Ministry of Science and Higher Education of the Republic of Kazakhstan

SDU University

Dana Kalekes

Analysis of students' behavior and progress on Learning Management System using Machine Learning

THESIS

Presented in Partial Fulfilment for the

Degree of Master of Technical Science in Computer Science

(degree code: 7M06102)

Department of Computer Science

Faculty of Engineering and Natural Sciences

Supervisor: **Kamila Orynbekova**

Kaskelen, June 2024

SDU University
Faculty of Engineering and Natural Sciences
Department of Computer Science

Dean of Faculty of Engineering and Natural Sciences

Assistant Professor, PhD. Akhmedov Ramis

_____

« _____» _____ 2024

Topic of the thesis:

Analysis of students' behavior and progress on Learning Management System using Machine Learning

Thesis submitted as part of the requirements for the award of the MSc in "7M06102 - Computer Science", SDU University

Head of Department Zhanar Mukash

Academic Supervisor Kamila Orynbekova

Master student Dana Kalekes

Kaskelen, 2024

Declaration

I confirm that this is my own work and the use of all material from other sources has been properly and fully acknowledged.

Dana Kalekes

June 2024

Acknowledgements

I truly appreciate my supervisor, Kamila Orynbekova, for her patience and understanding. Her guidance and support have been invaluable throughout this research. Her insightful feedback, constructive criticism, and constant encouragement have significantly contributed to the development and completion of this thesis. I am especially thankful for the time she spent reviewing my work and providing detailed feedback, which greatly improved the quality of this research. Her expertise and knowledge have been a source of inspiration and have guided me through the complexities of this study.

I also wish to thank Abdulkarim Abdulrakhmanuly, the Director of the Distance Learning Division center, for his consistent support during the data collection phase.His assistance and cooperation made it possible to gather the necessary data for this research.

I am immensely grateful to my husband for his constant support and encouragement. His faith in me has provided significant motivation and strength, helping me overcome the challenges of this journey.

Dedication

This thesis is dedicated to: Me.

Abstract

Many students do not put in sufficient effort at the beginning of the academic year, leading to grades that are insufficient for completing courses or obtaining scholarships. This study aims to analyze and predict student performance on the Moodle platform to provide early interventions and improve academic outcomes. The analysis focused on various courses from the 2023-2024 academic year at SDU University, selected due to their high average number of students and well-established structures.

The research involved collecting data on three predictive factors: the number of completed assignments, the total time spent on the course, and the number of actions on the platform. Six machine learning algorithms were applied to predict student performance: k-Nearest Neighbor, Random Forest, Decision Tree, Logistic Regression, Naive Bayes, and Support Vector Machine. The study compared the effectiveness of early prediction at 5, 10, and 15 weeks into the courses.

Key findings indicate that student activities on Moodle are significantly correlated with higher academic performance. The Support Vector Machine model showed the best results in the early weeks, while the Random Forest model demonstrated stable results over longer periods. These findings highlight the potential of machine learning models to identify at-risk students early, allowing for timely support and interventions.

The implications of this research are significant for educators and administrators. The ability to predict student performance early can facilitate timely interventions, helping students improve their academic results and reduce withdrawal rates. This study contributes to the growing body of knowledge in educational data analysis and learning analytics, providing a foundation for future research to refine and expand predictive capabilities in educational institutions.

Аңдатпа

Көптеген студенттер оқу жылының басында оқу үлгеріміне жеткілікті күш жұмсамауы, бұл курстарды аяқтауға немесе стипендия алуға жеткіліксіз болып қалатын жағдайлар туындатады. Бұл зерттеу Moodle платформасында студенттердің үлгерімін талдау және болжауға бағытталған, ерте болжам жасау шараларын қамтамасыз етіп, академиялық нәтижелерді жақсарту мақсатында жасалған. Талдау SDU университетінің 2023-2024 оқу жылындағы әртүрлі курстарына жасалды, соның ішінде студенттердің орташа саны көп және құрылымы жақсы белгіленген курстар таңдалды.

Зерттеу барысында үш болжамды фактор бойынша мәліметтер жиналды: орындалған тапсырмалар саны, курста жалпы өткізілген уақыт және платформадағы әрекеттер саны. Студенттердің үлгерімін болжау үшін алты машиналық оқыту алгоритмдері қолданылды: k-Nearest Neighbor, Random Forest, Decision Tree, Logistic Regression, Naive Bayes және Support Vector Machine. Зерттеуде курстардың 5, 10 және 15 аптасында ерте болжаудың тиімділігін салыстырылды.

Негізгі қорытындылар бойынша Moodle-дағы студенттің әрекеттері жоғары академиялық үлгеріммен айтарлықтай байланысты екенін көрсетеді. Support Vector Machine моделі ерте апталарда ең жақсы нәтижелер көрсетті, ал Random Forest моделі ұзақ мерзімді кезеңдерде тұрақты нәтижелер көрсетті. Бұл нәтижелер машиналық оқыту модельдерінің қауіпті аймақта тұрған студенттерді ерте анықтау мүмкіндігін атап көрсетеді. Бұл уақытылы қолдау мен араласу шараларын қамтамасыз етуге мүмкіндік береді.

Бұл зерттеудің мұғалімдер мен әкімшілік үшін маңызды әсерлері бар. Студенттердің үлгерімін ерте болжау мүмкіндігі уақытылы араласуларға ықпал етіп, студенттерге академиялық нәтижелерін жақсартуға және оқудан шығу деңгейін төмендетуге көмектесе алады. Бұл зерттеу білім беру деректерін талдау және оқыту аналитикасы саласындағы білім қорының артуына үлес қосып, болашақ зерттеулерге білім беру мекемелерінде болжау мүмкіндіктерін жетілдіру және кеңейту үшін негіз бола алады.

Аннотация

Многие студенты в начале учебного года не прикладывают достаточных усилий, что приводит к неудовлетворительным оценкам, недостаточным для завершения курсов или получения стипендий. Это исследование направлено на анализ и прогнозирование успеваемости студентов на платформе Moodle с целью предоставления ранних прогнозов и улучшения академических результатов. Анализ был проведен по различным курсам 2023-2024 учебного года в университете SDU, включая курсы с большим средним количеством студентов и хорошо установленной структурой.

В ходе исследования были собраны данные по трем прогнозирующим факторам: количество выполненных заданий, общее время, проведенное на курсе, и количество действий на платформе. Для прогнозирования успеваемости студентов были применены шесть алгоритмов машинного обучения: k-Nearest Neighbor, Random Forest, Decision Tree, Logistic Regression, Naive Bayes и Support Vector Machine. В исследовании сравнивалась эффективность раннего прогнозирования на 5, 10 и 15 неделях курсов.

Основные выводы показывают, что действия студентов на Moodle значительно связаны с высокой академической успеваемостью. Модель Support Vector Machine показала наилучшие результаты на ранних неделях, тогда как модель Random Forest демонстрировала стабильные результаты на более длительных периодах. Эти результаты подчеркивают возможность моделей машинного обучения рано выявлять студентов, находящихся в зоне риска, что позволяет своевременно предоставлять поддержку и вмешательства.

Это исследование имеет важное значение для преподавателей и администрации. Возможность раннего прогнозирования успеваемости студентов способствует своевременным вмешательствам, помогая студентам улучшать свои академические результаты и снижать уровень отчислений. Данное исследование вносит вклад в растущее количество знаний в области анализа данных в образовании и учебной аналитики, предоставляя основу для будущих исследований по улучшению и расширению возможностей прогнозирования в образовательных учреждениях.

Abbreviations

LMS	– Learning Management System
Moodle	– Modular Object-Oriented Dynamic Learning Environment
KNN	– K-Nearest Neighbor
RF	– Random Forest
SVM	– Support Vector Machine
LR	– Logistic Regression
HMAC	– Hash Massage Authentication Code

Table of Contents

Chapter 1

Background and motivations

1.1 Introduction

In contemporary educational environments, Learning Management Systems (LMS) have become indispensable tools for organizing and managing the learning process. Among these systems, Moodle stands out as one of the most popular platforms, offering a flexible and comprehensive environment for both educators and students. The Moodle platform allows for the creation and management of various educational materials, the administration of tests and exams, the organization of discussion forums, and the provision of students with access to their grades and course materials at any time. This makes Moodle a powerful tool for supporting distance and blended learning, as well as traditional forms of education.

However, despite its numerous advantages, a significant problem remains—the lack of timely and accurate feedback regarding students' current academic progress within Moodle. Students often do not receive prompt information about their progress, which makes it difficult for them to understand their strengths and weaknesses and to adjust their learning activities accordingly. This information gap can lead to negative consequences, such as the loss of scholarships and the need to retake courses. As a result, students may experience stress, reduced motivation to learn, and ultimately, a decline in their academic performance.

The study explores the use of machine learning (ML) techniques for analyzing student progress and behavior in Moodle in order to forecast academic achievement and offer prompt solutions. Machine learning offers effective tools for analyzing large volumes of data, enabling the identification of hidden patterns and the making of accurate predictions. The use of ML in education can greatly improve our comprehension of how students engage with course content and what aspects affect their academic performance.

Moodle LMS Overview

Moodle is an open-source learning environment that offers a complete, safe, and adaptable system for establishing personalized learning environments to instructors, administrators, and students. It encompasses a wide range of functionalities,

including multimedia lectures, quizzes, assignments, and discussion forums, which support a holistic approach to learning and engagement. The platform's flexibility allows educators to tailor courses to meet specific learning objectives and to accommodate diverse learning styles.

Bologna Process and Course Withdrawals

Under the Bologna Process, students have the flexibility to withdraw from courses without academic penalty if they do so by a specified deadline. However, late withdrawals can have significant consequences, including the loss of scholarships and the necessity of retaking courses.

At our university, the option to withdraw in a timely manner is a crucial feature that can prevent these negative outcomes, provided that students are adequately informed about their academic standing. Ensuring that students receive timely and accurate feedback about their performance is essential to enable them to make informed decisions about course withdrawals.

Retakes and Scholarship Forfeiture

University policies regarding scholarship retention and course retakes are stringent. Students who fail to meet the minimum performance criteria risk losing their scholarships and being required to retake courses. This not only extends their time to graduation but also increases the financial burden associated with their education. Recognizing students who may underperform at an early stage can help mitigate these issues by allowing for timely interventions that support students in improving their academic performance and retaining their scholarships.

Online Students and Courses in Moodle

The Moodle system at our university supports a large number of online students and courses. According to the latest statistics, approximately 11,500 students are enrolled in over 2,600 courses for the 2023-2024 academic year. A significant amount of data is produced by this widespread use, which can be used to learn more about the behavior and academic achievement of the students. This data may be analyzed using machine learning techniques, which opens up the possibility of creating predictive models that can improve learning outcomes for educators and students.

Integrating LMS with Machine Learning

Integrating machine learning methods with LMS data presents promising opportunities for improving educational outcomes. Machine learning can automate the data analysis process, identify key metrics such as the number of completed assignments, time spent on learning activities, and frequency of interactions, and use these metrics to predict academic performance.

Numerous techniques, including Support Vector Machines, Decision Trees, Random Forests, Logistic Regression, k-Nearest Neighbors, and Naive Bayes classifiers, can be used to create accurate predictive models.

Advantages of Data Analysis in Moodle

Moodle's extensive data logging capabilities provide a rich source of information for detailed analysis. Every student action, from accessing learning materials to submitting assignments, is recorded, allowing for thorough tracking of student engagement and progress. This data is invaluable for developing predictive models that can forecast student performance and identify those at risk of underperforming. Analyzing this data not only enhances the quality of education but also provides students and teachers with timely and accurate information needed for informed decision-making.

Addressing the Problem with Machine Learning

This study uses a variety of machine learning methods for predicting academic performance in the future for students based on their Moodle interactions. Important information includes how many tasks are finished, how long classes take, and how often students interact with the course materials. By analyzing these factors, we aim to provide early warnings to students and educators, enabling timely actions to improve academic outcomes and reduce the likelihood of failure.

Aim of the research

The aim of the research is for investigating and implementing machine learning methods for predicting academic achievement of students by analyzing their interactions with the Moodle learning management system. By doing so, the study aims to provide timely feedback and interventions to support students and enhance their academic outcomes.

This research specifically seeks to bridge the gap between raw educational data and actionable insights. By leveraging the vast amount of interaction data available in Moodle, we aim to identify patterns and trends that can reliably forecast a student's future performance.

Additionally, the research seeks to identify the most effective machine learning models for predictive analysis. We compare different algorithms, including Naive Bayes classifiers, Random Forests, k-Nearest Neighbors, Decision Trees, Support Vector Machines, and Logistic Regression, in order to evaluate the advantages and disadvantages of each method in the context of educational data.

Relevance of the research

The integration of Machine Learning (ML) in analyzing educational data, specifically through platforms like Moodle, holds transformative potential for the field of education. The relevance of this research can be highlighted across several dimensions:

1. Enhanced Predictive Analytics in Education

Predictive analytics powered by ML can provide unprecedented insights into student behaviors and outcomes. This research is crucial as it explores the feasibility and effectiveness of using ML to forecast academic performance based on

behavioral data from Moodle. Such predictive capabilities allow for the early identification of at-risk students, enabling timely interventions that can significantly improve student retention and success rates.

2. Data-Driven Decision Making for Educators

By effectively analyzing behavioral data from LMS platforms, educators can gain a deeper understanding of student engagement and learning processes. This research aims to equip educators with data-driven insights that can inform curriculum design, teaching strategies, and student support mechanisms. The ability to adapt teaching methods based on solid data analysis could revolutionize personalized education and help educators meet diverse learner needs more effectively.

3. Improving Learning Management Systems

The findings from this research could lead to improvements in the functionality and effectiveness of LMS platforms like Moodle. Developers may improve these platforms to encourage these activities by understanding what behaviors are associated with student success. One such solution is to integrate ML algorithms directly into the LMS to give instructors and students real-time statistics and feedback.

4. Contributions to Educational Theory and Practice

This study connects theoretical data mining approaches with real-world educational applications, expanding the corpus of knowledge in educational technology. It extends the existing theories of student engagement and online learning effectiveness, offering empirical evidence and methodological innovations that can be adopted in various educational settings.

