.11: Hyperparameters for Machine Learning (ML) Models Implemented under Experiment 2F.

Model	**Hyperparameters**
MLP	activation: logistic, learning_rate_init: 0.001, solver: sgd, iter=500
DT	criterion: gini, min_samples_leaf: 100, min_samples_split: 2, splitter: best
k-NN	algorithm: auto, leaf_size: 10, n_neighbors: 100, weights: uniform
RF	criterion: gini, min_samples_leaf: 100, min_samples_split: 50, n_estimators: 10
XGBoost	booster: gblinear, learning_rate: 0.1, max_depth: 50, n_estimators: 2
CATBoost	depth: 4, iterations: 150, l2_leaf_reg: 1, learning_rate: 0.001
SVC	degree: 1, gamma: scale



(a) MLP (b) DT (c) k-NN (d) RF

(e) XGBoost (f) CATBoost (g) SVC

Figure 5.23: Confusion Metrics of Implemented Machine Learning (ML) Models for Classification of Students' Performance using Coursera Introduction to Programming 1 Dataset.

5.6.3 Results

Table 5.12 and Figures 5.23 and 5.24 provide numerical and graphical representations of the results, respectively. Using data from an Introduction to Programming 1 course on Coursera. Table 5.12 displays the quantitative results of extensive testing, conducted on the several ML models that were deployed for the purpose of predicting students' grades in that course. It is clear from the data that the XGBoost model performed best, with a classification accuracy of 0.80. It was found that CATBoost was the least accurate model, scoring a mere 0.71. The overall performance of the models showed that the dataset was more difficult than expected and that the trained models failed to adequately capture the most important features.

Table 5.12: Quantitative Test Results for Machine Learning (ML) Classification of Students' Performance using Coursera Introduction to Programming 1 Dataset.

Models	Accuracy	F1 Score	Precision	Recall	Sensitivity	Specificity
MLP	0.78	0.72	0.88	0.78	0.79	0.76
DT	0.79	0.74	0.76	0.79	0.81	0.76
k-NN	0.77	0.73	0.86	0.77	0.82	0.72
RF	0.76	0.72	0.76	0.77	0.80	0.66
XGBoost	0.80	0.76	0.89	0.81	0.81	0.81
CATBoost	0.71	0.64	0.83	0.70	0.75	0.66
SVC	0.77	0.73	0.85	0.77	0.80	0.74

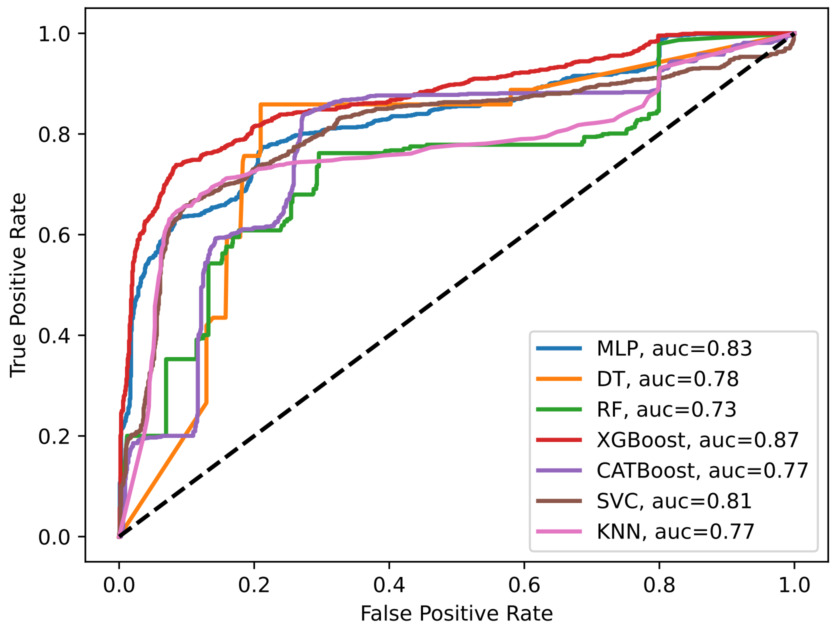


Figure 5.24: ROC Curves of the Implemented Machine Learning (ML) Models for the Classification of Students' Performance using Coursera Introduction to Programming 1 Dataset.

The confusion matrices displayed in Figure 5.23 were also used to evaluate the models' performance in terms of Type I and Type II errors. Type II error was reduced to 19% with XGBoost. However, the Type I error was minimised to 18% with the k-NN model. XGBoost was deemed the best classifier for the whole Introduction to Programming 1 course. Figure 5.24's ROC curves corroborate this; XGBoost's is one of the smoothest curves, with an AUC of 0.87.

5.7 Experiment 2G: Introduction to Programming 2

This section describes the ML research conducted using the Introduction to Programming 2 course dataset from Coursera. Insightful comparisons of the performance of MLP, DT, RF, XGBoost, CATBoost, k-NN, and SVC ML models are provided.

5.7.1 Dataset

Coursera data samples from the "Introduction to Programming 2" course comprised the dataset used for this enquiry. The number of input features and output variables were identical to those of the Coursera dataset. The correlation map between the input features and the target variable is depicted in Figure 5.25. The map reveals that *hits_count* and *Course Grade* are the two most correlated parameters, but *assessment_type_id_7* and *video_duration* are the least correlated. Regarding the target variable distribution of the dataset (see Figure 5.26), an imbalance between the "Pass" and "Fail" classes was identified.

5.7.2 Experimental Settings

In terms of programming language, package, data pre-processing, dataset partitioning and evaluation measures, the same experimental protocols as described in Section 4.1.2 were adhered to during the investigations. In Table 5.13, the only difference was the values of the hyperparameters, generated by the Grid Search.

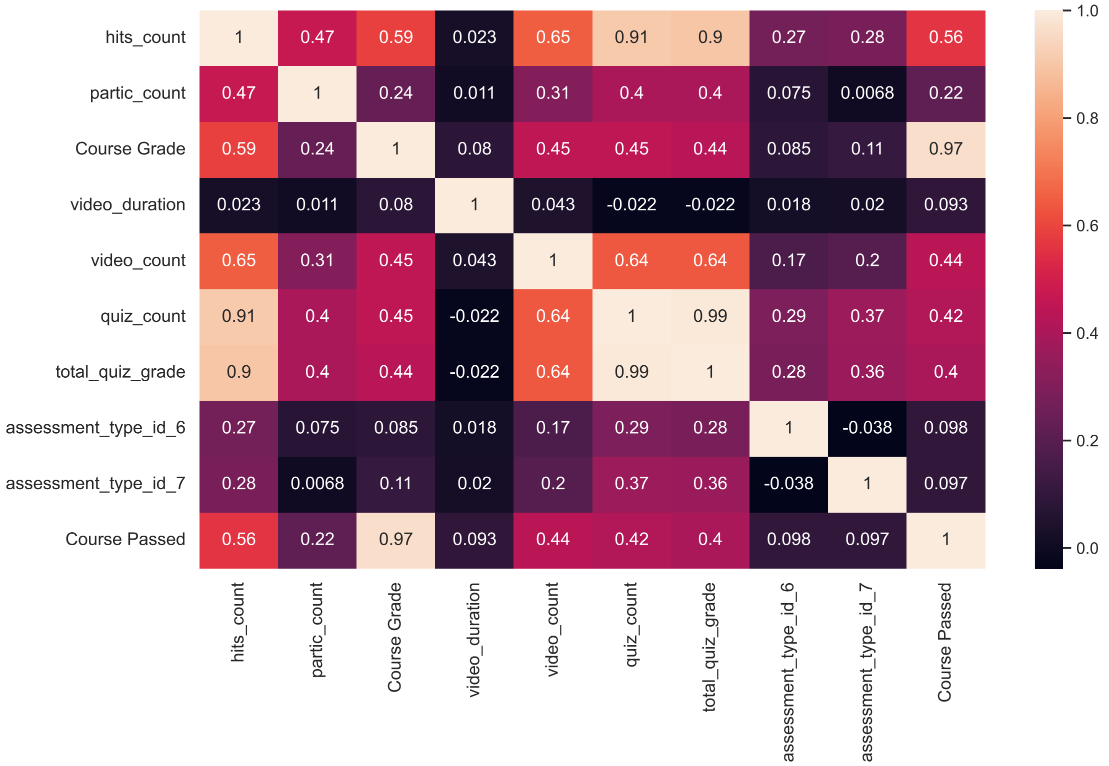


Figure 5.25: Feature Correlation Map for Coursera Introduction to Programming 2 Dataset.

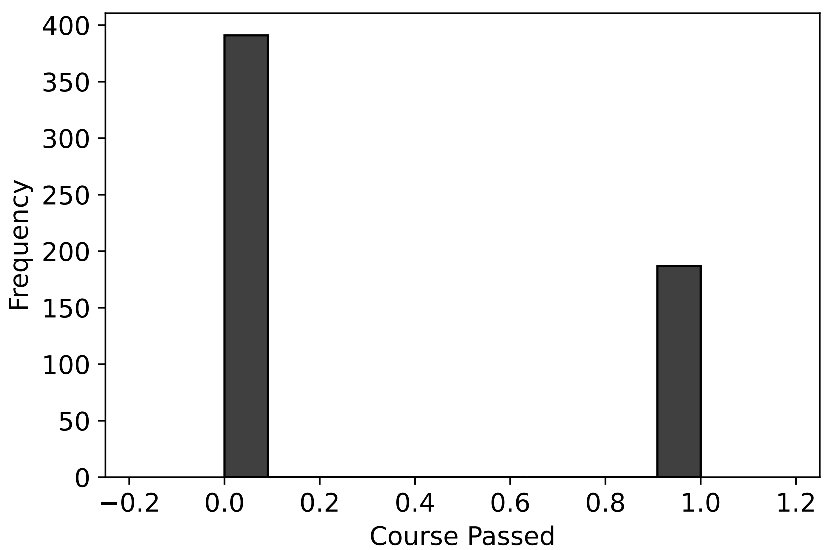


Figure 5.26: Class Distribution for Coursera Introduction to Programming 2 Dataset.

Table 5.13: Hyperparameters for Machine Learning (ML) Models Implemented under Experiment 2G.

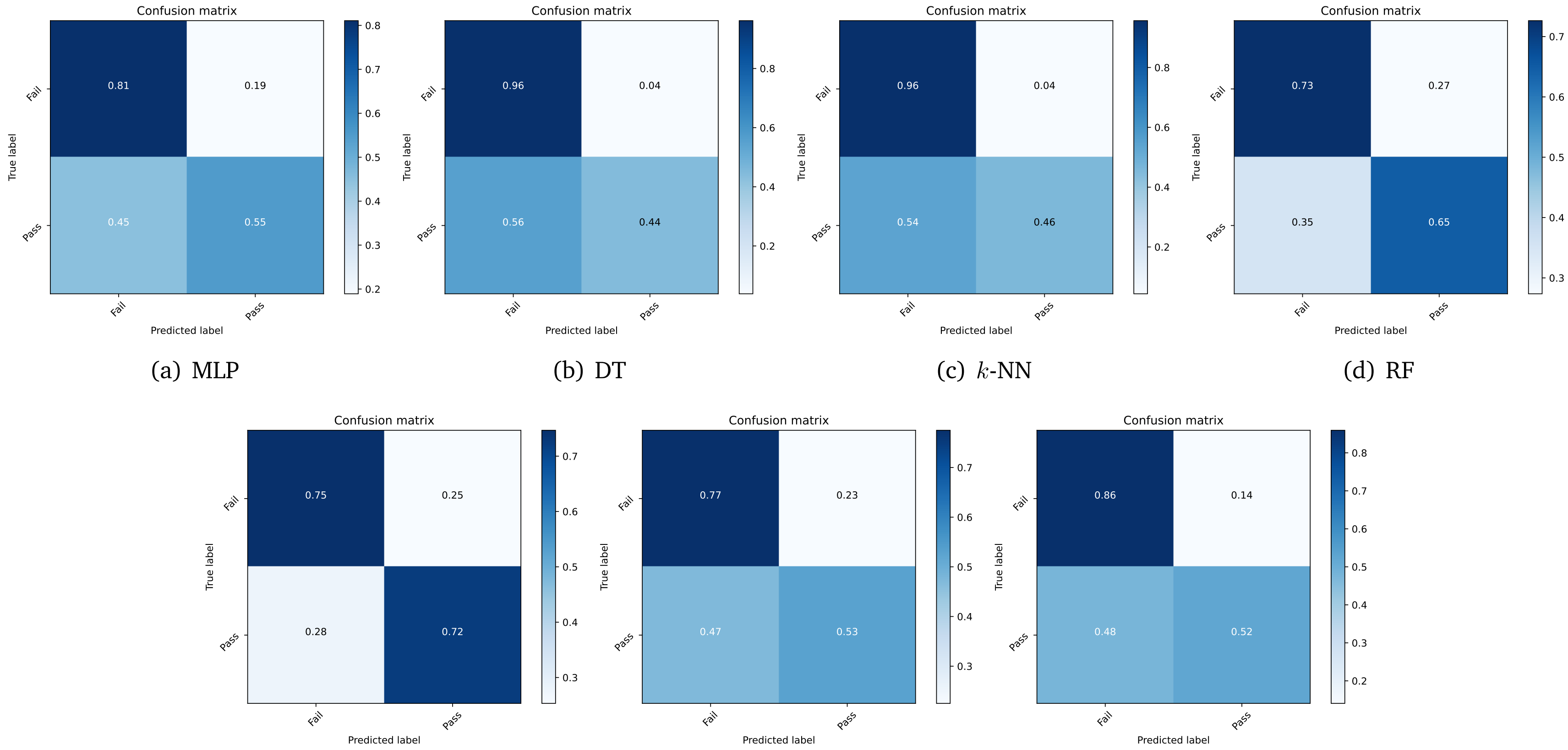
Model	Hyperparameters
MLP	activation: relu, learning_rate_init: 0.001, solver: sgd, iter=500
DT	criterion: gini, min_samples_leaf: 50, min_samples_split: 2, splitter: random
k-NN	algorithm: auto, leaf_size: 10, n_neighbors: 100, weights: uniform
RF	criterion: gini, min_samples_leaf: 100, min_samples_split: 30, n_estimators: 100
XGBoost	booster: gblinear, learning_rate: 0.1, max_depth: 10, n_estimators: 2
CATBoost	depth: 4, iterations: 150, l2_leaf_reg: 0.5, learning_rate: 0.01
SVC	degree: 1, gamma: scale

5.7.3 Results

Table 5.14 and Figures 5.27 and 5.28 provide numerical and graphical representations of the results, respectively. Using data from a Introduction to Programming 2 course on Coursera, Table 5.14 displays the quantitative results of extensive testing conducted on the several ML models that were deployed for the purpose of predicting students' grades in that course. Data from the table shows that the k-NN model performed best in terms of classification accuracy. It was found that CATBoost was the least accurate model, scoring a mere 0.70. The overall performance of the models showed that the dataset was more difficult than expected and that the trained models failed to adequately capture the most important aspects. These difficulties may be related to dataset quality and dataset complexity, which need to be explored based on the explainability of the models.

Table 5.14: Quantitative Test Results for Machine Learning (ML) Classification of Student Performance using Coursera Introduction to Programming 2 Dataset.

Models	Accuracy	F1 Score	Precision	Recall	Sensitivity	Specificity
MLP	0.73	0.62	0.75	0.68	0.81	0.55
DT	0.67	0.56	0.60	0.65	0.96	0.44
k-NN	0.80	0.67	0.77	0.71	0.96	0.46
RF	0.73	0.65	0.74	0.71	0.73	0.65
XGBoost	0.74	0.67	0.75	0.74	0.75	0.72
CATBoost	0.70	0.56	0.63	0.65	0.77	0.53
SVC	0.75	0.64	0.74	0.69	0.86	0.52



(a) MLP (b) DT (c) k-NN (d) RF

(e) XGBoost (f) CATBoost (g) SVC

Figure 5.27: Confusion Metrics of Implemented Machine Learning (ML) Models for Classification of Students' Performance using, Coursera Introduction to Programming 2 Dataset.

The confusion matrices displayed in Figure 5.27 were also used to evaluate the models' performance in terms of Type I and Type II errors. The results show that XGBoost has the lowest Type II error (at 28%). But the k-NN model had the lowest Type I error, at 4%. As far as class distribution goes, XGBoost was said to be the best classifier for the Introduction to Programming 2 course as a whole. This is supported by the ROC curves shown in Figure 5.28, where XGBoost has an AUC that places it among the smoothest curves.

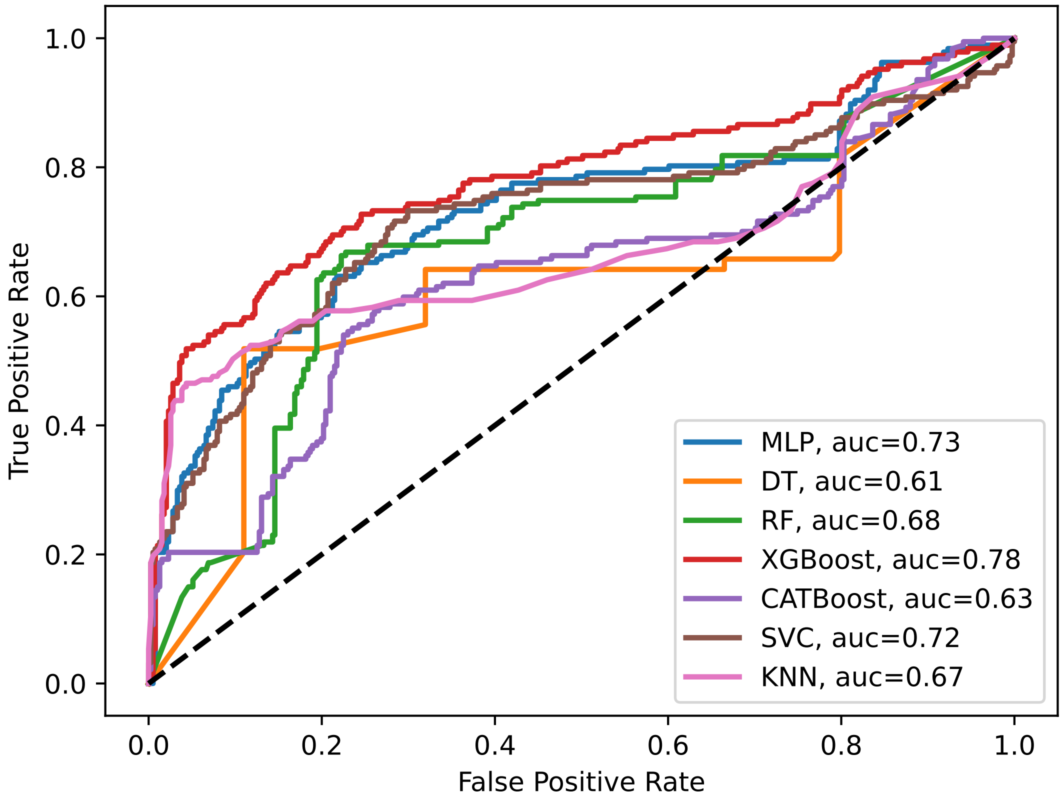


Figure 5.28: ROC Curves of the Implemented Machine Learning (ML) Models for the Classification of Students Performance using Coursera Introduction to Programming 2 Dataset.

5.8 Experiment 2H: Web Development

This section contains information on the ML research, conducted using the Coursera Web Development course dataset. Important insights are presented after comparing the performance of MLP, DT, RF, XGBoost, CATBoost, k-NN and SVC ML models.

5.8.1 Dataset

The dataset utilised for this analysis consists of samples from the Coursera dataset connected to the "Web Development" course. The number of input features and output variables were identical to those of the Coursera dataset. The correlation map between the input features and the target variable is depicted in Figure 5.29. The map reveals that *hits_count* and *Course Grade* are the two most correlated parameters, but *assessment_type_id_6* and *video_duration* are the least correlated. Regarding the target variable distribution of the dataset (see Figure 5.30), an imbalance between the "Pass" and "Fail" classes was identified.

5.8.2 Experimental Settings

In terms of programming language, package, data pre-processing, dataset partitioning and evaluation measures, the identical experimental protocols as described in Section 4.1.2 were adhered to during the research. In Table 5.15, the only difference was the values of the hyperparameters, generated by the Grid search.

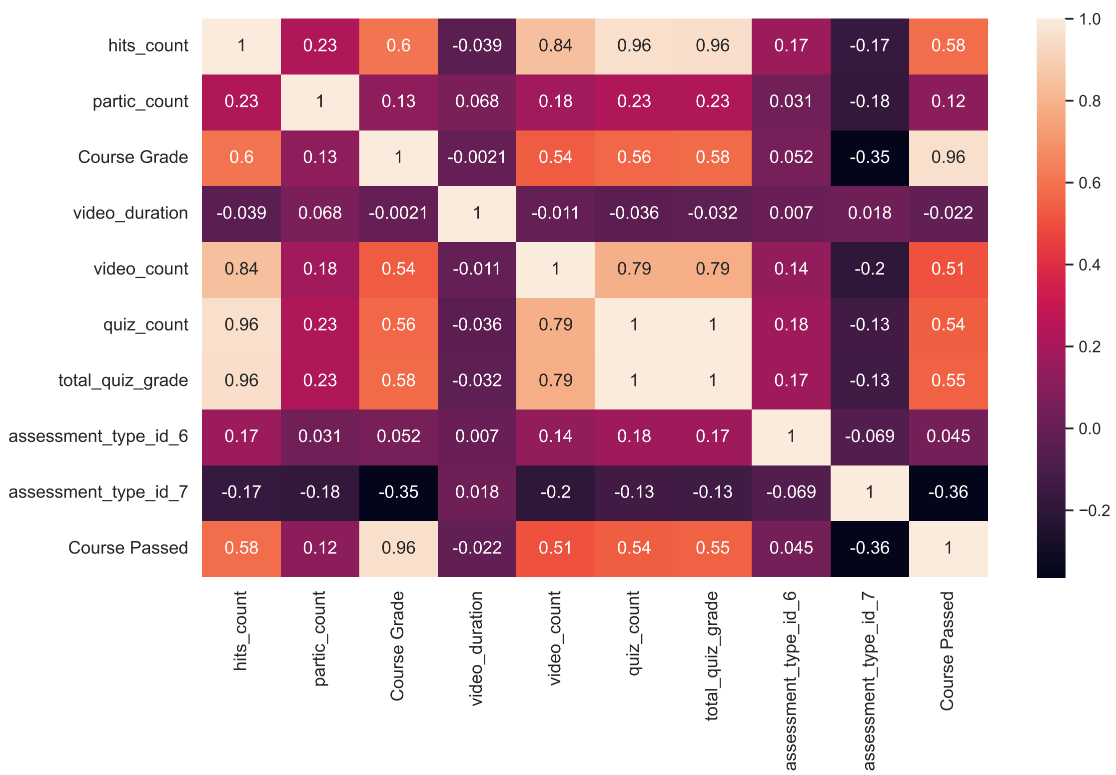


Figure 5.29: Feature Correlation Map for Coursera Web Development Dataset.

