الجمهورية الجزائرية الديمقراطية الشعبية

People's Democratic Republic of Algeria

وزارة التعليم العالي والبحث العلمي

Ministry of Higher Education and Scientific Research

جامعة غرداية

University of Ghardaia

Registration n°
/…../…../…../…../…..

كلية العلوم والتكنولوجيا

Faculty of Science and Technology

قسم الرياضيات و الإعلام الآلي

Department of Mathematics and Computer Science

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

Master

Domain: Mathematis and Computer Science

Field : Computer Science

Specialty: Intelligent Systems for Knowledge Extraction

Topic

Improving Student Performance Prediction using Artificial Intelligence

Presented by:

Maria Kerboub & Aicha Heriat

Publicly defended on : 23/06/2024

Jury members :

Mr. Slimane Bellaouar	**MCA**	**Univ. Ghardaia**	**President**
Mr. Youcef Mahdjoub	**MAA**	**Univ. Ghardaia**	**Examiner**
Ms. Asma Bouchekouf	**MAA**	**Univ. Ghardaia**	**Supervisor**
Mr. Houssem Eddine Degha	**MCB**	**Univ. Ghardaia**	**Co-Supervisor**

Academic year : 2023/2024

Acknowledgment

We express our most sincere gratitude and thanks to the gracious and merciful God, who granted us the strength and courage to complete this modest work. Our thanks go, on the one hand, to Dr. Asma Bouchekouf and Dr. Houssem Eddine Degha for their guidance. We also thank our colleagues and acquaintances for their friendship and collaboration. Finally, we also thank all those who, directly or indirectly, contributed to the completion of this modest work.

I dedicate this work to my dear parents, may God protect them. To my dear siblings Amina, Asma, Ali, Hamza, Aicha, Mousa, Salah, And my sister-in-law Marwa and their children Soumaya, Rahaf, Al-Mu'tasim Billah, Jasser, Sidra, Raghad, Israa, Adam, and Mahomed and Raed_mounir. To all my professors, who contributed to our education. To all my friends Nadia, Marwa, Fatima, Nesrine, Aicha, Darine, and my dear Friend before my companion, Aicha.

Thank you.

KERBOUB MARIA

It is with great pleasure that I dedicate this work to my dear parents, my brothers Walid, Mohamed Abderazak, Ahmed Abdelhadi, to all the members of my family, my grandparents, my aunts and uncles, my cousins, and special thanks to all my professors, and my close friends, and my colleague and sister Maria. Thank you from the bottom of my heart.

HERIAT AICHA

ABSTRACT

In recent years, improvements in predicting student performance have garnered significant attention due to their applications in education, personalized academic support, and optimizing educational resources. Predicting student outcomes using artificial intelligence (AI) holds immense potential for teachers to proactively identify and support at-risk students. In this thesis, we explore this exciting field by employing various machine learning algorithms with an enhanced approach to model optimization. Specifically, we implement hyperparameter tuning using grid search to ensure optimal configurations for four classifiers: Support Vector Machines (SVM), Multilayer Perceptrons (MLP), Decision Trees, and Artificial Neural Networks (ANNs). This comprehensive analysis aims to identify the most effective model for predicting student performance in the selected dataset. The results reveal that the Artificial Neural Network model achieved an accuracy of 98%, demonstrating its superiority in performance prediction.

Keywords: Student performance prediction, Artificial Neural Networks (ANN), Support Vector Machines (SVM), Multilayer Perceptron (MLP), Decision Tree (DT), Machine Learning, Deep Learning, Grid Search.

RÉSUMÉ

Ces dernières années, les améliorations dans la prédiction des performances des étudiants ont suscité un intérêt considérable en raison de leurs applications dans l'éducation, le soutien académique personnalisé et l'optimisation des ressources éducatives. La prédiction des résultats des étudiants en utilisant l'intelligence artificielle (IA) offre un potentiel immense pour les enseignants afin d'identifier et de soutenir de manière proactive les étudiants à risque. Dans cette thèse, nous explorons ce domaine passionnant en utilisant divers algorithmes d'apprentissage automatique avec une approche améliorée pour l'optimisation des modèles. Plus précisément, nous mettons en œuvre un réglage des hyperparamètres en utilisant la recherche en grille pour garantir des configurations optimales pour quatre classificateurs : les machines à vecteurs de support (SVM), les perceptrons multicouches (MLP), les arbres de décision et les réseaux de neurones artificiels (ANN). Cette analyse complète vise à identifier le modèle le plus efficace pour prédire les performances des étudiants dans notre ensemble de données. Les résultats révèlent que le modèle de réseau de neurones artificiels a atteint une précision de 98 %, démontrant sa supériorité dans la prédiction des performances.

Mots-clés : Prédiction de la performance des étudiants, Réseaux de neurones artificiels, Machine à vecteurs de support, Perceptron multicouche, Apprentissage automatique, Apprentissage profond, Recherche en grille.

ملخص

في السنوات الأخيرة، حظيت تحسينات توقع أداء الطلاب باهتمام كبير نظراً لتطبيقاتها في التعليم، الدعم الأكاديمي الشخصي، وتحسين الموارد التعليمية. يُعتبر التنبؤ بنتائج الطلاب باستخدام الذكاء الاصطناعي ذو إمكانات هائلة للمدرسين لتحديد ودعم الطلاب المعرضين للخطر بشكل استباقي. في هذه الأطروحة، نستكشف هذا المجال المثير من خلال استخدام خوارزميات تعلم الآلة المختلفة مع نهج محسن لتعزيز النماذج. على وجه التحديد، نقوم بتنفيذ ضبط المعلمات الفائقة باستخدام البحث الشبكي لضمان تكوينات مثلى لأربعة مصنفات: آلات الدعم الفائق، الشبكات العصبية متعددة الطبقات، أشجار القرار، والشبكات العصبية الاصطناعية. تهدف هذه التحليل الشامل إلى تحديد النموذج الأكثر فعالية لتوقع أداء الطلاب في مجموعة البيانات الخاصة بنا. تكشف النتائج أن نموذج الشبكة العصبية الاصطناعية حقق دقة بنسبة 98%، مما يظهر تفوقه في توقع الأداء .

الكلمات المفتاحية: توقع أداء الطلاب، الشبكات العصبية الاصطناعية، آلة الدعم الفائق، الشبكات العصبية متعددة الطبقات، التعلم الآلي، التعلم العميق، البحث الشبكي، أشجار القرار

Contents

List of Figures

List of Tables

List of Acronyms

SVM Support Vector Machine

DL Deep Learning

ML Machine Learning

ANN Artificial Neural Network

SPP Student Performance Prediction

MLP Multilayer Perceptron

DT Decision Tree

KNN KNearest Neighbour

NB Naïve Bayes

LR Logistic Regression

NN Neural Network

DNN Deep Neural Network

AUC Area Under Curve

RF Random Forests

ROC Receiver Operating Characteristic

MSE Mean Squared Error

MAE Mean Absolute Error

R^2 R-squared

LMS Learning Management System

TLA Training Learning Architecture

AI Artificial Intelligence

SIS Student Information System

LDA Linear Discriminant Analysis

SVC Support Vector Classifier

ReLU Rectified Linear Unit

DM Data Mining

TPR True Positive Rate

TNR True Negative Rate

General Introduction

Context and Motivation

The field of education plays a pivotal role in the advancement of nations, serving as a fundamental tool for success in life. Educational institutions endeavor to provide quality education to their students to foster an enriching learning environment. The success of students has become a paramount metric for educational institutions. They continuously strive to enhance learning experiences, improve academic outcomes, and mitigate student attrition. A key strategy in achieving these goals is the prediction of student performance before examinations.

This brings us to the concept of student performance prediction, utilizing academic and demographic data, among other factors, to forecast future academic outcomes using machine learning(ML) algorithms. Existing solutions for predicting student performance primarily use traditional machine learning techniques such as Logistic Regression, Random Forest, and Support Vector Machines. While these models have demonstrated effectiveness [37], they often require extensive feature engineering and may not capture complex data relationships. Furthermore, deep learning models like Artificial Neural Networks(ANN) have shown promise but are computationally intensive and require large datasets [32]. This study aims to bridge these gaps by providing a comparative analysis of different models, highlighting their strengths and weaknesses, and suggesting the most effective approaches for student performance prediction.

Goals and approach

In this study, we aim to evaluate four distinct machine learning algorithms on the same dataset to compare their effectiveness in solving the problem of predicting student performance. By evaluating and contrasting the performance of these algorithms, we seek to identify the most suitable approach for addressing this critical educational challenge. To address the limitations of existing solutions, our study aims to implement and evaluate various machine learning models to predict student performance accurately. By comparing traditional machine learning models with deep learning approaches, we seek to identify the most effective model for this task. Furthermore, we will compare the best-performing model with baseline models to establish its relative effectiveness.

Organization

The present document is organized into three chapters.

Chapter 1: This section presents the foundational concepts that are essential for understanding student success predictions.

Chapter 2: In this section, we review related work and various research studies conducted in the past on predicting student success.

Chapter 3: outlines the research design and methodology.

Chapter 4: presents the implementation and evaluation results, along with the discussion.

Chapter 1

Fundamental Concepts

1.1 Introduction

In this chapter, we present the fundamental concepts necessary for understanding the rest of the report. We begin with the definition of education, its historical development, and its role in society. Next, we define student performance and its various dimensions. We discuss the impact of student success on educational quality and its associated benefits. Finally, we explore the importance of predicting student performance and relevant techniques in machine learning and deep learning.

1.2 Definition of Education

This section delves into the meaning of education, tracing its historical journey, exploring modern interpretations, and examining its pivotal role in society.

1.2.1 What is Education?

Education is a process of facilitating learning, or the acquisition of knowledge, skills, values, beliefs, and habits. Educational methods include teaching, training, storytelling, discussion, and directed research. Education frequently takes place under the guidance of educators, but learners may also educate themselves in a process called autodidactic learning [9] .

1.2.2 Historical Perspective

Historically, education has evolved from informal, community-based teaching to formal, institutionalized systems. Early education often took place within families and communities, focusing on practical skills and moral values[12] . Over time, the establishment of schools and universities formalized the process, emphasizing academic knowledge and intellectual development [53].

1.2.3 Modern Definitions and Concepts

Modern education encompasses a broad spectrum of learning activities, both formal and informal. It is not confined to the walls of classrooms but extends to online platforms and lifelong learning environments. Education today is seen as a fundamental human right and a crucial driver of social and economic development [9][27].

1.2.4 The Role of Education in Society

Education serves multiple roles in society. It promotes individual growth, economic development, and social cohesion. Educated individuals tend to have better employment opportunities and are more likely to participate in civic activities. Furthermore, education is essential for fostering critical thinking, innovation, and the ability to adapt to changing societal needs [43].

1.3 Definition of Student Performance

1.3.1 Dimensions of Student Performance: Academic and Beyond

Student performance is a multifaceted construct that encompasses various dimensions of a student's academic and non-academic life. It can be broadly categorized into academic achievement, behavioral aspects, and emotional and psychological factors [59].