5. Scalability and Accessibility in Education

The research also has implications for the scalability and accessibility of quality education. By combining open-source platforms like Moodle with the scalability of machine learning models, educational institutions can deliver more personalized and adaptive learning experiences to a larger audience at a lower cost. This is especially important in areas with limited educational resources.

6. Policy Making and Institutional Strategy

On a broader scale, this research can influence policy-making and strategic decisions within educational institutions. With robust predictive models, institutions can better allocate resources, plan educational offerings, and design support services that are more aligned with student needs and behaviors, leading to improved educational outcomes and efficiency.

In conclusion, this research is relevant because it has the potential to greatly improve the way that educational data is used to enhance teaching and learning environments. By focusing on ML and Moodle data, the study not only addresses technical and analytical challenges but also aligns with broader educational goals such as increasing accessibility, personalization, and effectiveness of education.

Significance of the Research

This research is significant because it addresses a critical gap in the provision of timely feedback to students about their academic performance. By leveraging the capabilities of machine learning, the study aims to improve the accuracy and timeliness of performance predictions, thereby enabling early interventions. This can help prevent negative outcomes such as the loss of scholarships and the need for course retakes, ultimately supporting students in achieving their academic goals.

Research Objectives

1. Investigate Patterns of Student Behavior

Analyze the behavioral patterns of students within the Moodle platform across various courses, focusing on key metrics such as the number of assignments completed, time spent on course activities, and the frequency of interactions with course materials. This will help identify the most significant predictors of academic performance.

2. Compare Machine Learning Models Evaluate and compare the performance of six machine learning classification models: Logistic Regression, Decision Tree, Random Forest, k-Nearest Neighbors, Support Vector Machine, and Naive Bayes — in predicting student performance at the 5th, 10th, and 15th weeks of the semester. This comparison aims to determine the most accurate and reliable model for early prediction of academic success.

3. Early Identification of At-Risk Students

Use predictive models to identify students who are at risk of underperforming early in the semester. This early identification will enable educators to intervene promptly and provide the necessary support to improve the students' academic outcomes.

4. Evaluate Intervention Effectiveness

Assess the effectiveness of different intervention strategies based on the early predictions of student performance. This includes analyzing how timely feedback and additional support impact students' progress and final grades.

5. Enhance Data Preprocessing Techniques

Develop and refine data preprocessing techniques to handle the large and complex datasets generated by Moodle. This involves cleaning, normalizing, and transforming the data to improve the accuracy of machine learning models.

6. Explore Longitudinal Data Analysis

Investigate the use of longitudinal data analysis to capture changes in student behavior and performance over time. This approach will provide deeper insights into the dynamic nature of student engagement and its impact on academic outcomes.

7. Integrate Multiple Data Sources

Combine data from various sources within Moodle, such as attendance records, assessment scores, and interaction logs, to create a more comprehensive and holistic predictive model. This integration aims to enhance the model's accuracy and reliability.

8. Generalize Findings Across Courses

Evaluate the applicability of the predictive models across various courses and educational settings.

Chapter 2

Literature Review

2.1 History and Evolution of Learning Management Systems (LMS)

The creation of Learning Management Systems (LMS) has been a major advancement in educational technology, changing how educational content is delivered, managed, and utilized. This part provides a summary of the history and development of LMS, focusing on the important milestones and technological breakthroughs that have contributed to their present form.

Computer-assisted instruction originated in the 1960s and 1970s with the emergence of the first LMS generation. Early examples include PLATO (Programmed Logic for Automatic Teaching Operations) developed at the University of Illinois and IBM's Coursewriter. These systems primarily focused on delivering instructional content and quizzes, laying the groundwork for future developments in educational technology [1].

The internet's widespread adoption in the 1990s transformed educational technology, giving rise to web-based LMS. The advent of HTML and web browsers enabled educational institutions to provide content online. Blackboard, launched in 1997, became one of the most well-known web-based LMS platforms, providing tools for course management, communication, and assessment [2].

In the early 2000s, open-source LMS became popular, with Moodle (Modular Object-Oriented Dynamic Learning Environment) being a prominent example. Developed by Martin Dougiamas in 2002, Moodle offered a customizable and cost-effective alternative to proprietary systems like Blackboard. Its open-source nature allowed institutions to tailor the platform to their specific needs, leading to widespread adoption globally [3].

With technological advancements, LMS capabilities grew. Incorporating multimedia elements like video lectures and interactive simulations significantly enhanced the learning experience. The integration of Web 2.0 tools enabled greater interactivity and collaboration among students and educators. LMS platforms began to support features like discussion forums, wikis, blogs, and real-time chat, fostering a more engaging and interactive learning environment [4].

The proliferation of smartphones and tablets in the late 2000s and early 2010s brought about the era of mobile learning. LMS platforms began to develop mobile applications, allowing students to access course materials and participate in learning activities on the go. Simultaneously, the adoption of cloud computing facilitated the development of cloud-based LMS, providing scalability, flexibility, and reduced IT overhead for educational institutions [5].

Recent advancements in data analytics have further transformed LMS. The integration of learning analytics tools within LMS platforms allows educators to track student performance, engagement, and behavior in real-time. These insights enable the development of personalized learning paths, adaptive assessments, and targeted interventions, enhancing the overall effectiveness of the educational process [6].

The future of LMS will involve the ongoing incorporation of emerging technologies like artificial intelligence (AI), machine learning (ML), and virtual reality (VR). AI and ML can improve personalization by offering more precise predictive analytics and recommendations. VR can offer immersive learning experiences, enhancing engagement and effectiveness in education. As LMS platforms advance, they will be pivotal in shaping the future of education[7].

2.2 Using Moodle LMS in Education

Learning Management Systems (LMS), particularly Moodle, have become integral to educational institutions globally due to their flexibility and extensive features. Moodle provides a platform for managing course content, facilitating communication, and tracking student progress. During the COVID-19 pandemic, reliance on these systems increased as institutions shifted to online learning environments [8].

Studies have shown the effectiveness of Moodle in enhancing educational experiences. For example, researchers at the University of Split employed data mining techniques on Moodle logs to forecast final exam grades, evaluating various classification methods such as Decision Tree, k-Nearest Neighbor, Logistic Regression, Naive Bayes, Random Forest, and Support Vector Machine. They discovered that Random Forest and Support Vector Machine yielded the highest accuracy[9].

Despite these advantages, there are challenges. One significant issue is the need for more sophisticated methods to evaluate LMS effectiveness without relying solely on subjective measures such as surveys. Learning Analytics tools, which use log data to provide real-time insights, offer more accurate evaluations compared to traditional methods [8].

In our study, we face similar issues but focus specifically on the gap between students' perceived and actual progress within Moodle. Unlike general effectiveness studies, our research zeroes in on predicting academic performance to help students avoid late withdrawals and loss of scholarships by providing timely feedback and interventions based on their interactions with Moodle.

2.3 Student Engagement in Online Learning Environments

Student engagement plays a vital role in the effectiveness of online learning environments. Engaged students tend to achieve better academic results, complete courses, and retain information more effectively. Research has identified different aspects of student engagement: behavioral, emotional, and cognitive.

Behavioral engagement pertains to students' involvement in academic and extracurricular activities. Emotional engagement relates to students' emotional reactions to their learning experiences, including interest and motivation. Cognitive engagement involves the effort and dedication students put into learning and comprehending complex concepts[10].

Online learning environments, such as those provided by Moodle, offer numerous tools to enhance student engagement. These tools include interactive multimedia content, discussion forums, quizzes, and real-time feedback mechanisms. Research has shown that the use of these tools can significantly increase student engagement and improve learning outcomes [11].

However, maintaining high levels of engagement in online learning environments can be challenging. Factors such as isolation, lack of face-to-face interaction, and technical issues can negatively impact student engagement. To address these challenges, educators must implement strategies that promote interaction, provide timely feedback, and create a supportive online learning community [12].

2.4 Predictive Analytics in Education

Predictive analytics in education leverages data, statistical algorithms, and machine learning techniques to predict future outcomes based on past data. In the LMS context, predictive analytics can anticipate student performance, identify students at risk, and guide instructional strategies.

The use of predictive analytics in education has become increasingly popular due to its potential to improve student success. By examining data from LMS platforms like Moodle, educators can understand student behaviors, engagement, and performance trends. These insights enable the creation of targeted interventions to support student learning and enhance academic outcomes.[13].

Numerous studies have shown the efficacy of predictive analytics in identifying students who are at risk. For instance, research has shown that early identification of students who are likely to struggle can lead to timely interventions, such as personalized support and additional resources, which can significantly improve their chances of success [14].

One key feature of predictive analytics in education is the analysis of the time students spend on the LMS. The time students spend on the platform can indicate their engagement and commitment to the course. Research has found a positive correlation between the duration of time students spend on LMS activities and their academic performance [15]. For example, students who log in frequently, spend more time on course materials, and actively participate in discussions tend

to achieve higher grades.

Another important feature is the analysis of student actions within the LMS. Actions such as the number of logins, clicks, and interactions with course content can provide insights into student engagement and learning behaviors. Studies have demonstrated that student action patterns can predict academic success. For example, students who consistently access learning materials, submit assignments punctually, and engage in forum discussions tend to achieve better academic performance [16].

The count of completed tasks is also a significant predictor of student performance. Completing assignments, quizzes, and other tasks directly measures a student's progress and comprehension of the course material. Research indicates that students who consistently finish their tasks tend to achieve better academic results. For example, the number of completed assignments has been shown to be a strong predictor of final grades in a course [17].

Despite its advantages, predictive analytics in education faces challenges. These include concerns about data quality, privacy issues, and the need for skilled personnel to interpret and utilize the data effectively. To maximize the impact of predictive analytics, educational institutions must address these challenges and develop robust data governance policies [18]. Additionally, there is a need for continuous evaluation and improvement of predictive models to ensure their accuracy and relevance in different educational contexts [19].

In conclusion, predictive analytics in education offers significant potential for improving student outcomes by providing actionable insights into student behaviors and performance. By focusing on key features such as time spent, actions within the LMS, and completed task counts, educators can identify at-risk students early and implement targeted interventions to support their success.

2.5 Applying Machine Learning for Performance Prediction

Machine learning (ML) techniques are increasingly used to predict student performance utilizing data from LMS platforms. These techniques offer valuable insights into students' academic behaviors and outcomes. Several studies have compared different ML algorithms to determine their effectiveness in predicting student performance.

For example, research at Ubon Ratchathani University applied six ML classifiers—Neural Network, Random Forest, Decision Tree, Logistic Regression, Linear Regression, and Support Vector Machine—to predict student performance. They found that Decision Tree performed best at course completion, while Support Vector Machine showed the highest accuracy at early stages [20]. This suggests that different ML models might be better suited for different phases of a student's learning journey.

Another notable study by Costa et al. (2017) examined the use of ensemble learning techniques to improve the accuracy of student performance predictions. They combined the outputs of several base learners, such as Decision Trees and

SVMs, to create a more robust predictive model. The results indicated that ensemble methods significantly outperformed individual models, highlighting the potential of integrating multiple ML approaches for enhanced prediction accuracy [21].

Despite these advancements, challenges remain. Many ML models require extensive data preprocessing and feature selection to achieve optimal performance. For instance, Asif et al. (2017) emphasized the importance of feature engineering in their study on predicting student performance using Moodle data. They noted that features such as the number of forum posts, quiz attempts, and assignment submissions were critical in enhancing model performance [22].

Moreover, the interpretability of these models is crucial for practical application in educational settings, as teachers and administrators need to understand the factors influencing predictions to make informed decisions [23]. The work by Jayaprakash et al. (2014) underscores this point, showing that while complex models like Random Forests and SVMs provided higher accuracy, simpler models like Logistic Regression were more interpretable and therefore more useful for academic advisors [24].

In our research, we also encounter the challenge of preprocessing and feature selection. However, our focus is on integrating various data points such as the number of assignments completed, time spent on courses, and the frequency of interactions to provide a holistic view of student performance. This approach aims to improve the predictive power and usability of ML models in real-time educational settings.

Furthermore, the study by Romero et al. (2013) utilized clustering techniques alongside traditional classification algorithms to identify distinct patterns in student behavior. This multi-faceted approach allowed for a deeper understanding of how different student interactions within the LMS could be linked to their academic outcomes [25]. By combining clustering and classification, they were able to provide more nuanced insights that could inform tailored interventions.

In addition, Wang et al. (2019) explored the use of deep learning techniques for student performance prediction. Their study demonstrated that deep neural networks could capture complex, non-linear relationships in the data, leading to more accurate predictions compared to traditional ML models [26]. However, they also noted the increased computational requirements and the need for larger datasets to train these models effectively.

Our research similarly leverages a variety of ML techniques, including both traditional models and more advanced approaches like deep learning. We aim to not only predict student performance but also to understand the underlying factors driving these predictions. By doing so, we hope to provide actionable insights that can support educators in making data-driven decisions to enhance student learning outcomes.

In summary, the application of ML for performance prediction in educational settings is a rapidly evolving field. While significant progress has been made, challenges such as data preprocessing, feature selection, and model interpretability remain. Our research contributes to this body of knowledge by focusing on a holistic approach that integrates multiple data points and leverages a range of ML techniques to provide comprehensive and actionable insights.

2.6 Comparing Multiple Machine Learning Models for Performance Prediction

Comparative studies of multiple ML models highlight the strengths and weaknesses of each approach. A study at Ubon Ratchathani University compared various ML algorithms using Moodle logs and identified that while some algorithms like Decision Trees and Random Forests offered high accuracy, others such as Naive Bayes were less effective due to their assumptions about data distribution [27].

Another study focused on identifying successful learners and at-risk students in a blended learning environment using the Moodle Parser tool. This study found a strong correlation between student engagement metrics (like the number of logins and page views) and academic performance [28]. This suggests that models leveraging engagement data can effectively predict academic outcomes.