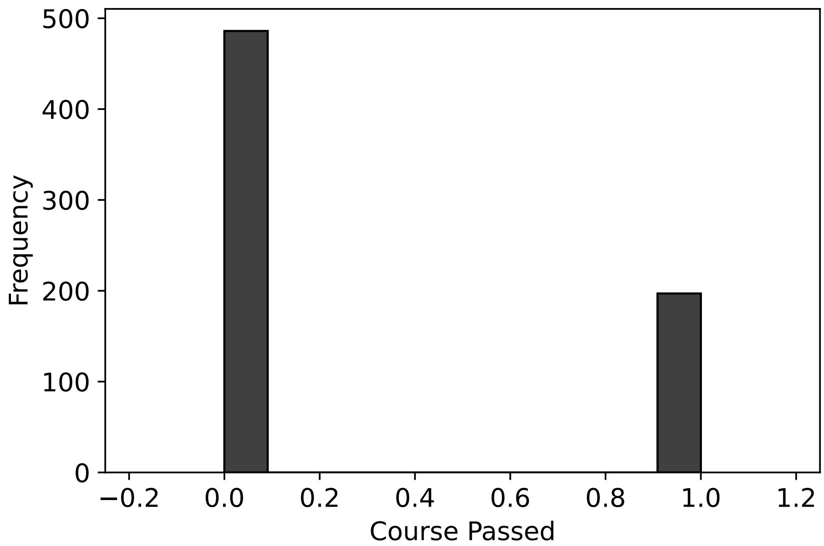


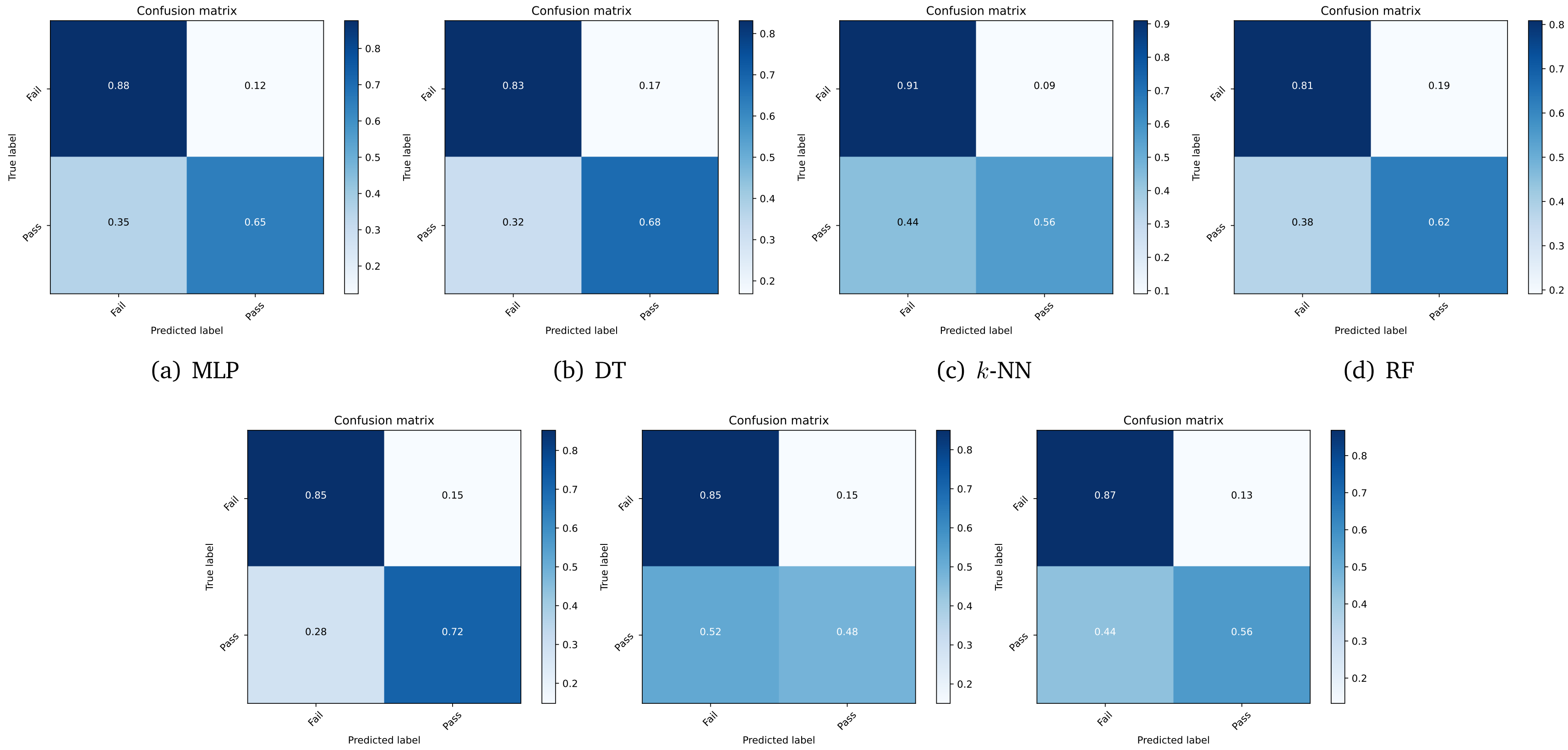
Figure 5.30: Class Distribution for Coursera Web Development Dataset.

Table 5.15: Hyperparameters for Machine Learning (ML) Models Implemented under Experiment 2H.

Model	Hyperparamters
MLP	activation: relu, learning_rate_init: 0.001, solver: sgd, iter=500
DT	criterion: gini, min_samples_leaf: 50, min_samples_split: 2, splitter: random
k-NN	algorithm: auto, leaf_size: 10, n_neighbors: 100, weights: uniform
RF	criterion: entropy, min_samples_leaf: 100, min_samples_split: 2, n_estimators: 10
XGBoost	booster: gblinear, learning_rate: 0.1, max_depth: 50, n_estimators: 10
CATBoost	depth: 4, iterations: 150, l2_leaf_reg: 0.5, learning_rate: 0.001
SVC	degree: 1, gamma: auto

5.8.3 Results

Table 5.16 and Figures 5.31 and 5.32 present the results numerically and graphically, respectively. Table 5.16 provides quantitative test results for the implemented ML models in the context of the performance prediction of students using the Discrete Mathematics course dataset from Coursera. Based on the table, the XGBoost model performed the best, with a classification accuracy of 0.81. With an accuracy of 0.74, CATBoost was deemed the poorest model. Overall, the performance of the models revealed the difficult nature of the dataset and the incapacity of the models to capture significant features during training.



(a) MLP (b) DT (c) k-NN (d) RF

(e) XGBoost (f) CATBoost (g) SVC

Figure 5.31: Confusion Metrics of Implemented Machine Learning (ML) Models for Classification of Students' Performance using Coursera Web Development Dataset.

Table 5.16: Quantitative Test Results for Machine Learning (ML) Classification of Students' Performance using Coursera Web Development Dataset.

Models	Accuracy	F1 Score	Precision	Recall	Sensitivity	Specificity
MLP	0.81	0.73	0.78	0.76	0.88	0.65
DT	0.79	0.69	0.77	0.73	0.83	0.68
k-NN	0.81	0.71	0.78	0.74	0.91	0.56
RF	0.76	0.64	0.77	0.72	0.81	0.62
XGBoost	0.81	0.75	0.84	0.78	0.85	0.72
CATBoost	0.74	0.61	0.64	0.67	0.85	0.48
SVC	0.78	0.67	0.84	0.71	0.87	0.56

Figure 5.31 depicts confusion matrices used to evaluate the performance of models in terms of Type I and Type II errors. It can be seen that XGBoost achieved the lowest Type II error of 28%. In contrast, the k-NN model had the lowest Type I error of 9%. Overall, XGBoost was identified as the best classifier for the Web Development course in terms of class distribution. This is also supported by the ROC curves shown in Figure 5.32, where XGBoost's AUC of 0.81 places it among the best curves.

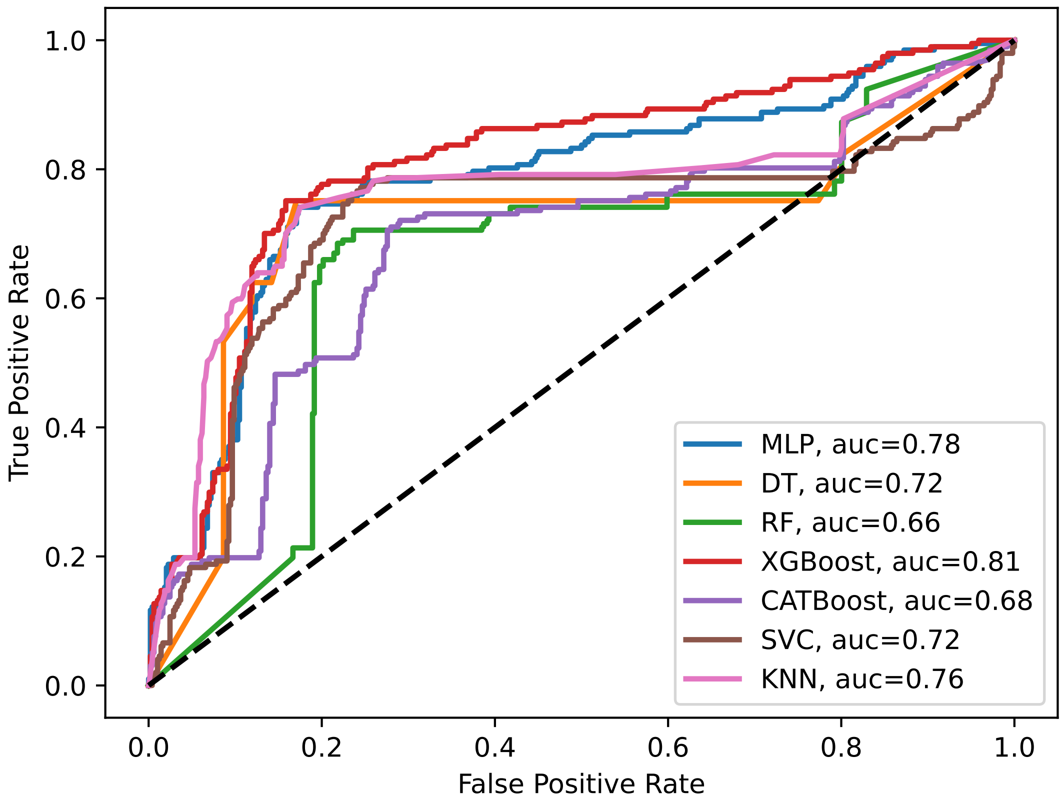


Figure 5.32: ROC Curves of the Implemented Machine Learning (ML) Models for the Classification of Students' Performance using Coursera Web Development Dataset.

5.9 Discussion on Results

The performance of ML models for individual courses within the Coursera dataset was evaluated and reported to be variable across multiple courses. We observed that the performance for the Computational Mathematics course was the best among all courses, while the performance for the How Computers Work course was the worst. Additionally, confusion matrix plots for all the courses indicated that most of the models struggled to predict one of the classes accurately. Although the overall average performance of the models was reported as good, they exhibited a characteristic of performing well for one class while performing poorly for the other, making them impractical for the problem at hand. One potential contributing factor to this behavior is the highly imbalanced nature of the available samples for training, which can hinder the models' ability to effectively learn from the minority class.

In addition to the class imbalance issue, several other factors could contribute to the degraded performance of machine learning models in this context. The complexity of the dataset, including intricate patterns, noise, and outliers, can pose challenges for models to generalize effectively. Poor dataset quality, with missing values or incorrect labels, can further hinder the model's learning process. Inadequate or biased features might fail to capture the necessary information for accurate predictions. Limited dataset samples can result in overfitting and insufficient generalization. Moreover, data distribution shifts, where there are significant differences between the training and deployment data, can also negatively impact model performance. Proper model complexity, hyperparameter tuning, and feature engineering are crucial considerations as well.

The comparison of performance across courses was conducted to gain insights into the behavior of ML models on different academic disciplines and specific course content. Understanding the differences in model performance across courses helps identify which courses might require more personalized interventions and educational strategies. For example, the observed superior performance for the Computational Mathematics course suggests that the content and structure of this course might be particularly well-suited for accurate performance prediction using ML models. On the other hand, the poor performance in the How Computers Work course may highlight potential challenges and limitations in current teaching methodologies or course design.

Regarding class imbalance, it indeed plays a critical role in the effectiveness of ML models. Class imbalance refers to the unequal distribution of samples among different classes in the dataset. In our study, the presence of a significant disparity in the number of students falling into "Pass" and "Fail" categories for individual courses resulted in class imbalance. The severity of class imbalance can impact model performance; when one class is substantially smaller than the other, models may become biased towards the majority class and struggle to predict the minority class accurately.

It can be clearly observed that, single model (i.e., XGBoost) stood out in all cases with the most distributed class performance among all and hence nominated to be the best performing model (see Table 5.17 for comparison). The best Type II error and AUC of 8% and 0.92 were observed for the Computational Mathematics course, respectively.

In summary, the results indicated that while certain ML models showed promising performance for individual courses, the class imbalance issue posed challenges for accurate prediction across all classes. The comparison of model performance across courses shed light on potential course-specific factors influencing students' performance and identified courses where ML models performed exceptionally well or poorly. Moreover, the discussion on class imbalance and other factors contributing to model performance helps in understanding the limitations and potential areas for improvement in our study.

Table 5.17: Performance Comparison of Best Performing Machine Learning (ML) Models for Each Course under Experiment 2.

Course Name	Best Model	Accuracy	Type II Error	AUC
Algorithms and Data Structures	XGBoost	0.80	40%	0.78
Computational Mathematics	XGBoost	0.87	8%	0.92
Discrete Mathematics	XGBoost	0.76	19%	0.82
Fundamentals of Computer Science	XGBoost	0.79	18%	0.82
How Computers Work	XGBoost	0.70	32%	0.68
Introduction to Programming 1	XGBoost	0.80	19%	0.87
Introduction to Programming 2	XGBoost	0.74	28%	0.78
Web Development	XGBoost	0.81	28%	0.81

While the literature review in Section 2.5 discussed aspects of class imbalance well, we acknowledge that we could have provided more specific details on the degree of imbalance in our dataset and its implications on model performance. A comprehensive analysis of class imbalance severity and its effect on ML models could have offered deeper insights into the challenges faced during the prediction task.

In conclusion,

5.10 Summary

This chapter provides the details about experiment 2, where ML models were used, under this research and the performance of the students' was classified under the Coursera datasets. The main aim of this experiment is to investigate the impact of each course on the ML performance in comparison to the combined all courses, comprising of the eight courses performance of ML models was compared course wise and also collectively at the end of experiment 2. From the analysis, it has been reported that ML models performed of computational Mathematics course and XGBOOST model emerged as the best among all the courses to provide most efficient class-wise performance for students' performance classification.

Further, using the data from an Introduction to Programming 1, Introduction to Programming 2 and Discrete Mathematics, the XGBOOST classifier model performed the best in all models. However, the class imbalance problem has been highlighted as the one of the prominent causes of the degraded performance of the ML models.

Chapter 6

Class Imbalance Problem

This chapter addresses the class imbalance problem in the Coursera dataset by using multiple sampling and generative models for improved ML classification performance. First, the impact of different SMOTE approaches has been investigated towards addressing the class imbalance problem. Later, the generative models are used to create synthetic data and the impact of synthetic data on the classification performance has been studied. For the comparative analysis, ML classification models including MLP, DT, RF, XGBoost, CATBoost, k-NN and SVC have been implemented.

6.1 Experiment 3A – Class Balancing using SMOTE Techniques

In this experiment, multiple SMOTE based class balancing techniques have been compared for ML student performance classification. For each SMOTE approach, the information about the balanced dataset, GridSearch hyperparameters and classification results are presented. At the end of this experiment, a comparison of top performing models for each SMOTE technique are compared to select the best model and to report the important insights.

6.1.1 Boderline SMOTE

Dataset and Hyperparameters

The Coursera dataset was balanced using the Borderline SMOTE technique for this investigation with having same number input features and same target variable as in the original dataset. The correlation map between input features and target variable is presented in Figure 6.1. It can be observed that *hits_count* and *Course Grade* are the two most correlated features while *assessment_type_id_7* and *video_duration* are the least correlated features, similar trend as observed in the original dataset.

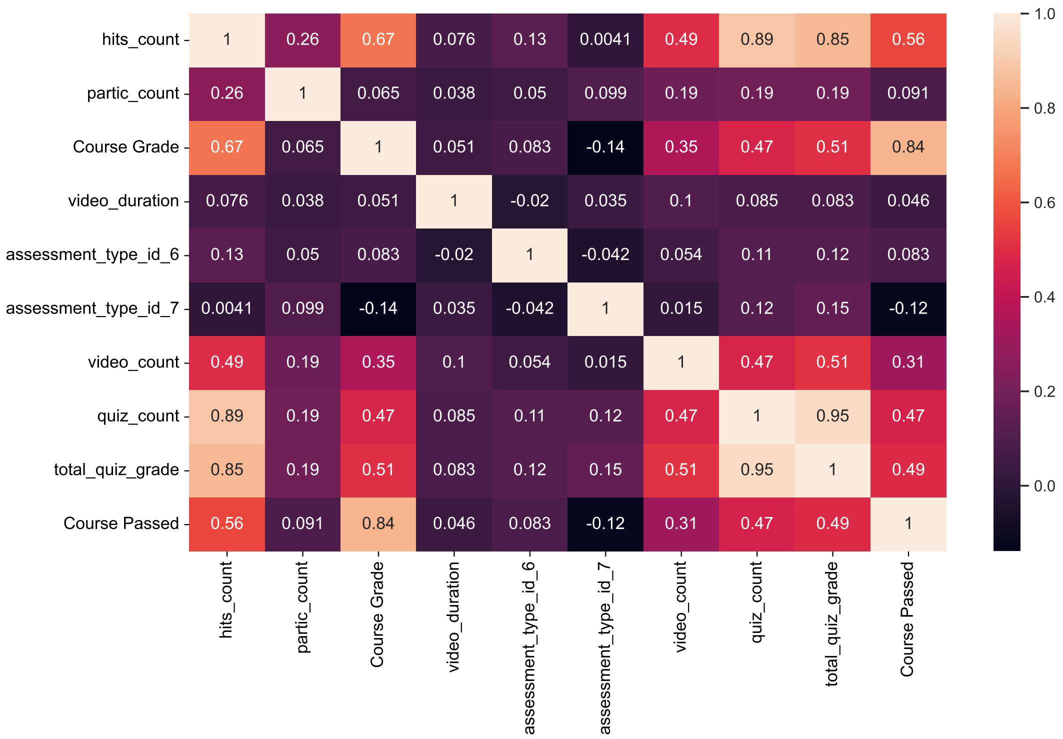


Figure 6.1: Feature Correlation Map for Coursera Dataset Balanced using Borderline SMOTE.

Experiments were conducted using the same experimental techniques as those described in Section 4.1.2, including the same programming language, package, data pre-processing, dataset splitting and evaluation measures. The only variation was in the GridSearch-reported hyperparameter values (Table 6.1).

Table 6.1: Hyperparameters for Machine Learning (ML) Models Implemented under Experiment 3A for Borderline SMOTE.

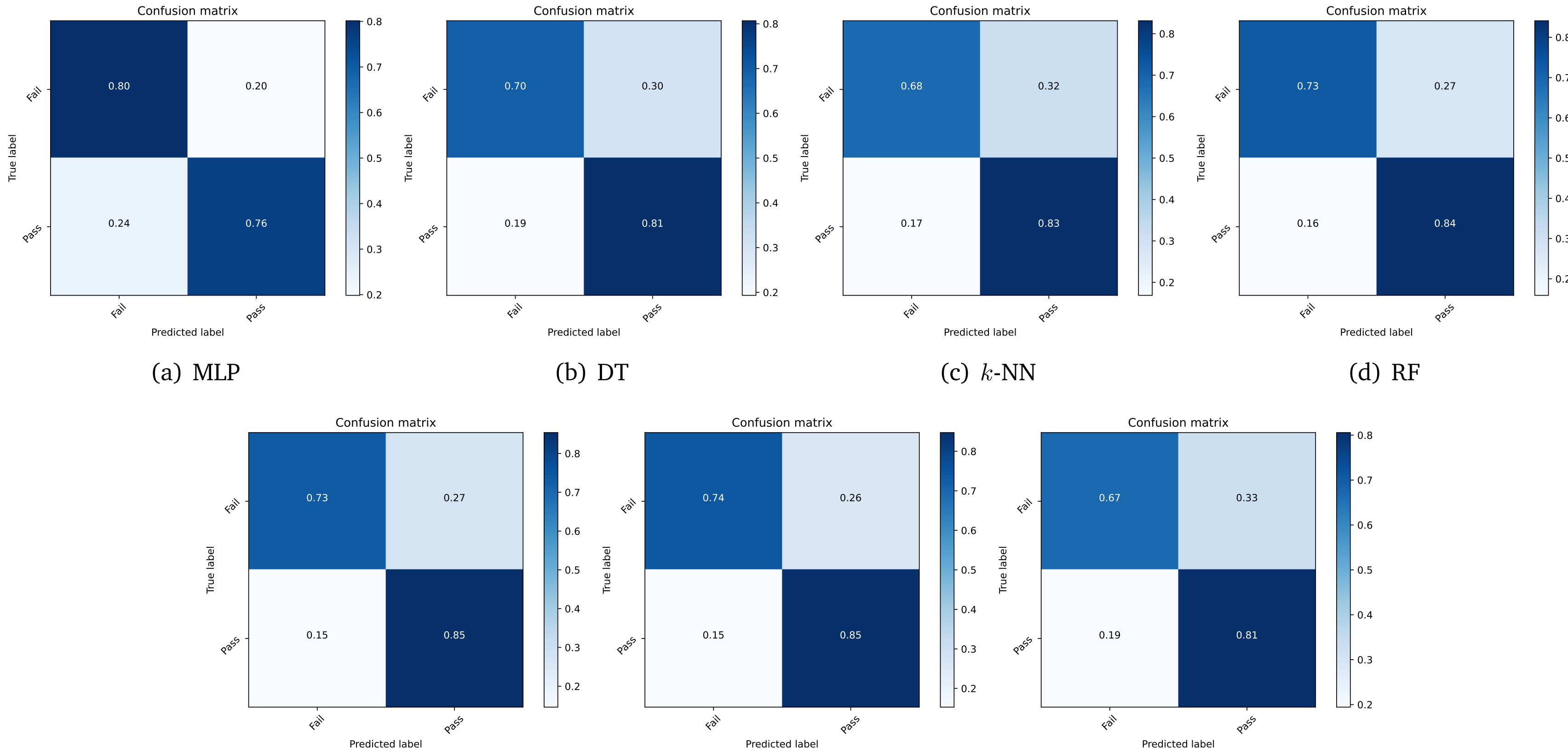
Model	Hyper parameters
MLP	activation: logistic, learning_rate_init: 0.001, solver: sgd, iter=500
DT	criterion: entropy, min_samples_leaf: 100, min_samples_split: 2, splitter: random
k-NN	algorithm: auto, leaf_size: 10, n_neighbors: 100, weights: uniform
RF	criterion: gini, min_samples_leaf: 100, min_samples_split: 10, n_estimators: 100
XGBoost	booster: gbtree, learning_rate: 0.1, max_depth: 3, n_estimators: 10
CATBoost	depth: 4, iterations: 150, l2_leaf_reg: 1, learning_rate: 0.001
SVC	degree: 1, gamma: scale

Classification Results

Table 6.2, Figures 6.2 and 6.3 provide numerical and graphical representations of the results, respectively. Table 6.2 displays the quantitative results of detailed experiments, conducted using the several ML models for the Bordeline SMOTE balanced dataset. It can be observed that CATBoost model was able to achieve the best accuracy of 0.80 while SVC was the least accurate with accuracy value of 0.74. Data balancing using the Borderline SMOTE did not result into much improved performance, which can be attributed to inability of Borderline SMOTE in efficiently sampling the existing data.

Table 6.2: Quantitative Test Results for Machine Learning (ML) Classification of Students, Performance using Coursera Dataset Balanced and Borderline SMOTE.

Models	Accuracy	F1 Score	Precision	Recall	Sensitivity	Specificity
MLP	0.78	0.78	0.78	0.78	0.80	0.76
DT	0.79	0.78	0.80	0.79	0.70	0.81
k-NN	0.76	0.75	0.77	0.76	0.68	0.83
RF	0.78	0.78	0.79	0.78	0.73	0.84
XGBoost	0.79	0.79	0.80	0.79	0.73	0.85
CATBoost	0.80	0.79	0.80	0.80	0.74	0.85
SVC	0.74	0.74	0.75	0.74	0.67	0.81



(a) MLP (b) DT (c) k-NN (d) RF

(e) XGBoost (f) CATBoost (g) SVC

Figure 6.2: Confusion Metrices of Implemented Machine Learning (ML) Models for Classification of Students, Performance using Coursera Dataset Balanced using Borderline SMOTE.

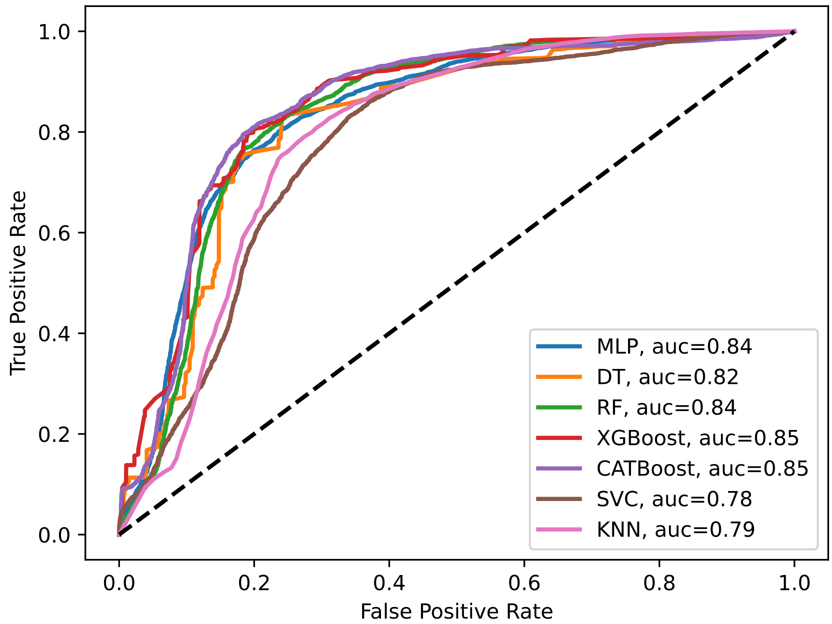


Figure 6.3: ROC Curves of the Implemented Machine Learning (ML) Models for the Classification of Students, Performance using Coursera Dataset Balanced using Borderline SMOTE.