1.3.2 Academic Achievement

Academic achievement is typically measured by grades, test scores, and the completion of academic milestones. It is often used as the primary indicator of student performance and success in educational settings (Kuh et al., 2006). High academic achievement is associated with better opportunities for higher education and employment [49].

1.3.3 Behavioral Aspects

Behavioral aspects of student performance include attendance, class participation, and engagement in school activities. These behaviors are crucial for academic success as they reflect a student's commitment and involvement in their education [20].

1.3.4 Emotional and Psychological Factors

Emotional and psychological factors such as motivation, self-esteem, and emotional well-being also play a significant role in student performance. These factors influence a student's ability to cope with academic challenges and persist in their studies [44].

1.4 The Impact of Student Success on Education

1.4.1 Enhancing Educational Quality

Student success is directly linked to the overall quality of education. High student performance can lead to improved school rankings, increased funding, and greater community support. Successful students also serve as role models, fostering a culture of achievement within educational institutions [24].

1.4.2 Economic and Social Benefits

The success of students has far-reaching economic and social benefits. Educated individuals contribute more effectively to the workforce, driving innovation and economic growth. Additionally, education promotes social mobility and reduces inequality by providing individuals with the skills and knowledge necessary to improve their socio-economic status [22].

1.4.3 Personal Development

Beyond academic and economic benefits, student success contributes to personal development. Education helps individuals develop critical thinking, problem-solving skills, and ethical values. These attributes are essential for personal fulfillment and active citizenship[36] .

1.5 Predicting Student Performance

In this instance, we attempt to estimate the value of a variable that relates to the student. The values that are usually expected in education are student performance, knowledge, scores, or marks. The value can be categorical or discrete (classification task)(**see Figure 1.1**) or numerical or continuous (regression job). Finding the relationship between a dependent variable and one or more independent variables is done using regression analysis. Individual things are grouped via classification based on quantitative traits they possess naturally or on a training set of already labelled items. The most common uses of DM in education are for student performance prediction [46].

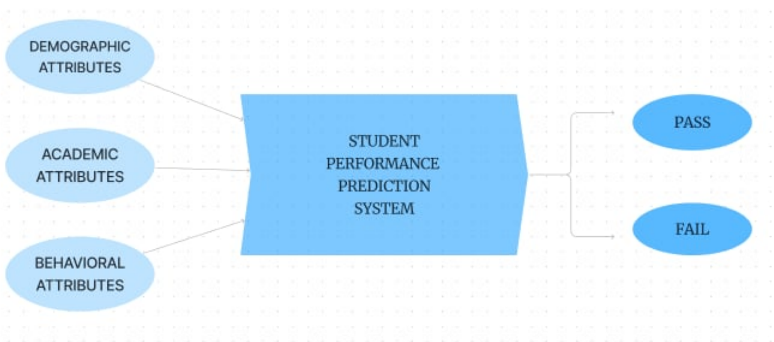


Figure 1.1: A student performance prediction system

1.6 The Importance of Student Performance Prediction

1.6.1 Identifying At-Risk Students

Predicting student performance is crucial for identifying at-risk students who may need additional support to succeed. Early identification allows educators to implement targeted interventions, thereby preventing dropout and improving retention rates[10] .

1.6.2 Personalizing Learning

Performance prediction enables personalized learning experiences tailored to individual student needs. By understanding a student's strengths and weaknesses, educators can customize instruction to enhance learning outcomes[35] .

1.6.3 Improving Educational Planning

Accurate predictions of student performance can aid in educational planning and policy-making. Schools can allocate resources more effectively, design better curricula, and improve teaching strategies based on predictive insights [16].

1.7 Deep Learning and Machine Learning

In this section, we will focus on deep learning and machine learning, delving into their models and discussing the main algorithms used for predicting student performance. We will begin with an overview of deep learning and machine learning, then move on to review the different models used in this context, and finally, we will discuss the most common and effective algorithms for predicting student performance[**?**].

1.7.1 Machine Learning Definition

Machine learning is a field of study that gives computers the ability to learn without being explicitly programmed, as defined by Arthur Samuel. According to Tom Mitchell, a computer program is said to learn from experience (E) with respect to some class of tasks (T) and performance measure (P) if its performance at those tasks, as measured by P, improves with experience. Ethem Alpaydin describes it as programming computers to optimize a performance criterion using example data or past experience. These definitions emphasize training computers to perform tasks intelligently by learning from repeated examples.

1. Overview of Machine Learning Approaches: Machine learning can be divided based on the nature of data labeling into:

   - Supervised Learning: This approach is used to estimate an unknown (input, output) mapping from known (input, output) samples, where the outputs are labeled. Examples include classification and regression.

- Unsupervised Learning: In this approach, the system is given input samples only, without any labeled outputs. Examples include clustering and probability density function estimation.
- Semi-supervised Learning: This combines supervised and unsupervised learning. Part of the data is labeled, and this labeled part is used to infer the labels of the unlabeled portion. Examples include text and image retrieval systems.

2. Other Classifications of Machine Learning:

- Transductive and Inductive Learning: Transductive learning involves inferring from specific training cases to specific testing cases, using either discrete labels (as in clustering) or continuous labels (as in manifold learning). Inductive learning aims to predict outputs from new inputs that the learner has not encountered before. An inductive bias is necessary for generalization beyond seen observations.
- Discriminative and Generative Models: Discriminative models measure the conditional probability of an output given deterministic inputs, such as neural networks or support vector machines. Generative models are fully probabilistic, using techniques like Bayesian networks or naïve Bayes to model the joint probability distribution of inputs and outputs.
- Reinforcement Learning: This approach involves an agent taking a sequence of actions to maximize cumulative rewards, such as winning a game of checkers. It is particularly useful for online learning applications where the environment is dynamic and the agent must learn through trial and error[18] .

1.7.2 Deep learning Definition

Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. These methods have dramatically improved the state-of-the-art in speech recognition, visual object recognition, object detection, and many other domains such as drug discovery and genomics.

Deep learning discovers intricate structure in large data sets by using the backpropagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer.

Deep convolutional nets have brought about breakthroughs in processing images, video, speech, and audio, whereas recurrent nets have shown effectiveness in sequential data such as text and speech [33] .

1.7.3 Support Vector Machines (SVM)

Are a sort of supervised learning algorithm that is used for classification and regression analysis. They determine the hyperplane that best splits the data into distinct classifications.

SVMs have the benefit of being able to handle high-dimensional data and non-linear boundaries by utilizing kernel functions . They are also useful when dealing with datasets with tiny sample numbers since they seek the largest feasible margin between various classes, leading in superior generalisation performance . One downside of SVMs is their sensitivity to the kernel function and the parameters associated with it, which necessitates careful tweaking. Moreover, SVMs may be computationally expensive and may demand a substantial amount of processing resources to train on huge datasets [28] **(see Figure 1.2)**.

$$w \cdot xi + b = 0 \tag{1.1}$$

such that:

$$yi.(xi.w + b) \geq 1, \forall i \tag{1.2}$$

$$yi.(xi.w + b) \geq -1, \forall i \tag{1.3}$$

where :

$$w = \text{weight associated with the features 1 to } z,$$
$$b = \text{set of points.}$$

but when the data is not linearly separable the kernel trick is used

The kernel function:

The kernel function is a mathematical trick that allows the SVM to perform a 'two-dimensional' classification of a set of originally one-dimensional data. In general, a kernel function projects data from a low-dimensional space to a space of higher dimension.

1. Linear Kernel Function The linear kernel function is commonly described as:

$$K(x_i, x_j) = x_i^\top x_j$$

2. Polynomial Kernel Function The polynomial kernel function is directional, i.e., the output depends on the direction of the two vectors in low-dimensional space. This is due to the dot product in the kernel. The magnitude of the output is also dependent on the magnitude of the vectors:

$$K(x_i, x_j) = (1 + x_i^\top x_j)^d$$

where d is the degree of the kernel function.

3. Radial Basis Function (RBF) The radial basis function is one of the most popular kernel functions. It adds a "bump" around each data point:

$$K(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right)$$

where $\gamma > 0$ is a kernel parameter.

4. Sigmoid Function The sigmoid kernel function is defined as:

$$K(x_i, x_j) = \tanh(\gamma x_i^\top x_j + r)^d$$

where γ, r, and d are kernel parameters[42].

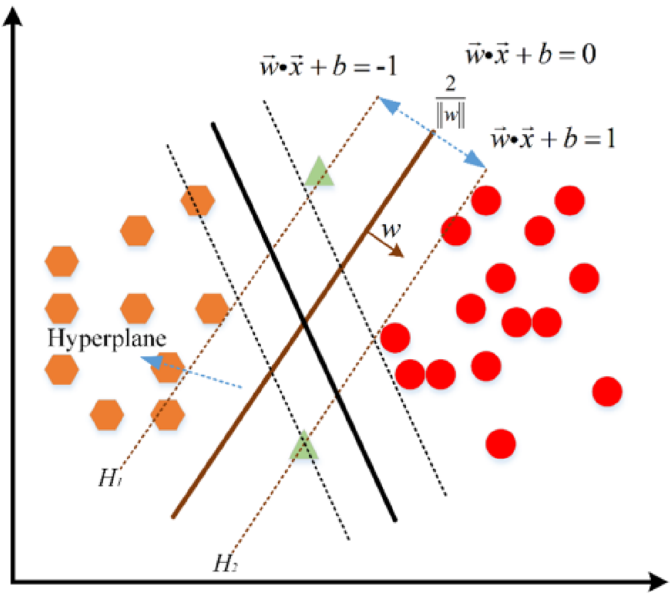


Figure 1.2: Schematic of the SVM Algorithm [26]

1.7.4 Decision Tree (DT)

Is a supervised learning algorithm used for both classification and regression tasks. It splits the data into subsets based on the value of input features, creating a tree **(see Figure 1.3)**like model of decisions. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from data features [23] [52][34] .

Key Components and Formulae:

Gini Impurity

The Gini Impurity is used to measure the impurity of a node. It is defined as:

$$\text{Gini}(t) = 1 - \sum_{i=1}^{C} p_i^2 \tag{1.4}$$

where p_i is the proportion of observations of class i at node t, and C is the number of classes.

Information Gain:

Information Gain measures the reduction in entropy or impurity from a dataset split and is used to select the attribute that best partitions the data. It is defined as:

$$\text{IG}(D, a) = \text{Entropy}(D) - \sum_{v \in \text{Values}(a)} \frac{|D_v|}{|D|} \text{Entropy}(D_v) \tag{1.5}$$

where:

- $\text{Entropy}(D) = -\sum_{i=1}^{C} p_i \log_2(p_i)$

- D is the dataset
- a is an attribute
- D_v is the subset of D where attribute a has value v
- p_i is the proportion of class i in dataset D

Entropy:

Entropy is a measure of randomness or impurity in the dataset and is defined as:

$$\text{Entropy}(D) = -\sum_{i=1}^{C} p_i \log_2(p_i) \tag{1.6}$$

where p_i is the proportion of class i in the dataset D.