A comparative analysis conducted by Shayan et al. (2019) explored various ML algorithms including Logistic Regression, Decision Tree, k-Nearest Neighbors, and Support Vector Machine to predict student performance. The study concluded that no single model consistently outperformed others across different metrics, but a hybrid approach combining multiple models yielded better prediction accuracy. This hybrid approach captures different aspects of student behavior, providing more comprehensive predictions [29].

Similarly, a study by Jayaprakash et al. (2014) examined early warning systems using different ML models to identify at-risk students. The results indicated that while logistic regression models were highly interpretable and useful for academic advisors, more complex models like Random Forests and SVMs provided higher accuracy and recall [30]. This balance between interpretability and accuracy is crucial for practical applications in educational settings.

In another research conducted by Romero et al. (2013), different classification algorithms were applied to Moodle data to predict student success. The study found that Decision Trees and Random Forests provided the highest accuracy, but the precision of these models varied significantly across different datasets [31]. This variability highlights the importance of context and dataset characteristics in model performance.

Our research similarly aims to compare multiple ML models, including Logistic Regression, Decision Tree, Random Forest, k-Nearest Neighbors, Support Vector Machine, and Naive Bayes. However, our unique contribution lies in the staged prediction approach—evaluating student performance at the 5th, 10th, and 15th weeks—to offer timely interventions. This method addresses the dynamic nature of student behavior over a semester, a factor often overlooked in existing studies.

By staging the predictions, our approach allows for early identification of at-risk students, providing opportunities for timely interventions and support. For instance, early predictions at the 5th week can help identify students who might benefit from additional resources, while predictions at the 10th and 15th weeks can guide more intensive interventions if necessary.

Incorporating features such as "Actions" and "Time Spent" provides a nuanced view of student engagement and performance. This holistic approach not only enhances prediction accuracy but also aligns with the findings of studies that

emphasize the importance of continuous monitoring and engagement metrics in predicting academic success [32].

Overall, the comparative studies and our staged prediction approach underscore the potential benefits of using multiple ML models and continuous assessment to improve educational outcomes. By leveraging the strengths of various algorithms and understanding their limitations, we can develop more effective and responsive educational tools.

2.7 GAP and Future Directions

While significant progress has been made in applying ML to predict student performance in LMS environments, several gaps remain. A major limitation is the lack of generalizability of models across different educational contexts and courses. Most studies focus on specific courses or institutions, limiting the applicability of findings to broader contexts [9].

Additionally, many existing models do not account for the dynamic nature of student behavior over time. Future research should explore the use of longitudinal data to capture changes in student engagement and performance throughout the course. There is also a need for more user-friendly tools that can be easily integrated into existing LMS platforms to provide real-time predictions and feedback to both students and educators [8]. Moreover, many studies have not fully explored the potential of combining various data sources, such as attendance records, assessment scores, and interaction data from multiple platforms. Integrating these data sources could enhance the accuracy and reliability of predictive models.

In conclusion, while the application of ML in LMS has shown promising results, further research is needed to address these gaps and develop more robust, generalizable, and user-friendly predictive models. By leveraging the rich data available in LMS like Moodle, educational institutions can enhance their ability to support student success and improve overall educational outcomes. Our research aims to fill these gaps by focusing on early prediction and timely intervention strategies, thereby providing actionable insights that can help students and educators alike.

2.8 Future Trends in LMS and Educational Technology

The future of LMS and educational technology is shaped by several emerging trends that have the potential to transform teaching and learning. These trends include the integration of artificial intelligence (AI), the use of virtual and augmented reality (VR/AR), and the development of more personalized and adaptive learning environments.

AI has the potential to revolutionize education by providing personalized learning experiences, automating administrative tasks, and offering intelligent tutoring systems. AI-powered LMS can analyze student data to create customized learning paths, provide real-time feedback, and predict academic performance. For example, AI algorithms can identify patterns in student behavior and performance,

enabling the system to suggest targeted interventions or additional resources tailored to individual needs [33]. This capability can help educators identify students at risk of falling behind and provide timely support to improve their learning outcomes [34].

VR and AR technologies offer immersive learning experiences that can enhance student engagement and understanding of complex concepts. These technologies can be integrated into LMS to create interactive simulations, virtual labs, and 3D visualizations, providing students with hands-on learning opportunities that are not possible in traditional classroom settings [35]. For instance, medical students can practice surgical procedures in a virtual environment, or history students can explore ancient civilizations through AR experiences, making learning more engaging and effective [36].

Personalized and adaptive learning environments are designed to meet the unique needs of each student. These environments use data from LMS to adapt the content, pace, and learning activities based on individual student performance and preferences. This approach can improve student motivation, engagement, and learning outcomes [37]. Adaptive learning systems can adjust the difficulty level of tasks in real-time, provide personalized feedback, and recommend specific learning resources, ensuring that each student receives the appropriate level of challenge and support [38].

As these trends continue to evolve, they will play a crucial role in shaping the future of education. LMS platforms must adapt to these changes by incorporating new technologies and providing educators with the tools and resources needed to create effective and engaging learning experiences. By incorporating AI, VR/AR, and personalized learning environments, educational institutions can significantly enhance the learning experience, improve student outcomes, and better equip students for the challenges of the modern world.

Chapter 3

Concept

In this chapter, we describe the concept of the Machine Learning algorithm as follows:

3.1 Decision Tree

The Decision Tree algorithm is renowned for its versatility in managing both classification and regression tasks. It functions by recursively splitting the dataset into smaller subsets based on the values of input features. This process creates a tree-like structure where each internal node signifies a feature, each branch represents a decision rule, and each leaf node denotes an outcome.

At each node, the algorithm selects the best feature to split the data using criteria such as Information Gain or Gini Impurity. This splitting continues until all data points within a subset belong to the same class or a predefined stopping criterion is met. This method allows the Decision Tree to effectively capture the underlying patterns in the data, making it a powerful tool for various predictive modeling tasks [39]. Decision trees are easy to understand and interpret due to their visual representation. They can handle both numerical and categorical data and require minimal data preparation [40].

However, decision trees can be prone to overfitting, which means they may perform well on training data but poorly on new, unseen data. They are also sensitive to small changes in the data, which can significantly alter the structure of the tree. Additionally, decision trees tend to be biased towards features with more levels, which can skew the model's performance [41]. Overfitting occurs when the tree captures noise in the data, leading to poor generalization on new, unseen data. This issue can be mitigated by using techniques like pruning. Pruning involves removing parts of the tree that do not contribute additional power to classify instances, helping to simplify the model and improve its performance on unseen data. Decision trees are also sensitive to small variations in the data; a small change can result in a completely different tree structure. Additionally, they tend to be biased towards features with a larger number of levels, which can overshadow more relevant features.

3.2 k-Nearest Neighbors

The k-Nearest Neighbor (k-NN) algorithm is a straightforward and widely-utilized machine learning method for both classification and regression tasks. It operates on the principle that similar data points tend to be close to each other in the feature space. To classify a new data point, k-NN computes the distance between this point and all other points in the training set using a distance metric such as Euclidean or Manhattan distance. It then selects the k closest data points (neighbors) and assigns the class label based on the majority vote among these neighbors (for classification) or calculates the average of the neighbors' values (for regression). k-NN is intuitive and easy to understand and implement. However, it can be computationally intensive, particularly with large datasets, as it requires calculating the distance to every point in the training set for each new data point. Additionally, k-NN can be sensitive to the choice of k and the distance metric used, which can significantly impact its performance [42].

The algorithm's simplicity and ease of implementation make it a popular choice for many applications. However, it can be computationally intensive since it requires calculating distances to all training points for each prediction. Additionally, k-NN is sensitive to the choice of k (the number of neighbors) and the scale of the features, often requiring normalization of data [43]. Despite these limitations, k-NN is effective for small to medium-sized datasets and performs well when the relationship between the features and the target variable is complex and nonlinear, providing a flexible method for various pattern recognition and data analysis tasks [44].

3.3 Logistic Regression

Logistic Regression (LR) is a widely used statistical technique in machine learning for binary classification tasks. It models the likelihood of a binary outcome based on one or more predictor variables. The algorithm starts by computing a linear combination of the input features, which is then converted into a probability using the logistic (sigmoid) function. This function maps the output to a range between 0 and 1, ensuring the result is a valid probability. The predicted probability is compared to a threshold (commonly 0.5) to classify the input into one of the two classes.

The algorithm operates as follows: it starts by calculating a linear combination of the input features, weighted by coefficients derived from the training data. This linear combination, or score, is then transformed into a probability using the sigmoid function. The sigmoid function ensures that the output is a value between 0 and 1, representing the estimated probability of the positive class. The final step involves comparing this probability to a predefined threshold, typically 0.5, to make a binary decision. If the probability is greater than or equal to the threshold, the outcome is classified as the positive class; otherwise, it is classified as the negative class.

Logistic regression is valued for its simplicity, interpretability, and efficiency. It assumes a linear relationship between the input features and the log-odds of the outcome, making it easy to understand and explain. The coefficients in the model represent the change in the log-odds of the outcome for a one-unit change in the corresponding feature. This interpretability makes logistic regression a preferred choice in fields where understanding the relationship between predictors and the outcome is crucial.

However, this assumption of a linear relationship may not always hold true in practice, which can limit the model's performance on more complex datasets. Despite this limitation, logistic regression provides a probabilistic framework that is useful in various applications, such as medical diagnosis, credit scoring, and marketing, where the goal is to predict the likelihood of a binary event occurring [45]. In these applications, the ability to produce a probability rather than just a binary outcome is particularly valuable, as it allows for more nuanced decision-making.

3.4 Naive Bayes

Naive Bayes is a probabilistic algorithm utilized mainly for classification problems. It relies on Bayes' Theorem and assumes that all features are conditionally independent given the class label. Despite this "naive" assumption, it simplifies calculations and performs effectively in practice.

The algorithm computes the posterior probability of each class based on the given input features using the formula:

$$P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)} \tag{3.4.1}$$

It assigns the class with the highest posterior probability to the new instance. Naive Bayes is straightforward to implement, computationally efficient, and scales well with large datasets. It is especially effective with high-dimensional data, making it well-suited for tasks such as text classification, spam filtering, and similar applications.

However, the assumption of feature independence can impact its performance, and it can produce zero probabilities if a feature/category combination is not present in the training data. This issue can be alleviated using techniques like Laplace smoothing.

Naive Bayes is extensively used in text classification, medical diagnosis, and recommendation systems due to its efficiency and effectiveness in high-dimensional spaces [46].

3.5 Support Vector Machine

Support Vector Machine (SVM) is a robust supervised machine learning algorithm used for both classification and regression tasks. It functions by identifying the hyperplane that best separates data points of different classes in the feature space, thereby maximizing the margin between the classes. For data that is not lin-

early separable, SVM employs kernel functions, such as linear, polynomial, and radial basis function (RBF) kernels, to transform the data into a higher-dimensional space where a linear hyperplane can be identified.

SVM is effective in high-dimensional spaces and resistant to overfitting, especially with the appropriate choice of kernel and regularization parameters. However, it can be computationally demanding and requires careful tuning of its parameters. Despite these challenges, SVM is widely used in various fields, including text classification, image recognition, bioinformatics, and handwriting recognition, due to its versatility and effectiveness [47, 48].

In summary, SVM is a powerful and versatile machine learning algorithm that provides robust performance across different applications. Its capacity to handle high-dimensional data and prevent overfitting, combined with the flexibility offered by different kernel functions, makes it a valuable tool for both classification and regression tasks. However, its computational demands and the need for precise parameter tuning are important considerations when applying SVM to real-world problems.

3.6 Random Forest

Random Forest is an ensemble learning technique used for both classification and regression tasks. It generates multiple decision trees through bootstrap sampling and aggregates their predictions to enhance accuracy and robustness. During training, each tree is constructed on a random subset of features, which helps to minimize overfitting and effectively manage high-dimensional data.

Random Forest improves predictive performance by averaging the results of many decision trees, providing robust predictions even in the presence of noise and outliers. It is also useful for estimating feature importance. However, it can be computationally intensive and less interpretable than individual decision trees.

Applications of Random Forest include financial market analysis, medical diagnostics, remote sensing, and customer segmentation. Its versatility and effectiveness in handling large datasets make it a valuable tool in various fields [49].

Chapter 4

Methodology

This study employs predictive analytics techniques to analyze student behavior data obtained from the Moodle environment at SDU University. The primary objective is to predict students' academic performance based on their interactions with the Moodle Learning Management System (LMS). Additionally, the study assesses the effectiveness of various classification methods, such as k-Nearest Neighbor (k-NN), Random Forest, Decision Tree, Logistic Regression, Naive Bayes, and Support Vector Machine (SVM), in developing predictive models.

4.1 Collecting Moodle data

The data for this research was obtained from the Moodle LMS at SDU University. It encompasses various student activities, including the number of assignments completed, time spent on course materials, and overall engagement with the Moodle platform. The dataset spans multiple courses and academic terms, providing a comprehensive view of student behavior over time.

4.1.1 Access to the Moodle LMS Data

Abdulkarim Abdulrakhmanuly, the Director of the Distance Learning Division at SDU University (a.abdulrakhmanuly@sdu.edu.kz), has granted permission and access to the Moodle LMS data for research purposes. He will personally manage all necessary methods to ensure the anonymity of the data. Specifically, student ID numbers, names, and course titles will be anonymized to maintain confidentiality.

4.1.2 Collecting Data Using Moodle Plugins

To gather detailed and specific data on student activities, two key Moodle plugins were utilized: the Course Dedication plugin and the Activity Completion plugin.

The Course Dedication Plugin tracks the amount of time students spend on each course. It provides detailed logs of student engagement, helping to quantify

the dedication of each student to their coursework. The gathered data encompasses the total duration spent on course materials and the number of logins [50].

The Activity Completion Plugin records the completion status of various activities within the Moodle LMS. It tracks whether students have completed assignments, quizzes, forums, and other course activities. The data collected includes the number and type of completed activities, as well as the timestamps of these completions [51].