The confusion matrices displayed in Figure 6.3 were also used to evaluate the models' performance in terms of Type I and Type II errors. XGBoost and CATBoost models were able to achieve best Type II error (i.e., 15%), while MLP was able to achieve the best Type I error (i.e., 20%). Overall, the CATBoost model was able to achieve the most distributed performance among classes. Figure 6.3 shows the ROC curve for all the implemented models and demonstrate the superiority of CATBoost and XGBoost models with AUC of 0.85.

6.1.2 SMOTE

Dataset and Hyperparameters

The Coursera dataset was balanced, using the SMOTE technique for this investigation with same number input features and same target variable as in the original dataset. The correlation map between input features and target variable is presented in Figure 6.4. It can be observed that *hits_count* and *Course Grade* are the two most correlated features while *assessment_type_id_6* and *partic_count* are the least correlated features, similar trend as observed in the original dataset.

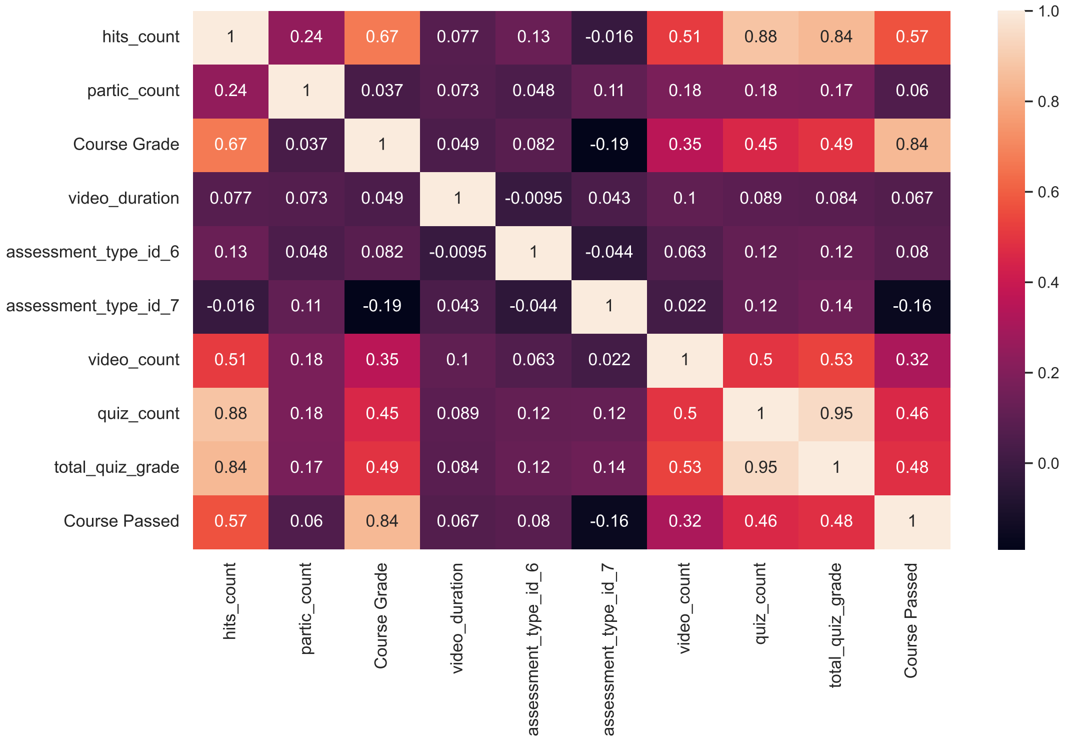
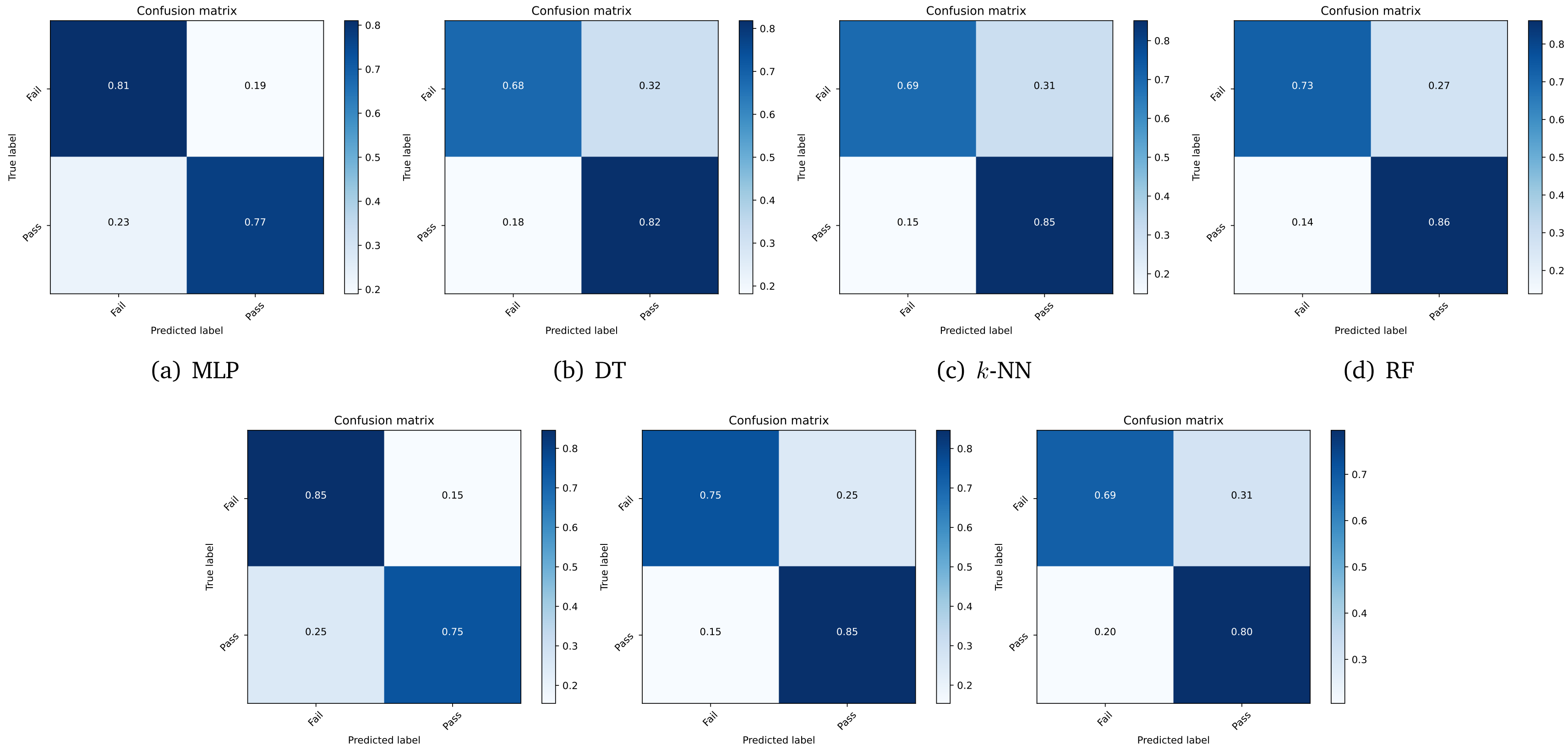


Figure 6.4: Feature Correlation Map for Coursera Dataset Balanced using SMOTE.

Experiments were conducted using the same experimental techniques as those described in Section 4.1.2, including the same programming language, package, data pre-processing, dataset splitting, and evaluation measures. The only variation was in the GridSearch-reported hyper parameter values (Table 6.3).

Table 6.3: Hyperparameters for Machine Learning (ML) Models Implemented under Experiment 3A for SMOTE.

Model	**Hyperparameters**
MLP	activation: logistic, learning_rate_init: 0.001, solver: sgd, iter=500
DT	criterion: entropy, min_samples_leaf: 100, min_samples_split: 2, splitter: random
k-NN	algorithm: auto, leaf_size: 10, n_neighbors: 50, weights: uniform
RF	criterion: gini, min_samples_leaf: 50, min_samples_split: 50, n_estimators: 30
XGBoost	booster: gblinear, learning_rate: 0.01, max_depth: 10, n_estimators: 100
CATBoost	depth: 4, iterations: 150, l2_leaf_reg: 0.5, learning_rate: 0.001
SVC	degree: 1, gamma: scale



(a) MLP (b) DT (c) k-NN (d) RF

(e) XGBoost (f) CATBoost (g) SVC

Figure 6.5: Confusion Metrices of Implemented Machine Learning (ML) Models for Classification of Students, Performance using Coursera Dataset Balanced using SMOTE.

Classification Results

Table 6.4 and Figures 6.5 and 6.6 provide numerical and graphical representations of the results respectively. Table 6.4 displays the quantitative results of detailed experiments conducted using the several ML models for the SMOTE balanced dataset. It can be observed that the CATBoost model was able to achieve the best accuracy of 0.80 while SVC was the least accurate with an accuracy value of 0.74. Data balancing using the SMOTE did not result into much improved performance, which can be attributed to inability of SMOTE in efficiently sampling the existing data.

Table 6.4: Quantitative Test Results for Machine Learning (ML) Classification of Students, Performance using Coursera Dataset Balanced using SMOTE.

Models	Accuracy	F1 Score	Precision	Recall	Sensitivity	Specificity
MLP	0.79	0.79	0.79	0.79	0.81	0.77
DT	0.76	0.76	0.77	0.76	0.68	0.82
k-NN	0.77	0.77	0.78	0.77	0.69	0.85
RF	0.79	0.78	0.79	0.79	0.73	0.86
XGBoost	0.80	0.80	0.80	0.80	0.85	0.75
CATBoost	0.80	0.80	0.80	0.80	0.75	0.85
SVC	0.74	0.74	0.75	0.74	0.89	0.80

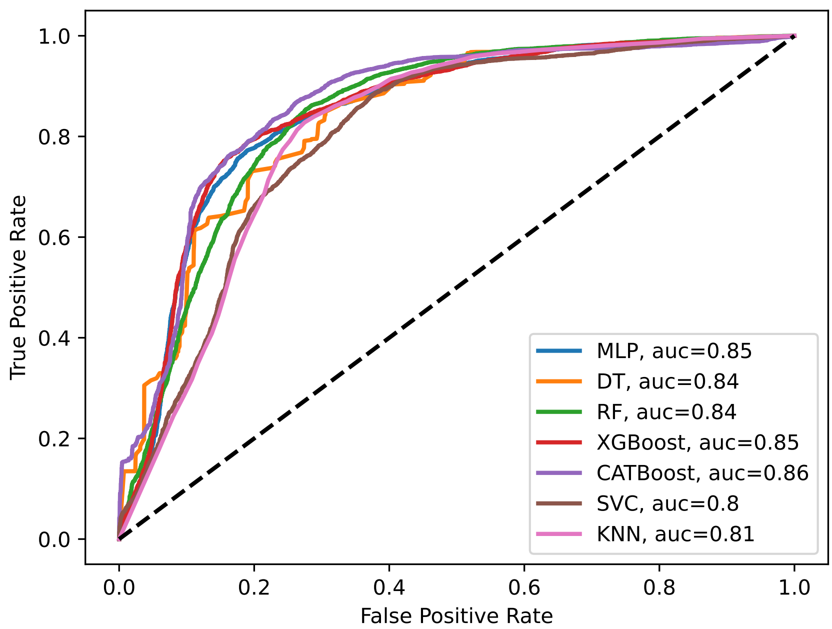


Figure 6.6: ROC Curves of the Implemented Machine Learning (ML) Models for the Classification of Students' Performance using Coursera Dataset Balanced using SMOTE.

The confusion matrices displayed in Figure 6.6 were also used to evaluate the models' performance in terms of Type I and Type II errors. RF model was able to achieve best Type II error (i.e., 14%), while XGBoost was able to achieve the best Type I error (i.e., 15%). Overall, the CATBoost model was able to achieve the most distributed performance among classes. Figure 6.3 shows the ROC curve for all the implemented models and demonstrate the superiority of CATBoost model with AUC of 0.86.

6.1.3 SMOTE NN

Dataset and Hyperparameters

The Coursera dataset was balanced using the SMOTE NN technique for this investigation with same number input features and same target variable as in the original dataset. The correlation map between input features and target variable is presented in Figure 6.7. It can be observed that *hits_count* and *Course Grade* are the two most correlated features while *assessment_type_id_6* and *video_duration* are the least correlated features, similar trend as observed in the original dataset.

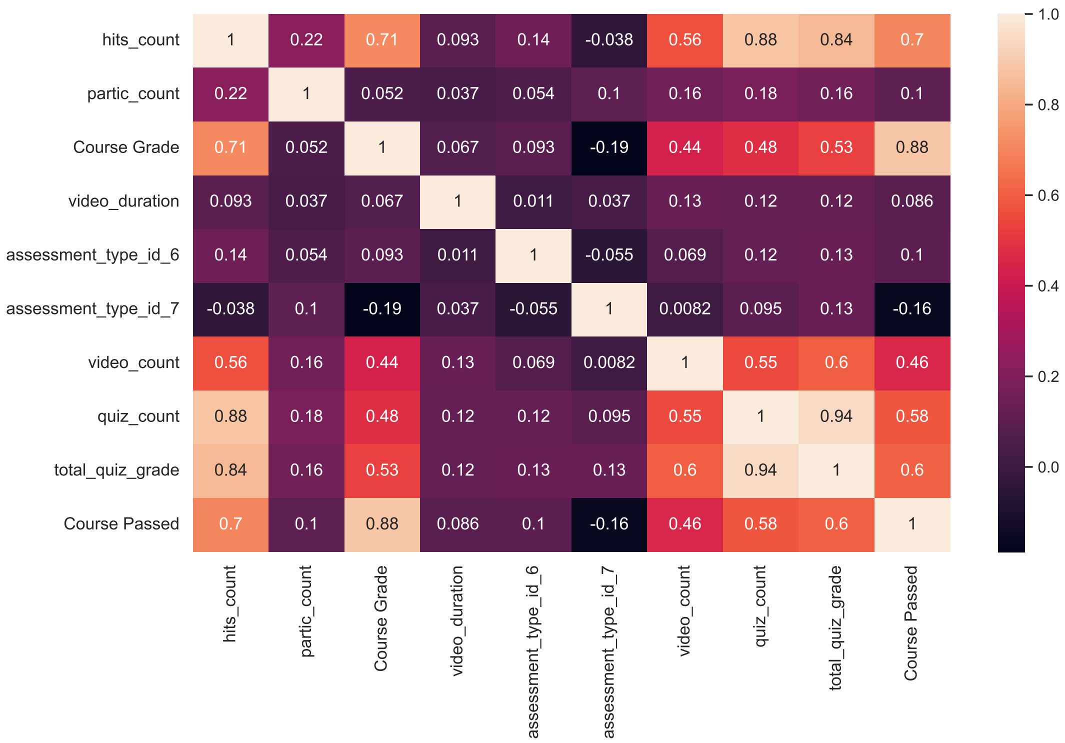
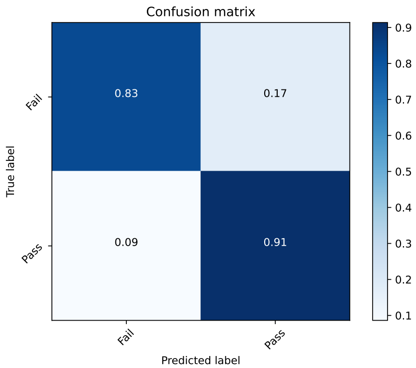
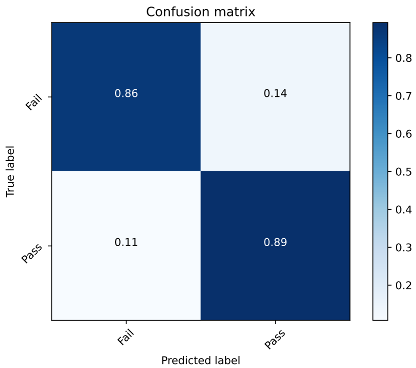


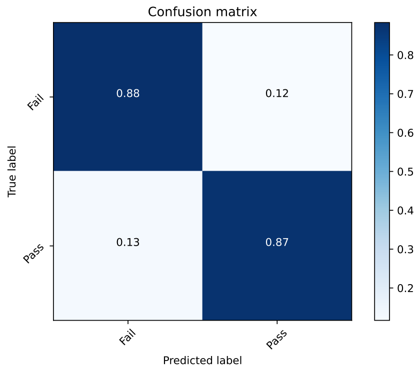
Figure 6.7: Feature Correlation Map for Coursera Dataset Balanced using SMOTE NN.



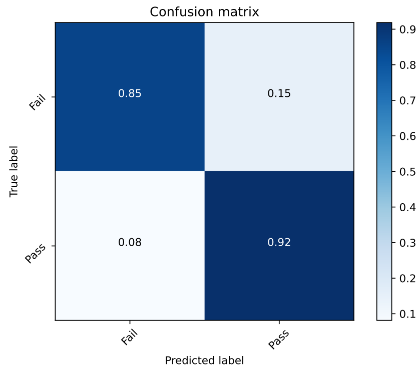
(a) MLP



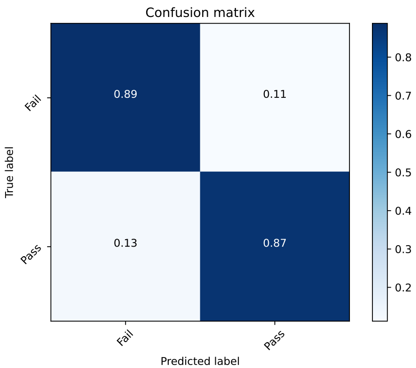
(b) DT



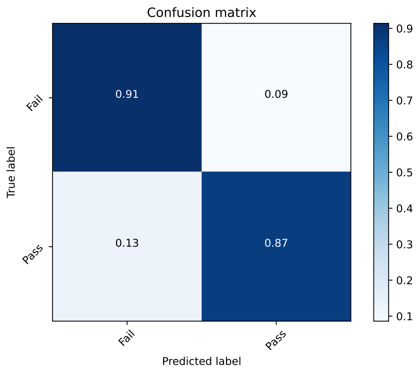
(c) k-NN



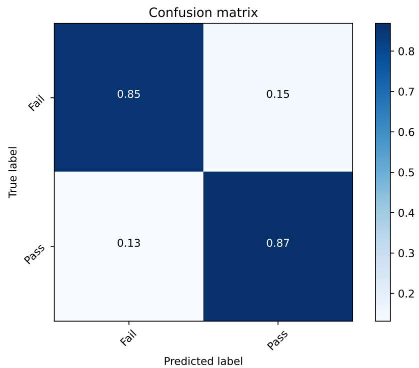
(d) RF



(e) XGBoost



(f) CATBoost



(g) SVC

Figure 6.8: Confusion Metrices of Implemented Machine Learning (ML) Models for Classification of Students' Performance using Coursera Dataset Balanced using SMOTE NN.

Experiments were conducted using the same experimental techniques as those described in Section 4.1.2, including the same programming language, package, data pre-processing, dataset splitting, and evaluation measures. The only variation was in the GridSearch reported hyperparameter values (Table 6.5).

Table 6.5: Hyperparameters for Machine Learning (ML) Models Implemented under Experiment 3A for SMOTE NN.

Model	Hyperparameters
MLP	activation: relu, learning_rate_init: 0.001, solver: adam, iter=500
DT	criterion: entropy, min_samples_leaf: 100, min_samples_split: 2, splitter: best
k-NN	algorithm: auto, leaf_size: 10, n_neighbors: 100, weights: distance
RF	criterion: gini, min_samples_leaf: 10, min_samples_split: 50, n_estimators: 30
XGBoost	booster: gbtree, learning_rate: 0.01, max_depth: 3, n_estimators: 100
CATBoost	depth: 4, iterations: 150, l2_leaf_reg: 1, learning_rate: 0.001
SVC	degree: 1, gamma: scale

Classification Results

Table 6.6 and Figures 6.8 and 6.9 provide numerical and graphical representations of the results, respectively. Table 6.6 displays the quantitative results of detailed experiments conducted using the several ML models for the SMOTE NN balanced dataset. It can be observed that the CATBoost model was able to achieve the best accuracy of 0.89 while SVC was the least accurate with an accuracy value of 0.86. Data balancing using the SMOTE NN was reported to enhance the classification performance, which may be attributed to ability of the NN in efficiently sampling the existing data samples.

Table 6.6: Quantitative Test Results for Machine Learning (ML) Classification of Students' Performance using Coursera Dataset Balanced using SMOTE NN.

Models	Accuracy	F1 Score	Precision	Recall	Sensitivity	Specificity
MLP	0.88	0.87	0.88	0.88	0.83	0.91
DT	0.88	0.88	0.88	0.88	0.86	0.89
k-NN	0.88	0.88	0.88	0.88	0.88	0.87
RF	0.89	0.89	0.89	0.89	0.85	0.92
XGBoost	0.88	0.88	0.88	0.88	0.89	0.87
CATBoost	0.89	0.89	0.89	0.89	0.91	0.87
SVC	0.86	0.86	0.87	0.86	0.85	0.87

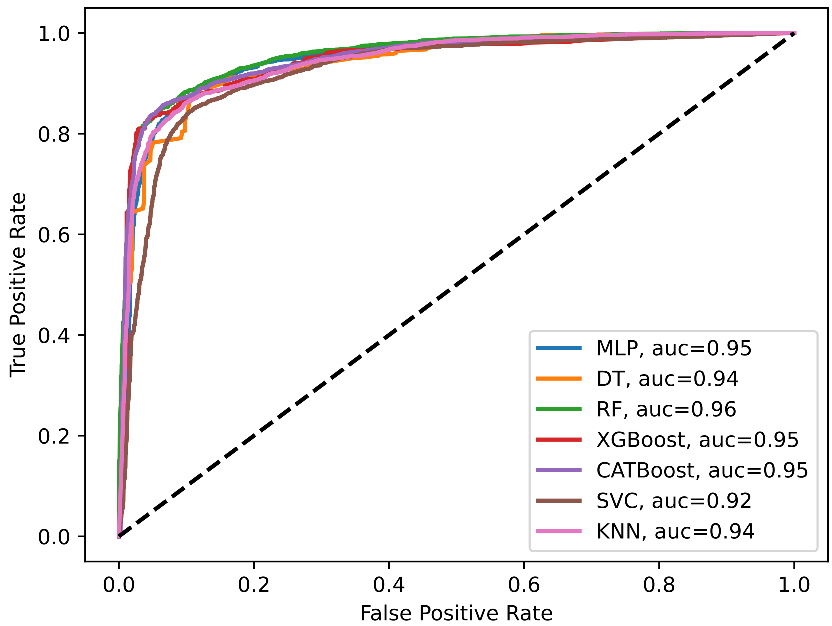


Figure 6.9: ROC Curves of the Implemented Machine Learning (ML) Models for the Classification of Students' Performance using Coursera Dataset Balanced using SMOTE NN.

The confusion matrices displayed in Figure 6.6 were also used to evaluate the models' performance in terms of Type I and Type II errors. RF model was able to achieve best Type II error (i.e., 9%), while CATBoost was able to achieve the best Type I error (i.e., 9%). Overall, the CATBoost model was able to achieve the most distributed performance among classes. Figure 6.3 shows the ROC curve for all the implemented models and demonstrate the superiority of the CATBoost model with AUC of 0.95.

6.1.4 SMOTE Tomek

Dataset and Hyperparameters

The Coursera dataset was balanced, using the SMOTE Tomek technique for this investigation with same number input features and same target variable as in the original dataset. The correlation map between input features and target variable is presented in Figure 6.10. It can be observed that *hits_count* and *Course Grade* are the two most correlated features while *assessment_type_id_6* and *partic_count* are the least correlated features, similar trend as observed in the original dataset.