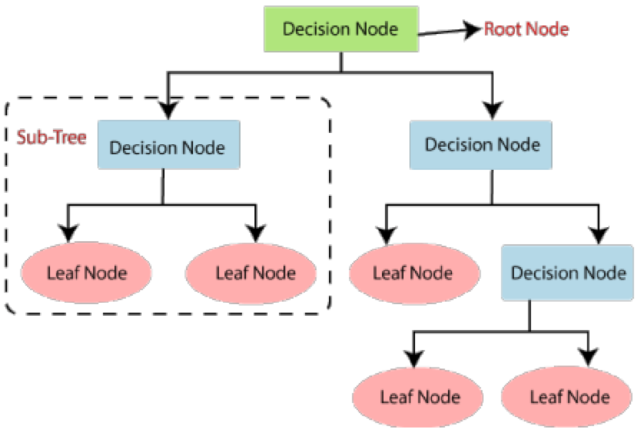


Figure 1.3: A Graphical Representation of Decision Tree [13]

1.7.5 Multilayer Perceptron (MLP)

The back propagation algorithm (including its variants) is the principle procedure for training multilayer perceptron's. In MLP neural network, each unit performs a biased weighted sum of inputs to them and passes this activation level through a transfer function to generate output **(see Figure 1.4)**. The most common activation functions in MLP are logistic and hyperbolic tangent sigmoid functions [51].

Forward Pass:

$$\text{Input layer:} \quad \mathbf{x} \in R^d \tag{1.7}$$

$$\text{Hidden layer:} \quad \mathbf{h}^{(l)} = f\left(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}\right), \quad l = 1, \ldots, L \tag{1.8}$$

$$\text{Output layer:} \quad \mathbf{y} = g\left(\mathbf{W}^{(L+1)}\mathbf{h}^{(L)} + \mathbf{b}^{(L+1)}\right) \tag{1.9}$$

where:

- $\mathbf{x}$ is the input vector
- $\mathbf{h}^{(l)}$ is the activation of the l-th hidden layer
- $\mathbf{W}^{(l)}$ and $\mathbf{b}^{(l)}$ are the weights and biases for the l-th layer
- $f(\cdot)$ is the activation function (e.g., ReLU, sigmoid)
- $g(\cdot)$ is the output activation function (e.g., softmax for classification)
- L is the number of hidden layers

Loss Function (log loss):

$$\mathcal{L}(\mathbf{y}, \mathbf{y}^*) = -\sum_{i=1}^{C} y_i^* \log(y_i) \tag{1.10}$$

where:

- $\mathbf{y}$ is the predicted output
- $\mathbf{y}^*$ is the true output (one-hot encoded for classification)
- C is the number of classes

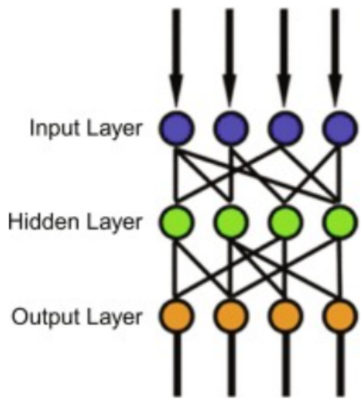


Figure 1.4: Architecture of the Multilayer Perceptron (MLP)[50]

1.7.6 Artificial Neural Networks (ANN)

An Artificial Neural Network (ANN) is a computational model inspired by the structure and functionality of the human brain. It is a mathematical model composed of intercon-nected processing units called artificial neurons or nodes. The basic building block of an ANN is a neuron, which receives inputs, applies weights to them, performs a mathematical operation, and generates an output. The outputs of neurons in one layer serve as inputs to the neurons in the next layer, forming a network of interconnected layers **(see Figure 1.5)**.

- **Input Layer :**

  It receives the initial data or features to be processed.

- **Hidden Layer(s):**

  These layers perform complex computations by applying weights to inputs and applying activation functions. They extract features and learn representations from the input data.

- **output Layer :**

  It produces the final output, which depends on the problem being solved. For example, in a classification task, the output layer may represent different classes, while in a regression task, it may produce a continuous value [47].

Key Components and Formulae:

Neuron Activation

The output of a single neuron is calculated as:

$$a_j = f\left(\sum_{i=1}^{n} w_{ij} x_i + b_j\right) \tag{1.11}$$

where:

- x_i are the input signals.
- w_{ij} are the weights associated with the inputs.
- b_j is the bias term.
- $f(\cdot)$ is the activation function (e.g., sigmoid, ReLU).
- a_j is the output of the neuron.

Feedforward Pass:

In a feedforward neural network, the data flows from the input layer to the output layer through hidden layers [21, 48]. For a network with L layers:

$$\mathbf{h}^{(l)} = f\left(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}\right), \quad l = 1, \ldots, L \tag{1.12}$$

$$\mathbf{y} = g\left(\mathbf{W}^{(L+1)}\mathbf{h}^{(L)} + \mathbf{b}^{(L+1)}\right) \tag{1.13}$$

where:

- $\mathbf{h}^{(0)}$ is the input vector $\mathbf{x}$.

- $\mathbf{h}^{(l)}$ is the activation of the l-th layer.
- $\mathbf{W}^{(l)}$ and $\mathbf{b}^{(l)}$ are the weights and biases for the l-th layer.
- $f(\cdot)$ is the activation function (e.g., ReLU).
- $g(\cdot)$ is the output activation function (e.g., softmax for classification).
- $\mathbf{y}$ is the output vector.

Loss Function(log loss):

The loss function measures the difference between the predicted output and the true output. For classification tasks, [21][48]the cross-entropy loss is often used:

$$\mathcal{L}(\mathbf{y}, \mathbf{y}^*) = -\sum_{i=1}^{C} y_i^* \log(y_i) \tag{1.14}$$

where:

- $\mathbf{y}$ is the predicted output.
- $\mathbf{y}^*$ is the true output (one-hot encoded for classification).
- C is the number of classes.

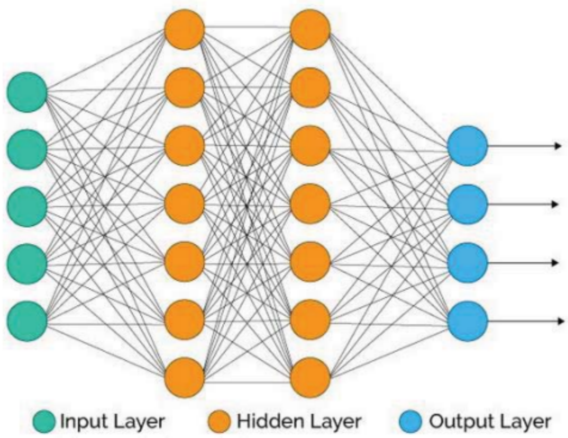


Figure 1.5: Architecture of the Artificial Neural Network (ANN) [15]

1.7.7 Grid search

This method is also called parameter sweeping, where we manually define a subset of hyperparameters and then use all possible parameter combinations for the specified parameter subsets **(see Figure 1.6)**. Each combination of hyperparameters is validated using cross-validation, and then the best-performing hyperparameter combination is selected. This is considered the simplest way of hyperparameter tuning [17].

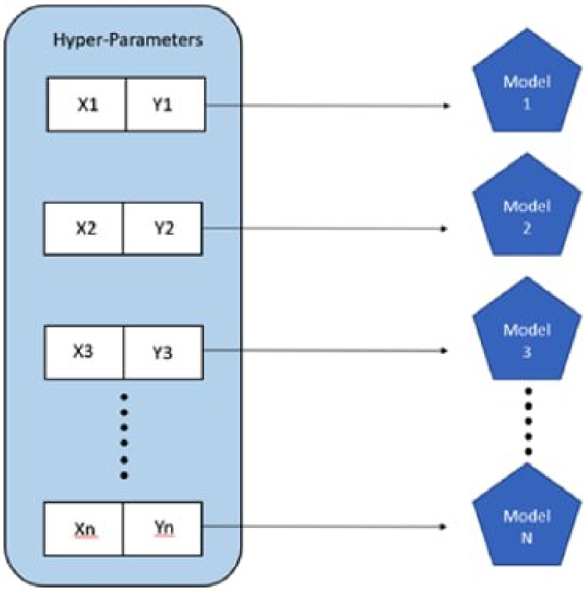


Figure 1.6: Manuel hyperparameter tuning .page 139 [7] .

1.8 Conclusion

In this chapter, we have laid the groundwork by exploring the fundamental concepts essential for understanding the subsequent sections of the report. We began by defining education and examining its evolution and societal role. We then delved into student performance, covering its various dimensions and the impact of student success on educational quality and broader benefits. Finally, we discussed the significance of predicting student performance and introduced relevant machine learning and deep learning techniques. This foundation prepares us to delve deeper into the specifics of our research, providing a clear context for analyzing and applying predictive models in education.

Chapter 2

State Of The Art

2.1 Introduction

This chapter reviews various studies in student performance prediction, focusing on the different machine learning algorithms and datasets employed. We introduce key studies and then present additional examples in tables summarizing titles, authors, classifiers, datasets, and accuracy rates. This structured overview highlights the strengths and weaknesses of existing approaches and sets the foundation for our comparative analysis.

2.2 Related works

In this section, we will first introduce some studies and then elaborate further with additional examples in the table **(table 2.1)** . A total of 17 papers were reviewed to gather insights into the methodologies and outcomes of existing research in predicting student performance. Common models in machine learning and deep learning include Support Vector Machines (SVM), Decision Trees (DT), Random Forests (RF), and Artificial Neural Networks (ANNs). Each model employs distinct techniques to classify data and make predictions.

The studies reviewed cover a wide range of approaches and datasets, highlighting the diversity in feature selection, preprocessing techniques, and evaluation metrics used. For instance, several papers focused on the importance of demographic and behavioral features in predicting academic success, while others emphasized the role of engagement metrics such as attendance and participation in online discussions. Additionally, the reviewed studies utilized various datasets, from small, institution-specific collections to larger, publicly available datasets like the one used in this research.

2.2.1 Support Vector Machines(SVM)

in the study conducted by[37],Support Vector Machines(SVM) emerged as the second-most proficient classifier in terms of achieving high accuracy rates. Out of the 19 papers surveyed, SVM was employed in five instances to prognosticate academic success. some of the applications of svm are conducted by [29], wherein SVM yielded accuracy rates of 90.3% and 75%, respectively. Such results underscore the promising efficacy of SVMs in forecasting student performance. Nonetheless, it is imperative to note the paucity of research endeavors dedicated to assessing the transferability and generalizability of SVM models across diverse educational settings.

2.2.2 Decision Trees and Random Forests

[55] achieved accuracies of 69% and 79% with decision tree and random forest models, respectively. Similarly, [30] obtained 66.85% accuracy with a decision tree model. While these studies showcase the usability of these models

2.2.3 Artificial Neural Networks(ANNs)

[32][14] reported accuracies of 84.8% and 93.81% using ANN models, highlighting their effectiveness in student performance prediction. However, ANNs are often criticized for being "black boxes," making it difficult to understand how they arrive at their predictions.