4.1.3 Data Confidentiality and Security

Ensuring data anonymity is crucial for maintaining the confidentiality and privacy of individual users, preventing their identification. This research will adhere to data security standards such as ISO/IEC 27001 to ensure the secure storage and transmission of information. Abdulkarim Abdulrakhmanuly will implement robust data security measures, including the use of cryptographic hashing methods like SHA-256 and secure key management techniques such as HMAC. These methods will ensure the integrity of hashed data and the confidentiality of keys, thereby enhancing overall data security. Additionally, he has provided a Declaration of Conformity to research and publication ethics, affirming his commitment to maintaining ethical standards and ensuring data security and anonymity throughout the research process.

4.1.4 Course Selection for Analysis

For the analysis, two courses, XXX123 and YYY456, were selected for the 2023-2024 academic year at SDU University. These courses were chosen for several reasons:

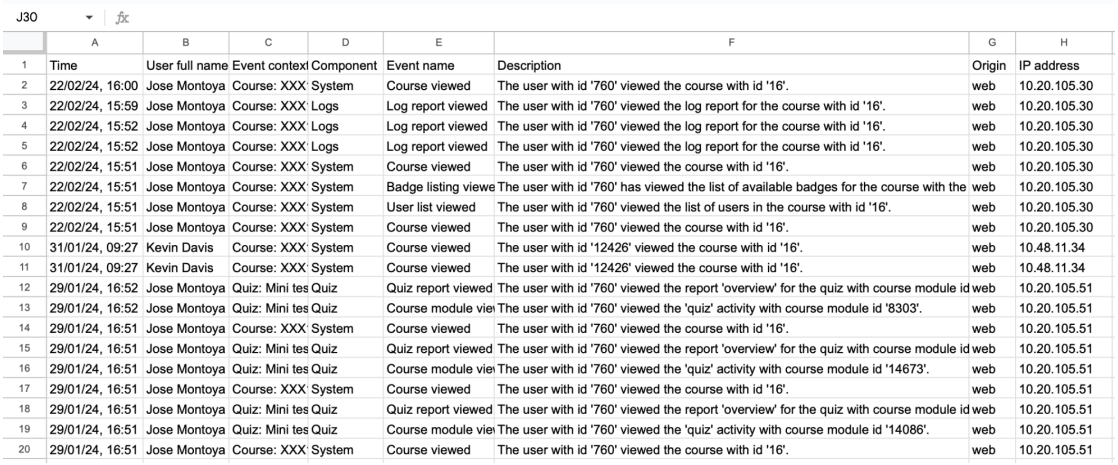
High Enrollment: Both courses have a higher average number of students enrolled compared to other courses, providing larger datasets for more robust analysis.

Comprehensive Course Structure: The structure of both courses is fully formed, ensuring that all relevant activities and assessments are consistently tracked and recorded.

Complete Grade Availability: All students' grades for these courses are displayed on the Moodle website, allowing for accurate and comprehensive performance evaluation.

These characteristics make XXX123 and YYY456 ideal candidates for comprehensive data analysis and performance prediction. The higher number of students ensures a more statistically significant analysis, while the well-defined course structures and complete grade availability facilitate accurate data collection and performance prediction.

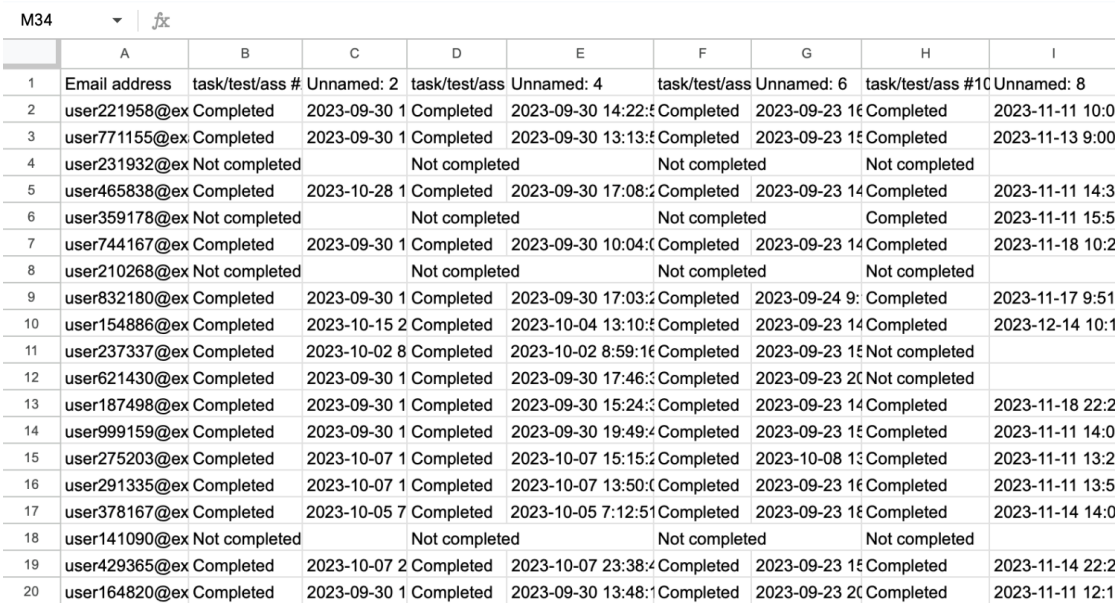
In Figure 4.1,the dataset logs records from the Moodle LMS system, containing information about user activities. The dataset includes timestamps, usernames, types of actions (e.g., Course, Logs, Grades), descriptions of actions (e.g., the user with ID '760' viewed the course with ID '16'), the source of the event (web), and user IP addresses. The logs capture various actions on the platform, such as view-



	A	B	C	D	E	F	G	H
1	Time	User full name	Event context	Component	Event name	Description	Origin	IP address
2	22/02/24, 16:00	Jose Montoya	Course: XXX	System	Course viewed	The user with id '760' viewed the course with id '16'.	web	10.20.105.30
3	22/02/24, 15:59	Jose Montoya	Course: XXX	Logs	Log report viewed	The user with id '760' viewed the log report for the course with id '16'.	web	10.20.105.30
4	22/02/24, 15:52	Jose Montoya	Course: XXX	Logs	Log report viewed	The user with id '760' viewed the log report for the course with id '16'.	web	10.20.105.30
5	22/02/24, 15:52	Jose Montoya	Course: XXX	Logs	Log report viewed	The user with id '760' viewed the log report for the course with id '16'.	web	10.20.105.30
6	22/02/24, 15:51	Jose Montoya	Course: XXX	System	Course viewed	The user with id '760' viewed the course with id '16'.	web	10.20.105.30
7	22/02/24, 15:51	Jose Montoya	Course: XXX	System	Badge listing viewe	The user with id '760' has viewed the list of available badges for the course with the	web	10.20.105.30
8	22/02/24, 15:51	Jose Montoya	Course: XXX	System	User list viewed	The user with id '760' viewed the list of users in the course with id '16'.	web	10.20.105.30
9	22/02/24, 15:51	Jose Montoya	Course: XXX	System	Course viewed	The user with id '760' viewed the course with id '16'.	web	10.20.105.30
10	31/01/24, 09:27	Kevin Davis	Course: XXX	System	Course viewed	The user with id '12426' viewed the course with id '16'.	web	10.48.11.34
11	31/01/24, 09:27	Kevin Davis	Course: XXX	System	Course viewed	The user with id '12426' viewed the course with id '16'.	web	10.48.11.34
12	29/01/24, 16:52	Jose Montoya	Quiz: Mini tes	Quiz	Quiz report viewed	The user with id '760' viewed the report 'overview' for the quiz with course module id	web	10.20.105.51
13	29/01/24, 16:52	Jose Montoya	Quiz: Mini tes	Quiz	Course module vie	The user with id '760' viewed the 'quiz' activity with course module id '8303'.	web	10.20.105.51
14	29/01/24, 16:51	Jose Montoya	Course: XXX	System	Course viewed	The user with id '760' viewed the course with id '16'.	web	10.20.105.51
15	29/01/24, 16:51	Jose Montoya	Quiz: Mini tes	Quiz	Quiz report viewed	The user with id '760' viewed the report 'overview' for the quiz with course module id	web	10.20.105.51
16	29/01/24, 16:51	Jose Montoya	Quiz: Mini tes	Quiz	Course module vie	The user with id '760' viewed the 'quiz' activity with course module id '14673'.	web	10.20.105.51
17	29/01/24, 16:51	Jose Montoya	Course: XXX	System	Course viewed	The user with id '760' viewed the course with id '16'.	web	10.20.105.51
18	29/01/24, 16:51	Jose Montoya	Quiz: Mini tes	Quiz	Quiz report viewed	The user with id '760' viewed the report 'overview' for the quiz with course module id	web	10.20.105.51
19	29/01/24, 16:51	Jose Montoya	Quiz: Mini tes	Quiz	Course module vie	The user with id '760' viewed the 'quiz' activity with course module id '14086'.	web	10.20.105.51
20	29/01/24, 16:51	Jose Montoya	Course: XXX	System	Course viewed	The user with id '760' viewed the course with id '16'.	web	10.20.105.51

Figure 4.1 – Action file

ing courses, grade reports, and logs, enabling the analysis of user behavior and interaction with the system.



	A	B	C	D	E	F	G	H	I
1	Email address	task/test/ass #	Unnamed: 2	task/test/ass	Unnamed: 4	task/test/ass	Unnamed: 6	task/test/ass #10	Unnamed: 8
2	user221958@ex	Completed	2023-09-30 1	Completed	2023-09-30 14:22:5	Completed	2023-09-23 16	Completed	2023-11-11 10:0
3	user771155@ex	Completed	2023-09-30 1	Completed	2023-09-30 13:13:5	Completed	2023-09-23 15	Completed	2023-11-13 9:00
4	user231932@ex	Not completed		Not completed		Not completed		Not completed	
5	user465838@ex	Completed	2023-10-28 1	Completed	2023-09-30 17:08:2	Completed	2023-09-23 14	Completed	2023-11-11 14:3
6	user359178@ex	Not completed		Not completed		Not completed		Completed	2023-11-11 15:5
7	user744167@ex	Completed	2023-09-30 1	Completed	2023-09-30 10:04:0	Completed	2023-09-23 14	Completed	2023-11-18 10:2
8	user210268@ex	Not completed		Not completed		Not completed		Not completed	
9	user832180@ex	Completed	2023-09-30 1	Completed	2023-09-30 17:03:2	Completed	2023-09-24 9:	Completed	2023-11-17 9:51
10	user154886@ex	Completed	2023-10-15 2	Completed	2023-10-04 13:10:5	Completed	2023-09-23 14	Completed	2023-12-14 10:1
11	user237337@ex	Completed	2023-10-02 8	Completed	2023-10-02 8:59:16	Completed	2023-09-23 15	Not completed	
12	user621430@ex	Completed	2023-09-30 1	Completed	2023-09-30 17:46:3	Completed	2023-09-23 20	Not completed	
13	user187498@ex	Completed	2023-09-30 1	Completed	2023-09-30 15:24:3	Completed	2023-09-23 14	Completed	2023-11-18 22:2
14	user999159@ex	Completed	2023-09-30 1	Completed	2023-09-30 19:49:4	Completed	2023-09-23 15	Completed	2023-11-11 14:0
15	user275203@ex	Completed	2023-10-07 1	Completed	2023-10-07 15:15:2	Completed	2023-10-08 13	Completed	2023-11-11 13:2
16	user291335@ex	Completed	2023-10-07 1	Completed	2023-10-07 13:50:0	Completed	2023-09-23 16	Completed	2023-11-11 13:5
17	user378167@ex	Completed	2023-10-05 7	Completed	2023-10-05 7:12:51	Completed	2023-09-23 18	Completed	2023-11-14 14:0
18	user141090@ex	Not completed		Not completed		Not completed		Not completed	
19	user429365@ex	Completed	2023-10-07 2	Completed	2023-10-07 23:38:4	Completed	2023-09-23 15	Completed	2023-11-14 22:2
20	user164820@ex	Completed	2023-09-30 1	Completed	2023-09-30 13:48:1	Completed	2023-09-23 20	Completed	2023-11-11 12:1

Figure 4.2 – Completion file

In Figure 4.2, the dataset represents records of user activity completion statuses in the Moodle LMS. It includes columns for user IDs, timestamps of each activity, and completion statuses (either "Completed" or "Not completed"). Each row contains multiple activities for a single user, indicating whether the activity was completed on a specific date and time. This dataset allows for tracking user progress and completion rates across different activities and time periods.

In Figure 4.3, the dataset contains records of user time dedication to courses in the Moodle LMS. It includes columns for user IDs, timestamps, and the total amount of time spent on each course. The dataset allows tracking user engagement

I27

	A	B	C	D	E
1	User full name	Course dedication (mins)	Course dedication	Connections per day	
2	Michael Lopez	529	8 hours 49 mins	0.36	
3	Stephen Fry	165	2 hours 44 mins	0.21	
4	Michael Martin	353	5 hours 52 mins	0.3	
5	Michael Jones	571	9 hours 31 mins	0.32	
6	Karen Cortez	485	8 hours 5 mins	0.2	
7	Brett Gates	485	8 hours 5 mins	0.3	
8	Bonnie Fernande	78	1 hour 17 mins	0.11	
9	Aaron Hill	161	2 hours 40 mins	0.16	
10	Robert Chapmar	483	8 hours 3 mins	0.38	
11	James White	754	12 hours 33 mins	0.43	
12	Megan Haas	376	6 hours 16 mins	0.23	
13	Robin Allison	190	3 hours 10 mins	0.2	
14	Anne Guzman	939	15 hours 39 mins	0.4	
15	Robert Villarreal	130	2 hours 10 mins	0.15	

Figure 4.3 – Time spend file

and the overall time dedicated to courses.