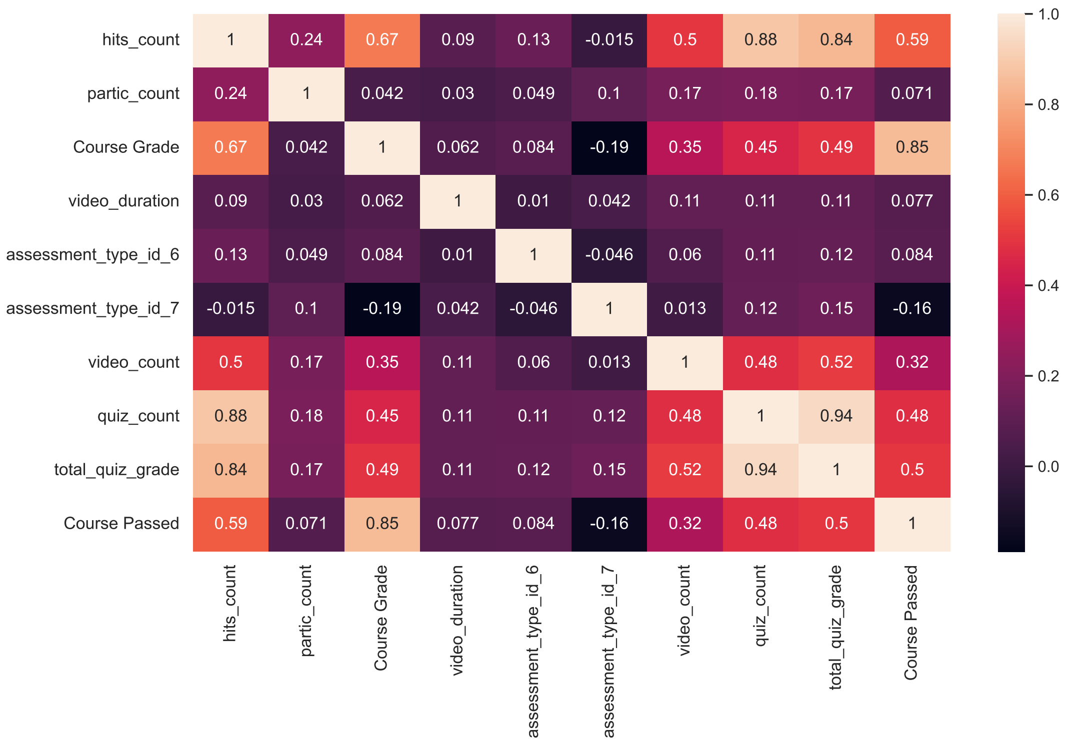
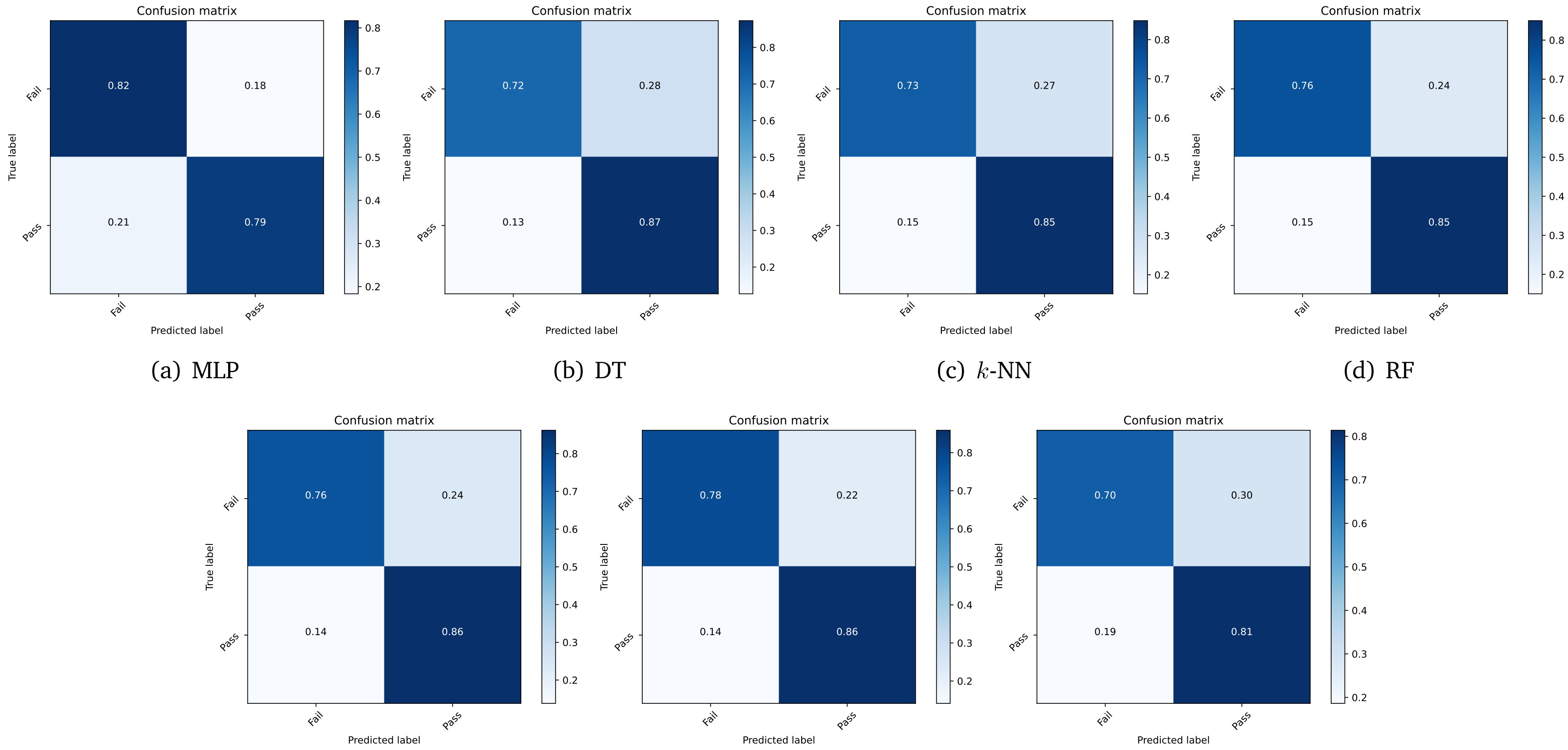


Figure 6.10: Feature Correlation Map for Coursera Dataset Balanced using SMOTE Tomek.

Experiments were conducted using the same experimental techniques as those described in Section 4.1.2, including the same programming language, package, data pre-processing, dataset splitting and evaluation measures. The only variation was in the GridSearch reported hyperparameters values (Table 6.7).

Table 6.7: Hyperparameters for Machine Learning (ML) Models Implemented under Experiment 3A for SMOTE Tomek.

Model	Hyper parameters
MLP	activation: logistic, learning_rate_init: 0.001, solver: sgd, iter=500
DT	criterion: gini, min_samples_leaf: 100, min_samples_split: 2, splitter: best
k-NN	algorithm: auto, leaf_size: 10, n_neighbors: 100, weights: uniform
RF	criterion: entropy, min_samples_leaf: 100, min_samples_split: 100, n_estimators: 10
XGBoost	booster: gbtree, learning_rate: 0.1, max_depth: 3, n_estimators: 10
CATBoost	depth: 4, iterations: 150, l2_leaf_reg: 0.5, learning_rate: 0.001
SVC	degree: 1, gamma: auto



(a) MLP (b) DT (c) k-NN (d) RF

(e) XGBoost (f) CATBoost (g) SVC

Figure 6.11: Confusion Metrices of Implemented Machine Learning (ML) Models for Classification of Students' Performance using Coursera Dataset Balanced using SMOTE Tomek.

Classification Results

Table 6.8 and Figures 6.11 and 6.12 provide numerical and graphical representations of the results, respectively. Table 6.8 displays the quantitative results of detailed experiments, conducted using the several ML models for the SMOTE Tomek balanced dataset. It can be observed that CATBoost model was able to achieve the best accuracy of 0.82 while SVC was the least accurate with an accuracy value of 0.76. Data balancing using the SMOTE Tomek slightly improved the performance in comparison to the original dataset.

Table 6.8: Quantitative Test Results for Machine Learning (ML) Classification of Students' Performance using Coursera Dataset Balanced using SMOTE Tomek.

Models	Accuracy	F1 Score	Precision	Recall	Sensitivity	Specificity
MLP	0.80	0.80	0.80	0.80	0.82	0.79
DT	0.79	0.79	0.81	0.80	0.72	0.87
k-NN	0.79	0.79	0.80	0.79	0.73	0.85
RF	0.81	0.81	0.82	0.81	0.76	0.85
XGBoost	0.81	0.81	0.82	0.81	0.76	0.86
CATBoost	0.82	0.82	0.82	0.82	0.78	0.86
SVC	0.76	0.76	0.77	0.76	0.70	0.81

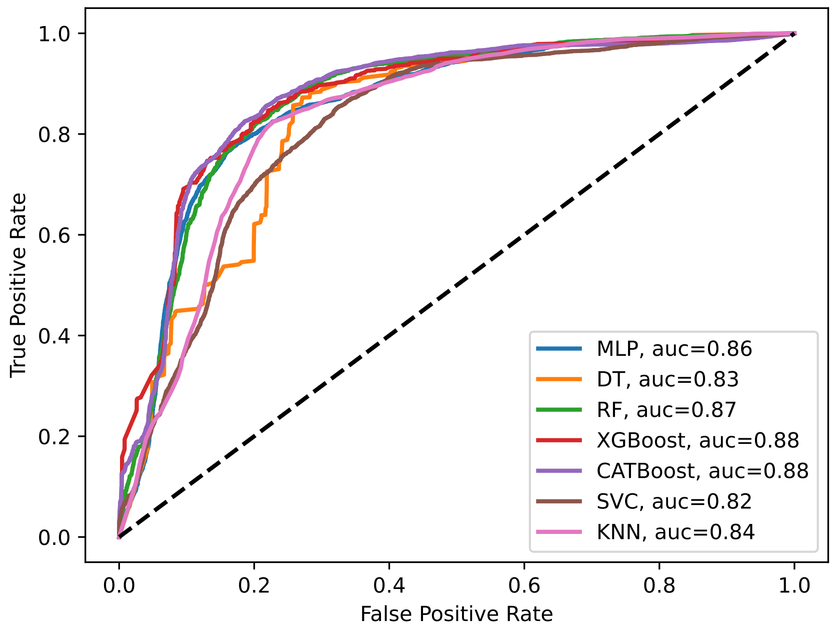


Figure 6.12: ROC Curves of the Implemented Machine Learning (ML) Models for the Classification of Students' Performance using Coursera Dataset Balanced using SMOTE Tomek.

The confusion matrices displayed in Figure 6.11 were also used to evaluate the models' performance in terms of Type I and Type II errors. DT model was able to achieved the best Type II error (i.e., 13%), while MLP was able to achieved the best Type I error (i.e., 18%). Overall, the CATBoost model was able to achieved the most distributed performance among classes. Figure 6.12 shows the ROC curve for all the implemented models and demonstrate the superiority of CATBoost model with AUC of 0.88.

6.1.5 SVM SMOTE

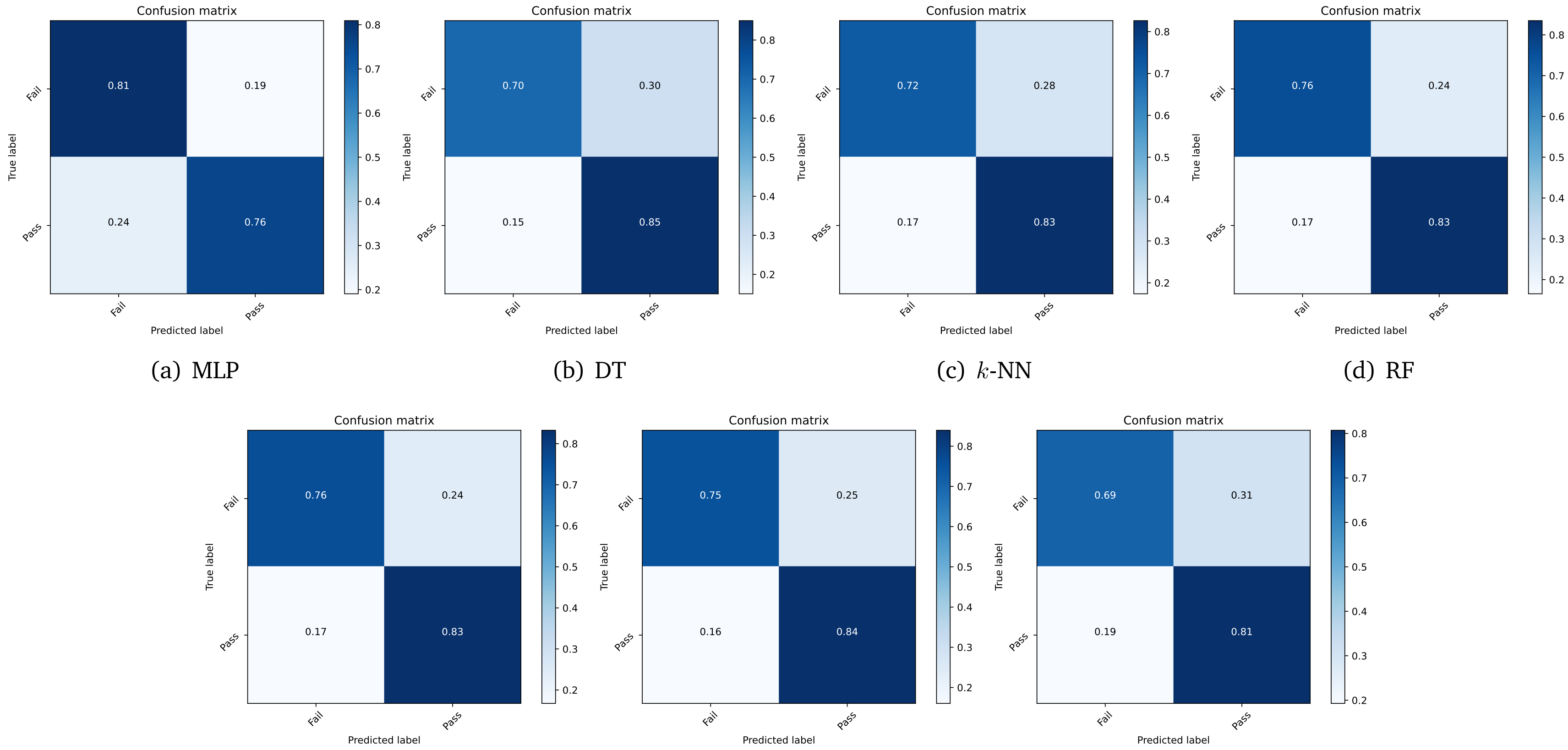
Dataset and Hyperparameters

The Coursera dataset was balanced using the SVM SMOTE technique for this investigation with same number input features and same target variable as in the original dataset. The correlation map between input features and target variable is presented in Figure 6.14. It can be observed that *hits_count* and *Course Grade* are the two most correlated features while *assessment_type_id_6* and *video_duration* are the least correlated features, similar trend as observed in the original dataset.

Experiments were conducted using the same experimental techniques as those described in Section 4.1.2, including the same programming language, package, data pre-processing, dataset splitting, and evaluation measures. The only variation was in the GridSearch reported hyperparameter values (Table 6.9).

Table 6.9: Hyperparameters for Machine Learning (ML) Models Implemented under Experiment 3A for SVM SMOTE.

Model	Hyperparameters
MLP	activation: logistic, learning_rate_init: 0.001, solver: sgd, iter=500
DT	criterion: gini, min_samples_leaf: 100, min_samples_split: 2, splitter: best
k-NN	algorithm: auto, leaf_size: 10, n_neighbors: 100, weights: uniform
RF	criterion: gini, min_samples_leaf: 100, min_samples_split: 2, n_estimators: 10
XGBoost	booster: gbtree, learning_rate: 0.01, max_depth: 3, n_estimators: 2
CATBoost	depth: 4, iterations: 150, l2_leaf_reg: 0.5, learning_rate: 0.001
SVC	degree: 1, gamma: scale



(a) MLP (b) DT (c) k-NN (d) RF

(e) XGBoost (f) CATBoost (g) SVC

Figure 6.13: Confusion Metrices of Implemented Machine Learning (ML) Models for Classification of Students' Performance using Coursera Dataset Balanced using SVM SMOTE.

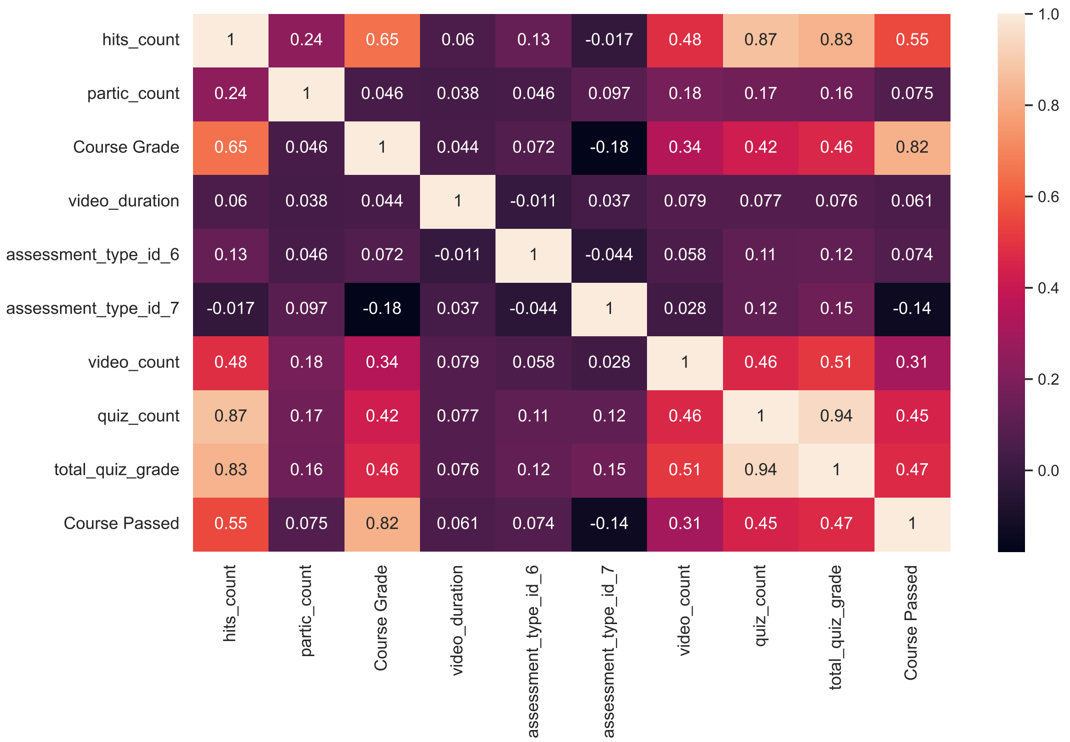


Figure 6.14: Feature Correlation Map for Coursera Dataset Balanced using SVM SMOTE.

Classification Results

Table 6.10 and Figures 6.13 and 6.15 provide numerical and graphical representations of the results, respectively. Table 6.10 displays the quantitative results of detailed experiments conducted using the several ML models for the SVM SMOTE balanced dataset. It can be observed that CATBoost model was able to achieve the best accuracy of 0.79 while SVC was the least accurate with an accuracy value of 0.75. Data balancing using the SVM SMOTE did not improve the performance in comparison to original dataset highlighting the inability of the SVM SMOTE in efficiently sampling the existing data samples.

The confusion matrices displayed in Figure 6.13 were also used to evaluate the models' performance in terms of Type I and Type II errors. DT model was able to achieve best Type II error (i.e., 15%), while MLP was able to achieve the best Type I error (i.e., 19%). Overall, the CATBoost model was able to achieve the most distributed performance among classes. Figure 6.15 shows the ROC curve for all the implemented models and demonstrates the superiority of CATBoost model with AUC of 0.85.

Table 6.10: Quantitative Test Results for Machine Learning (ML) Classification of Students' Performance using Coursera Dataset Balanced using SVM SMOTE.

Models	Accuracy	F1 Score	Precision	Recall	Sensitivity	Specificity
MLP	0.79	0.79	0.79	0.79	0.81	0.76
DT	0.78	0.77	0.78	0.78	0.70	0.85
k-NN	0.77	0.77	0.78	0.77	0.72	0.83
RF	0.78	0.78	0.79	0.78	0.76	0.83
XGBoost	0.79	0.79	0.80	0.79	0.76	0.83
CATBoost	0.79	0.79	0.80	0.79	0.75	0.84
SVC	0.75	0.74	0.76	0.75	0.69	0.81

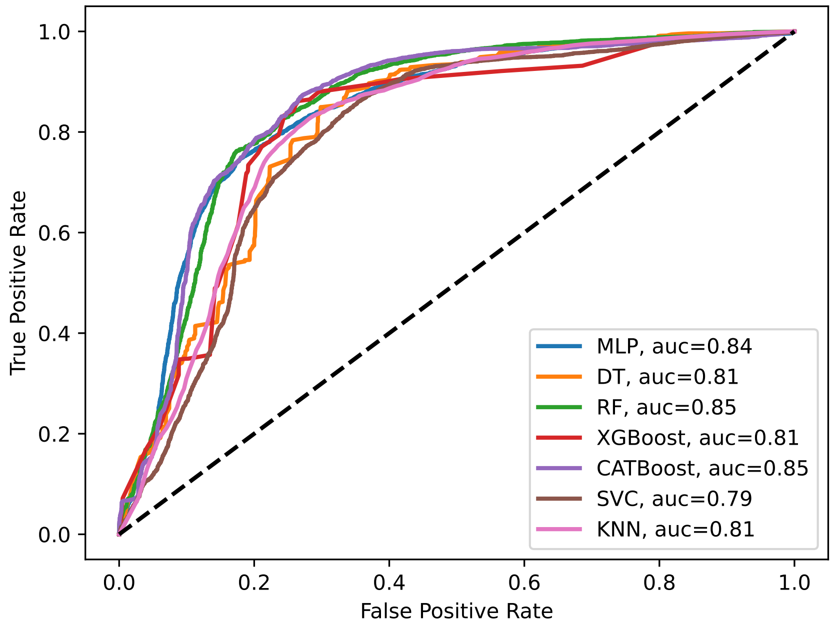


Figure 6.15: ROC Curves of the Implemented Machine Learning (ML) Models for the Classification of Students' Performance using Coursera Dataset Balanced using SVM SMOTE.

6.1.6 Performance Comparison

The performance of ML models for the balanced Coursera dataset was evaluated and compared to highlight the best. Performance for the Coursera dataset balanced using the SMOTE NN was observed to be the best, while, the performance for the Coursera dataset balanced using SMOTE and SVM SMOTE techniques was reported as worst.

Overall, it can be observed that SMOTE NN and SMOTE Tomek techniques were able to provide the better performances in comparison to original dataset, demonstrating their superiority among other implemented for dataset balancing. It can be clearly observed that, single model (i.e., CATBoost) stood out in majority cases with most distributed class performance among all and hence nominated to be the best performing model (see Table 6.11 for comparison). The best Type II error and AUC of 8% and 0.92 were observed for the SMOTE NN balanced Coursera dataset, respectively.

Table 6.11: Performance Comparison of Best Performing Machine Learning (ML) Models for Each Course under Experiment 2.

SMOTE Technique	Best Model	Accuracy	Type II Error	AUC
Borderline SMOTE	CATBoost	0.80	15%	0.85
SMOTE	CATBoost	0.80	15%	0.86
SMOTE NN	CATBoost	0.89	13%	0.95
SMOTE Tomek	CATBoost	0.82	14%	0.88
SVM SMOTE	CATBoost	0.79	16%	0.85

6.2 Experiment 3B - Variational Auto-Encoder

In addition to the SMOTE techniques, a latest approach of using VAE generated synthetic data was also investigated. In this experiment, VAE was used to generated the simulated dataset for Coursera dataset and performance of multiple ML models such as MLP, DT, RF, XGBoost, CATBoost, k-NN and SVC was compared to discuss if VAE proved useful or not. In this experiment, the Variational Auto Encoder (VAE)model was trained to generate the simulated samples from the Coursera dataset and performance of ML models was evaluated on the VAEsimulated dataset.

Experimental Protocols for VAE

The original unbalanced Coursera dataset was used to train the VAE model and generate the artificial samples which were used for this experiment. Experiments were conducted using the same experimental techniques as those described in Section 4.1.2, including the same programming language, package, data pre-processing. However, the structure of VAE was developed using the TensoFlow and Keras packages. For both the encoder and decoder sides, a single hidden layer of fully connected 100 neurons was used with Rectified Linear Unit (ReLU) activation. A custom loss function and gradient optimization was implemented for the training of the proposed VAE. Model was trained for 200 epochs with Adam optimizer (i.e, learnign rate of 0.001). In the end, the classification performance was reported as the accuracy of model to classify the students' performance.