Table 2.1: Overview of Related Studies on Student Performance Prediction(Part 1)

Study	Year	Dataset	Model	Accuracy	Evaluation
[31]	2023	xAPI-Edu-Data	LR	NO	NO
[57]	2023	xAPI-Edu-Data	LR,DT,RF	NO	MAE R2
[41]	2021	xAPI-Edu-Data	DNN	84.3%	cost function
[60]	2023	xAPI-Edu-Data	conventional feed-forward neural networks	91.95%	Recall F1-score
[56]	2019	xAPI-Edu-Data	DNN	85%	Accuracy Precision Recall F1-score
[6]	2020	NO	Neural Network Algorithm	76.8%	Precision, Sensitivity Recall
[5]	2022	Students' data and their final exam results	RF	91.7%	Accuracy R^2
[39]	2023	xAPI-Edu-Data	RF	82.29%	Accuracy Recall F1-score
[11]	2018	xAPI-Edu-Data	DNN	84.3%	Acciracy

Table 2.2: Overview of Related Studies on Student Performance Prediction(Part 2)

[14]	2023	The Open University Learning Analytics dataset	ANN	93.81%	Accuracy Precision Recall F1-score
[58]	2022	Student Information System (SIS) dataset	LR , RF,SVM, NN,NB,KNN	70% 75%	Accuracy Precision Recall F1-score AUC
[32]	2019	NO	NN	84.8%	Error performance Regression Error histogram Confusion matrix ROC AUC
[40]	2022	NO	KNN, NB	NO	MSE Regression analysie Error histogram Confusion matrix Roc
[29]	2022	Belgrade Metropolitan University's EMS and LMS	KNN,SVM NB,DT,LR LDA	NO	Accuracy Confusion matrix
[45]	2019	Darwin Li's student performance dataset from Nottingham Trent International College	SVM,MLP DT,RF,NB	80% 84%	Accuracy
[54][	2019	student dataset of 2018 academic year obtained from the Department of Networking and System Security in KSIT	NN,MLP	73.68%.	Accuracy
[30]	2018	NO	DT	66.9%	Accuracy Precision Recall F-measure

2.2.4 Discussion :

The related works tables**(Table 2.1)**and **(Table 2.2)**presents a detailed comparison of various studies focusing on student performance prediction using different machine learning models, datasets, and evaluation metrics. The studies employ a range of models, including Artificial Neural Networks(ANN), Logistic Regression(LR), Random Forest(RF), Support Vector Machine(SVM), Decision Tree(DT), and K-Nearest Neighbors(KNN). These models are evaluated across multiple datasets, most notably the xAPI-Edu-Data dataset, which is widely used due to its comprehensive attributes covering demographic, academic, and behavioral aspects of students.

In terms of performance metrics, accuracy is the most commonly reported measure, with studies also frequently reporting precision, recall, and F1 score to provide a more rounded assessment of model performance. For instance, the ANN model achieved the highest accuracy(89.87%) in a study using The Open University Learning Analytics dataset. Logistic Regression and Random Forest also showed strong performance with accuracy rates around 84.% and 82.75%, respectively. SVM and KNN models demonstrated notable accuracy, such as 84.87% and 83.78%, respectively. However, the Decision Tree model showed relatively lower accuracy(66.87%).

Overall, the related works highlight the importance of selecting appropriate models and datasets to optimize prediction accuracy. Our study builds upon these findings by integrating both traditional machine learning and deep learning models, comparing them, and employing a thorough evaluation framework. This approach aims to enhance predictive power and provide actionable insights for educators to support student performance improvement.

2.3 Conclusion

This chapter reviewed key studies on student performance prediction using various machine learning models. We compared classifiers, datasets, and accuracy rates, emphasizing the importance of selecting appropriate models. These insights will guide our implementation and evaluation of different models in the next chapters.

Chapter 3

Methodology

3.1 Introduction

This chapter presents the methodology employed in this research to achieve the objectives of training and comparing four different models for predicting students' performance using a dataset. It outlines the research design, data collection process, data splitting for training and testing, model training, and accurate model evaluation. Through this comprehensive approach, our goal was to identify the model that provides the most accurate and reliable predictions of student performance.

3.2 Research Design

The research design for this study follows a systematic and empirical approach. It involves using a publicly available dataset found on Kaggle, which contains information related to students' academic performance. This dataset includes various features such as gender, number of visits to educational resources, number of discussion participations, parents' satisfaction with the school, student absences, etc. The study aims to analyze these features and identify patterns and factors affecting students' performance by using machine learning and deep learning models to predict academic performance based on the available data.

3.3 Proposed System

The provided flowchart outlines a machine learning workflow for predicting student performance using educational data, specifically the xAPI-Edu-Data. The workflow begins with data collection, where the appropriate educational data is selected. In this case, xAPI-Edu-Data is chosen, containing various features related to students' academic activities and background information that are essential for building a predictive model.

In the preprocessing stage, categorical features are converted to numerical values using techniques such as one-hot encoding or label encoding, ensuring the data is in a suitable format for the models. This is followed by the grid search step, which aims to tune the hyperparameters of the models by systematically working through multiple combinations of parameter values to determine the optimal settings that enhance model performance.

In the training stage, several machine learning algorithms are trained, including Support Vector Machine (SVM), Decision Tree(DT), Multi-Layer Perceptron(MLP), and Artificial Neural Network(ANN). These models are then evaluated using multiple metrics such as accuracy, precision, recall, and F1-Score to determine the model's effectiveness in prediction.

After evaluation, the best model is selected based on the results of the aforementioned metrics. The performance of all trained models is compared, and the model with the highest scores is chosen. Finally, the selected model is used to predict student performance on new, unseen data. This step involves applying the model to the data to generate predictions about students' future academic outcomes.

This process requires ensuring data quality by removing missing or erroneous values and feature engineering to create new features or modify existing ones to better capture the underlying patterns in the data. Model interpretability should also be considered, as simpler models like decision trees might be preferred for understanding the reasoning behind predictions, especially in educational contexts. Ethical considerations are also crucial to ensure the privacy and security of student information and the responsible use of predictions. By following this workflow, one can systematically develop, train, and evaluate machine learning models to predict student performance, ultimately aiding educational institutions in identifying and supporting students who may need additional assistance(**see Figure 3.1**) .

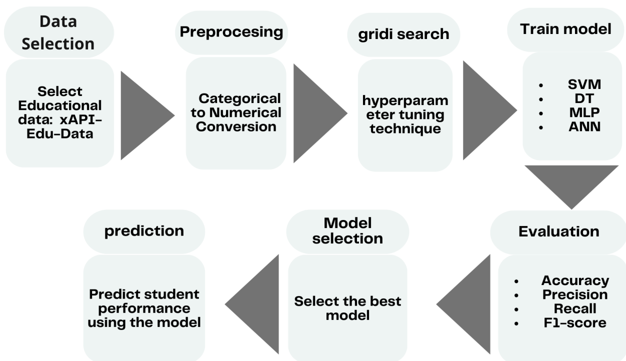


Figure 3.1: Architecture of the proposed system .

3.3.1 Diagram sequence

The diagram sequence shows the process of hyperparameter tuning and model selection for predicting student performance. Models like SVM, MLP, Decision Tree, and ANN are tuned using Grid Search. After training, models are evaluated using metrics such as accuracy, precision, recall, and F1 score. The best configurations are compared, and the top-performing model is selected for the prediction system **(see Figure 3.2)**.

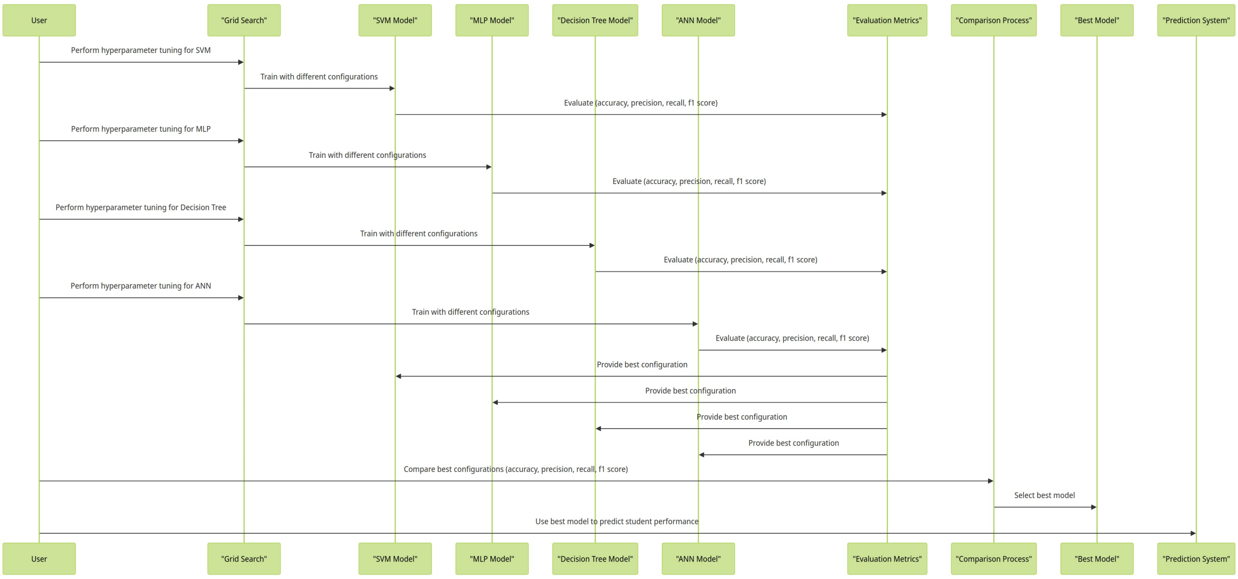


Figure 3.2: Sequence diagram.

3.4 Data Preparation

3.4.1 Data Selection

In this research, the well-known Kaggle dataset "Students' Academic Performance Dataset(xAPI-Educational Mining Dataset)" was utilized. This dataset was collected from the Kalboard 360 learning management system(LMS). Kalboard 360 is a multi-agent LMS designed to facilitate learning through advanced technology, providing users with synchronous access to educational resources from any Internet-connected device.

The data was gathered using a learner activity tracking tool known as the Experience API(xAPI), a component of the Training and Learning Architecture(TLA) that enables monitoring of learning progress and learner actions such as reading articles or watching training videos. The xAPI assists learning activity providers in identifying the learner, activity, and objects that describe a learning experience.

The dataset comprises 480 student records and 16 features, categorized into demographic features(e.g., gender and nationality), academic background features(e.g., educational stage, grade level, and section), and behavioral features(e.g., raising hand in class, accessing resources, answering parent surveys, and school satisfaction) the classification is shown in **(Table 3.1)** . The students include 305 males and 175 females from various origins, such as Kuwait, Jordan, Palestine, Iraq, Lebanon, Tunisia, Saudi Arabia, Egypt, Syria, USA, Iran, Libya, Morocco, and Venezuela.

Data collection occurred over two educational semesters, with 245 student records collected during the first semester and 235 during the second semester. The dataset also includes a school attendance feature, categorizing students based on their absence days: 191 students exceeded 7 absence days, while 289 students had absence days under 7.

Additionally, a new category of features related to parent involvement in the educational process is included. This feature includes two sub-features: Parent Answering Survey and Parent School Satisfaction. Among the parents, 270 answered the survey, while 210 did not, and 292 expressed satisfaction with the school, while 188 did not [8] .