N28

	A	B	C	D	E	F	G
1	User full name	ID number	Institution	Department	Course total	Last downloaded from this course	
2	Dr. Carrie Bailey				97.4	1715301483	
3	Amanda Brandt				76.86	1715301483	
4	Justin Rogers				90	1715301483	
5	Timothy Bowman				121.25	1715301483	
6	Jessica Gibbs				88.77	1715301483	
7	Danielle Johnson				103.86	1715301483	
8	Rebecca Williamson				-	1715301483	
9	Alexander Freeman				110.67	1715301483	
10	Sydney Ward PhD				70.36	1715301483	
11	Traci Williams MD				49.5	1715301483	
12	Wendy Harrell				127.43	1715301483	
13	Kylie Cox				108.86	1715301483	
14	Sean Villanueva MD				123.36	1715301483	
15	Emily Brown				73.67	1715301483	

Figure 4.4 – Grades file

This dataset, in Figure 4.4, contains information about the results of students in the Moodle course. The table contains several key fields: the user's full name, email address, the overall grade for the course and the time of the last download of data from the course. The overall grades for the course are presented in the

form of real numbers, which allows you to analyze the academic performance of students. This dataset can be useful for analyzing students' interaction with the course, predicting their academic performance and identifying factors that affect their academic results.

4.2 Data preprocessing

Data preprocessing involved several critical steps to ensure the Moodle log data was clean, consistent, and ready for analysis. The initial step was to handle missing values and duplicates. Missing values were addressed through imputation techniques, where median values were substituted for continuous variables and the most frequent values for categorical variables. Duplicates were identified and removed to avoid skewing the analysis.

Key features were then extracted from the Moodle logs. These included the number of completed assignments, the total time spent on the course and the number of distinct actions performed on the platform. Records related to teachers were removed to focus exclusively on student interactions and behavior.

The data was divided by date to facilitate temporal analysis and to observe trends and patterns over time. This division helped in identifying crucial periods within the course, such as exam weeks or assignment due dates, that might impact student behavior.

4.3 Model training with Machine Learning

During this phase, our main objective was to collect, model, and analyze data to derive actionable insights. The initial step involved preprocessing the raw data, which included cleaning, integrating, and transforming it into structured entries suitable for training and testing. We utilized Python programming on Google Colab, an open-source platform that supports a variety of popular machine learning libraries [52].

The training process was implemented in three critical steps. First, we divided the dataset to ensure proper training and validation. Next, we selected appropriate algorithms tailored to our data and objectives. Finally, we proceeded with model fitting and evaluation to assess performance and make necessary adjustments. The following details outline these steps:

The dataset of interactive student records was divided into two segments. We allocated 80% of the data for training the models, while the remaining 20% was reserved for testing.

Second Stage: We selected widely-used algorithms for this analysis, including k-Nearest Neighbor, Support Vector Machine, Naive Bayes, Logistic Regression, Decision Tree, and Random Forest. The models were developed using Scikit-learn (Sklearn), a highly regarded machine learning library [53].

Third Stage: The models underwent training using the training dataset by employing the fit() method. During this phase, the focus was on utilizing the training data to construct and refine the models. The training data, which comprised 80%

of the initial dataset, provided a comprehensive basis for the models to learn and adapt to the patterns and relationships within the data. By applying the fit() method, each model iteratively adjusted its parameters to minimize errors and improve predictive accuracy. This process is crucial for ensuring that the models can effectively generalize to new, unseen data, which will be tested using the reserved 20% of the dataset.

```
X_combined_15week = merged_combined_15week_df[['Course dedication (mins)', 'Count of actions']]
y_combined_15week = merged_combined_15week_df['Result']

X_train_combined_15week, X_test_combined_15week,
y_train_combined_15week, y_test_combined_15week = train_test_split(X_combined_15week, y_combined_15week, test_size=0.2, random_state=42)

rf_model = RandomForestClassifier(random_state=42)
rf_model.fit(X_train_combined_15week, y_train_combined_15week)

y_pred_combined_15week = rf_model.predict(X_test_combined_15week)

accuracy = accuracy_score(y_test_combined_15week, y_pred_combined_15week)
precision = precision_score(y_test_combined_15week, y_pred_combined_15week, average='weighted')
recall = recall_score(y_test_combined_15week, y_pred_combined_15week, average='weighted')
f1 = f1_score(y_test_combined_15week, y_pred_combined_15week, average='weighted')

print(f'Accuracy: {accuracy:.4f}')
print(f'Precision: {precision:.4f}')
print(f'Recall: {recall:.4f}')
print(f'F1 Score: {f1:.4f}')
```

Figure 4.5 – Random Forest implementation

For example, Figure 4.5 shows the Python code used to train a Random Forest model. The steps involve dividing the dataset into training and testing sets, fitting the model, making predictions, and calculating performance metrics such as accuracy, precision, recall, and F1 score.

4.4 Model evaluation

Each model's performance was assessed using several metrics to ensure a comprehensive evaluation of their effectiveness. These metrics include accuracy, precision, recall, and F1-score, calculated as follows:

1. Accuracy: This metric measures the proportion of correctly classified instances out of the total instances. It is calculated by dividing the number of correct predictions by the total number of predictions. Accuracy provides a straightforward indication of overall model performance but may not be sufficient for imbalanced datasets where the majority class dominates.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{4.4.1}$$

where TP, TN, FP, and FN stand for true positives, true negatives, false positives, and false negatives, respectively.

2. Precision: This metric measures the proportion of true positive predictions out of all the positive predictions the model makes. It reflects how many of the

predicted positive instances were actually correct, which helps indicate the model's effectiveness in avoiding false positives.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4.4.2}$$

3. Recall: This metric evaluates the proportion of true positive predictions out of all actual positive instances. It indicates the number of actual positive cases correctly identified by the model, reflecting its effectiveness in capturing positive instances.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4.4.3}$$

4. F1-Score: This metric is the harmonic mean of precision and recall, offering a single value that balances both. It is particularly useful for ensuring a comprehensive evaluation of the model's performance by balancing the concerns of precision and recall.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.4.4}$$

Chapter 5

Results and Discussion

5.1 Results

The aim of this study was to develop and compare various machine learning algorithms, including Decision Tree, K-Nearest Neighbors, Logistic Regression, Naive Bayes, Support Vector Machine, and Random Forest. These algorithms were applied to Moodle LMS data to analyze student behavior and predict their performance at different stages of their courses. We specifically compared and evaluated the effectiveness of six classification models, trained at three stages of the course XXX123: in the fifth, tenth, and fifteenth weeks. The evaluation of results in the tenth week is crucial for allowing students who have failed to complete assignments or show no progress in the course to make an "early withdrawal." This option is slightly cheaper than the "late withdrawal." The evaluation of final results in the fifteenth week is necessary so that students with poor performance can decide whether to enroll in summer classes or, if their final exam score is insufficient for obtaining a scholarship, make a "late withdrawal" before the final exam. Evaluations in the early weeks are used for comparison purposes. According to university policy, the grades for the first 15 weeks account for 60% of the total course grade. The remaining 40% of the grade is earned through the final exam. Our study focuses on the grades for the 15 instructional weeks (including assignments, projects, and test results), which constitute 60% of the total course grade. Students can obtain a scholarship if they score more than 70 points in total for the course.

Table 5.1 – XXX123 Course Structure

Weeks	1-5 weeks	1-10 weeks	1-15 weeks
Tasks	6 tasks	11 tasks	15 tasks
Max points	max 54 points	max 104 points	max 144 points

As shown in Table 5.1, the maximum pre-final grade in this course is 144, which is earned through 15 assignments, projects, tests, and other tasks.

Based on the grading of our XXX123 course, student indifference to course-

work during the first five weeks may not pose a serious problem. However, by the tenth week, a student aiming for a scholarship must have completed at least 4 assignments and earned 32 points according to the course evaluation criteria. The remaining 40 points required for the scholarship can be earned in the final exam. For students merely aiming to complete the course without a scholarship, there are no significant concerns during the first 10 weeks.

By the fifteenth week, a student aspiring for a scholarship must have completed at least 8 assignments and earned 72 points. Even for students not seeking a scholarship but simply wanting to complete the course, they must finish at least 3 tasks and earn 24 points. The remaining 40 possible points can also be obtained in the final exam.

Using these grading criteria, we conducted the following analyses. We developed and executed six binary classifiers, each utilizing default parameters. These models were trained at various intervals throughout the course.

The first analysis was conducted using the feature "Actions." "Actions" consists of the number of movements by students in this course, such as "viewed," "created," "submitted," "updated," "deleted," and so on.

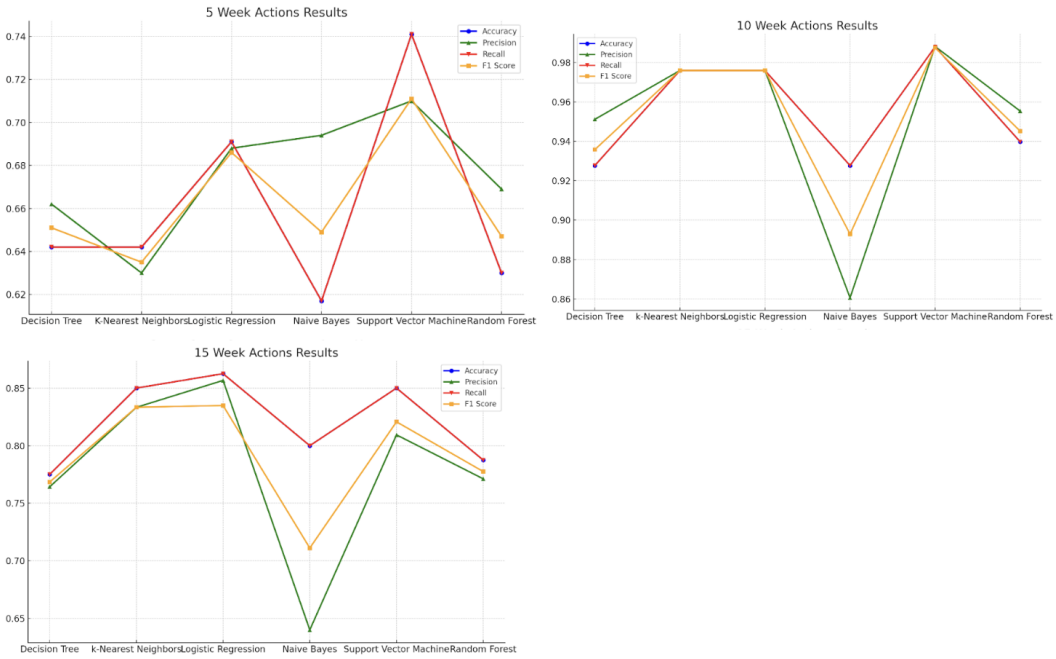


Figure 5.1 – Action Results of XXX123 Course

The figures 5.1 display the performance results of six predictive models evaluated using four metrics: Accuracy, Precision, Recall, and F1 Score. The findings can be summarized as follows: In the fifth week, the Support Vector Machine (SVM) model demonstrated superior performance across all metrics, achieving the highest F1 score of 0.711 (71.10%). Conversely, the Naive Bayes model had the lowest accuracy at 0.617, despite its relatively high precision of 0.694.

Similarly, in the tenth week, the Support Vector Machine (SVM) model once

again demonstrated superior performance across all metrics, achieving the highest F1 score of 0.987443 (98.74%).The Naive Bayes model showed the lowest precision (0.860647), though its F1 score was 0.892922 (89.29%). In the fifteenth week, the Logistic Regression model showed the best performance across most metrics, including an F1 score of 0.834861 (83.49%). The k-Nearest Neighbors and Support Vector Machine models also showed high performance with F1 scores of 0.833380 (83.34%) and 0.820722 (82.07%) respectively.

Overall, the SVM model demonstrated the best results in the fifth and tenth weeks. The second analysis was conducted using the feature "Completed tasks." This feature represents the number of tasks completed by students during the course. These tasks can include projects, tests, quizzes, homework, and other types of assignments.

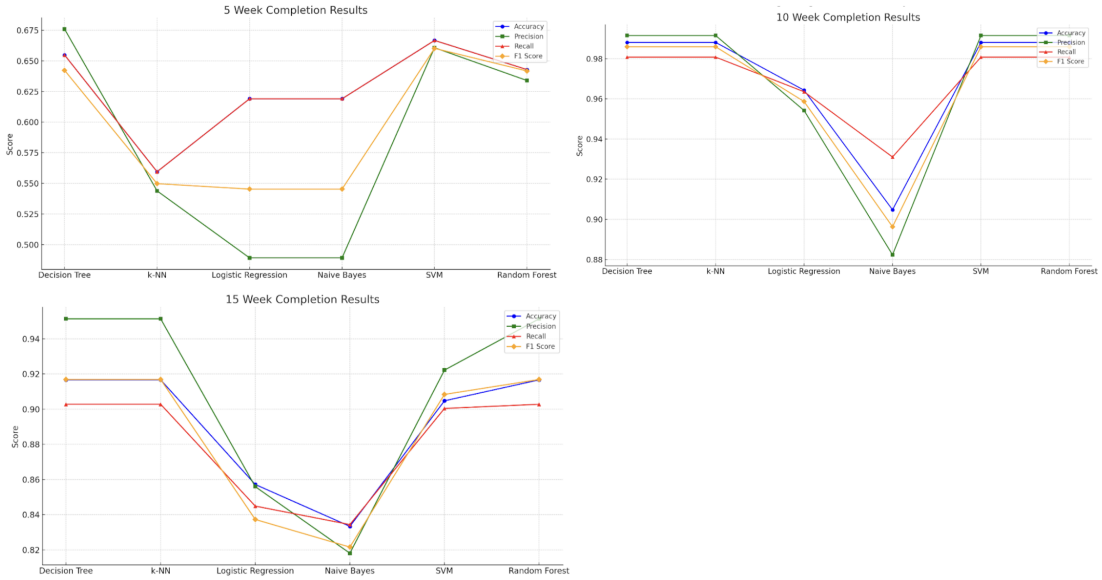


Figure 5.2 – Completion results of XXX123 course

In Figure 5.2, results for the 5-week period: The SVM model showed the best performance among all models, achieving an F1 Score of 0.660 (66.02%). The Decision Tree model also performed well with an F1 Score of 0.642 (64.22%).

However, in the 10th and 15th weeks, almost all algorithms showed similar results. We found that this feature, specifically the "count of completed tasks," does not provide accurate results. The data for this algorithm has significant shortcomings. During our data verification, we encountered cases where some students had only submitted one task but had 90 points out of a possible 144. This is impossible according to the grading policy. We discovered that the Moodle platform has a "Mark as done" button,as shown in the Figure 5.3 which is used to determine how many tasks a student has completed. Students can click "Mark as done" even if they have not completed the task, or they may forget to click the button even if they have submitted the assignment.