Classification Results

This section presents the experimental results for the ML models in classifying the students performance using the VAE simulated data. The quantitative results are presented in Table 6.12 and indicates the improved performance in comparison to SMOTE techniques. From the results, it can be observed that CATBoost was the best-performing model with a classification accuracy of 94%, while the XGboost and MLP models were the second best with a classification accuracy of 87%.

Table 6.12: Quantitative Test Results for Machine Learning (ML) Classification of Students' Performance using VAE Simulated Coursera Dataset.

Model	k-NN	SVC	RF	DT	CATBoost	XGBoost	MLP
Accuracy	0.83	0.81	0.86	0.86	0.94	0.87	0.87

6.3 Experiment 3C – Synthetic Data using GANs

This experiment investigates the scope of artificially generated data using generative models to improve the performance of ML models for predicting of students' performance on VLE. Under this experiment, GANs are used to generate the artificial data and the performance of multiple ML models, including MLP, DT, RF, XGBoost, CATBoost, k-NN and SVC, compared for the Coursera dataset to understand the effectiveness of generated data.

6.3.1 GAN Simulated Coursera

In this investigation, the GANmodel was trained to generate the simulated samples from the Coursera dataset and the performance of ML models was evaluated on the GANsimulated dataset.

Dataset and Hyperparameters

The original unbalanced Coursera dataset was used to train the GAN model and generate the artificial samples used for this investigation and referred to as GAN Simulated Coursera. The correlation map between input features and target variable is presented in Figure 6.16. It can be observed that *video_count* and *Course Grade* are the two most correlated features. In contrast, *partic_count* and *video_duration* are the least correlated features, a similar trend as observed in the original dataset.

Experiments were conducted using the same experimental techniques as those described in Section 4.1.2, including the same programming language, package, data pre-processing, dataset splitting and evaluation measures. The only variation was in the GridSearch reported hyperparameter values (Table 6.13).

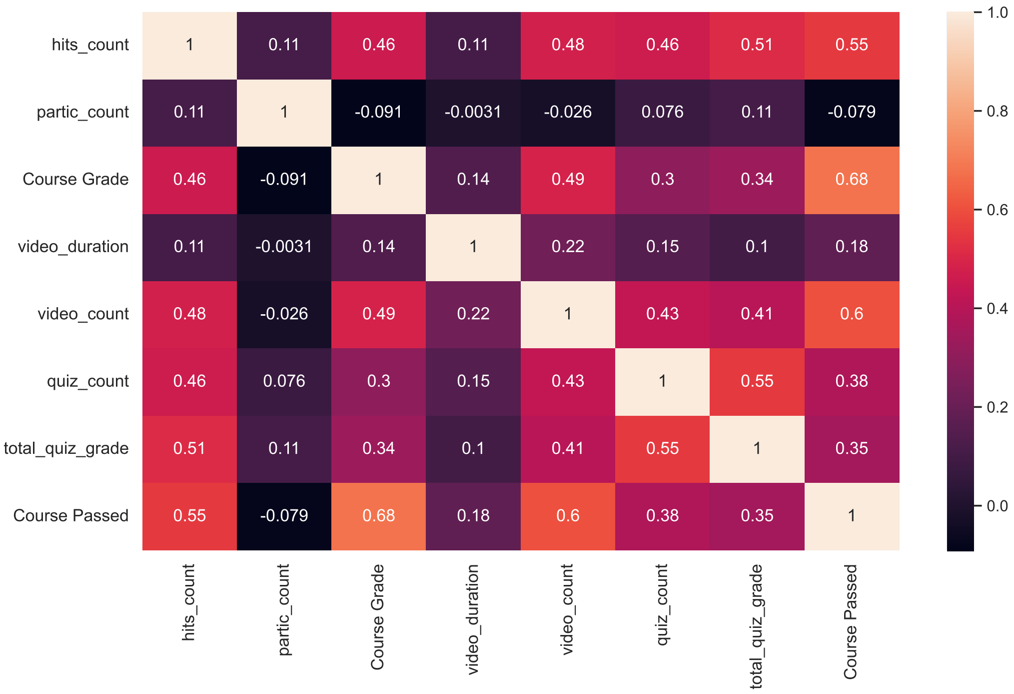


Figure 6.16: Feature Correlation Map for GAN Simulated Coursera Dataset.

Classification Results

Table 6.14 and Figures 6.17 and 6.18 provide numerical and graphical representations of the results, respectively. Table 6.14 displays the quantitative results of detailed experiments conducted using the several ML models for the GAN simulated Coursera dataset. It can be observed that the MLP model was able to achieve the best accuracy of 0.87 while DT was the least accurate with an accuracy value of 0.82. Simulated data was able to improve the overall accuracy of the ML models.

The confusion matrices displayed in Figure 6.17 were also used to evaluate the models' performance in terms of Type I and Type II errors. SVC model was able to achieve best Type II error (i.e., 34%), while CATBoost was able to achieve the best Type I error (i.e., 6%). Overall, the XGBoost model was able to achieve the most distributed performance among classes. Figure 6.18 shows the ROC curve for all the implemented models and demonstrates the superiority of XGBoost model with AUC of 0.89.

Table 6.13: Hyperparameters for Machine Learning (ML) Models Implemented under Experiment 3B for GAN Simulated Coursera.

Model	Hyperparamters
MLP	activation: logistic, learning_rate_init: 0.01, solver: sgd, iter=500
DT	criterion: entropy, min_samples_leaf: 50, min_samples_split: 2, splitter: random
k-NN	algorithm: auto, leaf_size: 10, n_neighbors: 100, weights: uniform
RF	criterion: gini, min_samples_leaf: 100, min_samples_split: 100, n_estimators: 30
XGBoost	booster: gblinear, learning_rate: 0.1, max_depth: 3, n_estimators: 100
CATBoost	depth: 4, iterations: 150, l2_leaf_reg: 0.5, learning_rate: 0.001
SVC	degree: 1, gamma: auto

Table 6.14: Quantitative Test Results for Machine Learning (ML) Classification of Students' Performance using GAN Simulated Coursera Dataset.

Models	Accuracy	F1 Score	Precision	Recall	Sensitivity	Specificity
MLP	0.87	0.80	0.84	0.79	0.93	0.65
DT	0.82	0.74	0.77	0.74	0.89	0.59
k-NN	0.85	0.77	0.83	0.77	0.92	0.62
RF	0.84	0.76	0.83	0.76	0.92	0.60
XGBoost	0.86	0.78	0.83	0.79	0.93	0.65
CATBoost	0.86	0.77	0.84	0.76	0.94	0.59
SVC	0.86	0.78	0.84	0.79	0.93	0.66

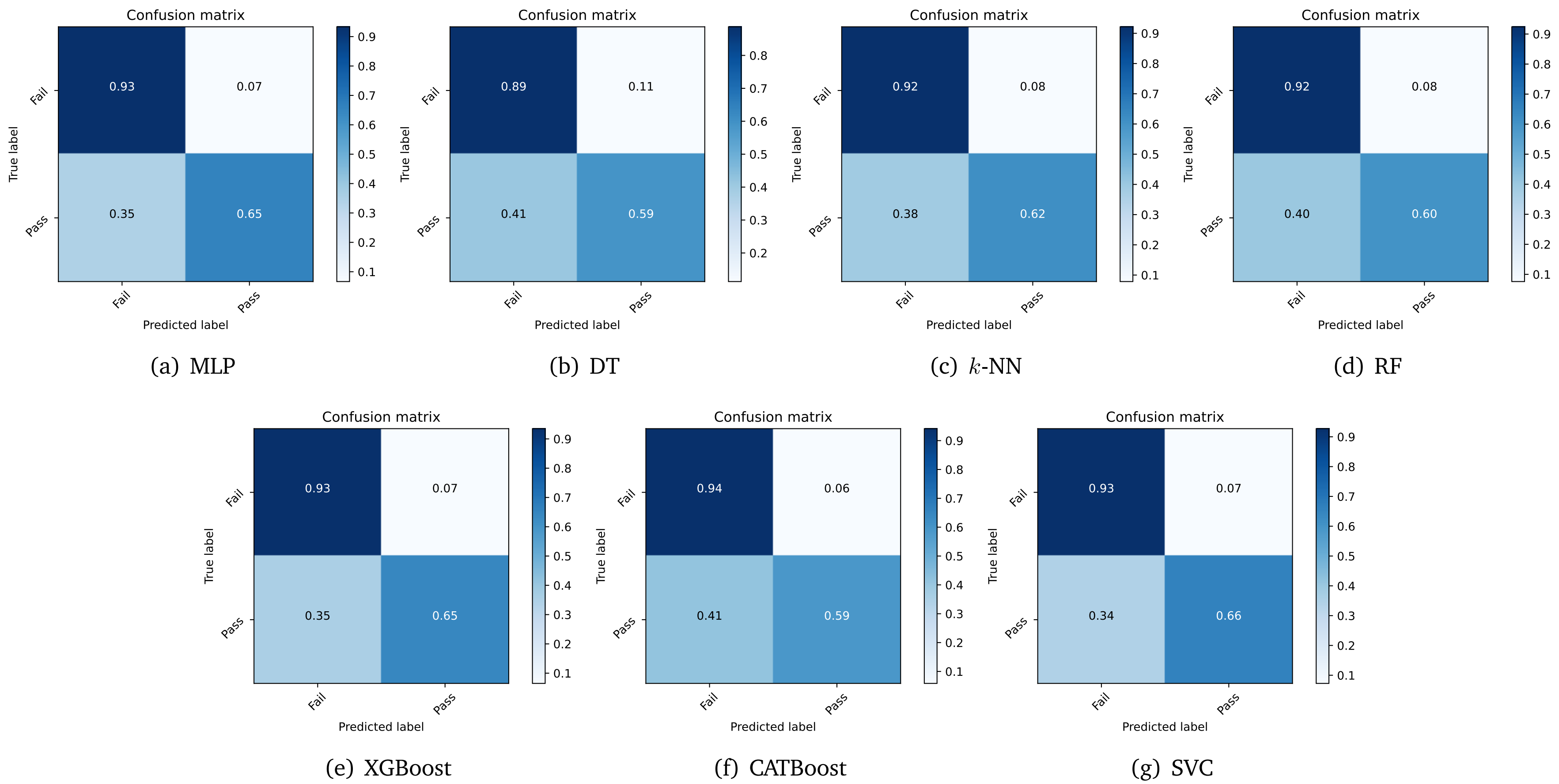


Figure 6.17: Confusion Metrices of Implemented Machine Learning (ML) Models for Classification of Students' Performance using GAN Simulated Coursera Dataset.

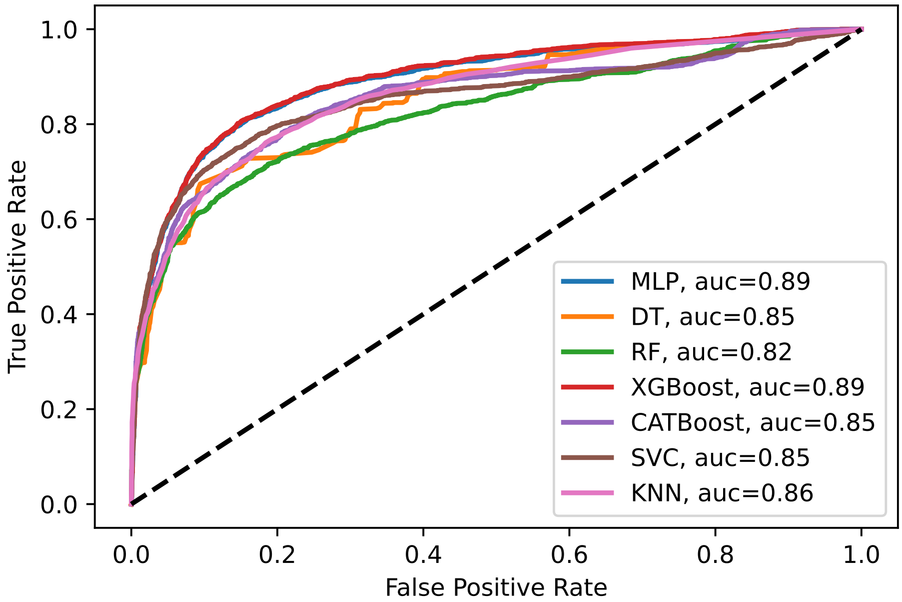


Figure 6.18: ROC Curves of the Implemented Machine Learning (ML) Models for the Classification of Students' Performance using GAN Simulated Coursera Dataset.

6.3.2 Coursera Original with GAN Simulated

In this investigation, the impact of the generated data using GAN was investigated on the original data. For this purpose, experiments were performed using original plus the GAN simulated data and the performance of ML models was investigated.

Dataset and Hyperparameters

Details about the GAN simulated and Original Coursera data have already been discussed in previous section. Experiments were conducted using the same experimental techniques as those described in Section 4.1.2, including the same programming language, package, data pre-processing, dataset splitting and evaluation measures. The only variation was in the GridSearch-reported hyperparameter values (Table 6.15).

Table 6.15: Hyperparameters for Machine Learning (ML) Models Implemented under Experiment 3B for Original+GAN Simulated Coursera Dataset.

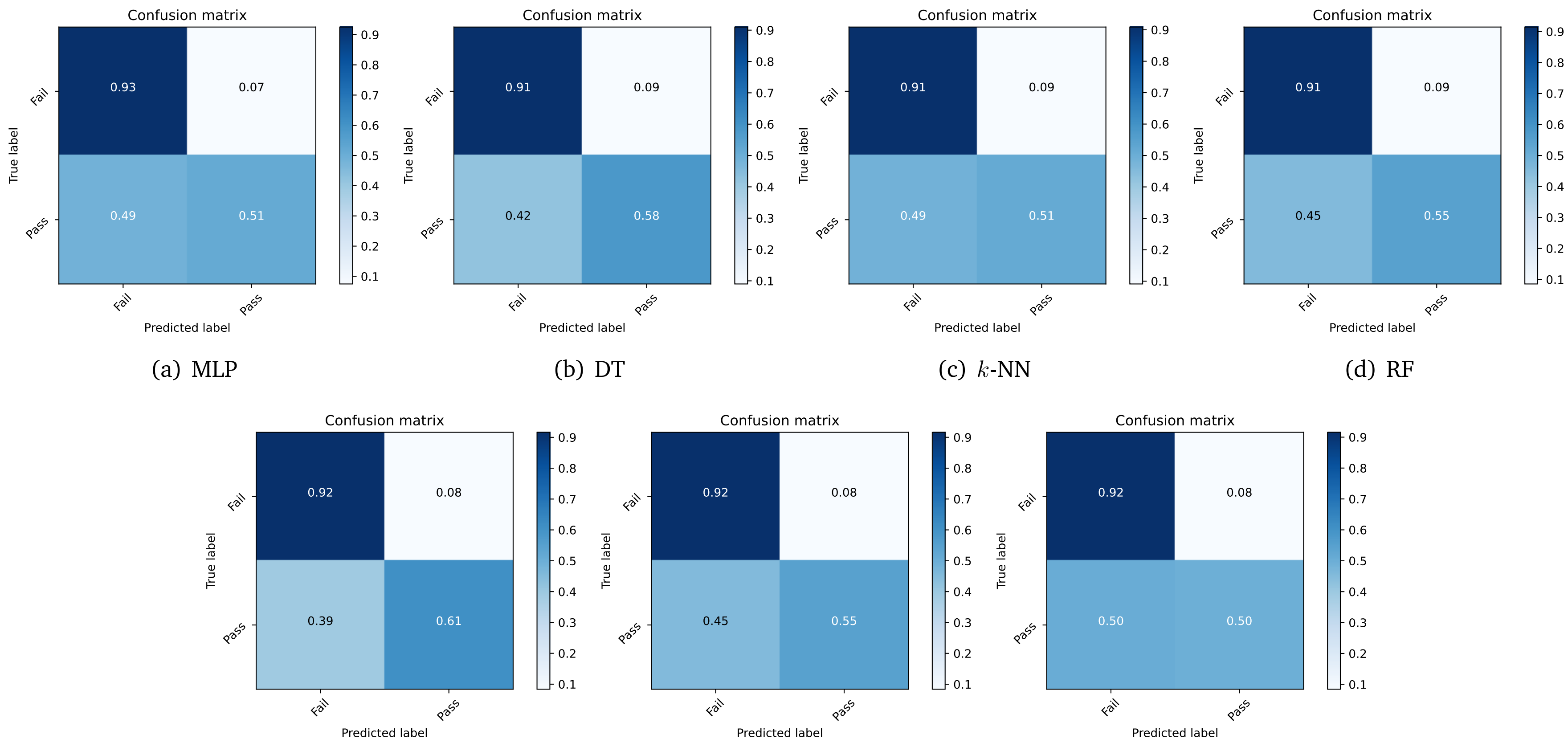
Model	Hyperparameters
MLP	activation: identity, learning_rate_init: 0.1, solver: adam, iter=500
DT	criterion: gini, min_samples_leaf: 100, min_samples_split: 2, splitter: best
k-NN	algorithm: auto, leaf_size: 10, n_neighbors: 100, weights: uniform
RF	criterion: gini, min_samples_leaf: 100, min_samples_split: 50, n_estimators: 10
XGBoost	booster: gblinear, learning_rate: 0.1, max_depth: 50, n_estimators: 10
CATBoost	depth: 4, iterations: 150, l2_leaf_reg: 0.5, learning_rate: 0.001
SVC	degree: 1, gamma: auto

Classification Results

Table 6.16 and Figures 6.19 and 6.20 provide numerical and graphical representations of the results, respectively. Table 6.16 displays quantitative results of detailed experiments conducted using the several ML models for the original plus the GAN simulated dataset. It can be observed that the XGBoost model was able to achieve the best accuracy of 0.83 while SVC and k-NN were the least accurate with an accuracy value of 0.80. Adding the simulated data in the original data in an unbalanced form did not help much in improving the performance. This suggested that class balancing is of more importance in comparison to increased data size for this specific case.

Table 6.16: Quantitative Test Results for Machine Learning (ML) Classification of Students' Performance using Original + GAN Simulated Coursera Dataset.

Models	Accuracy	F1 Score	Precision	Recall	Sensitivity	Specificity
MLP	0.80	0.72	0.79	0.71	0.93	0.51
DT	0.82	0.75	0.78	0.74	0.90	0.58
k-NN	0.80	0.71	0.77	0.71	0.91	0.51
RF	0.82	0.74	0.79	0.74	0.91	0.55
XGBoost	0.83	0.77	0.82	0.76	0.92	0.60
CATBoost	0.81	0.74	0.78	0.73	0.92	0.55
SVC	0.80	0.71	0.77	0.71	0.92	0.50



(a) MLP (b) DT (c) k-NN (d) RF

(e) XGBoost (f) CATBoost (g) SVC

Figure 6.19: Confusion Metrics of Implemented Machine Learning (ML) Models for Classification of Students' Performance using Original + GAN Simulated Coursera Dataset.

The confusion matrices displayed in Figure 6.19 were also used to evaluate the models' performance in terms of Type I and Type II errors. XGBoost model was able to achieve best Type II error (i.e., 39%), while MLP was able to achieve the best Type I error (i.e., 7%). Overall, the XGBoost model was able to achieve the most distributed performance among classes. Figure 6.20 shows the ROC curve for all the implemented models and demonstrates the superiority of XGBoost model with AUC of 0.87.

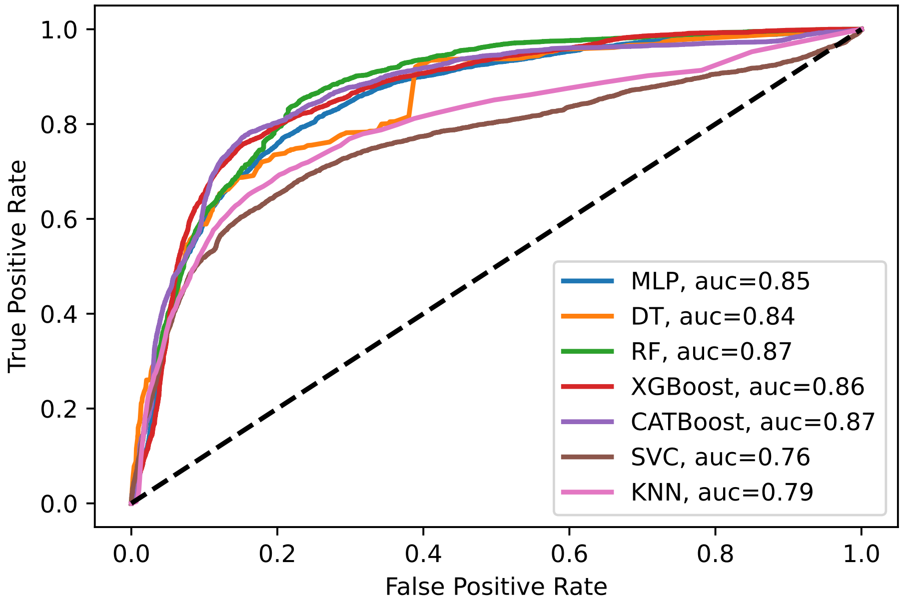


Figure 6.20: ROC Curves of the Implemented Machine Learning (ML) Models for the Classification of Students' Performance using Original + GAN Simulated Coursera Dataset.

6.3.3 Coursera with SMOTE-GAN

In this investigation, GAN model was trained using the SMOTENN balanced Coursera dataset and the performance of ML models was evaluated.

Dataset and Hyperparameters

The Coursera dataset was first balanced using the SMOTENN approach (i.e., identified best from Experiment 3A) and then GAN model was trained to generate the simulated data samples. The correlation map between input features and target variable is presented in Figure 6.14. It can be observed that *hits_count* and *Course Grade* are the two most correlated features while *partic_count* and *video_duration* are the least correlated features, similar trend as observed in the original dataset.

Experiments were conducted using the same experimental techniques as those described in Section 4.1.2, including the same programming language, package, data pre-processing, dataset splitting and evaluation measures. The only variation was in the GridSearch-reported hyperparameter values (Table 6.18).