Table 3.1: Students' Attribute Classification

Attribute Classification	Attribute	Explanation
Demographic Attributes	Gender Nationality Place of Birth Relation	Statistical data such as age, gender
Academic Attributes	Stage ID Grade ID Section ID Topic Semester	Data related to student academic activities
Behavioral Attributes	Raise Hands Visited Resources Announcement Views Discussion Parent Answering Survey Parent School Satisfaction Student Absent Days	Student Engagement withLMS

3.4.2 Data preprocessing

Since the dataset can be classified as Secondary Data collected from multiple publicly available sources so Data has been previously transformed, modified, cleansed its doesn't need any handling missing values or outliers so we did prepared to make it compatible . Initially, we encountered categorical features in our dataset. To make them compatible with machine learning algorithms, we applied the get dummies technique. This process converts categorical values into numerical representations.

3.4.3 Data Exploration and Visualization

The curve illustrates the numbers of students in each performance category represented by different colors: red for medium performance, green for low performance, and blue for high performance. The analysis indicates that the majority have average performance, with a minority having high or low performance**(Figure 3.3)** .

The conclusions suggest the necessity of improving the performance of students with low performance by providing more support and guidance. Recommendations include providing additional support, using diverse teaching methods, offering extra learning opportunities, and regularly assessing performance with constructive feedback. In conclusion, the analysis provides valuable information for enhancing students' performance and potential future improvements, emphasizing the need for further analysis to gain a deeper understanding of student performance.

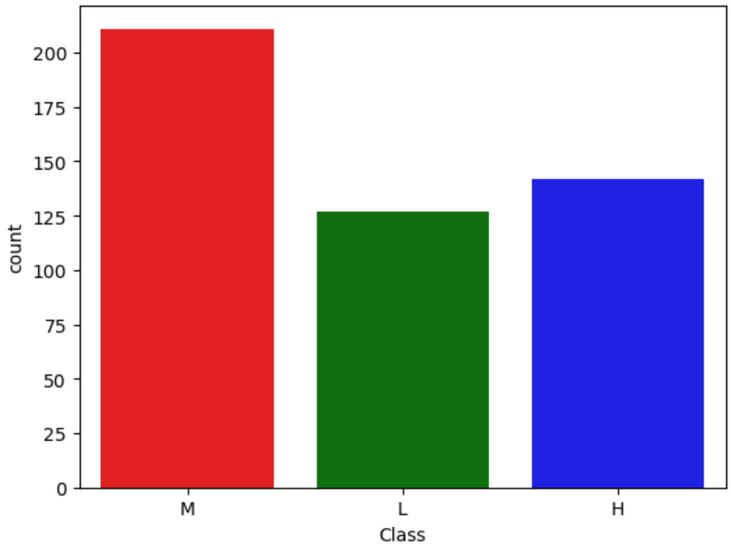


Figure 3.3: Count of Students in Each Class.

The graph shows an analysis of student performance on xAPI-Edu-Data. It's a double bar chart representing student classes(M = Middle, L = Lower, H = Higher) on the horizontal axis and the number of students on the vertical axis. Red bars represent male students, while blue bars represent female students. Overall, there's nearly equal gender representation across all classes, with slightly more males in the middle class and slightly more females in the higher class. Fewer students are observed in the lower class compared to the middle and higher classes **(Figure 3.4)**.

In summary, while gender representation is balanced overall, variations exist across different classes. These differences may stem from factors like students' interests, capabilities, or gender-specific opportunities..

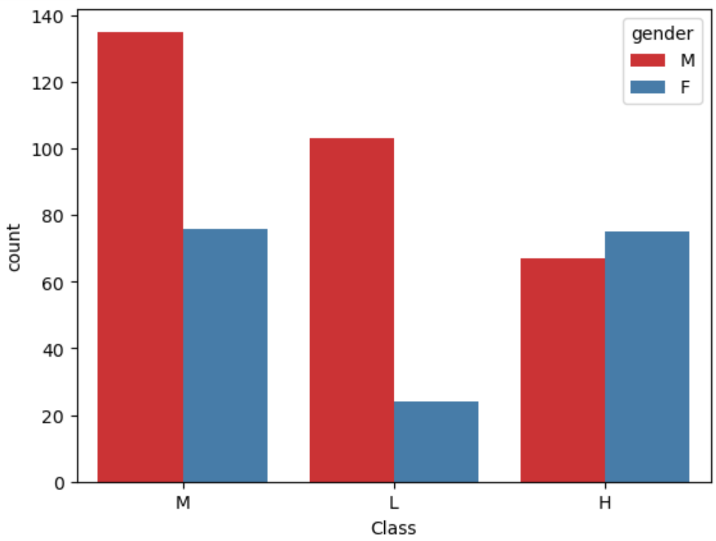


Figure 3.4: Count of Students in Each Class by Gender.

3.4.4 Data Splitting

Before diving into model training, we split the dataset into two parts:

- Training Data: Comprising 80% of the dataset, this portion is used to train our machine learning models.
- Testing Data: The remaining 20% serves as an independent set for evaluating model performance.

For deep learning techniques, we further split the training data into:

- Training set:80% of the training data , used to train the deep learning models.
- Validation set:20% of the training data , used to tune hyperparameters and prevent overfitting by validating the model during training.

3.4.5 Feature Selection

Not all features contribute equally to predicting student performance. To identify the most influential features, we employed the Random Forest Classifier.

The classifier revealed that the following features significantly impact performance:

- Gender
- Raised Hands
- Visited Resources
- Announcements View

- Discussion Participation

By selecting these crucial features, which are shown in **(Figure 3.5)**, we ensure that our model focuses on the most relevant aspects of student behavior. This step is essential for accurate performance prediction

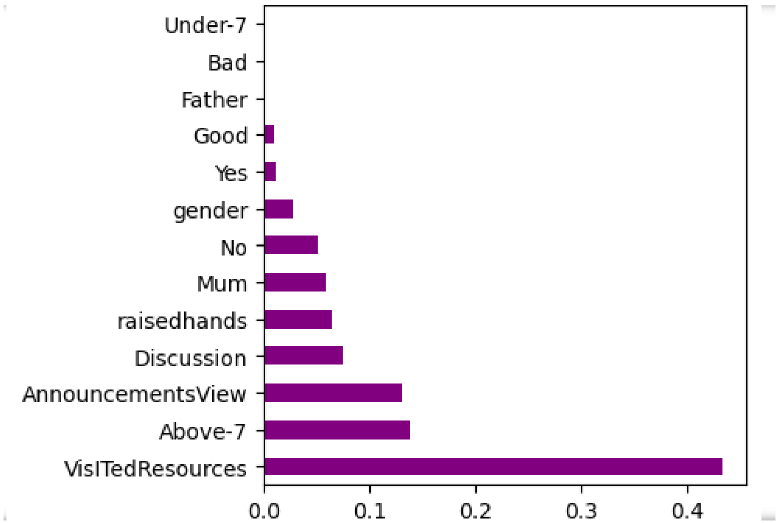


Figure 3.5: Plot of features importance.

3.5 Model Selection

Four different algorithms were selected for training and comparison in this study. The selection process was based on their popularity and demonstrated performance in student performance prediction tasks. The chosen algorithms include Support Vector Machine(SVM), Multi-Layer Perceptron(MLP), Decision Tree(DT), and Artificial Neural Network(ANN), each with its own unique algorithms and architectures. The selection of diverse algorithms enable a comprehensive analysis of their performance and comparison.

3.6 Grid search for hyperparameters

To garanti that we are training the models with best configurations we employed the grid search for hyperparameters(hyperparameters tuning technique). The Grid Search conducts hyperparameter tuning for each model(SVM, MLP, Decision Tree, ANN) After tuning, each model will be evaluated using evaluation metrics to determine its performance with the tuned hyperparameters.

3.7 Model Training and Evaluation

The selected models, including Support Vector Machines (SVM), Multilayer Perceptrons (MLP), Decision Trees (DT), and Artificial Neural Networks (ANN), were trained using the preprocessed dataset. During the training phase, we conducted hyperparameter tuning using grid search to optimize the models' performance and prevent overfitting. The dataset was split into training (80%) and testing sets (20%) to evaluate the models' effectiveness in predicting student performance.

The models were evaluated using various performance metrics, including accuracy, precision, recall, and F1 score, to ensure a comprehensive assessment. Computational efficiency and resource requirements were also considered to assess the models' practical feasibility. Detailed explanations of the training process, hyperparameter tuning, and evaluation metrics are provided in the next chapter.

3.8 Conclusion

This chapter outlined the methodology used in this research to predict students' performance using four models. It included data selection, splitting, model selection, feature selection, training, and evaluation. The aim was to identify the most accurate classifier. This methodology sets the stage for the following chapter, where the results and analysis will be discussed.

Chapter 4

Implementation And Evaluation

4.1 Introduction

The implementation of predictive models for student performance is essential for identifying students at risk of underperforming and providing timely interventions. Given the growing reliance on data-driven decision-making in education, leveraging machine learning techniques can offer significant improvements over traditional methods.

The primary goal is to implement and evaluate machine learning models to predict student performance accurately. Specific objectives include:

- Comparing the performance of different algorithms.
- Identifying the most effective model for accurate predictions.
- Comparing the best performing model with baseline models.

This chapter contain four sections the implementation setup then the evaluations and comparisons and We conclude by discussing the obtained results

.

4.2 Implementation Setup

In this section, we will explore the details of the development environment and the programming language used to implement our system. Additionally, we will delve into the training and testing procedures, the dataset used[8] , and provide in-depth insights into the architecture we have designed.

4.2.1 Development Environment

Anaconda

Anaconda is a comprehensive platform for data science and statistical analysis that includes Python as well as a wide range of libraries and tools used in data analysis and software development. I used Anaconda to create a stable and powerful development environment for software development and testing.

Conda offers a variety of features, including:

- Improvement of Python programming skills.
- Development of artificial intelligence applications using popular Python libraries such as Keras, and TensorFlow.
- Utilization of the Jupyter Notebook development environment, which requires no prior configuration.

4.2.2 Programming Language and Libraries

In this section, we will provide an introduction to the Python programming language and the libraries that have been used to implement the Student Performance Prediction (SPP) models.

Python

The Python programming language has become extremely popular among programmers and is now considered the most widely used programming language. Its popularity extends to various domains, including infrastructure management, data analysis, and software development. The appeal of Python lies in its ability to allow developers to focus on their tasks without being burdened by implementation complexities. Unlike previous languages, Python frees developers from strict syntactic limitations, enabling them to write code more efficiently. Consequently, Python offers a faster development experience compared to other languages.

Libraries used

- **TensorFlow** The TensorFlow library was used to define the core com-ponents for all of our architectures. Its primary purpose is to implement machine and deep learning algorithms. In addition, it offers a high de-gree of adaptability in its application to create neural networks[ten].
- **Keras** it is used with TensorFlow. We used this library to implement the different layers, the activation functions, and the preparation of the training base[ten].