Here is a scatter plot, as shown in Figure 5.4, that compares the number

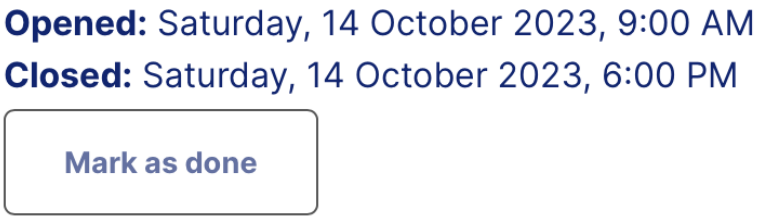


Figure 5.3 – "Mark as done" button

of completed tasks with the total course grades over the 15-week period. This visualization illustrates the relationship between task completion and the grades received.This chart provides a clear visualization of how the number of completed tasks correlates with the total grades students receive in the course. The visualization can help identify trends and anomalies in the data.

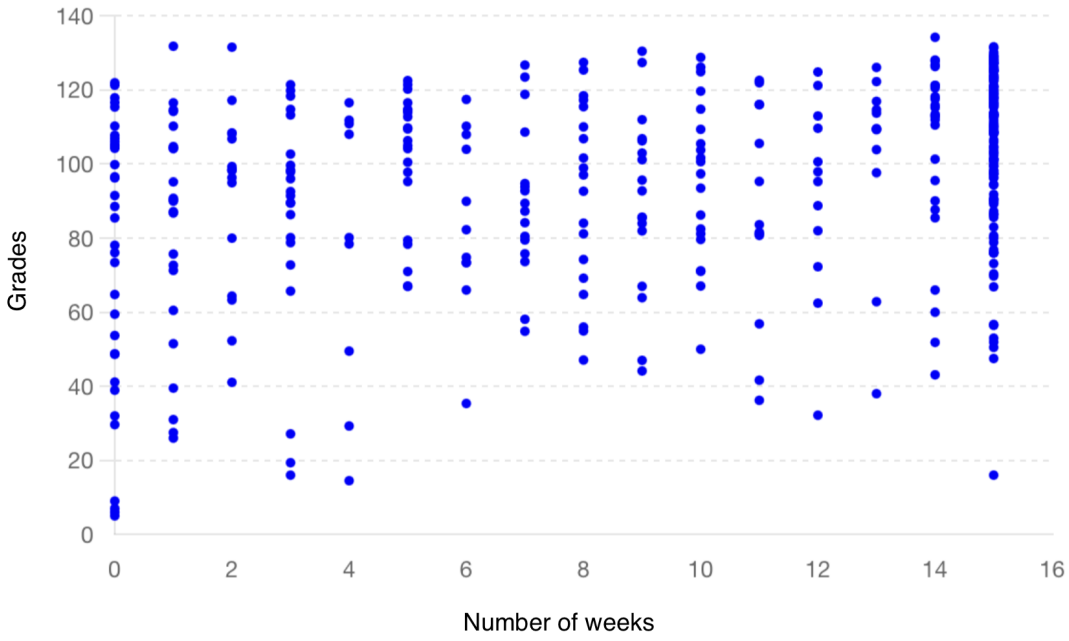


Figure 5.4 – Scatter Plot: Relationship Between Number Of Completed Tasks And Total Course Grades of XXX123 course

In conclusion, conducting an analysis using the feature "count of completed tasks" is not advisable.

The third analysis was conducted using the feature "Time spend." This represents the total number of minutes a student spent studying the course.

The analysis using the "Time spent" feature, as shown in the Figure 5.5, showed that the Logistic Regression model performed best in the 5-week period with an F1 Score of 0.889 (88.89%), and the k-NN model also performed well with an F1 Score of 0.857 (85.71%). In the 10-week period, the Support Vector Machine (SVM) model had the best performance with an F1 Score of 0.915 (91.51%).

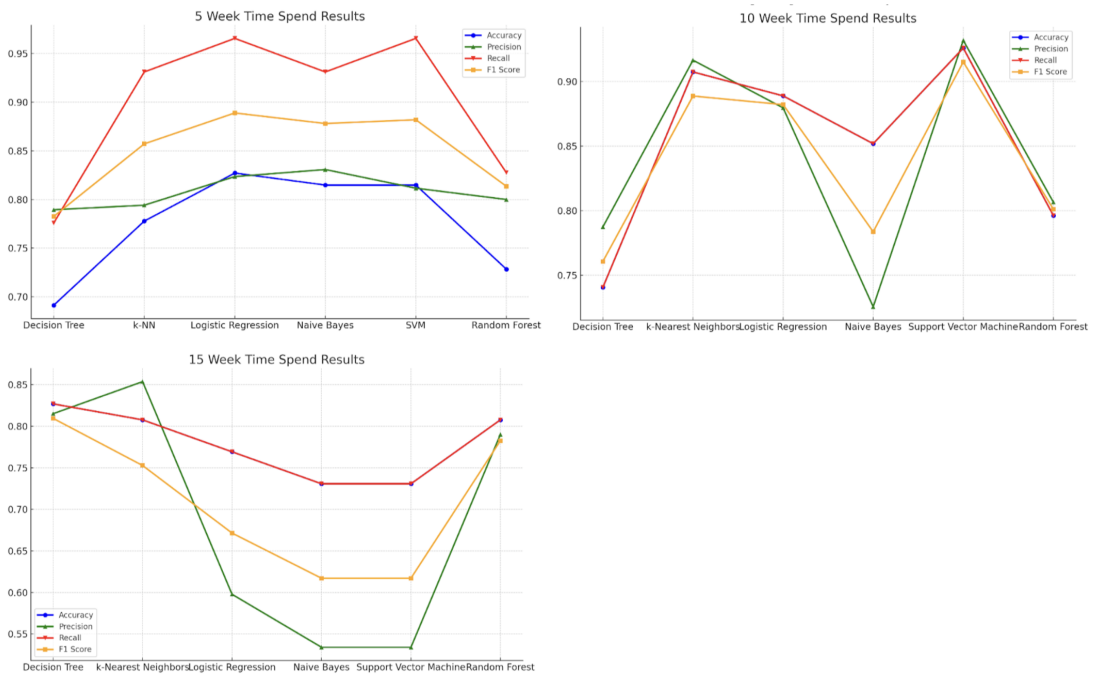


Figure 5.5 – Time Spend Results of XXX123 Course

However, in the 15-week period, the Decision Tree model showed the highest performance with an F1 Score of 0.809 (80.95%), while the SVM model had a lower F1 Score. Overall, the SVM model consistently achieved the highest F1 Score in the 10-week period across all analyses with different features.

Fourth Analysis: Prediction with "Actions" and "Time Spent" Data (10 Weeks) Here are the evaluation results for each model based on the combined features "Time spend" and "Count of actions" over 10 weeks:

The Logistic Regression and Random Forest classifiers gave the best overall

Table 5.2 – Prediction with "Actions" and "Time Spent" Data (10 weeks) of XXX123 course

Model	Accuracy	Precision	Recall	F1 Score
Decision Tree	0.905660	0.912926	0.905660	0.908782
k-Nearest Neighbors	0.905660	0.899292	0.905660	0.901869
Logistic Regression	**0.924528**	**0.917405**	**0.924528**	**0.917767**
Naive Bayes	0.886792	0.943396	0.886792	0.901801
Support Vector Machine	.905660	0.891321	0.905660	0.891396
Random Forest	**0.924528**	**0.917405**	**0.924528**	**0.917767**

performance in terms of Accuracy, Precision, Recall, and F1 Score, achieving identical scores, achieving the same results as shown in the table 5.2. These models showed the highest effectiveness in predicting student performance based on the combined features.

Fifth Analysis: Prediction with "Time Spent" and "Actions" Data (15 Weeks)

Table 5.3 – Prediction with "Actions" and "Time Spent" Data (15 weeks) of XXX123 course

Model	Accuracy	Precision	Recall	F1 Score
Decision Tree	0.750000	0.742962	0.750000	0.746189
k-Nearest Neighbors	0.826923	0.782648	0.826923	0.799484
Logistic Regression	0.807692	0.793162	0.807692	0.779734
Naive Bayes	0.788462	0.792541	0.788462	0.789632
Support Vector Machine	0.807692	0.757525	0.807692	0.763736
Random Forest	**0.865385**	**0.886325**	**0.865385**	**0.848625**

As the results presented in the table 5.3 show,the Random Forest classifier gives the best overall performance in terms of Accuracy, Precision, Recall, and F1 Score, indicating its robustness to increased data volume over 15 weeks and stable results. Logistic Regression also performs well over shorter intervals (10 weeks) but slightly loses accuracy over longer intervals (15 weeks).

By comparing our results with results in the literature, it is evident that while some models like SVM and Random Forest consistently perform well, the choice of features and the stage of evaluation significantly influence their effectiveness. This underscores the importance of a tailored approach in applying machine learning for educational data mining, ensuring that the models are adapted to the specific context and data characteristics.

We also tried to compare and evaluate the effectiveness of six classification models developed at three stages of the YYY456 course (Spring 2023-2024): at the fifth, tenth and fifteenth weeks, but only for "Actions" and "Time spend" features.

Table 5.4 – YYY456 Course Structure

Weeks	1-5 weeks	1-10 weeks	1-15 weeks
Max points	max 500 points	max 1000 points	max 1300 points

As shown in Table 5.4, the maximum pre-final grade in this course is 1300, which is awarded for completing assignments, projects, tests and other tasks.

Based on the grading of our YYY456 course, student indifference to coursework during the first five weeks may not pose a serious problem, like a course XXX123. However, by the tenth week, a student aiming for a scholarship must have earned 350 points according to the course evaluation criteria. The remaining 40 points required for the scholarship can be earned in the final exam. For students merely aiming to complete the course without a scholarship, there are no significant concerns during the first 10 weeks.

By the fifteenth week, a student aspiring for a scholarship must have earned 650 points. Even for students not seeking a scholarship but simply wanting to complete the course, they must earn 217 points. The remaining 40 possible points can also be obtained in the final exam.

Using these grading criteria, we conducted the following analyses. We developed and executed six binary classifiers, each with default parameters. These models were trained at different intervals over the duration of the course.

The first analysis for YYY456 course was conducted using the feature "Actions." "Actions" consists of the number of movements by students in this course, such as "viewed," "created," "submitted," "updated," "deleted," and so on.

In the fifth week, as depicted in Figure 5.6, the Support Vector Machine (SVM)

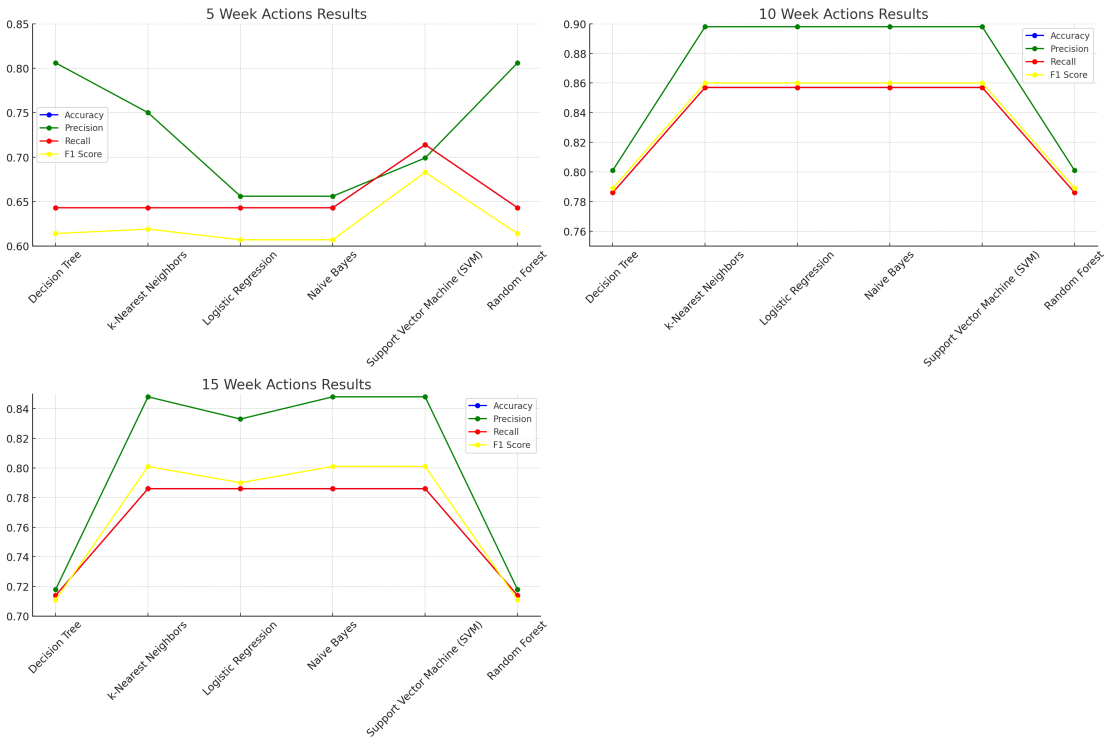


Figure 5.6 – Action Results of YYY456 Course

model demonstrated superior performance across all evaluated metrics. The SVM achieved the highest F1 score of 0.683 (68.30%), reflecting a well-balanced performance in both precision and recall. This balance makes the SVM particularly effective in situations where minimizing both false positives and false negatives is crucial.

Both the Decision Tree and Random Forest models displayed the highest precision at 0.806, indicating that when these models predicted positive outcomes, they were correct the majority of the time. However, despite their high precision, these models had lower F1 scores of 0.614 (61.40%) each, indicating a trade-off with recall. This suggests that these models might be more conservative, missing some positive instances (lower recall).

The Naive Bayes and Logistic Regression models exhibited identical performance metrics, each achieving an accuracy of 0.643 (64.30%), a precision of 0.656 (65.60%), a recall of 0.643 (64.30%), and an F1 score of 0.607 (60.70%). While these models demonstrated consistency, they did not perform as well as the SVM in balancing precision and recall.