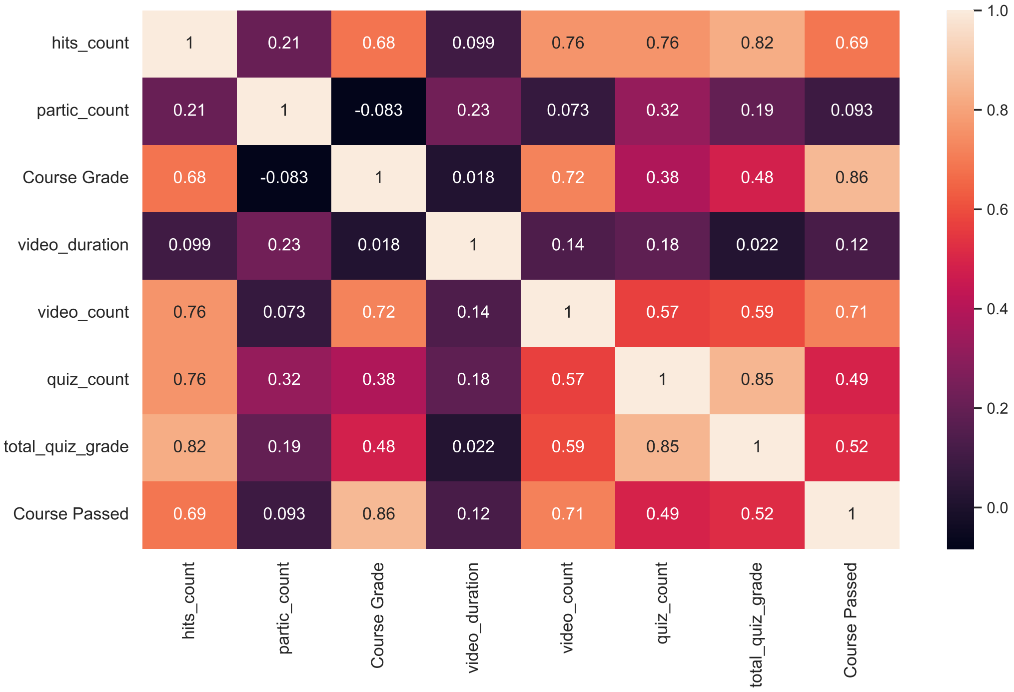
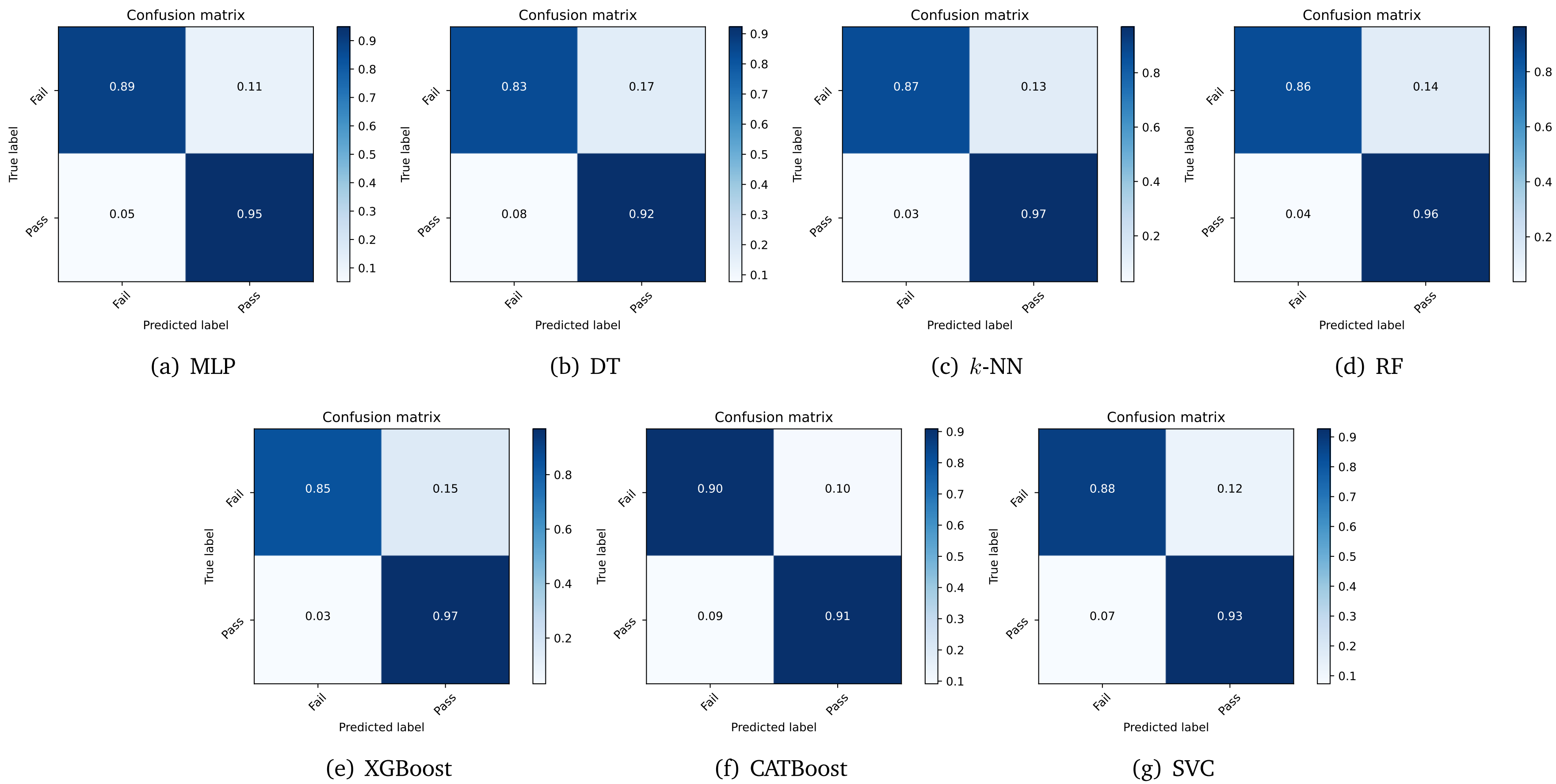


Figure 6.21: Feature Correlation Map for SMOTENN-GAN Coursera Dataset.

Table 6.17: Hyperparameters for Machine Learning (ML) Models Implemented under Experiment 3B for SMOTENN-GAN Coursera Dataset.

Model	**Hyperparameters**
MLP	activation: relu, learning_rate_init: 0.1, solver: adam, iter=500
DT	criterion: gini, min_samples_leaf: 1, min_samples_split: 50, splitter: random
k-NN	algorithm: auto, leaf_size: 10, n_neighbors: 5, weights: distance
RF	criterion: entropy, min_samples_leaf: 1, min_samples_split: 2, n_estimators: 50
XGBoost	booster: gbtree, learning_rate: 0.1, max_depth: 10, n_estimators: 100
CATBoost	depth: 5, iterations: 150, l2_leaf_reg: 1, learning_rate: 0.001
SVC	degree: 1, gamma: auto



(a) MLP

(b) DT

(c) k-NN

(d) RF

(e) XGBoost

(f) CATBoost

(g) SVC

Figure 6.22: Confusion Metrics of Implemented Machine Learning (ML) Models for Classification of Students' Performance using SMOTENN-GAN Coursera Dataset.

Classification Results

Table 6.10 and Figures 6.22 and 6.23 provide numerical and graphical representations of the results, respectively. Table 6.10 displays the quantitative results of detailed experiments conducted using the several ML models for the SMOTENN-GAN Coursera dataset. It can be observed that the MLP model was able to achieve the best accuracy of 0.92 while DT was the least accurate with an accuracy value of 0.88. A significant improvement in results was improved for the case when the original data was first balanced using the SMOTENN technique prior to generating simulated data.

The confusion matrices displayed in Figure 6.22 were also used to evaluate the models' performance in terms of Type I and Type II errors. XGBoost model was able to achieve best Type II error (i.e., 3%), while CATBoost was able to achieve the best Type I error (i.e., 10%). Overall, the MLP model was able to achieve the most distributed performance among classes. Figure 6.23 shows the ROC curve for all the implemented models and demonstrates the superiority of CATBoost model with AUC of 0.97.

Table 6.18: Quantitative Test Results for Machine Learning (ML) Classification of Students' Performance using SMOTENN-GAN Coursera Dataset.

Models	Accuracy	F1 Score	Precision	Recall	Sensitivity	Specificity
MLP	0.92	0.92	0.92	0.92	0.89	0.95
DT	0.88	0.88	0.89	0.88	0.83	0.92
k-NN	0.92	0.92	0.93	0.92	0.87	0.97
RF	0.92	0.91	0.93	0.91	0.86	0.96
XGBoost	0.91	0.91	0.92	0.91	0.85	0.97
CATBoost	0.91	0.90	0.91	0.91	0.90	0.91
SVC	0.90	0.90	0.91	0.90	0.88	0.93

6.3.4 Performance Comparison

Performance of ML models for different simulated dataset combinations was evaluated and compared to identify the best. It can be observed that SMOTENN-GAN simulated dataset was able to achieve the best performance while the unbalanced GAN simulated dataset did not improve the performance when mixed with the original dataset. Overall, for SMOTENN-GAN case, MLP model was able to achieve the best ML classification performance (see Table 6.19 for comparison). The best Type

II error and AUC of 8% and 0.92 were observed for the SMOTENN-GAN simulated dataset, respectively.

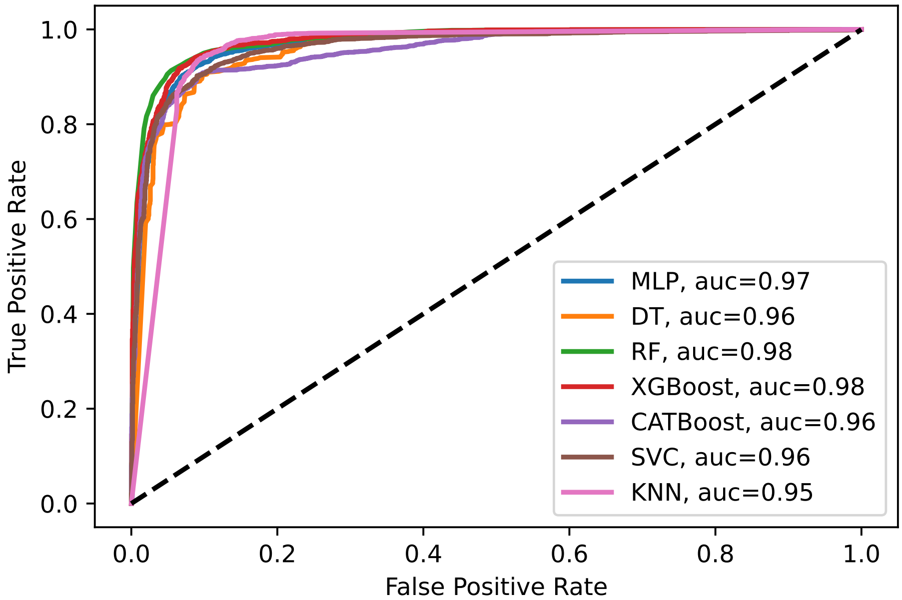


Figure 6.23: ROC Curves of the Implemented Machine Learning (ML) Models for the Classification of Students' Performance using SMOTENN-GAN Coursera Dataset.

Table 6.19: Performance Comparison of Best Performing Machine Learning (ML) Models for Each Course under Experiment 2.

Simulated Dataset Setup	Best Model	Accuracy	Type II Error	AUC
GAN Simulated Dataset	XGBoost	0.86	35%	0.89
Original + GAN Simulated Dataset	XGBoost	0.83	39%	0.86
SMOTENN-GAN Simulated Dataset	MLP	0.92	4%	0.97

6.4 Summary

This chapter provides the summary of the results, concluded from different experiments, performed and done under the context of predicting students' performance on *VLE* by using ML models like *MLP, DT, RF, k-NN, XGBOOST,CATBOOST* and *SVT*. Two techniques used to handle imbalance data in this chapter. First *VAE* applied on the original data to generate samples from minority class. The results showed improvement in *ML* performance for VAE model compared with original Coursera dataset. Furthermore, the improvemnt based from the increase in samples that provide to the models. The VAE focused on generating new samples from the data distribution, which is represented in the latent space after encoding the original data. The second technique used to state the novelty of deploying models in balancing data set field is *GAN*. The results showed almost better performance compared to *VAE*. Handling imbalanced data set by *GAN* that generates samples from minority classes in the Coursera dataset. The results showed significant improvement in performance using *GAN*.

Chapter 7

Conclusion

This chapter presents the summary of results abducted from the experiments performed under the scope of predicting student performance on VLE using ML models within different datasets. The chapter is structured to provide the information about how the formulated Research Questions (RQ) were addressed and what findings were achieved. Furthermore, the chapter also outlines potential future research directions in this context.

7.1 Conclusion

The research performed in this aimed to perform a detailed review of the literature with regards to the student performance prediction on VLE. The goal was to identify the significant factors in OULAD and Coursera datasets. Furthermore, the class imbalance problem in data mining was also discussed in detail and experiments were performed to address the issue. The knowledge gained from the research activities was then used to implement applications and evaluate various solutions for imbalanced data sets.

To address the RQ1 (*What are the most influential and significant features related to prediction of students' performance over VLE?*),this research question answered by applied ML algorithms on two datasets (i.e., OULAD and Coursera). There were four experiments done under this question (i.e., Experiment 1A, 1B, 1C and 1D) as follows:

- Experiment 1A: The significant factors affect students' performance in OULAD is (where the student lived during the module presentation and number of previous attempts which is the number of times the student has attempted a specific module). These were the two most correlated features which impact the performance of prediction student performance in OULAD. Overall, the ML applied on OULAD, the CATBoost achieved the Type I error of 77%, a very high values, On the other hand, MLP was able to achieve the least Type II error of only 5%. Overall, based on the distribution of Type I and Type II errors, the CATBoost model preformed best among all. The ROC curves for the implemented models and confirms that all models performed in a similar capacity, with CATBoost a bit better end with an AUC of 0.6. The research done by Hussain et al. [156] also backed these findings by illustrating that the CatBoost model acquired an accuracy and precision of 93%.

- Experiment 1B: The significant factors in combined Coursera dataset is the total number of mouse hits student did while using the VLE, and quiz count is the number of quizzes student had during the module). The total number of mouse clicks the student press during log in on VLE. This indicates the student engagement and participation on the modue. In addition, it could show that the low number of mouse clicks indicates low student engagement on the module. These were the two most correlated features which impact the performance of prediction models in Coursera. The total number of Student' mouse hits correlated with the student' performance. It could be a positive relationship between the number of student' mouse clicks and the duration student spent on the Coursera module, which leads to significant impact on student' performance. Comparable to Experiment 1A, the performance of ML models was approximate to each other with accuracy towards 80%. Overall, a higher accuracy was spotted for the Coursera dataset classification compared to OULAD.

- Experiment 1C: After applied feature engineering on combined Coursera dataset, a new feature was added as performance divided into three target (Low, Medium, and High). The XGBoost and CATBoost were the best performing models with accuracy of 81%. While MLP and RF were observed as the second best with accuracy of 80%.

  Type II error is defined as the "instances from High class predicted as Low class". While the Type I error is defined as the "instances from Low class predicted as H class". Overall, the RF model was able to get the minimum value of Type II error 2%, while the DT model got the best value of Type I error 1%. Overall, it can be observed that Medium class was most often miss-classified as the High class and same is the case where High class was miss-classified as the Medium class. Overall, in terms of class distribution, DT model can be nominated as the best given that it offers the least miss-classifications between the distant classes. Similar results were also recorded in research conducted by Bayani et al. [157] and Massaoudi et al. [158] that showed that the CatBoost model has the highest performance accuracy of almost 91.02% and 98.36%, respectively. DT model was also reported to have an accuracy of 88.52% when compared with DT, CART, JR [64].

- Experiment 1D: Ensemble methods applied into two ways as stacking and voting, the ML used in this method (DT, k-NN, and XGBoost) were implemented on the Coursera datasets. The ensemble of DT, k-NN and XGBoost by the stacking method achieved the classification accuracy of 97%, while by the voting method achieved 85%. This indicates a significant improvement in compared individual models implemented previously in Experiment 1B. From the results, it can be reported that the combination of multiple models reported a better performance, however, may have impacted the inference speed and use of computational resource. Similar findings were presented by [81] that have adopted combination of models for their study due to their better performance.

The relevance of VLEs encouraging learners to adopt remote learning has significantly enhanced by the pandemic. Daily, a tremendous amount of information is produced as learners use VLEs to carry out various tasks and retrieve educational resources. The data collected should be handled and analyzed by the appropriate ML program to be helpful. The main areas of application for ML techniques are EDM. To reveal new correlations in raw information and create a predictive model for making prognostications, including estimating academic achievement, lower academic achievement, involvement, etc., ML methods are frequently utilized. It is crucial to use the most effective and suitable ML technique to achieve the highest academic achievement in a virtual learning environment. Teachers and supervisors could employ the ML approaches and the strategies highlighted in this dissertation to enhance the training resources and give learners well-informed advice, enhancing and customizing their academic opportunities.

In conclusion, the experiment 1 results showcased the potential of ML models in predicting students' performance within VLEs, utilizing both the OULAD and Coursera datasets. While both datasets are valuable resources for data-driven solutions in education, they differ in their composition and scope. The OULAD dataset offers a comprehensive and diverse set of features, including demographic, performance, and behavioral data, enabling in-depth analysis of student behavior and learning outcomes. In contrast, the Coursera dataset provides specific information on course progress, assessments, grades, discussions, and learner demographics from a narrower range of courses. Leveraging the strengths of these datasets and addressing critical factors such as class imbalance, dataset complexity, and data quality will enhance the accuracy and effectiveness of ML models in predicting student success. Further exploration of course-specific factors influencing model performance will pave the way for personalized interventions and improved educational strategies in the future.

To address the RQ2 (*To what extent students' performance can be predicted using classification models on VLE?*), a detailed experiment was performed using the range of ML models (see Experiment 1 and Experiment 2). The purpose of Experiment 1 was to investigate the extent to which students' performance can be predicted using classification models in VLE. The prediction models were based on features related to course design, teaching style, student/teacher access patterns, and student life circumstances. In addition, the experiment explored the use of feature engineering to improve the classification results on the Coursera dataset. Followings important findings were reported from Experiment 1:

- The results of Experiment 1 indicated that the performance of the various ML models was relatively similar, with slightly lower results overall. However, the use of feature engineering led to improved performance on the Coursera dataset, with a classification accuracy of 81%. This suggests that there may be potential for improving the models, approach, and data used in predicting students' performance in a VLE. The literature review has also highlighted several researches comparing the efficiency of models for predicting students' performance in the VLEs. For example, Hooshyar et al. [159] applied various models on the procrastination behavior of students to assess their productivity. It was found that among all the applied models, NN was the most effective model, with 96% accuracy for categorical features. The results also indicated that LSVM is the most efficient model, with a classification accuracy of 95%. Similarly, in another research, Tomasevic et al. [69] evaluated the effectiveness of different modeling approaches to analyze the impact of different features on students' performance. The findings showed that SVM had a classification accuracy of 96.4% when demographic, performance and engagement features were applied. Moreover, the highest classification accuracy of 92.54% was observed for the deep learning model when applied to features like students' performance section information and activity participation [85]. In contrast, research also has reported SVR as the most efficient algorithm for the prediction while BP as the most effective algorithm for the classification [84].

However, most studies have reported that SVM has the highest classification accuracy when features like student background performance demographic, engagement, course activities and past performance are used to estimate the students' performance in the virtual learning environment. For example, it is observed that SVM has a classification accuracy of 96% [81], 96.4% [69], and 95% [86] in different studies. It is also evidence from earlier research that DNN is also a very effective model for predicting students' performance in VLEs when features like performance and participation demographics, clickstream behavior and assessment performance are applied. The results show that DNN has a highest classification accuracy of 92.54% [85], 95.34% [70] and 93% [87] in different investigations. On the other hand, experiment 1 has shown that different models applied to assist students' performance in the VLEs have a classification accuracy of almost 65%. This implied a need for enhancement in the applied models and data sets used to forecast students' performance in a virtual learning environment.

- One key finding from Experiment 1 was the impact of the class imbalance problem on the performance of the ML models. The class imbalance problem refers to a situation in which the number of instances belonging to one class (the minority class) is significantly lower than the number of instances belonging to the other class (the majority class). This can lead to ML models being biased towards the majority class, resulting in poor performance when predicting the minority class. In Experiment 1, the class imbalance problem was identified as the main reason for the degraded performance of the ML models. This finding highlights the need for further research on methods to address the class imbalance problem in the context of predicting students' performance in a VLE. These results are in accordance with earlier researches. Numerous studies [160]; [161] have found that the situation of class imbalance negatively affects ML models. Prior research has recommended balancing the datasets from the positive and negative classes to improve the predicted accuracy using Catboost to reprocess the entire dataset [162]. To achieve equality across samples from various classes, academics often use one of three techniques: under-sampling the majority class, over-sampling the minority class or combining the two [163]. As most ML systems bias toward the majority class, imbalanced datasets face a significant problem. It is noticeable that the minority class faces substantial academic challenges and incurs exorbitant prices due to misrepresentation. The imbalanced datasets in this subject have made it extremely difficult to forecast learners' progress, and the various interpolation techniques must be compared [164]. Imbalanced data characterization is a common observation recorded in the domain of DM and ML. The distribution of variables in imbalanced data differs substantially between classes because imbalance data classification is a form of supervised learning. There are two types of classes observed in data distribution. The first type is a positive class or the minority class, which has variables with less iteration. The second type of class, the negative or majority class is one with variables with more observations. These classes of data distribution are present in binary or multi-class settings and prone to imbalanced data classification [165]. ML models are typically built on scenarios where the distribution of the information is reasonably balanced. When presented with uneven data classification, most ML models can develop varying levels of flaws or might subsequently prove ineffective [166]. Thus, there is relevant explanatory value and practical utility in improving the evaluation and comprehension of ML models for classifying imbalanced data.

- Overall, the results of Experiment 1 suggest that further investigation is needed to fully understand the factors that influence prediction performance and to develop more effective approaches to predicting students' performance in a VLE. While the ML models in Experiment 1 achieved relatively similar performance, with the potential for improvement through the use of feature engineering, the impact of the class imbalance problem indicates the need for more research on methods to address this issue.

In Experiment 2, ML models were applied individually to each course within the Coursera dataset in order to identify course-specific patterns and to further investigate the class imbalance problem. The significance of features from each course was determined using correlation maps, and the impact of each feature was interpreted from the results. Following findings were reported from the Experiment 2:

- The results of Experiment 2 showed that the XGBoost model achieved the best classification performance, with a score of 87% for the Computational Mathematics course and a Type II error rate of only 8%. This suggests that the data quality for the Computational Mathematics course was much better in comparison to the other courses. The better performance of the XGBoost model compared to the MLP model may be due to the presence of high correlated relationships, for which conventional models tend to perform better than deep learning models. The results are also backed by the published literature. A literature review also showed that the classification accuracy of XGBoost is 97.98% when performance and activity features are employed [85]. Similarly, research has indicated that the accuracy of MLP for custom-collected datasets is 92.54% when performance and engagement feature engineering is applied [85].

- From the experiments, class imbalance was identified as one of the main reasons for the degraded performance of ML models on the existing datasets. . This is also supported by previous research. As [167] have illustrated that class imbalance has a negative impact on the ML models' performance.

The literature review has revealed that different models have different extents of predicting students' performance in VLE. For example, multi-regression models have an accuracy of almost 0.147 RMSE [79], the k-NN model has an accuracy of 88% [80], SVM models showed an accuracy of 86% [81], and DT, CART, JRIP, GBT, and NBC models are accurate upto 88.52% [64]. These extents to measure the performance of students showed the accuracy of these models. These models have employed various performance indicators to assess the students' efficiency in VLE. Some of these indicators are past performance, interaction with LMS, course activities, mouse clicks, mouse overs, students' backgrounds, academic records, demographic, and class participation. This showed that students more engaged with the learning management system are more productive than others. Similarly, students who frequently visit the course materials provided in the VLEs perform better than others. This suggested that the syllabus for that specific class particularly addressed their educational needs. Similarly, students with sound educational records and past achievements also tend to be more engaged during classroom in VLE.

The researcher found that when all models were applied in a similar capacity, CATBoost outperformed the other models with an AUC of 0.6. The experiment showed that when the performance of different ML models was approximated to each other, an accuracy of 80% was observed. A higher accuracy was spotted for the Coursera dataset classification compared to OULAD. The experiment also indicated that the CATBoost and XGBoost are the most efficient models, with an accuracy of 81%. It was also found that MLP and RF are the second-best-performing models, with an accuracy of 80%. Whereas in the case of class distribution DT model can be considered the best performing since it provides the least miss-classifications between the remote classes. The research suggested a combination of different models of better efficiency, but they can have certain limitations in their performance. The efficiency of the different ML models was generally comparable, with marginally weaker outcomes. However, feature engineering resulted in better performance, with a classification accuracy of 81%. This implies that there might be room for improvement in the models, strategy, and information utilized to forecast students' performance in a VLE.

To address the RQ3 (*To what extent balancing techniques can be used to improve the class imbalance problem in existing VLE datasets?*), a detailed investigation was preformed (see Experiment 3A) using multiple SMOTE techniques for class balancing to demonstrate the impact on the prediction performance of ML models. Followings findings were reported from the experiments:

- The results of Experiment 3A showed that the SMOTE NN and SMOTE Tomek balancing techniques performed best among the implemented methods. For the SMOTE NN balanced Coursera dataset, the CATBoost model achieved a classification accuracy of 89% and a Type II error rate of 8%. The SMOTE NN method was able to effectively sample the original data, leading to improved performance of the ML model. Similar findings are reported by Kumar et al. [164]. They found that SMOTEEN, with k-NN, was determined to be an optimal model with an accuracy of 96.25%. Batista et al. [168] also reported that for datasets with a modest quantity of positive cases, Smote + Tomek and Smote + ENN both produced excellent outcomes. Cavalcanti et al. [162] employed SMOTE to restructure samples with variable input concentrations. They then utilized the balanced data to determine if the teachers' evaluation was of a strong lineup. The researchers attained close to 87% accuracy, which is a 2% enhancement over the algorithm without SMOTE.

- The SVM SMOTE technique was reported to be the worst in sampling the original data. This may be due to its inability to capture the significant patterns from the data in comparison to the other methods. For the SVM SMOTE balanced dataset, the CATBoost model achieved a classification accuracy of 79% and a Type II error rate of 16%. These results suggest that the use of different class balancing techniques can impact the performance of ML models in predicting students' performance in a VLE. The results are in accordance with the findings of Hlosta et al. [169]. They proved that XG-Boost has great prediction accuracy. These results also supported the findings of Nasir et al. [163] that showed that with a proportion of 94.51, the CATBoost model was the highest accurate classification model.

Finally, to address the RQ4 (*To what extent available techniques can be used to improve the class imbalance problem in existing VLE datasets?*), Experiment 3B and 3C were performed on Coursera dataset. Experiment 3B where VAE applied to generate samples from original data, the CAT-Boost was the best-performing model with an accuracy of 94%, while the XGboost and MLP models were the second best accuracy of 87%. The least ML performing models was SVC with an accuracy of 81%.