- **NumPy** We utilized this library to adapt the input types according to the configurations of the employed models, which were specifically designed to handle multidimensional arrays or matrices, along with ma-thematical functions that operate on these arrays. This package was particularly employed for window extraction and image scanning purposes[num] .

- **Sklearn** The sklearn library, one of the most advantageous ML pa- ckages in Python, provides numerous powerful techniques for machine learning and statistical modeling. These techniques include dimensio-nality reduction, regression, classification, and clustering [sci].

- **Pandas:** is a data analysis open-source and processing tool developed in the language of Python. It is flexible, powerful, fast, and simple to use[pan].

4.2.3 Model Building

This section details the iterative process of model development, including parameter optimization through grid search, to enhance the predictive capabilities of the models.

Support Vector Machines (SVM)

In this iterative process, a grid search is conducted to optimize the parameters for a Support Vector Machine (SVM) model. Different combinations of parameters such as 'C' (regularization parameter), 'kernel', and 'gamma' are defined. The grid search systematically evaluates each combination by creating and training an SVM model with the training data. The model is then tested with a separate testing dataset to assess its performance using metrics like accuracy, precision, F1 score, and ROC-AUC score. The results are organized into a table**(Table 4.1)** for easy comparison of the SVM model's performance under different parameter setups. This approach helps identify the optimal parameters for achieving accurate predictions with the SVM model. The model building process is illustrated in **(Figure 4.1)**..

```python
# Define the parameter grid for SVM
parameters = [
    {'C': 0.1, 'kernel': 'linear', 'gamma': 'scale'},
    {'C': 1, 'kernel': 'poly', 'gamma': 'scale'},
    {'C': 10, 'kernel': 'rbf', 'gamma': 'scale'},
    {'C': 0.1, 'kernel': 'sigmoid', 'gamma': 'auto'},
    {'C': 1, 'kernel': 'rbf', 'gamma': 'auto'}
]

# Store results
results = []

for params in parameters:
    model = SVC(C=params['C'], kernel=params['kernel'], gamma=params['gamma'], probability=True)  # Set probability=True
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    y_prob = model.predict_proba(X_test)  # Use predict_proba for probabilities
    accuracy = accuracy_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred, average='weighted')
    recall = recall_score(y_test, y_pred, average='weighted')
    f1 = f1_score(y_test, y_pred, average='weighted')
    roc_auc = roc_auc_score(y_test, y_prob, multi_class='ovr')  # Use y_prob for ROC-AUC

    results.append({
        'C': params['C'],
        'Kernel': params['kernel'],
        'Gamma': params['gamma'],
        'Accuracy': accuracy,
        'Precision': precision,
        'Recall': recall,
        'F1 Score': f1,
        'ROC-AUC': roc_auc
    })

svm_results_df = pd.DataFrame(results)
print(svm_results_df)
```

Figure 4.1: Simple Building Support vector machine

Table 4.1: SVM Model Grid Parameters

C	Kernel	Gamma	Accuracy	Precision	Recall	F1 Score	ROC-AUC
0.1	linear	scale	0.729167	0.735819	0.729167	0.725198	0.921000
1.0	poly	scale	0.718750	0.720231	0.718750	0.716948	0.905217
10.0	rbf	scale	0.781250	0.776929	0.781250	0.775193	0.901050
0.1	sigmoid	auto	0.739583	0.746799	0.739583	0.733724	0.905077
1.0	rbf	auto	0.729167	0.728573	0.729167	0.726172	0.910986

The SVM model demonstrates varying performance across different parameter configurations, with the best accuracy achieved when C=10.0, kernel='rbf', and gamma='scale'.

Multilayer Perceptron (MLP)

For the Multilayer Perceptron (MLP) model, different hyperparameter combinations are defined, including the number of hidden layers, activation functions, and optimization methods. An empty list is created to store the results from testing each combination. Each set of hyperparameters is used to build an MLP model, which is then trained with the data and evaluated using performance metrics like accuracy and precision. These results are organized into a table(**Table 4.2**) to compare the performance of the MLP model under different setups, guiding the selection of the most effective configuration for accurate student performance predictions. The model building process is shown in (**Figure 4.2**).

```python
from sklearn.neural_network import MLPClassifier

# Define the parameter grid for MLP
parameters = [
    {'hidden_layer_sizes': (50,), 'activation': 'tanh', 'solver': 'sgd'},
    {'hidden_layer_sizes': (100,), 'activation': 'relu', 'solver': 'adam'},
    {'hidden_layer_sizes': (50, 50), 'activation': 'relu', 'solver': 'sgd'}
]

# Store results
results = []

for params in parameters:
    model = MLPClassifier(hidden_layer_sizes=params['hidden_layer_sizes'], activation=params['activation'],
                          solver=params['solver'], max_iter=500)
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
     accuracy = accuracy_score(y_test, y_pred)
     precision = precision_score(y_test, y_pred, average='weighted')
     recall = recall_score(y_test, y_pred, average='weighted')
     f1 = f1_score(y_test, y_pred, average='weighted')
     roc_auc = roc_auc_score(y_test, model.predict_proba(X_test), multi_class='ovr')

     results.append({
         'Hidden Layers': params['hidden_layer_sizes'],
         'Activation': params['activation'],
         'Solver': params['solver'],
         'Accuracy': accuracy,
         'Precision': precision,
         'Recall': recall,
         'F1 Score': f1,
         'ROC-AUC': roc_auc
     })

mlp_results_df = pd.DataFrame(results)
print(mlp_results_df)
```

Figure 4.2: Simple Building Multilayer Perceptron

Table 4.2: MLP Model Parameters and Performance

Hidden Layers	Activation	Solver	Accuracy	Precision	Recall	F1 Score	ROC-AUC
(50,)	tanh	sgd	0.719	0.728	0.719	0.716	0.917
(100,)	relu	adam	0.771	0.771	0.771	0.770	0.914
(50, 50)	relu	sgd	0.750	0.753	0.750	0.746	0.920

The MLP model demonstrates improved accuracy with different configurations of hidden layers and activation functions. Notably, the configuration with 1 hidden layers of 100 neurons, 'relu' activation, and 'adam' solver achieved the highest accuracy.

Decision Tree (DT)

A parameter grid is defined for the Decision Tree model, consisting of different hyperparameter combinations such as criteria for measuring the quality of a split, maximum tree depth, and the minimum number of samples required to split an internal node. An empty list is initialized to store the evaluation results for each hyperparameter combination. Each set of parameters is used to build a Decision Tree model, which is then trained and evaluated using metrics such as accuracy, precision, recall, F1 score, and ROC-AUC. The results are organized into a table **(Table 4.3)**, facilitating the comparison of different parameter configurations and helping to identify the most effective setup for predicting student performance with the Decision Tree model. The model building process is illustrated in **(Figure 4.3)**.

```python
from sklearn.tree import DecisionTreeClassifier

# Define the parameter grid for Decision Tree
parameters = [
    {'criterion': 'gini', 'max_depth': None, 'min_samples_split': 2},
    {'criterion': 'entropy', 'max_depth': 10, 'min_samples_split': 5},
    {'criterion': 'gini', 'max_depth': 20, 'min_samples_split': 10}
]

# Store results
results = []

for params in parameters:
    model = DecisionTreeClassifier(criterion=params['criterion'], max_depth=params['max_depth'],
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

     accuracy = accuracy_score(y_test, y_pred)
     precision = precision_score(y_test, y_pred, average='weighted')
     recall = recall_score(y_test, y_pred, average='weighted')
     f1 = f1_score(y_test, y_pred, average='weighted')
     roc_auc = roc_auc_score(y_test, model.predict_proba(X_test), multi_class='ovr')

     results.append({
         'Criterion': params['criterion'],
         'Max Depth': params['max_depth'],
         'Min Samples Split': params['min_samples_split'],
         'Accuracy': accuracy,
         'Precision': precision,
         'Recall': recall,
         'F1 Score': f1,
         'ROC-AUC': roc_auc
     })

dt_results_df = pd.DataFrame(results)
print(dt_results_df)
```

Figure 4.3: Simple Building Decision Tree

Table 4.3: DT Model Parameters and Performance

Criterion	Max Depth	Min Samples Split	Accuracy	Precision	Recall	F1 Score	ROC-AUC
gini	NaN	2	0.698	0.703	0.698	0.684	0.769
entropy	10.0	5	0708	0.708	0.708	0.706	0.806
gini	20.0	10	0.698	0.698	0.698	0.689	0.816

The Decision Tree model exhibits varying performance based on different parameter settings. The configuration with criterion='entropy', max depth=10, and min samples split=5 achieved the highest accuracy.

Artificial Neural Networks (ANN):

we start with building a model(1) then after evaluating it we built another model (2)

Model 1:

The first model utilized an Artificial Neural Network (ANN) architecture, comprising three layers. The initial layer featured 64 neurons with a sigmoid activation function, matching the input dimension to the number of features in the scaled training data. The subsequent layer contained 32 neurons, also with a sigmoid activation function. Finally, the output layer consisted of three neurons with a softmax activation function, facilitating classification into three classes shown in (**Figure 4.4)**. This model achieved an accuracy of 82%

```
# Build the ANN model
model = Sequential()
model.add(Dense(units=64, activation='sigmoid', input_dim=X_train_scaled.shape[1]))
model.add(Dense(units=32, activation='sigmoid'))
model.add(Dense(units=3, activation='softmax'))  # 3 output units for 3 classes
```

(a) Model ANN

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 64)	1,088
dense_1 (Dense)	(None, 32)	2,080
dense_2 (Dense)	(None, 3)	99

Total params: 9,803 (38.30 KB)

Trainable params: 3,267 (12.76 KB)

Non-trainable params: 0 (0.00 B)

Optimizer params: 6,536 (25.54 KB)

(b) Structure of ANN model

Figure 4.4: Simple Building ANN model 1

Model 2:

To enhance performance further, we introduced Model 2, a more intricate ANN configuration. This model commenced with an input layer matching the feature count of the scaled training data. It comprised three hidden layers, each employing Rectified Linear Unit (ReLU) activation functions and L2 regularization to mitigate overfitting. The first hidden layer featured 128 neurons and a dropout rate of 0.5. The second hidden layer contained 64 neurons, also with a dropout rate of 0.5. The third hidden layer comprised 32 neurons. The output layer retained three neurons with a softmax activation function for classification shown in **(Figure 4.5)** . Through this augmented architecture, Model 2 achieved a remarkable accuracy of 98%.