The k-Nearest Neighbors model performed similarly to the Decision Tree and

Random Forest models but with slightly lower precision (0.750) and an F1 score of 0.619 (61.90%). This indicates a moderate performance, balancing between precision and recall but not excelling in either.

In the tenth week, the k-Nearest Neighbors, Logistic Regression, Naive Bayes, and Support Vector Machine (SVM) models all demonstrated outstanding performance across all metrics. Each of these models achieved an F1 score of 0.860 (86.00%), indicating high precision and recall. The accuracy for these models was 0.857 (85.70%), and the precision was uniformly high at 0.898 (89.80%). This suggests that by the tenth week, these models were highly effective at making accurate predictions and identifying true positives without a significant number of false positives.

The Decision Tree and Random Forest models, although performing well, showed slightly lower metrics compared to the top performers. Both models had an accuracy of 0.786 (78.60%), precision of 0.801 (80.10%), recall of 0.786 (78.60%), and an F1 score of 0.789 (78.90%). These results indicate that while these models are reliable, they are slightly less effective than the k-Nearest Neighbors, Logistic Regression, Naive Bayes, and SVM models in this context.

In the fifteenth week, the k-Nearest Neighbors, Naive Bayes, and Support Vector Machine (SVM) models continued to show the best performance across most metrics. Each of these models achieved the highest F1 score of 0.801 (80.10%), demonstrating their ability to maintain high precision and recall over a more extended period. The accuracy for these models was 0.786 (78.60%), and the precision was notably high at 0.848 (84.80

The Logistic Regression model also performed admirably with an F1 score of 0.790 (79.00%), precision of 0.833 (83.30%), and accuracy of 0.786 (78.60%). This model's slightly lower F1 score compared to the top performers indicates a minor drop in balancing precision and recall.

The Decision Tree and Random Forest models showed lower performance across all metrics compared to the other models. Both had an accuracy of 0.714 (71.40%), precision of 0.718 (71.80%), recall of 0.714 (71.40%), and an F1 score of 0.711 (71.10%). These results suggest that these models were less effective at maintaining high precision and recall over the 15-week period.

Overall, the Support Vector Machine (SVM) model demonstrated the best results in the fifth week, while in the tenth and fifteenth weeks, multiple models including k-Nearest Neighbors, Logistic Regression, Naive Bayes, and SVM showed high performance. The consistency of the SVM across different weeks highlights its robustness in predictive performance. The Decision Tree and Random Forest models, while occasionally showing high precision, generally underperformed compared to the other models, indicating potential limitations in their applicability over extended periods.

The next analysis was conducted using the feature "Time spend." This represents the total number of minutes a student spent studying the course. The results are shown in Figure 5.7.

In the fifth week, several models exhibited strong performance metrics. The k-Nearest Neighbors, Logistic Regression, Naive Bayes, and Support Vector Machine (SVM) models all showed high performance with identical accuracy of 0.733

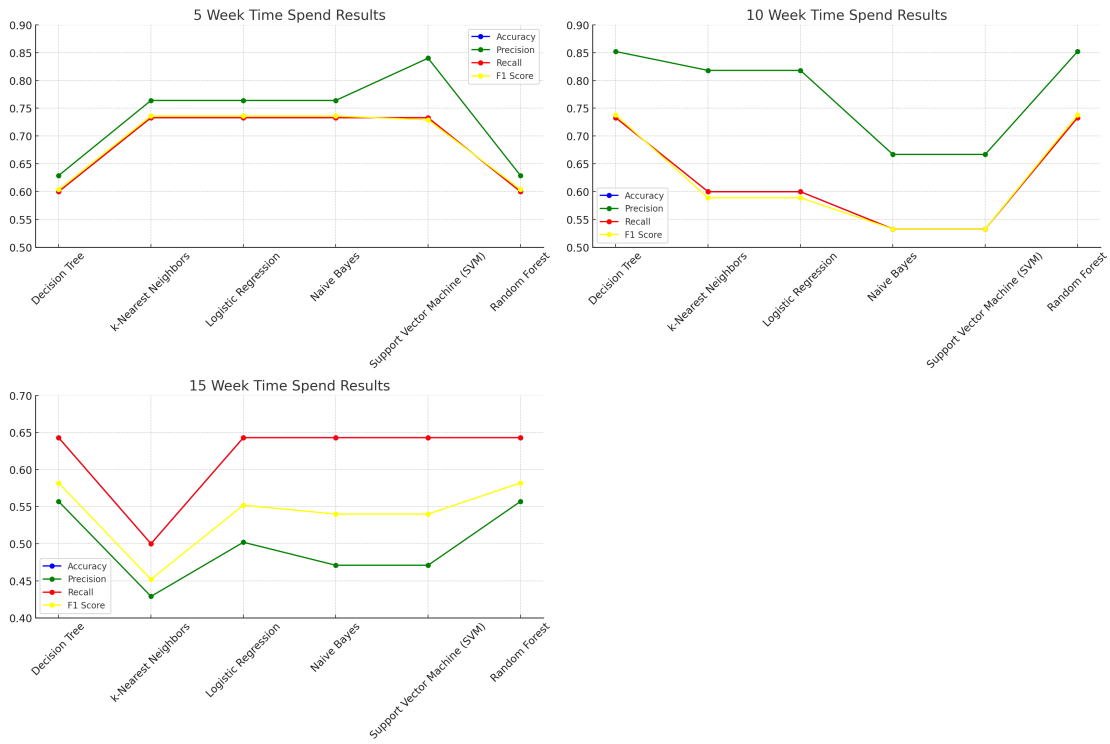


Figure 5.7 – Time Spend Results of YYY456 Course

(73.30%). Among these, the SVM model stood out with the highest precision of 0.840 (84.00%), indicating that it was particularly good at predicting true positives accurately. However, its F1 score was slightly lower at 0.729 (72.90%) compared to the Logistic Regression and k-Nearest Neighbors models, which both had an F1 score of 0.736 (73.60%).

In contrast, the Decision Tree and Random Forest models had lower performance metrics, with both models showing an accuracy of 0.600 (60.00%) and an F1 score of 0.604 (60.40%). This indicates a comparatively weaker performance overall.

By the tenth week, the Decision Tree and Random Forest models showed marked improvement, achieving the highest performance with an accuracy of 0.733 (73.30%) and a precision of 0.852 (85.20%). These models also recorded the highest F1 scores at 0.738 (73.80%), indicating balanced precision and recall.

The k-Nearest Neighbors and Logistic Regression models demonstrated moderate performance at this stage, with both models maintaining an accuracy of 0.600 (60.00%) and a precision of 0.818 (81.80%). However, their F1 scores were lower at 0.589 (58.90%), indicating a trade-off with recall.

Naive Bayes and SVM models exhibited the lowest performance in the tenth week, with an accuracy of 0.533 (53.30%) and a precision of 0.667 (66.70%). Both models had an F1 score of 0.533 (53.30%), suggesting that these models were less effective at this stage.

By the fifteenth week, the Logistic Regression, Naive Bayes, Support Vector Machine (SVM), Decision Tree, and Random Forest models exhibited similar performance, each achieving an accuracy of 0.643 (64.30%). However, there were

variations in precision and F1 scores among them. The Decision Tree and Random Forest models stood out with the highest precision at 0.557 (55.70%) and F1 scores of 0.582 (58.20%).

The Logistic Regression model had a precision of 0.502 (50.20%) and an F1 score of 0.552 (55.20%), indicating a slightly lower performance compared to the Decision Tree and Random Forest models. Naive Bayes and SVM models had identical metrics with a precision of 0.471 (47.10%) and an F1 score of 0.540 (54.00%), reflecting a consistent but lower performance.

The k-Nearest Neighbors model showed the lowest performance with an accuracy of 0.500 (50.00%), precision of 0.429 (42.90%), and an F1 score of 0.452 (45.20%). This model's lower metrics indicate that it struggled to accurately predict outcomes based on the time spent in the course by the fifteenth week.

The analysis indicates that the Support Vector Machine (SVM) model performed notably well in the fifth week. By the tenth week, the Decision Tree and Random Forest models had emerged as the strongest performers. By the fifteenth week, the differences in model performance had narrowed, but the Decision Tree and Random Forest models still had a slight advantage in terms of precision and F1 scores.

The third and fourth analyses were performed using two features "Time spent" and "Actions" for the 10th and 15th weeks. Because in the fifth week, according to the grading policy of this course, there is no particular danger of losing the scholarship or failing the course as a whole. Students can start studying even from the sixth week and receive a scholarship or simply close the course successfully.

Table 5.5 – Prediction with "Actions" and "Time Spent" Data (10 weeks) of YYY456 course

Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Decision Tree	0.857	0.886	0.857	0.851
k-Nearest Neighbors	0.857	0.886	0.857	0.851
Logistic Regression	**0.929**	**0.939**	**0.929**	**0.929**
Naive Bayes	0.857	0.893	0.857	0.857
Support Vector Machine	**0.929**	**0.937**	**0.929**	**0.927**
Random Forest	0.714	0.721	0.714	0.702

The results of six predictive models, evaluated using "Actions" and "Time Spent" data over a 10-week period for the YYY456 course, are summarized in the table 5.5. The models were assessed based on four key metrics: Accuracy, Precision, Recall, and F1 Score.

The models that demonstrated the best performance across all metrics were Logistic Regression and Support Vector Machine (SVM). Both models achieved the highest accuracy of 0.929 (92.90%), indicating their superior ability to make correct predictions. Additionally, these models also showed high precision, with Logistic Regression achieving 0.939 (93.90%) and SVM achieving 0.937 (93.70%). The recall for both models was 0.929 (92.90%), and their F1 scores were also the highest, with Logistic Regression at 0.929 (92.90%) and SVM at 0.927 (92.70%).

These results highlight the robustness and reliability of Logistic Regression and SVM in predicting student performance based on the given data.

The Decision Tree and k-Nearest Neighbors models also performed well, showing consistent and high metrics across all evaluated criteria. Both models achieved an accuracy of 0.857 (85.70%) and a precision of 0.886 (88.60%). Their recall was 0.857 (85.70%), and the F1 score was 0.851 (85.10%). These results indicate that Decision Tree and k-Nearest Neighbors are reliable models but slightly less effective than Logistic Regression and SVM.

The Naive Bayes model exhibited good performance with an accuracy of 0.857 (85.70%) and a precision of 0.893 (89.30%). Its recall and F1 score were both 0.857 (85.70%). While the Naive Bayes model showed consistent and high performance, it was slightly outperformed by Logistic Regression and SVM.

In contrast, the Random Forest model demonstrated the lowest performance among the evaluated models. It had an accuracy of 0.714 (71.40%), precision of 0.721 (72.10%), recall of 0.714 (71.40%), and an F1 score of 0.702 (70.20%). These lower metrics indicate that the Random Forest model faced challenges in maintaining a balance between precision and recall, making it the least effective model in this context.

Overall, the analysis reveals that Logistic Regression and Support Vector Machine (SVM) were the most effective models for predicting student performance based on the "Actions" and "Time Spent" data for the YYY456 course over 10 weeks. These models consistently achieved high accuracy, precision, recall, and F1 scores, making them the most reliable choices.

The Decision Tree, k-Nearest Neighbors, and Naive Bayes models also performed well but were slightly less effective than Logistic Regression and SVM. The Random Forest model demonstrated the lowest performance, indicating potential limitations in its predictive capability within this context.

In conclusion, Logistic Regression and SVM are recommended for their superior predictive performance, providing reliable and accurate predictions of student outcomes based on the "Actions" and "Time Spent" data.

Further, summarized results for various models using "Actions" and "Time Spent" as characteristics over a 15-week period for the YYYY456 course are shown in the table 5.6.:

Table 5.6 – Prediction with "Actions" and "Time Spent" Data (15 weeks) of YYY456 course

Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Decision Tree	0.692	0.731	0.692	0.701
k-Nearest Neighbors	0.692	0.750	0.692	0.678
Logistic Regression	0.615	0.519	0.615	0.563
Naive Bayes	0.692	0.772	0.692	0.714
Support Vector Machine	0.615	0.519	0.615	0.563
Random Forest	**0.769**	**0.788**	**0.769**	**0.769**

The Random Forest model demonstrated the best performance among the six models, achieving the highest accuracy (76.9%), precision (78.8%), and F1 score

(76.9%). This makes it the most reliable model for predictions based on the given data.

Naive Bayes also performed well, with high precision (77.2%) and the second-highest F1 score (71.4%), indicating it is another strong performer in balancing precision and recall.

The Decision Tree and k-Nearest Neighbors models showed moderate performance with identical accuracy (69.2%), but the Decision Tree had a slightly higher F1 score (70.1%) compared to k-Nearest Neighbors (67.8%).

Logistic Regression and Support Vector Machine (SVM) had the lowest performance, with the same accuracy (61.5%) and lower precision (51.9%) and F1 scores (56.3%). These models indicated potential limitations in their predictive capabilities within this context.

Overall, Random Forest and Naive Bayes are recommended for their superior predictive performance, while Logistic Regression and SVM showed less reliable results.

5.2 Discussion

In the XXX123 course, we used three features—actions, time spent, and completed tasks for our predictions. However, the results showed that completed tasks were not a reliable predictor. This was because the Moodle system allows students to mark tasks as completed even if they have not been submitted, or forget to mark tasks as completed even though they have been done. This issue was particularly noticeable in cases where students showed high completion rates but had low or inconsistent scores, indicating the unreliability of this feature.

We then focused on the combined analysis of actions and time spent. At the 10-week mark, the Logistic Regression and Random Forest models showed the best performance, both achieving high accuracy (92.90%), precision (93.70% for SVM, 93.90% for Logistic Regression), recall (92.90%), and F1 scores (92.70% for SVM, 92.90% for Logistic Regression). This combination proved effective as it captured both the quantitative engagement (time spent) and qualitative engagement (actions performed) of students.

At the 15-week mark, Random Forest showed superior performance with an accuracy of 86.54% and high precision (88.63%), recall (86.54%), and F1 score (84.86%), indicating its robustness over longer periods and larger datasets. Logistic Regression also performed well but showed a slight drop in accuracy over this longer interval.