Experiment 3C where generative model (i.e., GAN) was used to create the synthetic data samples and their impact on the classification performance was studied. The proposed approach of using the GAN with SMOTE NN indicated its ability and scope in tackling the class-imbalance problem in comparison to solely using the sampling techniques. Followings findings were reported from the experiments:

- The performance for SMOTE NN balanced dataset combined with the GAN simulated dataset (SMOTENN-GAN) was reported to achieve the best performance in comparison to the unbalanced dataset simulated with GAN. The MLP model was able to achieve the classification accuracy of 92% with only 4% Type II error. Chen et al. [170] also stated that there are several resampling models that can be used to address the imbalanced dataset challenges. These included RUS, oversampling with SMOTE, and hybrid sampling (SMOTEENN).
- GAN was combined with SMOTENN to combat the class imbalance problem at the data level. The outcome of this novel approach was the introduction of an informed over sampling technique that reduces the number of instances in the minority class without removing useful information.

Class imbalance is a common problem in real-world datasets that can negatively impact the performance of ML models. In this research, multiple SMOTE techniques were implemented to address the class imbalance problem in an experiment and their impact on classification performance was investigated. The results showed that the SMOTE-NN technique was effective in improving the classification performance.

Additionally, a generative model (GAN) was used to balance the data by generating samples of the minority class to reach balanced classes. The proposed techniques' results demonstrated better performance than the original datasets, with the SMOTE-NN-GAN approach achieving around 90% classification accuracy. Overall, this research highlights the effectiveness of using generative models to balance imbalanced datasets and improves ML performance. The SMOTE-NN technique in particular was found to be a promising approach, as it reduces the number of instances in the minority class without removing useful information. It is worth noting, however, that these results are specific to the dataset and methods used in this study and may not necessarily generalize to other datasets and scenarios.

To conclude, imbalanced datasets are a prevalent issue in prediction models since they can lead to under-represented minority samples being badly equipped by the ML model, compromising the predicted prediction accuracy of the approach. In the era of big data, using modeling techniques for virtual learning is becoming more and more prevalent. The biases of the prediction accuracy, though, might make it impossible for minority group learners to benefit from the identical quality classroom management as a consequence of using the ML model.

7.2 Implications for Practice

ML models trained for the prediction of students' performance has significant implications for practice in education field. The presented results and trained model in this research can be efficiently used by the educators, administrators and policymakers to improve the current learning experience in VLE. There are examples in literature where a ML integrated system and/or dashboard is introduced to efficiently monitor and predict the perofrmance of students [98, 99, 100, 101, 102, 103, 104]. A few of the proposals have been discussed in this section about how the trained models can be effectively used:

- Identification of Early at Risk Students: The ML predictive models developed in this thesis can be effectively used to identify the course dropouts by integrating them in the university LMS system and monitoring real-time student data including assessment scores, demographic information and course engagement. Once identified, educators can implement targeted interventions to those students and can improve on the dropout rates. These interventions may include personalized guidance, additional resources, mentoring programs, or referral to student support services. The timely identification and intervention of at-risk students can significantly improve their academic performance, increase retention rates, and contribute to their overall success.

- Improvement in Curriculum and Teaching: The analysis of the ML models' performance on different datasets and courses can provide valuable insights for curriculum and teaching improvement. Educational institutions can utilize these insights to identify areas where students consistently struggle or excel, enabling them to refine course materials, modify teaching strategies, or introduce new teaching techniques.

- Allocation of Resources: The insights from the ML model predictions can assist educational institutions in resource allocation and planning. By identifying courses where students are more likely to face difficulties or fail, institutions can allocate additional resources, such as specialized faculty, teaching assistants, or interactive learning materials. This data-driven approach optimizes resource utilization and ensures that adequate support is provided where it is most needed. Furthermore, the predictive models can guide educational institutions in strategic planning and decision-making. By analyzing historical data and predicting future student performance, institutions can anticipate changes in enrollment patterns, identify emerging educational trends, and align their resources and offerings accordingly. This proactive approach enables institutions to stay responsive to evolving student needs and demands.

Let us consider a complete scenario about how one of the proposals mentioned above can be executed to enhance the virtual learning experience of students at a University. The presented scenario is similar to the proposed integrated systems by Islam and Mahmud [103], Susnjak et al. [104], Xin and Singh [102] and Cechinel et al. [101]. The best performing ML model (e.g., XGBoost) will be integrated into the university's LMS or similar platform. The LMS system collects the data related student including course engagement, assignment submission, assessment marks, participation and demographic information. The ML model will in real-time analyse the performance of student based on the collected data in terms of potential dropout. Whenever the model will identify that a student is at risk of dropout, it will send alerts to respective educators and advisors. Educators can than will be able to reach out to student for any counselling and personalized guidance to address their challenges. Based on the predictions and insights generated by the model, educators can adapt the learning experience for individual students. They can recommend relevant supplementary materials, suggest alternative learning pathways, or provide additional practice opportunities to help students overcome their weaknesses and reinforce their strengths. Educators can utilize the model's predictions to suggest elective courses, specialized programs, or interdisciplinary learning opportunities that align with students' interests and career goals. This personalized learning approach ensures that students receive tailored guidance and resources, enhancing their motivation, engagement, and overall learning outcomes.

7.3 Research Limitations

While the research presented in this thesis provides valuable insights and practical implications of ML based student performance prediction, it is essential to acknowledge certain limitations that need to be considered.

- Lack of Explainability: One of the inherent challenges of ML models, particularly complex algorithms like MLP and XGBoost, is their lack of explainability. While these models can accurately predict student performance, understanding the underlying reasons or factors driving those predictions can be challenging. This lack of interpretability hinders one's ability to gain insights into the specific features or variables influencing the predictions. Consequently, it becomes difficult to provide comprehensive explanations to stakeholders and make informed decisions based on the model's outcomes. Exploring the model towards understanding why model performed better or worse is still an active area of research.
- Data Availability and Quality: The performance of ML models heavily depends on the quality and availability of data. In this thesis, the performance prediction models were trained on benchmark VLE datasets (i.e., OULAD, Coursera). However, these datasets may not fully represent the diversity of educational contexts and institutions. Moreover, the quality of data, including missing

values, inconsistencies, or biases, can affect the model's accuracy and generalizability. Careful consideration and preprocessing of data are essential to mitigate these limitations.

- Class Imbalance: The research acknowledges the presence of class imbalance as a factor that affects the performance of ML models. The imbalance can lead to biased predictions, where the model tends to favor the majority class. Although experiments were conducted to address class imbalance using SMOTE and generative models, further research and exploration of advanced techniques for addressing class imbalance are needed to improve the model's performance.

- External Factors and Contextual Variations: The predictive models developed in this research focus on analyzing student performance within virtual learning environments. However, student performance can be influenced by various external factors, such as personal circumstances, socioeconomic background, or external events. These external factors may not be adequately captured or considered in the dataset used for training the models. It is crucial to recognize that the predictions derived from the models may not fully account for all contextual variations and external influences.

- Ethical Considerations: The use of predictive models in education raises ethical considerations related to privacy, bias, and fairness. The analysis and prediction of student performance involve handling sensitive data, such as demographic information and academic records. Ensuring the ethical use, storage, and protection of this data is paramount. Additionally, there is a need to continuously evaluate the models for biases and fairness, particularly to prevent potential discrimination based on factors such as gender, race, or socioeconomic status.

7.4 Future Work

The research problem pointed out in this study and the experiments conducted in this study suggest that more directions can be identified for investigating further solutions to investigate student performance prediction, and the class imbalance problem. This section lists the most important potential directions for future work.

- Investigating student performance prediction from a regression perspective: Regression models aim to predict a continuous output (e.g., a student's final grade) based on a set of input features (e.g., course engagement, demographic characteristics). Comparing the performance of ML models using regression techniques with those using classification techniques could provide insights into the best approach for predicting student performance in a VLE. Additionally, exploring the role of generative models (e.g., generative adversarial networks) in improving the performance of regression models could provide new approaches for this task.

- Using other datasets to investigate student performance: In addition to the OULAD and Coursera datasets used in this study, there are many other datasets available that could be used to investigate student performance in a VLE. For example, a dataset focused on students at risk of failing a course could be used to identify factors that contribute to this risk and to develop strategies for mitigating it. This could involve applying different ML models to the data and analyzing the results to identify patterns and trends.

- Investigating optimization algorithms to address the class imbalance problem: Optimization algorithms, such as genetic algorithms and simulated annealing, are a type of heuristic search algorithm that can be used to find solutions to complex problems. These algorithms could be used to address the class imbalance problem by identifying more effective approaches to balancing the data and improving the performance of ML models.

- Exploring different feature engineering techniques: Feature engineering involves identifying and selecting the most relevant features for a prediction task, as well as developing new features based on the available data. Different feature engineering techniques could be explored to see how they impact the performance of ML models in predicting student performance in a VLE.

- Investigating the impact of students' life circumstances: Student performance in a VLE can be influenced by a wide range of personal, social, and economic factors. Collecting and analyzing data on these factors could provide insights into how they impact student performance, and inform the development of more effective prediction models. This could involve examining the relationships between different variables and analyzing the results to identify patterns and trends.

Bibliography

[1] D. Fradkin and I. Muchnik, “Support Vector Machines for Classification,” *DIMACS series in discrete mathematics and theoretical computer science*, vol. 70, pp. 13–20, 2006.

[2] J. M. Crotty, “Distance Learning Has Been Around Since 1892, You Big MOOC,” section: Tech. [Online]. Available: https://www.forbes.com/sites/jamesmarshallcrotty/2012/11/14/distance-learning-has-been-around-since-1892-you-big-mooc/

[3] W. L. Bedwell and E. Salas, “Computer-based training: capitalizing on lessons learned,” *International Journal of Training and Development*, vol. 14, no. 3, pp. 239–249, 2010.

[4] B. Cope and M. Kalantzis, “A little history of e-learning: finding new ways to learn in the plato computer education system, 1959–1976,” *History of Education*, pp. 1–32, 2023.

[5] H. A. Dhaif, “Can computers teach languages,” in *English teaching forum*, vol. 27, no. 3, 1989, pp. 17–19.

[6] G. Bulman and R. W. Fairlie, “Technology and Education: Computers, Software, and the Internet,” in *Handbook of the Economics of Education*. Elsevier, 2016, vol. 5, pp. 239–280.

[7] A. Szymkowiak, B. Melović, M. Dabić, K. Jeganathan, and G. S. Kundi, “Information Technology and Gen Z: The Role of Teachers, the Internet, and Technology in the Education of Young People,” *Technology in Society*, vol. 65, p. 101565, 2021.

[8] M. Weller, “Twenty years of edtech,” *Educause Review Online*, vol. 53, no. 4, pp. 34–48, 2018.

[9] W. Al-Rahmi, A. Aldraiweesh, N. Yahaya, Y. B. Kamin, and A. M. Zeki, "Massive open online courses (moocs): Data on higher education," *Data in brief*, vol. 22, pp. 118–125, 2019.

[10] E. Pantò and A. Comas-Quinn, "The challenge of open education," *Journal of E-learning and Knowledge Society*, vol. 9, no. 1, pp. 11–22, 2013.

[11] M. Jantjies, T. Moodley, and R. Maart, "Experiential learning through virtual and augmented reality in higher education," in *Proceedings of the 2018 international conference on education technology management*, 2018, pp. 42–45.

[12] F. Gavilanes-Sagnay, E. Loza-Aguirre, D. Riofrío-Luzcando, and M. Segura-Morales, "A Systematic Literature Review of Indicators for the Understanding of Interactions in Virtual Learning Environments," in *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, 2018, pp. 596–600.

[13] P. Alves, L. Miranda, and C. Morais, "The Importance of Virtual Learning Environments in Higher Education," 2016, iSBN: 9781466688032 Pages: 404-425 Publisher: IGI Global. [Online]. Available: www.igi-global.com/chapter/the-importance-of-virtual-learning-environments-in-higher-education/142761

[14] J. Jani, R. Muszali, S. Nathan, M. S. ABDULLAH, and M. Shihah, "Blended Learning Approach Using Frog Vle Platform Towards Students' Achievement In Teaching Games For Understanding," *Journal of Applied and Fundamental Sciences*, vol. 10, pp. 1131–1151, Jan. 2018.

[15] C. Torres Martín, C. Acal, M. El Honrani, and Á. C. Mingorance Estrada, "Impact on the Virtual Learning Environment Due to COVID-19," *Sustainability*, vol. 13, no. 2, p. 582, 2021.

[16] S. Kurbakova, Z. Volkova, and A. Kurbakov, "Virtual Learning and Educational Environment: New Opportunities and Challenges Under the COVID-19 Pandemic," in *2020 The 4th International Conference on Education and Multimedia Technology*, 2020, pp. 167–171.

[17] I. Chatziralli, C. V. Ventura, S. Touhami, R. Reynolds, M. Nassisi, T. Weinberg, K. Pakzad-Vaezi, D. Anaya, M. Mustapha, A. Plant, M. Yuan, and A. Loewenstein, "Transforming Ophthalmic Education into Virtual Learning During COVID-19 Pandemic: A Global Perspective," *Eye*, pp.

1–8, Jul. 2020, publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s41433-020-1080-0

[18] A. H. Darusman and Y. Omar, "Enhancing Student Engagement in VLE Platform: Student Perceptions Towards Programming Course Learning Resources," *PSYCHOLOGY AND EDUCATION*, vol. 58, no. 1, pp. 5607–5612, 2021.

[19] C. Vegega, P. Pytel, and M. F. Pollo-Cattaneo, "Application of the Requirements Elicitation Process for the Construction of Intelligent System-Based Predictive Models in the Education Area," in *Applied Informatics*, ser. Communications in Computer and Information Science, H. Florez, M. Leon, J. M. Diaz-Nafria, and S. Belli, Eds. Cham: Springer International Publishing, 2019, pp. 43–58.

[20] R. López-Chila, J. Llerena-Izquierdo, and N. Sumba-Nacipucha, "Using Exam-View to Create Questionnaires for Online Evaluation in VLEs," in *2021 Second International Conference on Information Systems and Software Technologies (ICI2ST)*. IEEE, 2021, pp. 3–9.

[21] R. A Sa'di, A. Abdelraziq, and T. A Sharadgah, "E-Assessment at Jordan's Universities in the Time of the COVID-19 Lockdown: Challenges and Solutions," *Arab World English Journal (AWEJ) Special Issue on Covid*, vol. 19, 2021.

[22] P. E. Román, E. O. Torres, and R. L. Hernández, "Virtual Learning Environments as a Continuous Assessment Tool in University Students," in *Educational Technology and the New World of Persistent Learning*. IGI Global, 2019, pp. 229–252.

[23] A. Hellas, P. Ihantola, A. Petersen, V. V. Ajanovski, M. Gutica, T. Hynninen, A. Knutas, J. Leinonen, C. Messom, and S. N. Liao, "Predicting Academic Performance: A Systematic Literature Review," in *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, ser. ITiCSE 2018 Companion. New York, NY, USA: Association for Computing Machinery, Jul. 2018, pp. 175–199. [Online]. Available: https://doi.org/10.1145/3293881.3295783

[24] A. M. Shahiri, W. Husain, and N. A. Rashid, "A Review on Predicting Student's Performance Using Data Mining Techniques," *Procedia Computer Science*, vol. 72, pp. 414–422, Jan. 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050915036182

[25] A. S. J. Abu Hammad, “Mining Educational Data to Analyze Students’ Performance (A Case with University College of Science and Technology Students),” *Central European Researchers Journal*, vol. 4, no. 2, 2018.

[26] K. C. Harper, K. Chen, and D. C. Yen, “Distance Learning, Virtual Classrooms, and Teaching Pedagogy in the Internet Environment,” *Technology in Society*, vol. 26, no. 4, pp. 585–598, Nov. 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0160791X04000545

[27] A. Al-Azawei, M. Abdul, and A. Al-Masoudy, “Predicting Learners’ Performance in Virtual Learning Environment (VLE) Based on Demographic, Behavioral and Engagement Antecedents,” *International Journal of Emerging Technologies in Learning (iJET)*, vol. 15, May 2020.

[28] H. Mellar, R. Peytcheva-Forsyth, S. Kocdar, A. Karadeniz, and B. Yovkova, “Addressing Cheating in e-Assessment Using Student Authentication and Authorship Checking Systems: Teachers’ Perspectives,” *International Journal for Educational Integrity*, vol. 14, p. 2, Feb. 2018.

[29] S. D. Johnson, S. R. Aragon, and N. Shaik, “Comparative analysis of learner satisfaction and learning outcomes in online and face-to-face learning environments,” *Journal of interactive learning research*, vol. 11, no. 1, pp. 29–49, 2000.

[30] M. L. Williams, K. Paprock, and B. Covington, *Distance learning: The essential guide*. Sage Publications, 1998.

[31] A. E. Udokang and M. O. Ogundele, “Comparative study of distance learning and conventional system on the teachers’ job performance in kwara state,” *Procedia-social and behavioral sciences*, vol. 141, pp. 1192–1195, 2014.

[32] R. Conijn, A. V. d. Beemt, and P. Cuijpers, “Predicting Student Performance in a Blended MOOC,” *Journal of Computer Assisted Learning*, vol. 34, no. 5, pp. 615–628, 2018. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/jcal.12270

[33] M. Hussain, W. Zhu, W. Zhang, and S. M. R. Abidi, “Student Engagement Predictions in an e-Learning System and Their Impact on Student Course Assessment Scores,” Oct. 2018, iSSN: 1687-5265 Pages: e6347186 Publisher: Hindawi Volume: 2018. [Online]. Available: https://www.hindawi.com/journals/cin/2018/6347186/

[34] C. A. Boulton, C. Kent, and H. T. P. Williams, "Virtual Learning Environment Engagement and Learning Outcomes at A 'Bricks-and-Mortar' University," *Computers & Education*, vol. 126, pp. 129–142, Nov. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S036013151830174X

[35] A. Çebi and T. Güyer, "Students' Interaction Patterns in Different Online Learning Activities and their Relationship with Motivation, Self-Regulated Learning Strategy and Learning Performance," *Education and Information Technologies*, vol. 25, no. 5, pp. 3975–3993, Sep. 2020. [Online]. Available: https://doi.org/10.1007/s10639-020-10151-1

[36] D. Moher, L. Shamseer, M. Clarke, D. Ghersi, A. Liberati, M. Petticrew, P. Shekelle, L. A. Stewart *et al.*, "Preferred Reporting Items for Systematic Review and Meta-Analysis Protocols (PRISMA-P) 2015 Statement," *Systematic reviews*, vol. 4, no. 1, p. 1, 2015, publisher: Springer.

[37] J. Bravo-Agapito, S. J. Romero, and S. Pamplona, "Early Prediction of Undergraduate Student's Academic Performance in Completely Online Learning: A Five-Year Study," *Computers in Human Behavior*, vol. 115, p. 106595, 2021.

[38] J. Sneha and G. Nagaraja, "Virtual learning environments-a survey," *arXiv preprint arXiv:1402.2404*, 2014.

[39] M. Piotrowski, "What is an e-learning platform?" in *Learning management system technologies and software solutions for online teaching: Tools and applications*. IGI Global, 2010, pp. 20–36.

[40] M. Tahir, G. Zaman, and S. Shah, "Using Caputo-fabrizio Derivative for the Transmission of Mathematical Model Epidemic Corona Virus," *SeMA Journal*, vol. 78, no. 1, pp. 119–136, 2021.

[41] S. Appana, "A Review of Benefits and Limitations of Online Learning in the Context of the Student, the Instructor and the Tenured Faculty," *International Journal on E-learning*, vol. 7, no. 1, pp. 5–22, 2008.

[42] M. Tahir, G. Zaman, and S. I. A. Shah, "Evaluation and Control Estimation Strategy for Three Acting Play Diseases with Six Control Variables," *Cogent Mathematics & Statistics*, vol. 7, no. 1, p. 1805871, 2020.

[43] D. Hijazi and A. AlNatour, “Online Learning Challenges Affecting Students of English in an Efl Context During Covid-19 Pandemic,” *International Journal of Education and Practice*, vol. 9, no. 2, pp. 379–395, 2021.

[44] D. T. H. Dung, “The Advantages and Disadvantages of Virtual Learning,” *IOSR Journal of Research & Method in Education*, vol. 10, no. 3, pp. 45–48, 2020.

[45] C. Heaton-Shrestha, S. May, and L. Burke, “Student Retention in Higher Education: What Role for Virtual Learning Environments?” *Journal of Further and Higher education*, vol. 33, no. 1, pp. 83–92, 2009.

[46] E. Lacka, T. Wong, and M. Y. Haddoud, “Can Digital Technologies Improve Students’ Efficiency? Exploring the Role of Virtual Learning Environment and Social Media use in Higher Education,” *Computers & Education*, vol. 163, p. 104099, 2021.

[47] C. Müller and T. Mildenberger, “Facilitating Flexible Learning by Replacing Classroom Time with an Online Learning Environment: A Systematic Review of Blended Learning in Higher Education,” *Educational Research Review*, vol. 34, p. 100394, 2021.

[48] A. Phelps and D. Vlachopoulos, “Successful Transition to Synchronous Learning Environments in Distance Education: A Research on Entry-Level Synchronous Facilitator Competencies,” *Education and Information Technologies*, vol. 25, no. 3, pp. 1511–1527, 2020.

[49] Z. Almahasees, K. Mohsen, and M. O. Amin, “Faculty’s and Students’ Perceptions of Online Learning During COVID-19,” in *Frontiers in Education*, vol. 6. Frontiers Media SA, 2021.

[50] A.-M. Korhonen, S. Ruhalahti, and M. Veermans, “The Online Learning Process and Scaffolding in Student Teachers’ Personal Learning Environments,” *Education and Information Technologies*, vol. 24, no. 1, pp. 755–779, 2019.

[51] A. L. McMullen, V. A. Bursuc, C. Doval, S. Grant, J. D. Grossberg, I. M. Jones, and S. Willey, “Strategies for Effective Online Teaching in Higher Education.” *Business Education Innovation Journal*, vol. 12, no. 2, 2020.

[52] A. Y. Huang, O. H. Lu, J. C. Huang, C. Yin, and S. J. Yang, “Predicting Students’ Academic Performance by Using Educational Big Data and Learning Analytics: Evaluation of Classification Methods and Learning Logs,” *Interactive Learning Environments*, vol. 28, no. 2, pp. 206–230, 2020.

[53] M. Kokoç and A. Altun, “Effects of Learner Interaction with Learning Dashboards on Academic Performance in an e-Learning Environment,” *Behaviour & Information Technology*, vol. 40, no. 2, pp. 161–175, 2021.

[54] Á. F. Agudo-Peregrina, S. Iglesias-Pradas, M. Á. Conde-González, and Á. Hernández-García, “Can We Predict Success from Log Data in VLEs? Classification of Interactions for Learning Analytics and their Relation with Performance in VLE-Supported F2F and Online Learning,” *Computers in human behavior*, vol. 31, pp. 542–550, 2014.

[55] B. Maraza-Quispe, O. M. Alejandro-Oviedo, W. Choquehuanca-Quispe, N. Caytuiro-Silva, and J. Herrera-Quispe, “Towards a Standardization of Learning Behavior Indicators in Virtual Environments,” *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 11, 2020.

[56] H. G. Bilgiç, “Evaluation of the Emergency Remote Teaching Process During the COVID-19 Pandemic: Implications Based on Students’ Views,” *Asian Journal of Distance Education*, vol. 16, no. 2, 2021.

[57] H. J. Kim, A. J. Hong, and H.-D. Song, “The Roles of Academic Engagement and Digital Readiness in Students’ Achievements in University e-Learning Environments,” *International Journal of Educational Technology in Higher Education*, vol. 16, no. 1, pp. 1–18, 2019.

[58] G.-J. Hwang, S.-Y. Wang, and C.-L. Lai, “Effects of a Social Regulation-Based Online Learning Framework on Students’ Learning Achievements and Behaviors in Mathematics,” *Computers & Education*, vol. 160, p. 104031, 2021.

[59] E. M. Queiroga, J. L. Lopes, K. Kappel, M. Aguiar, R. M. Araújo, R. Munoz, R. Villarroel, and C. Cechinel, “A Learning Analytics Approach to Identify Students At Risk of Dropout: A Case Study with a Technical Distance Education Course,” *Applied Sciences*, vol. 10, no. 11, p. 3998, 2020.

[60] A. Francescucci and L. Rohani, “Exclusively Synchronous Online (VIRI) Learning: The Impact on Student Performance and Engagement Outcomes,” *Journal of marketing Education*, vol. 41, no. 1, pp. 60–69, 2019.

[61] T. Calders and M. Pechenizkiy, “Introduction to the Special Section on Educational Data Mining,” *Acm Sigkdd Explorations Newsletter*, vol. 13, no. 2, pp. 3–6, 2012.