This progression demonstrates the iterative nature of model refinement, showcasing the efficacy of augmenting complexity to enhance predictive performance.

```python
# Build the ANN model
model2 = Sequential()
model2.add(Dense(units=128, activation='relu', kernel_regularizer=regularizers.l2(0.001), input_dim=X_train_scaled.shape[1]))
model2.add(Dropout(0.5))  # Adding dropout for regularization
model2.add(Dense(units=64, activation='relu', kernel_regularizer=regularizers.l2(0.001)))
model2.add(Dropout(0.5))  # Adding dropout for regularization
model2.add(Dense(units=32, activation='relu', kernel_regularizer=regularizers.l2(0.001)))
model2.add(Dense(units=3, activation='softmax'))  # 3 output units for 3 classes
```

(a) Model ANN

Model: "sequential_2"

Layer (type)	Output Shape	Param #
dense_3 (Dense)	(None, 128)	2,304
dropout (Dropout)	(None, 128)	0
dense_4 (Dense)	(None, 64)	8,256
dropout_1 (Dropout)	(None, 64)	0
dense_5 (Dense)	(None, 32)	2,080
dense_6 (Dense)	(None, 3)	99

Total params: 38,219 (149.30 KB)

Trainable params: 12,739 (49.76 KB)

Non-trainable params: 0 (0.00 B)

Optimizer params: 25,480 (99.54 KB)

(b) Structure of ANN model

Figure 4.5: Simple Building ANN model 2

4.3 Evaluation and comparisons

4.3.1 Evaluation Metrics

In the process of training a deep learning and machine learning model, assessing its quality through evaluation metrics is crucial, and there exist various measures to do so. For our student performance prediction model, we utilize accuracy, recall, precision, and F1-Score as the evaluation metrics.

- Accuracy:In general, the accuracy metric measures the ratio of correct predictions over the total number of instances evaluated [25]. The formula for accuracy is:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{4.1}$$

- Precision:is used to measure the positive patterns that are correctly predicted from the total predicted patterns in a positive class [25] . The formula for precision is:

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

- Recall: is used to measure the fraction of positive patterns that are correctly classified [25] . The formula for the recall is:

$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

  where

  - TP: the number of true positives in the dataset.
  - TN: the number of true negatives in the dataset.
  - FP: the number of false positives in the dataset.

– FN: the number of false negatives in the dataset.

- F1-Measure: This metric represents the harmonic mean between recall and precision values [25] . The formula for precision is:

$$F1 - Measure(FM) = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4.4}$$

- ROC:curve evaluates classification model performance in binary classification by illustrating the relationship between True Positive Rate (TPR) and False Positive Rate (FPR) across different thresholds, plotted with FPR on the X-axis and TPR on the Y-axis [19] .

- TPR : is a measure that represents the proportion of true positives correctly identified out of all the actual positive instances [55]. It is also known as sensitivity.The formula for TPR is:

$$TPR = \frac{TP}{TP + FN} \tag{4.5}$$

- TNR: is the proportion of true negatives and complete number of negatives we have anticipated [55] . It is also known as specificity.The formula for TNR is:

$$TNR = \frac{TN}{TN + FP} \tag{4.6}$$

Once all the SPP models have been trained, we obtained a set of results. These results have been compiled and organized in the following table **(Table 4.4)**:

Table 4.4: Comparison of Model Performance

Model	Accuracy	Precision	Recall
SVM	0.78125	H 0.75 L 0.82 M 0.76	H 0.64 L 1.00 M 0.72
MLP	0.7917	H 0.83 L 0.81 M 0.76	H 0.71 L 0.89 M 0.78
Decision Tree	0.7083	H 0.72 L 0.77 M 0.65	H 0.64 L 0.86 M 0.65
ANN	0.98	H 1.00 L 0.97 M 1.00	H 0.96 L 1.00 M 1.00

Based on the information presented in the table, it can be concluded that the ANN model yields the best results shown in **(Figure 4.6)**.

We will provide some visualizations of the ANN model:

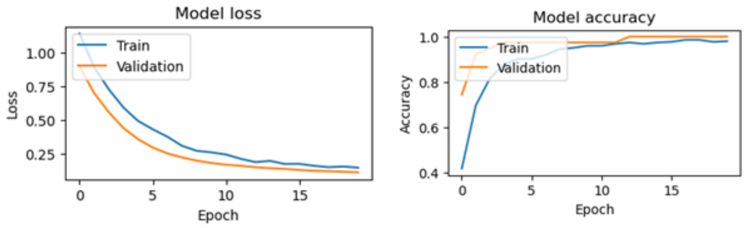
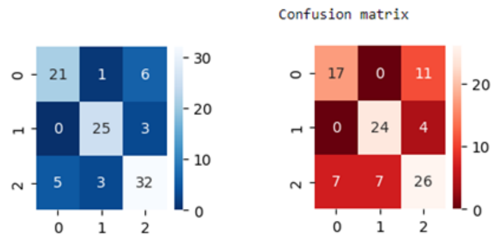


Figure 4.6: Visual Accuracy and Loss Results

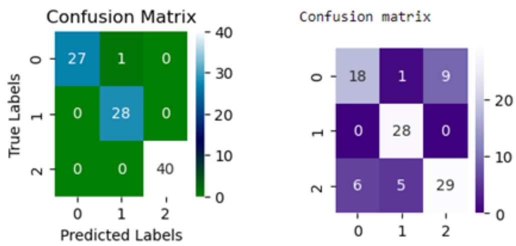
The chart on the left illustrates the increasing accuracy, while the chart on the right depicts the decreasing loss. It is worth noting that the training loss rate is higher than the validation loss rate. The validation accuracy is nearly identical to the training accuracy. As the number of iterations increases, the activity recognition accuracy rate for both training and validation steadily converges to one. Additionally, the loss rate gradually approaches zero, indicating a gradual improvement in the accuracy of activity recognition.

Here are the confusion matrices for the best model, which illustrate the model's performance in classifying the data shown in **(Figure 4.7) and (Figure 4.8)** .



(a)Model MLP (b)Model DT

Figure 4.7: Confusion Matrix for DT and MLP



(a)Model ANN (b) Model SVM

Figure 4.8: Confusion Matrix for ANN and SVM

4.3.2 Baseline Models:

To establish a benchmark for our proposed ANN model, we opted for Logistic Regression and K-Nearest Neighbors (KNN) decision tree, random forest, support vector machine (svm) as baseline models. This selection aligns with the research conducted by [38] , where these models were employed for student performance prediction using data with similar characteristics to our own.

Based on the provided comparison table **(Table 4.5)**, the performance of various classification models is evaluated across a range of metrics, including precision, recall, F1-score, and overall accuracy. Each of these metrics offers a different perspective on the performance of the models and how they interact with the data. Let's discuss these results in more detail:

- KNN (K-Nearest Neighbors):

  - The KNN model achieves moderate performance across all metrics, with precision ranging from 0.63 to 0.75 and recall ranging from 0.62 to 0.94. However, its F1-score ranges from 0.62 to 0.72, indicating some imbalance between precision and recall. The overall accuracy of the KNN model is 0.73.

- Random Forest:

  - The Random Forest model demonstrates better performance compared to KNN, with precision ranging from 0.65 to 0.82 and recall ranging from 0.72 to 0.94. The F1-score for Random Forest ranges from 0.68 to 0.74. The overall accuracy of the Random Forest model is 0.76.

- SVM (Support Vector Machine):

  - The SVM model shows competitive performance, with precision ranging from 0.68 to 0.83 and recall ranging from 0.72 to 0.88. The F1-score for SVM ranges from 0.70 to 0.75. The overall accuracy of the SVM model is 0.75.

- Logistic Regression:

  - The Logistic Regression model achieves consistent performance, with precision ranging from 0.68 to 0.82 and recall ranging from 0.67 to 0.91. The F1-score for Logistic Regression ranges from 0.68 to 0.75. The overall accuracy of the Logistic Regression model is 0.75.

- Decision Tree (Gini and Entropy):

- Decision Tree models exhibit varying performance. The Gini-based Decision Tree achieves precision ranging from 0.65 to 0.79 and recall ranging from 0.67 to 0.88, with an overall accuracy of 0.73. The Entropy-based Decision Tree achieves precision ranging from 0.67 to 0.72 and recall ranging from 0.46 to 0.85, with an overall accuracy of 0.68.

- Proposed System:

  - The proposed system outperforms all other models across all metrics. It achieves perfect precision (1.00) and recall for most classes and a high F1-score ranging from 0.98 to 1.00. The overall accuracy of the proposed system is 0.98, indicating its superior performance in accurately classifying data into high, medium, and low classes.

In conclusion, the comparison table highlights the effectiveness of the proposed system, which achieves superior performance compared to other classification models. Its high precision, recall, F1-score, and overall accuracy demonstrate its capability to accurately classify data across different classes.

Table 4.5: Comparison of Model Performance with Baseline Models

Model	Class	Precision	Recall	F1-score	Accuracy
KNN	H	0.63	0.62	0.62	0.73
	L	0.78	0.94	0.85	
	M	0.75	0.69	0.72	
Random forest	H	0.65	0.72	0.68	0.76
	L	0.82	0.94	0.88	
	M	0.79	0.69	0.74	
SVM	H	0.68	0.72	0.70	0.75
	L	0.83	0.88	0.86	
	M	0.78	0.73	0.75	
Regression logistique	H	0.68	0.67	0.68	0.75
	L	0.82	0.91	0.86	
	M	0.76	0.73	0.75	
Decision tree gini	H	0.65	0.67	0.66	0.73
	L	0.79	0.88	0.83	
	M	0.74	0.69	0.72	
Decision tree entropy	H	0.67	0.46	0.55	0.68
	L	0.72	0.85	0.78	
	M	0.66	0.72	0.69	
Proposed system	H	1.00	0.96	0.98	0.98
	L	0.97	1.00	0.98	
	M	1.00	1.00	1.00	

4.4 Conclusion

This chapter is divided into two parts. In the first part, we introduced the working environment, programming language, and training and testing procedures. We also provided details on the structures of the different models and presented the results in a table. Based on the obtained results, the ANN model demonstrated the best performance in detecting and identifying student performance prediction in intelligent environments. These results have provided valuable insights and perceptions for further analysis.

General Conclusion

In this thesis, we aimed to address the issue of student performance prediction using machine learning models. The problem at hand is the difficulty educational institutions face in predicting student performance accurately, which is crucial for early intervention and improving student outcomes. Our primary objectives were to compare different algorithms, identify the most effective model, and assess its relative effectiveness against baseline models.

To achieve these objectives, we employed a comprehensive methodology that included data collection, preprocessing, and feature engineering to prepare the dataset for analysis. We then implemented and fine-tuned four machine learning models: Support Vector Machines(SVM), Multilayer Perceptrons(MLP), Decision Trees, and Artificial Neural Networks(ANNs). Hyper-parameter tuning was performed using grid search to optimize the performance of these models. The models were evaluated based on their accuracy, precision, recall, and F1 score to determine the most effective approach for predicting student performance.

Our analysis revealed that the Artificial Neural Network(ANN) model outperformed other models in terms of accuracy and predictive power. Logistic Regression and Random Forest also demonstrated strong performance, whereas Decision Trees showed relatively lower accuracy.

These findings suggest that ANN's ability to handle complex data relationships makes it a superior choice for predicting student performance. This aligns with previous research indicating the robustness of neural networks in educational data analysis.

The results of this study can inform educational institutions in developing more effective predictive analytics tools. By leveraging accurate models, educators can identify at-risk students earlier and implement targeted interventions.

One limitation of this study is the reliance on a single dataset, which may affect the generalizability of the results. Future research should explore the use of diverse datasets to validate the findings and consider ensemble learning techniques, as well as investigating the impact of additional features.