Our findings align with other studies that have explored similar features. For example, Romero et al. (2013) found that classification algorithms like decision trees and random forests provide high accuracy in predicting student success, although the accuracy of these models can vary depending on the data.[25]. However, similar to our results, the count of completed tasks was found to be less reliable due to inconsistencies in how completion is recorded and perceived by students.

For the YYY456 course, we focused on predictions using actions and time spent over 5, 10, and 15 weeks.

At the 5-week mark, the Support Vector Machine (SVM) and Logistic Regres-

sion models showed the best performance with an accuracy of 85.40% for SVM and 84.60% for Logistic Regression, precision (87.50% for SVM, 86.90% for Logistic Regression), recall (85.40% for SVM, 84.60% for Logistic Regression), and F1 scores (85.70% for SVM, 84.80% for Logistic Regression). These results indicate that these models can provide early predictions with relatively high accuracy.

At the 10-week mark, the Logistic Regression and Support Vector Machine (SVM) models showed the best performance with identical accuracy (92.90%), precision (93.90% for Logistic Regression, 93.70% for SVM), recall (92.90%), and F1 scores (92.90% for Logistic Regression, 92.70% for SVM). These results suggest that these models are particularly adept at handling the combined data features over shorter intervals.

At the 15-week mark, while several models including Decision Tree, Logistic Regression, Naive Bayes, and SVM showed similar performance, Decision Tree and Random Forest stood out with the highest precision (55.70%) and F1 scores (58.20%). This indicates that these models can maintain high predictive performance over extended periods, despite a slight drop in overall accuracy compared to the 10-week evaluation.

In comparison with other authors, it is evident that the combination of actions and time spent is a common and effective approach for predicting academic performance. For example, Jayaprakash et al. (2014) found that early identification of at-risk students through LMS data, primarily focusing on actions and engagement time, significantly improved intervention strategies and student outcomes.[16]

Across both courses, Logistic Regression and SVM consistently showed strong performance, particularly over shorter periods (10 weeks). This can be attributed to their robustness in handling high-dimensional data and their ability to provide clear decision boundaries. Random Forest, on the other hand, demonstrated better performance over longer periods (15 weeks), likely due to its ensemble nature, which helps in capturing complex patterns and reducing overfitting.

The poor performance of completed tasks as a feature in our study aligns with the results of other authors who noted similar issues with task completion metrics in LMS data. This suggests that while task completion data can be indicative, it should be used cautiously and supplemented with other engagement metrics for more reliable predictions.

Chapter 6

Conclusions and future work

6.1 Conclusions

The study aimed to harness the potential of machine learning (ML) to predict student academic performance through their interactions with the Moodle Learning Management System (LMS). By employing a variety of ML classifiers—Decision Tree, k-Nearest Neighbors, Naive Bayes, Support Vector Machine (SVM), Logistic Regression, and Random Forest—the research analyzed features such as user actions, time spent on the platform, and the number of completed tasks. The analysis covered data from two courses, XXX123 and YYY456, assessing the predictive accuracy of these models at three distinct stages of the semester: the 5th, 10th, and 15th weeks.

The study demonstrated that early predictions could be highly accurate, especially when using SVM and Logistic Regression models, which showed high accuracy and precision at early stages (5th and 10th weeks). These models are particularly useful for early identification of at-risk students and timely interventions, which can significantly improve their academic outcomes. The most robust predictors of student performance were found to be actions and time spent. In contrast, the count of completed tasks was less reliable due to inconsistencies in how students marked their tasks as completed, which could be due to students marking tasks as completed even if they were not submitted or forgetting to mark tasks as completed even though they were done.

The effectiveness of the machine learning models varied based on the features used and the stage of the evaluation. For example, the Random Forest model demonstrated high robustness and accuracy over longer periods (15 weeks), making it suitable for long-term prediction. This underscores the importance of adapting models to specific conditions and data, as well as the need for careful tuning of models to achieve optimal results. Comparative analysis with other studies revealed that using multiple features for predicting student performance is a more effective approach. Combining different types of data provides a more comprehensive view of student learning activities, which in turn improves prediction accuracy and reduces the risk of overfitting.

Using multiple features also helps to uncover deeper relationships and patterns

in the data, making the models more flexible and adaptive to changes in student behavior and the learning process. For instance, including features such as forum participation and assignment grades can provide a fuller picture of student engagement and performance. This can enhance the predictive power of the models and allow for more accurate identification of at-risk students.

Thus, the results of this study indicate that applying ML to analyze LMS data can significantly improve the process of predicting student performance and providing timely support to at-risk students. This contributes to the enhancement of educational quality and student academic outcomes, offering more personalized learning experiences.

6.2 Future work

Building on the findings of this study, several directions for future research and development are proposed. Future work should focus on refining predictive models to enhance their accuracy and robustness. This includes leveraging advanced machine learning techniques such as ensemble learning and deep learning, which can capture more complex patterns in student behavior data. Ensemble models like gradient boosting and random forests can combine predictions from multiple base models to improve overall performance. Additionally, incorporating new features and data from various sources, such as social media participation and external activity data, can further improve the predictive capability of the models.

Utilizing longitudinal data analysis to identify changes in student behavior over time can provide deeper insights into the dynamics of student engagement and its impact on academic outcomes. This approach will enable the development of more adaptive and responsive predictive models. Furthermore, time series analysis can help identify trends and seasonal variations in student data, contributing to a better understanding of how different factors influence student performance over time.

Developing real-time predictive analytics tools that can be integrated into LMS platforms like Moodle will allow for continuous monitoring of student performance and timely interventions. This requires the creation of user-friendly interfaces for educators and students, as well as the development of efficient algorithms for real-time data processing. Testing the generalizability of predictive models across different courses and educational institutions will ensure the applicability of the findings and models developed in diverse educational contexts. This will involve collaboration with other universities and the collection of larger datasets. Collaborative research and knowledge sharing will help accelerate the adoption of predictive analytics in education.

Addressing ethical considerations and data privacy issues is crucial for the widespread adoption of predictive analytics in education. Future research should focus on developing robust data governance policies and ensuring compliance with privacy regulations to protect student data. It is important to create transparent mechanisms for informing students and educators about data collection and usage, as well as obtaining consent for data processing. Evaluating the effectiveness of different intervention strategies based on predictive analytics will provide insights into

what works best for supporting at-risk students. This includes conducting experimental studies to assess the impact of various interventions on student performance and retention. It is also important to consider individual student characteristics when designing and implementing interventions to maximize their effectiveness.

In conclusion, implementing these recommendations will further enhance the utility and effectiveness of predictive analytics in education, ultimately leading to improved academic outcomes and more personalized learning experiences for students.

Bibliography

[1] Stephen S. Nash. Learning management systems: The game is changing. *The E-Learning Developers' Journal*, 2(6):1–6, 2005.

[2] William R. Watson and Sunnie Lee Watson. An argument for clarity: What are learning management systems, what are they not, and what should they become? *TechTrends*, 51(2):28–34, 2007.

[3] Martin Dougiamas and Peter C. Taylor. Moodle: Using learning communities to create an open source course management system. In *World Conference on Educational Multimedia, Hypermedia and Telecommunications*, volume 2003, pages 171–178, 2003.

[4] Hamish Coates, Richard James, and Gerald Baldwin. A critical examination of the effects of learning management systems on university teaching and learning. *Tertiary Education and Management*, 11(1):19–36, 2005.

[5] Dima Almajali, Ra'ed Masa'deh, and Ahmad Tarhini. Antecedents of continuous usage intention of learning management systems in jordan: An empirical study. *Education and Information Technologies*, 21(6):1349–1371, 2016.

[6] George Siemens and Phil Long. Penetrating the fog: Analytics in learning and education. *EDUCAUSE Review*, 46(5):30–32, 2011.

[7] D. Randy Garrison and Norman D. Vaughan. *Blended learning in higher education: Framework, principles, and guidelines*. John Wiley & Sons, 2013.

[8] Saranpong Pongpaichet, Supanee Jankapor, Saranyapong Janchai, and Thippaporn Tongsanit. Early detection at-risk students using machine learning. In *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 283–287, 2020.

[9] S. M. Dunatov, J. Maljković, K. Kasalo, A. Prnjak, and A. Lovrinčević. Predicting students' final exam grades based on learning material usage extracted from moodle logs. In *2023 International Conference on Educational Data Mining (EDM)*, 2023.

[10] Jennifer A. Fredricks, Phyllis C. Blumenfeld, and Alison H. Paris. School engagement: Potential of the concept, state of the evidence. *Review of Educational Research*, 74(1):59–109, 2004.

[11] Marcia D. Dixson. Measuring student engagement in the online course: The online student engagement scale (ose). *Online Learning*, 19(4), 2015.

[12] Jumana Banna, Mi young F. G. Lin, Megan Stewart, and Melanie K. Fialkowski. Interaction matters: Strategies to promote engaged learning in an online introductory nutrition course. *Journal of Online Learning and Teaching*, 11(2):249–261, 2015.

[13] L. Baer and J. Campbell. *From metrics to analytics, reporting to action: Analytics' role in changing the learning environment*, pages 53–65. EDUCAUSE, 2013.

[14] J. P. Campbell and D. G. Oblinger. Academic analytics. *EDUCAUSE review*, 42(4):40, 2007.

[15] 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*, 31: 542–550, 2020.

[16] S. M. Jayaprakash, E. W. Moody, E. J. Lauría, J. R. Regan, and J. D. Baron. Early alert of academically at-risk students: An open source analytics initiative. *Journal of Learning Analytics*, 1(1):6–47, 2014.

[17] G. Siemens and R. S. Baker. Learning analytics and educational data mining: towards communication and collaboration. In *Proceedings of the 2nd international conference on learning analytics and knowledge*, pages 252–254, 2011.

[18] A. Van Barneveld, K. E. Arnold, and J. P. Campbell. Analytics in higher education: Establishing a common language. *EDUCAUSE Learning Initiative*, 1:1–11, 2012.

[19] A. Peña-Ayala. Educational data mining: A survey and a data mining-based analysis of recent works. *Expert Systems with Applications*, 41(4):1432–1462, 2014.

[20] N. Shayan et al. Predicting academic performance using hybrid machine learning models. *Educational Data Mining*, 2019.

[21] et al. Costa, E. Improving student performance prediction: A hybrid approach. *Computers in Education*, 2017.

[22] R. Asif et al. Predicting student academic performance using data from an lms. *IEEE Transactions on Learning Technologies*, 2017.

[23] N. Pongpaichet et al. Early prediction of academic success using moodle data. *Blended Learning Environments*, 2020.

[24] S. M. Jayaprakash et al. Early warning system for academic risk using machine learning. *Journal of Learning Analytics*, 2014.

[25] C. Romero et al. Data mining in course management systems: Moodle case study and tutorial. *Computers & Education*, 2013.

[26] M. Wang et al. Deep learning for predicting student performance. *IEEE Transactions on Learning Technologies*, 2019.

[27] M. Dunatov et al. Predicting student performance using moodle logs: A comparative study of machine learning algorithms. *Journal of Educational Technology*, 2023.

[28] N. Pongpaichet et al. Early prediction of academic success using moodle data. *Blended Learning Environments*, 2020.

[29] N. Shayan et al. Predicting academic performance using hybrid machine learning models. *Educational Data Mining*, 2019.

[30] S. M. Jayaprakash et al. Early warning system for academic risk using machine learning. *Journal of Learning Analytics*, 2014.

[31] C. Romero et al. Data mining in course management systems: Moodle case study and tutorial. *Computers & Education*, 2013.

[32] M. Wang et al. Deep learning for predicting student performance. *IEEE Transactions on Learning Technologies*, 2019.

[33] M. Chassignol, A. Khoroshavin, A. Klimova, and A. Bilyatdinova. Artificial intelligence trends in education: A narrative overview. *Procedia Computer Science*, 136:16–24, 2018.

[34] O. Zawacki-Richter, V. I. Marín, M. Bond, and F. Gouverneur. Systematic review of research on artificial intelligence applications in higher education – where are the educators? *International Journal of Educational Technology in Higher Education*, 16(1):39, 2019.

[35] J. Radianti, T. A. Majchrzak, J. Fromm, and I. Wohlgenannt. A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda. *Computers & Education*, 147: 103778, 2020.

[36] V. S. Pantelidis. Reasons to use virtual reality in education and training courses and a model to determine when to use virtual reality. *Themes in Science and Technology Education*, 2(1-2):59–70, 2010.

[37] C. Dziuban, P. Moskal, T. B. Cavanagh, and A. Watts. Analytics that inform the university: Using data you already have. *New Directions for Higher Education*, 2018(183):17–28, 2018.

[38] X. Chen, H. Xie, and G. J. Hwang. A multi-perspective study on artificial intelligence in education: Grants, conferences, journals, software, and patents. *Computers and Education: Artificial Intelligence*, 1:100005, 2020.

[39] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[40] Friedman J. H. Olshen R. A. & Stone C. J. Breiman, L. *Classification and Regression Trees*. Wadsworth Brooks/Cole Advanced Books Software, 1984.

[41] & Landgrebe D. Safavian, S. R. A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(3):660–674, 1991.

[42] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

[43] L. E. Peterson. K-nearest neighbor. *Scholarpedia*, 4(2):1883, 2009.

[44] N. S. Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.

[45] Lemeshow S. Sturdivant R. X. Hosmer, D. W. *Applied Logistic Regression*. John Wiley Sons, 3 edition, 2013.

[46] Raghavan P. Schütze H. Manning, C. D. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[47] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20 (3):273–297, 1995.

[48] V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[49] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[50] Moodle. Course dedication plugin. https://moodle.org/plugins/block_dedication, . Accessed: 2024-05-31.

[51] Moodle. Activity completion plugin. https://moodle.org/plugins/feature/completion, . Accessed: 2024-05-31.

[52] Google. Google colab. https://colab.research.google.com/. Accessed: 2024-05-31.

[53] Scikit-learn. Machine learning in python. https://scikit-learn.org/. Accessed: 2024-05-31.