[62] A. V. Manjarres, L. G. M. Sandoval, and M. J. S. Suárez, "Data Mining Techniques Applied in Educational Environments: Literature Review," *Digital Education Review*, vol. 0, no. 33, pp. 235–266, Jun. 2018, number: 33. [Online]. Available: https://revistes.ub.edu/index.php/der/article/view/18067

[63] A. Dutt, M. A. Ismail, and T. Herawan, "A Systematic Review on Educational Data Mining," *Ieee Access*, vol. 5, pp. 15 991–16 005, 2017.

[64] M. Hussain, W. Zhu, W. Zhang, and S. M. R. Abidi, "Student engagement predictions in an e-learning system and their impact on student course assessment scores," *Computational intelligence and neuroscience*, vol. 2018, 2018.

[65] O. Casquero, R. Ovelar, J. Romo, M. Benito, and M. Alberdi, "Students' Personal Networks in Virtual and Personal Learning Environments: A Case Study in Higher Education Using Learning Analytics Approach," *Interactive Learning Environments*, vol. 24, no. 1, pp. 49–67, 2016.

[66] P. Alves, L. Miranda, and C. Morais, "The influence of virtual learning environments in students' performance," *Universal Journal of Educational Research*, vol. 5, no. 3, pp. 517–527, 2017.

[67] D. Buenaño-Fernandez, W. Villegas-CH, and S. Luján-Mora, "The Use of Tools of Data Mining to Decision Making in Engineering Education — A Systematic Mapping Study," *Computer Applications in Engineering Education*, vol. 27, no. 3, pp. 744–758, 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cae.22100

[68] A. I. Adekitan and O. Salau, "The Impact of Engineering Students' Performance in the First Three Years on their Graduation Result using Educational Data Mining," *Heliyon*, vol. 5, no. 2, p. e01250, 2019.

[69] N. Tomasevic, N. Gvozdenovic, and S. Vranes, "An Overview and Comparison of Supervised Data Mining Techniques for Student Exam Performance Prediction," *Computers & Education*, vol. 143, p. 103676, Jan. 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360131519302295

[70] S. Hussain, Z. Muhsen, Y. Salal, P. Theodorou, F. Kurtoğlu, and G. Hazarika, "Prediction Model on Student Performance based on Internal Assessment us-

ing Deep Learning," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 14, p. 4, Apr. 2019.

[71] S. Ajibade, N. Ahmad, and S. M. Shamsuddin, "Educational Data Mining: Enhancement of Student Performance model using Ensemble Methods," *IOP Conference Series: Materials Science and Engineering*, vol. 551, p. 012061, Aug. 2019.

[72] Royal Society (Great Britain), *Machine Learning: the Power and Promise of Computers that Learn by Example*. The Royal Society, 2017, oCLC: 1016323791.

[73] J. Brockman, *The next fifty years: Science in the first half of the twenty-first century*. Vintage, 2007.

[74] R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, *Machine Learning: An Artificial Intelligence approach*. Springer Science & Business Media, 2013.

[75] C. Korkmaz and A.-P. Correia, "A Review of Research on Machine Learning in Educational Technology," *Educational Media International*, vol. 56, no. 3, pp. 250–267, 2019.

[76] X. Chen, H. Xie, D. Zou, and G.-J. Hwang, "Application and Theory Gaps During the Rise of Artificial Intelligence in Education," *Computers and Education: Artificial Intelligence*, vol. 1, p. 100002, 2020.

[77] X. Cheng, J. Sun, and A. Zarifis, "Artificial Intelligence and Deep Learning in Educational Technology Research and Practice," pp. 1653–1656, 2020.

[78] T. Bikmukhametov and J. Jäschke, "First Principles and Machine Learning Virtual Flow Metering: A Literature Review," *Journal of Petroleum Science and Engineering*, vol. 184, p. 106487, 2020.

[79] A. Elbadrawy, R. Studham, and G. Karypis, "Personalized Multi-Regression Models for Predicting Students' Performance in Course Activities," in *5th International Learning Analytics & Knowledge Conference (LAK15)*, Mar. 2015.

[80] M. Yee-King, A. Grimalt-Reynes, and M. d'Inverno, "Predicting Student Grades from Online, Collaborative Social Learning Metrics Using K-NN." in *EDM*, 2016, pp. 654–655.

[81] H. Al-Shehri, A. Al-Qarni, L. Al-Saati, A. Batoaq, H. Badukhen, S. Alrashed, J. Alhiyafi, and S. O. Olatunji, "Student Performance Prediction using Support Vector Machine and K-Nearest Neighbor," in *2017 IEEE 30th canadian conference on electrical and computer engineering (CCECE)*, 2017, pp. 1–4, tex.organization: IEEE.

[82] Z. Iqbal, J. Qadir, A. N. Mian, and F. Kamiran, "Machine Learning Based Student Grade Prediction: A Case Study," *arXiv preprint arXiv:1708.08744*, 2017.

[83] H. Heuer and A. Breiter, "Student Success Prediction and the Trade-Off Between Big Data and Data Minimization," *DeLFI 2018-Die 16. E-Learning Fachtagung Informatik*, 2018, publisher: Gesellschaft für Informatik eV.

[84] B. Sekeroglu, K. Dimililer, and K. Tuncal, "Student Performance Prediction and Classification Using Machine Learning Algorithms," in *ICEIT 2019: Proceedings of the 2019 8th International Conference on Educational and Information Technology*, Mar. 2019, pp. 7–11.

[85] M. El Fouki, N. Aknin, and K. E. El Kadiri, "Multidimensional Approach Based on Deep Learning to Improve the Prediction Performance of DNN Models," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 14, no. 02, pp. 30–41, 2019.

[86] D. Hooshyar, M. Pedaste, Y. Yang, L. Malva, G.-J. Hwang, M. Wang, H. Lim, and D. Delev, "From Gaming to Computational Thinking: An Adaptive Educational Computer Game-Based Learning Approach," *Journal of Educational Computing Research*, p. 0735633120965919, Oct. 2020, publisher: SAGE Publications Inc. [Online]. Available: https://doi.org/10.1177/0735633120965919

[87] H. Waheed, S.-U. Hassan, N. R. Aljohani, J. Hardman, S. Alelyani, and R. Nawaz, "Predicting Academic Performance of Students from VLE Big Data Using Deep Learning Models," *Computers in Human Behavior*, vol. 104, p. 106189, 2020, publisher: Elsevier.

[88] M. Tahir, S. I. A. Shah, G. Zaman, and T. Khan, "Stability Behaviour of Mathematical Model MERS Corona Virus Spread in Population," *Filomat*, vol. 33, no. 12, pp. 3947–3960, 2019.

[89] C. M. Paulo Alves, Luísa Miranda, "The Influence of Virtual Learning Environments in Students' Performance," *Universal Journal of Educational Research*, vol. 5, no. 3, pp. 517–527, 2017.

[90] M. Tahir, N. Anwar, S. I. A. Shah, and T. Khan, "Modeling and Stability Analysis of Epidemic Expansion Disease Ebola Virus with Implications Prevention in Population," *Cogent Biology*, vol. 5, no. 1, p. 1619219, 2019.

[91] M. Z. Wong, K. Kunii, M. Baylis, W. H. Ong, P. Kroupa, and S. Koller, "Synthetic Dataset Generation for Object-to-Model Deep Learning in Industrial Applications," *PeerJ Computer Science*, vol. 5, p. e222, 2019.

[92] R. Heyburn, R. R. Bond, M. Black, M. Mulvenna, J. Wallace, D. Rankin, and B. Cleland, "Machine Learning Using Synthetic and Real Data: Similarity of Evaluation Metrics for Different Healthcare Datasets and for Different Algorithms," in *Data Science and Knowledge Engineering for Sensing Decision Support: Proceedings of the 13th International FLINS Conference (FLINS 2018)*. World Scientific, 2018, pp. 1281–1291.

[93] T. N. Arvanitis, S. White, S. Harrison, R. Chaplin, and G. Despotou, "A Method for Machine Learning Generation of Realistic Synthetic Datasets for Validating Healthcare Applications," *Health Informatics Journal*, vol. 28, no. 2, p. 14604582221077000, 2022.

[94] Y. Lou, S. Li, N. Liu, and R. Liu, "Seismic Volumetric Dip Estimation Via a Supervised Deep Learning Model by Integrating Realistic Synthetic Data Sets," *Journal of Petroleum Science and Engineering*, vol. 218, p. 111021, 2022.

[95] B. K. Baradwaj and S. Pal, "Mining Educational Data to Analyze Students' Performance," *arXiv preprint arXiv:1201.3417*, 2012.

[96] X. Guo, Y. Yin, C. Dong, G. Yang, and G. Zhou, "On the Class Imbalance Problem," in *2008 Fourth international conference on natural computation*, vol. 4. IEEE, 2008, pp. 192–201.

[97] R. Barandela, J. S. Sánchez, V. Garcıa, and E. Rangel, "Strategies for Learning in Class Imbalance Problems," *Pattern Recognition*, vol. 36, no. 3, pp. 849–851, 2003.

[98] C. Herodotou, B. Rienties, A. Boroowa, Z. Zdrahal, and M. Hlosta, "A large-scale implementation of predictive learning analytics in higher education:

The teachers' role and perspective," *Educational Technology Research and Development*, vol. 67, pp. 1273–1306, 2019.

[99] C. Herodotou, G. Naydenova, A. Boroowa, A. Gilmour, and B. Rienties, "How can predictive learning analytics and motivational interventions increase student retention and enhance administrative support in distance education?" *Journal of Learning Analytics*, vol. 7, no. 2, pp. 72–83, 2020.

[100] C. E. Calvert, "Developing a model and applications for probabilities of student success: a case study of predictive analytics," *Open Learning: The Journal of Open, Distance and e-Learning*, vol. 29, no. 2, pp. 160–173, 2014.

[101] C. Cechinel, M. D. F. Dos Santos, C. Barrozo, J. E. Schardosim, E. de Vila, V. Ramos, T. Primo, R. Munoz, and E. M. Queiroga, "A learning analytics dashboard for moodle: Implementing machine learning techniques to early detect students at risk of failure," in *2021 XVI Latin American Conference on Learning Technologies (LACLO)*. IEEE, 2021, pp. 130–136.

[102] O. K. Xin and D. Singh, "Development of learning analytics dashboard based on moodle learning management system," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 7, 2021.

[103] S. Islam and H. Mahmud, "An intelligence learner management system using learning analytics and machine learning," in *2020 12th International Conference on Education Technology and Computers*, 2020, pp. 120–125.

[104] T. Susnjak, G. S. Ramaswami, and A. Mathrani, "Learning analytics dashboard: a tool for providing actionable insights to learners," *International Journal of Educational Technology in Higher Education*, vol. 19, no. 1, p. 12, 2022.

[105] B. Rienties, Q. Nguyen, W. Holmes, and K. Reedy, "A review of ten years of implementation and research in aligning learning design with learning analytics at the open university uk," *Interaction Design and Architecture (s)*, vol. 33, pp. 134–154, 2017.

[106] B. Rienties and L. Toetenel, "The impact of learning design on student behaviour, satisfaction and performance: A cross-institutional comparison across 151 modules," *Computers in Human Behavior*, vol. 60, pp. 333–341, 2016.

[107] L. Toetenel and B. Rienties, "Analysing 157 learning designs using learning analytic approaches as a means to evaluate the impact of pedagogical decision making," *British Journal of Educational Technology*, vol. 47, no. 5, pp. 981–992, 2016.

[108] B. Rienties, L. Toetenel, and A. Bryan, "" scaling up" learning design: impact of learning design activities on lms behavior and performance," in *Proceedings of the Fifth International Conference on Learning Analytics and Knowledge*, 2015, pp. 315–319.

[109] B. Rienties, W. Kaper, K. Struyven, D. Tempelaar, L. Van Gastel, S. Vrancken, M. Jasińska, and E. Virgailaitė-Mečkauskaitė, "A review of the role of information communication technology and course design in transitional education practices," *Interactive Learning Environments*, vol. 20, no. 6, pp. 563–581, 2012.

[110] J. Kuzilek, M. Hlosta, and Z. Zdrahal, "Open University Learning Analytics Dataset," *Scientific data*, vol. 4, p. 170171, 2017.

[111] Y. B. Rajabalee, M. I. Santally, and F. Rennie, "Modeling Students' Performances in Activity-Based e-Learning From a Learning Analytics Perspective: Implications and Relevance for Learning Design," *International Journal of Distance Education Technologies (IJDET)*, vol. 18, no. 4, pp. 71–93, 2020.

[112] R. S. Sutton and A. G. Barto, *Reinforcement Learning, Second Edition: An Introduction*. MIT Press, Nov. 2018, google-Books-ID: uWV0DwAAQBAJ.

[113] E. Alpaydin, *Introduction to Machine Learning*. MIT press, 2020.

[114] E. Fix, *Discriminatory Analysis: Onparametric Discrimination, Consistency Properties*. USAF School of Aviation Medicine, 1951.

[115] O. Anava and K. Levy, "k*-nearest Neighbors: From Global to Local," *Advances in neural information processing systems*, vol. 29, 2019.

[116] T. Cover and P. Hart, "Nearest Neighbor Pattern Classification," *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21–27, 1967.

[117] M. E. Hellman, "The Nearest Neighbor Classification Rule with a Reject Option," *IEEE Transactions on Systems Science and Cybernetics*, vol. 6, no. 3, pp. 179–185, 1970.

[118] K. Fukunaga and L. Hostetler, "K-Nearest-Neighbor Bayes-Risk Estimation," *IEEE Transactions on Information Theory*, vol. 21, no. 3, pp. 285–293, 1975.

[119] S. A. Dudani, "The Distance-Weighted k-Nearest-Neighbor Rule," *IEEE Transactions on Systems, Man, and Cybernetics*, no. 4, pp. 325–327, 1976.

[120] S. Bermejo and J. Cabestany, "Adaptive Soft k-Nearest-Neighbour Classifiers," *Pattern Recognition*, vol. 33, no. 12, pp. 1999–2005, 2000.

[121] J. M. Keller, M. R. Gray, and J. A. Givens, "A fuzzy k-nearest neighbor algorithm," *IEEE transactions on systems, man, and cybernetics*, no. 4, pp. 580–585, 1985.

[122] J. Heo, "The Distance Formula for the Derivation Problem," *Indian Journal of Pure and Applied Mathematics*, vol. 46, pp. 489–494, 2020.

[123] D. Coomans and D. L. Massart, "Alternative K-Nearest Neighbour Rules in Supervised Pattern Recognition: Part 1. k-Nearest Neighbour Classification by Using Alternative Voting Rules," *Analytica Chimica Acta*, vol. 136, pp. 15–27, 1982.

[124] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A Training Algorithm for Optimal Margin Classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*, 1992, pp. 144–152.

[125] Y. Pu, D. B. Apel, and H. Xu, "Rockburst Prediction in Kimberlite with Unsupervised Learning Method and Support Vector Classifier," *Tunnelling and Underground Space Technology*, vol. 90, pp. 12–18, 2019.

[126] X. Huang, L. Shi, and J. A. Suykens, "Support Vector Machine Classifier with Pinball Loss," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 5, pp. 984–997, 2013.

[127] C. Yuan and D. Casasent, "A Novel Support Vector Classifier with Better Rejection Performance," in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, vol. 1. IEEE, 2003, pp. I–I.

[128] I. Steinwart and A. Christmann, *Support Vector Machines*. Springer Science & Business Media, 2008.

[129] S. Suthaharan, "Decision Tree Learning," in *Machine Learning Models and Algorithms for Big Data Classification*. Springer, 2016, pp. 237–269.

[130] D. V. Patil and R. Bichkar, "Issues in Optimization of Decision Tree Learning: A Survey," *International Journal of Applied Information Systems (IJAIS), New York, USA*, vol. 3, no. 5, 2012.

[131] D. Che, Q. Liu, K. Rasheed, and X. Tao, "Decision Tree and Ensemble Learning Algorithms with their Applications in Bioinformatics," *Software tools and algorithms for biological systems*, pp. 191–199, 2011.

[132] A. E. Khedr, A. M. Idrees, and A. I. El Seddawy, "Enhancing Iterative Dichotomiser 3 Algorithm for Classification Decision Tree," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 6, no. 2, pp. 70–79, 2016.

[133] M. van Diepen and P. H. Franses, "Evaluating Chi-Squared Automatic Interaction detection," *Information Systems*, vol. 31, no. 8, pp. 814–831, 2006.

[134] M. A. Razi and K. Athappilly, "A Comparative Predictive Analysis of Neural Networks (NNs), Nonlinear Regression and Classification and Regression Tree (CART) Models," *Expert systems with applications*, vol. 29, no. 1, pp. 65–74, 2005.

[135] J. H. Friedman, "Multivariate Adaptive Regression Splines," *The annals of statistics*, pp. 1–67, 1991.

[136] R. A. Disha and S. Waheed, "Performance Analysis of Machine Learning Models for Intrusion Detection System Using Gini Impurity-Based Weighted Random Forest (GIWRF) Feature Selection Technique," *Cybersecurity*, vol. 5, no. 1, pp. 1–22, 2022.

[137] Y. Yuan, L. Wu, and X. Zhang, "Gini-Impurity Index Analysis," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 3154–3169, 2021.

[138] M. Sarstedt and E. Mooi, "Regression Analysis," in *A Concise Guide to Market Research*. Springer, 2019, pp. 209–256.

[139] W. Yao and L. Li, "A New Regression Model: Modal Linear Regression," *Scandinavian Journal of Statistics*, vol. 41, no. 3, pp. 656–671, 2014.

[140] X. Su, X. Yan, and C.-L. Tsai, “Linear Regression,” *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 4, no. 3, pp. 275–294, 2012.

[141] G. P. Gerdan and R. E. Deakin, “Transforming Cartesian Coordinates X, Y, Z to Geographical Coordinates ϕ, λ, h,” *Australian surveyor*, vol. 44, no. 1, pp. 55–63, 1999.

[142] B. Russell, *Introduction to Mathematical Philosophy*. Taylor & Francis, 2022.

[143] E. H. Houssein, M. A. Mahdy, D. Shebl, A. Manzoor, R. Sarkar, and W. M. Mohamed, “An Efficient Slime Mould Algorithm for Solving Multi-Objective Optimization Problems,” *Expert Systems with Applications*, vol. 187, p. 115870, 2022.

[144] M. Karimi-Mamaghan, M. Mohammadi, P. Meyer, A. M. Karimi-Mamaghan, and E.-G. Talbi, “Machine Learning at the Service of Meta-heuristics for Solving Combinatorial Optimization Problems: A State-of-the-Art,” *European Journal of Operational Research*, vol. 296, no. 2, pp. 393–422, 2022.

[145] G. P. Anila. M, “Least Square Regression for Prediction Problems in Machine Learning using R,” *International Journal of Engineering and Technology*, vol. 7, no. 3, pp. 3–12, 2022.

[146] M. O. A. Abu-Shawiesh and N. Saeed, “Monitoring Process Variability using Decile Mean Standard Deviation,” *Quality and Reliability Engineering International*, vol. 38, no. 1, pp. 501–515, 2022.

[147] A. Abraham, “Artificial Neural Networks,” *Handbook of measuring system design*, 2005.

[148] K. Mehrotra, C. K. Mohan, and S. Ranka, *Elements of Artificial Neural Networks*. MIT press, 1997.

[149] A. Krogh, “What Are Artificial Neural Networks?” *Nature biotechnology*, vol. 26, no. 2, pp. 195–197, 2008.

[150] I. A. Basheer and M. Hajmeer, “Artificial Neural Networks: Fundamentals, Computing, Design, and Application,” *Journal of microbiological methods*, vol. 43, no. 1, pp. 3–31, 2000.

[151] H. Mansourifar and W. Shi, “Deep Synthetic Minority Over-Sampling Technique,” *arXiv preprint arXiv:2003.09788*, 2020.

[152] D. Saxena and J. Cao, “Generative Adversarial Networks (GANs) Challenges, Solutions, and Future Directions,” *ACM Computing Surveys (CSUR)*, vol. 54, no. 3, pp. 1–42, 2021.

[153] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, “Generative Adversarial Networks: An Overview,” *IEEE signal processing magazine*, vol. 35, no. 1, pp. 53–65, 2018.

[154] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F.-Y. Wang, “Generative Adversarial Networks: Introduction and Outlook,” *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 4, pp. 588–598, 2017.

[155] M. Sokolova and G. Lapalme, “A Systematic Analysis of Performance Measures for Classification Tasks,” *Information processing & management*, vol. 45, no. 4, pp. 427–437, 2009.

[156] S. Hussain, M. W. Mustafa, T. A. Jumani, S. K. Baloch, H. Alotaibi, I. Khan, and A. Khan, “A Novel Feature Engineered-CatBoost-based Supervised Machine Learning Framework for Electricity Theft Detection,” *Energy Reports*, vol. 7, pp. 4425–4436, 2021.

[157] A. Bayani, A. Hosseini, F. Asadi, B. Hatami, K. Kavousi, M. Aria, and M. R. Zali, “Identifying Predictors of Varices Grading in Patients with Cirrhosis Using Ensemble Learning,” *Clinical Chemistry and Laboratory Medicine (CCLM)*, vol. 60, no. 12, pp. 1938–1945, 2022.

[158] M. Massaoudi, H. Abu-Rub, S. S. Refaat, I. Chihi, and F. S. Oueslati, “An Effective Ensemble Learning Approach-based Grid Stability Assessment and Classification,” in *2021 IEEE Kansas Power and Energy Conference (KPEC)*. IEEE, 2021, pp. 1–6.

[159] D. Hooshyar, M. Pedaste, and Y. Yang, “Mining Educational Data to Predict Students’ Performance through Procrastination Behavior,” *Entropy*, vol. 22, no. 1, p. 12, Jan. 2020, number: 1 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/1099-4300/22/1/12

[160] F. Thabtah, S. Hammoud, F. Kamalov, and A. Gonsalves, “Data Imbalance in Classification: Experimental Evaluation,” *Information Sciences*, vol. 513, pp. 429–441, 2020.

[161] G. Figueroa, Y.-S. Chen, N. Avila, and C.-C. Chu, "Improved Practices in Machine Learning Algorithms for NTL Detection with Imbalanced Data," in *2017 IEEE Power & Energy Society General Meeting*. IEEE, 2017, pp. 1–5.

[162] A. P. Cavalcanti, A. Diego, R. F. Mello, K. Mangaroska, A. Nascimento, F. Freitas, and D. Gašević, "How Good is My Feedback? A Content Analysis of Written Feedback," in *Proceedings of the tenth international conference on learning analytics & knowledge*, 2020, pp. 428–437.

[163] N. Nasir, A. Kansal, O. Alshaltone, F. Barneih, M. Sameer, A. Shanableh, and A. Al-Shamma'a, "Water Quality Classification using Machine Learning Algorithms," *Journal of Water Process Engineering*, vol. 48, p. 102920, 2022.

[164] V. Kumar, G. S. Lalotra, P. Sasikala, D. S. Rajput, R. Kaluri, K. Lakshmanna, M. Shorfuzzaman, A. Alsufyani, and M. Uddin, "Addressing Binary Classification Over Class Imbalanced Clinical Datasets using Computationally Intelligent Techniques," in *Healthcare*, vol. 10, no. 7. MDPI, 2022, p. 1293.

[165] S. Belarouci and M. A. Chikh, "Medical Imbalanced Data Classification," *Adv. Sci. Technol. Eng. Syst. J*, vol. 2, no. 3, pp. 116–124, 2017.

[166] R. Ghorbani and R. Ghousi, "Comparing Different Resampling Methods in Predicting Students' Performance using Machine Learning Techniques," *IEEE Access*, vol. 8, pp. 67 899–67 911, 2020.

[167] H. Patel, D. S. Rajput, O. P. Stan, and L. C. Miclea, "A New Fuzzy Adaptive Algorithm to Classify Imbalanced Data," *CMC-COMPUTERS MATERIALS & CONTINUA*, vol. 70, no. 1, pp. 73–89, 2022.

[168] G. E. Batista, R. C. Prati, and M. C. Monard, "A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data," *ACM SIGKDD explorations newsletter*, vol. 6, no. 1, pp. 20–29, 2004.

[169] M. Hlosta, Z. Zdrahal, and J. Zendulka, "Ouroboros: Early Identification of At-risk Students without Models Based on Legacy Data," in *Proceedings of the seventh international learning analytics & knowledge conference*, 2017, pp. 6–15.

[170] H.-C. Chen, E. Prasetyo, S. S. Kusumawardani, S.-S. Tseng, T.-L. Kung, K.-Y. Wang *et al.*, "Learning Performance Prediction with Imbalanced Virtual Learning Environment Students' Interactions Data," in *International Conference on*

Innovative Mobile and Internet Services in Ubiquitous Computing. Springer, 2021, pp. 330–340.