In conclusion, this thesis demonstrates the effectiveness of machine learning models, particularly ANNs, in predicting student performance. The insights gained from this research contribute to the development of more accurate and reliable educational tools, ultimately enhancing student outcomes. 1.2

Bibliography

[num] Numpy. `https://numpy.org/`. Accessed: 26-May-2024.

[pan] Pandas. `https://pandas.pydata.org`. Accessed: 26-May-2024.

[sci] scikit-learn. `https://scikit-learn.org/stable/index.html`. Accessed: 26-May-2024.

[ten] Tensorflow. `https://www.tensorflow.org/?hl=fr`. Accessed: 26-May-2024.

[5] AbdElrahman, A., Soliman, T. H. A., Taloba, A. I., and Farghally, M. F. (2022). A predictive model for student performance in classrooms using student interactions with an etextbook.

[6] Abubakaria, M. S., Arifin, F., and Hungilo, G. G. (2020). Predicting students' academic performance in educational data mining based on deep learning using tensorflow. *Int. J. Educ. Manage. Eng.(IJEME)*, 10(6):27–33.

[7] Ajgaonkar, S. (2022). *Practical Automated Machine Learning Using H2O. ai: Discover the power of automated machine learning, from experimentation through to deployment to production*. Packt Publishing Ltd.

[8] Aljarah, I. (2016). Students' academic performance dataset. `https://www.kaggle.com/aljarah/xAPI-Edu-Data`. Accessed: 21-Jul-2024.

[9] Antoninis, M., April, D., Barakat, B., Bella, N., D'Addio, A. C., Eck, M., Endrizzi, F., Joshi, P., Kubacka, K., McWilliam, A., et al. (2020). All means all: An introduction to the 2020 global education monitoring report on inclusion. *Prospects*, 49:103–109.

[10] Arnold, K. E. and Pistilli, M. D. (2012). Course signals at purdue: Using learning analytics to increase student success. In *Proceedings of the 2nd international conference on learning analytics and knowledge*, pages 267–270.

[11] Bendangnuksung, P. P. (2018). Students' performance prediction using deep neural network. *International Journal of Applied Engineering Research*, 13(2):1171–1176.

[12] Bowen, J. (1972). A history of western education.

[13] Buraimoh, E. F., Ajoodha, R., and Padayachee, K. (2021). *Predicting student success using student engagement in the online component of a blended-learning course*. PhD thesis, MS thesis, School Comput. Sci. Appl. Math., Dept. Sci., Univ. Witwatersrand ....

[14] Chavez, H., Chavez-Arias, B., Contreras-Rosas, S., Alvarez-Rodríguez, J. M., and Raymundo, C. (2023). Artificial neural network model to predict student performance using nonpersonal information. In *Frontiers in Education*, volume 8, page 1106679. Frontiers Media SA.

[15] Dastres, R. and Soori, M. (2021). Artificial neural network systems. *International Journal of Imaging and Robotics (IJIR)*, 21(2):13–25.

[16] Delen, D. (2011). Predicting student attrition with data mining methods. *Journal of College Student Retention: Research, Theory & Practice*, 13(1):17–35.

[17] Devisetty, U. K. (2022). *Deep Learning for Genomics: Data-driven approaches for genomics applications in life sciences and biotechnology*. Packt Publishing.

[18] El Naqa, I. and Murphy, M. J. (2015). *What is machine learning?* Springer.

[19] Fawcett, T. (2006). An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874.

[20] Finn, J. D. and Zimmer, K. S. (2012). Student engagement: What is it? why does it matter? In *Handbook of research on student engagement*, pages 97–131. Springer.

[21] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

[22] Hanushek, E. A. and Woessmann, L. (2023). *The knowledge capital of nations: Education and the economics of growth*. MIT press.

[23] Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

[24] Hattie, J. (2008). *Visible learning: A synthesis of over 800 meta-analyses relating to achievement*. routledge.

[25] Hossin, M. and Sulaiman, M. N. (2015). A review on evaluation metrics for data classification evaluations. *International journal of data mining & knowledge management process*, 5(2):1.

[26] Hu, J., Zhou, T., Ma, S., Yang, D., Guo, M., and Huang, P. (2022). Rock mass classification prediction model using heuristic algorithms and support vector machines: a case study of chambishi copper mine. *Scientific Reports*, 12(1):928.

[27] Indicators, O. (2013). Education at a glance 2016 oecd indicators.

[28] Jennifer, P., Contractor, S., M.R, E., V.V, K., and V, R. K. (2023). Student performance prediction using ml: A survey.

[29] Jović, J., Kisić, E., Raspopović Milic, M., Domazet, D., and Chandra, K. (2022). Prediction of student academic performance using machine learning algorithms.

[30] Kaunang, F. J. and Rotikan, R. (2018). Students' academic performance prediction using data mining. In *2018 Third International Conference on Informatics and Computing (ICIC)*, pages 1–5. IEEE.

[31] Kumar, N. and Kumar Nimbark, H. (2023). Classification model development for performance analysis of lms using machine learning algorithm and blockchain framework. *Journal Name*, 12(03).

[32] Lau, E. T., Sun, L., and Yang, Q. (2019). Modelling, prediction and classification of student academic performance using artificial neural networks. *SN Applied Sciences*, 1(9):982.

[33] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.

[34] Maimon, O. Z. and Rokach, L. (2014). *Data mining with decision trees: theory and applications*, volume 81. World scientific.

[35] Means, B., Toyama, Y., Murphy, R., Bakia, M., and Jones, K. (2009). Evaluation of evidence-based practices in online learning: A meta-analysis and review of online learning studies.

[36] Mezirow, J. (1997). Transformative learning: Theory to practice. *New directions for adult and continuing education*, 1997(74):5–12.

[37] Nawang, H., Makhtar, M., and Hamzah, W. (2021). A systematic literature review on student performance predictions. *International Journal of Advanced Technology and Engineering Exploration*, 8(84):1441–1453.

[38] Nirmala, M., Selvi, T. S., and Saravanan, V. (2021). Student academic performance prediction under various machine learning classification algorithms. *Int. J. Res. Appl. Sci. Eng. Technol.*, 9(11):221–237.

[39] Niu, Y., Zhou, T., Li, Z., and Liu, H. (2023). Student grade prediction model based on rfe_rf and integrated learning voting algorithm. In *2023 2nd International Conference on Educational Innovation and Multimedia Technology (EIMT 2023)*, pages 1193–1201. Atlantis Press.

[40] Padma, E., Yasin, T., and Srivatsav, M. (2022). Students performance prediction using machine learning algorithms.

[41] Patil, M. S., Sahasrabuddhe, D., and Pujari, V. (2021). Understudies' performance prediction using deep neural network. *Journal Name*, 11(3).

[42] Patle, A. and Chouhan, D. S. (2013). Svm kernel functions for classification. In *2013 International conference on advances in technology and engineering (ICATE)*, pages 1–9. IEEE.

[43] Psacharopoulos, G. and Patrinos, H. A. (2018). Returns to investment in education: a decennial review of the global literature. *Education Economics*, 26(5):445–458.

[44] Richardson, M., Abraham, C., and Bond, R. (2012). Psychological correlates of university students' academic performance: a systematic review and meta-analysis. *Psychological bulletin*, 138(2):353.

[45] Rimadana, M. R., Kusumawardani, S. S., Santosa, P. I., and Erwianda, M. S. F. (2019). Predicting student academic performance using machine learning and time management skill data. In *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, pages 511–515. IEEE.

[46] Saa, A. A. (2016). Educational data mining & students' performance prediction. *International journal of advanced computer science and applications*, 7(5).

[47] Saadi, F. and Lakas, N. E. (2023). *Human Activity Recognition approach for Energy Efficiency based on Deep Learning*. PhD thesis, university ghardaia.

[48] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61:85–117.

[49] Schneider, M. and Preckel, F. (2017). Variables associated with achievement in higher education: A systematic review of meta-analyses. *Psychological bulletin*, 143(6):565.

[50] Singh, V., Singh, D., and Kumar, H. (2024). Efficient application of deep neural networks for identifying small and multiple weed patches using drone images. *IEEE Access*.

[51] Sultana, J., Usha, M., and Farquad, M. (2018). An efficient deep learning method to predict students performance. *Higher Education Quality Assurance and Enhancement*.

[52] Tan, P.-N., Steinbach, M., Karpatne, A., and Kumar, V. (2020). *Introduction to Data Mining*. Pearson, 2nd edition.

[53] Tyack, D. B. and Cuban, L. (1997). *Tinkering toward utopia: A century of public school reform*. Harvard University Press.

[54] Umar, M. A. (2019). Student academic performance prediction using artificial neural networks: A case study. *International Journal of Computer Applications*, 178(48):24–29.

[55] Vijayalakshmi, V. and Venkatachalapathy, K. (2019a). Comparison of predicting student's performance using machine learning algorithms. *International Journal of Intelligent Systems and Applications*, 10(12):34.

[56] Vijayalakshmi, V. and Venkatachalapathy, K. (2019b). Deep neural network for multi-class prediction of student performance in educational data. *International Journal of Recent Technology and Engineering*, 8(2):5073–5081.

[57] Xiao, W. and Hu, J. (2023). Analyzing effective factors of online learning performance by interpreting machine learning models. *IEEE Access*, 11:132435–132447.

[58] Yağcı, M. (2022). Educational data mining: prediction of students' academic performance using machine learning algorithms. *Smart Learning Environments*, 9(1):11.

[59] York, T. T., Gibson, C., and Rankin, S. (2019). Defining and measuring academic success. *Practical assessment, research, and evaluation*, 20(1):5.

[60] Yurtkan, K., Adalier, A., and Umut, T. (2023). Student success prediction using feedforward neural networks. *Romanian Journal of Information Science and Technology*, 26(2):121–136.

République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'Enseignement Supérieur et de La Recherche Scientifique
كلية العلوم والتكنولوجيا
Faculté des Sciences et de la Technologie
قسم الرياضيات و الإعلام الآلي
Département des Mathématiques & de l'Informatique
جامعة غرداية
Université de Ghardaia

شهادة الترخيص بالإيداع

أنا الأستاذ : سليمان بالاعور

بصفتي رئيس لجنة المناقشة والمسؤول عن تصحيح مذكرة الماستر الموسومة ب:

Improving Student Performance Prediction using Artificial Intelligence

من إنجاز الطالبة: ماريا كربوب (Maria Kerboub)

والطالبة: عائشة هريات (Aicha Heriat)

الكلية: العلوم والتكنولوجيا
القسم: الرياضيات والإعلام الآلي
الشعبة: إعلام آلي

التخصص: الأنظمة الذكية لاستخراج المعارف
تاريخ المناقشة: 2024/26/23

أشهد أن الطلبة قاموا بالتصحيحات المطلوبة من طرف لجنة المناقشة وأن المطابقة بين النسخة الورقية والالكترونية استوفيت جميع شروطها.

مصادقة رئيس القسم

رئيس قسم الرياضيات و الإعلام الآلي
الحاج موسى ياسين

جامعة غرداية
قسم الرياضيات و الاعلام الآلي
كلية العلوم والتكنولوجيا

إمضاء المسؤول عن التصحيح

سليمان بالاعور