DOI: https://doi.org/10.52756/ijerr.2024.v40spl.005 *Int. J. Exp. Res. Rev.*, Special Vol. 40: 56-72 (2024)

Original Article | Peer Reviewed | Open Access

International Journal of Experimental Research and Review (IJERR)

ISSN: 2455-4855 (Online)
www.iaph.in

Predicting Student Academic Performance Using Neural Networks: Analyzing the Impact of Transfer Functions, Momentum and Learning Rate

Mini Agarwal* and Dr. Bharat Bhushan Agarwal

CSE Department, IFTM University, Moradabad, India

E-mail/Orcid Id:

MA, miniagarwal21@gmail.com, https://orcid.org/0000-0002-7233-1787;
BBA, bharatagarwal9@gmail.com, https://orcid.org/0000-0001-5519-3434

Article History:

Received: 18th Sept., 2023
Accepted: 14th June, 2024
Published: 30th June, 2024

Keywords:

Artificial Neural Network (ANN), Back Propagation (BP), Prediction, Multilayer Perceptron (MLP), COVID

How to cite this Article:
Mini Agarwal and Bharat Bhushan Agarwal (2024). Predicting Student Academic Performance Using Neural Networks: Analyzing the Impact of Transfer Functions, Momentum, and Learning Rate. *International Journal of Experimental Research and Review*, *40*(spl.), 56-72.
DOI: https://doi.org/10.52756/ijerr.2024.v40spl.005

Abstract: Artificial Neural Networks (ANN) demonstrate a compelling application of AI in predicting student performance, a critical aspect for both students and educators. Accurate forecasting of student achievements enables educators to monitor progress effectively, allowing educational institutions to optimize outcomes and improve student results. This study focuses on leveraging ANN for predictive analytics in student performance. Through a detailed evaluation of transfer functions, optimizers, learning rates, and momentum values, the model achieves an impressive 98% accuracy with specific configurations: a learning rate of 0.005, momentum of 0.7, Sigmoid transfer function, and SGD optimizer. Additionally, the study performs a comparative analysis of various Machine Learning Algorithms, including ANN, Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Decision Trees (DT), Naive Bayes (NB), and Logistic Regression (LR). Using data from 689 B.Tech students at IP University, the analysis reveals that ANN outperforms other algorithms with an accuracy of 97%. This high accuracy demonstrates the potential of ANN in educational settings, providing a valuable tool for educators to enhance student performance and outcomes.

Introduction

Technology is advancing rapidly, permeating both our daily lives and professional endeavors. This relentless progression has triggered notable transformations in the education sector, rooted in the fields of data mining and the application of artificial intelligence (AI). Predicting student performance is a very typical task in the education field, and different methods, algorithms, and approaches have been researched and applied to the use of machine learning (ML), educational data mining (EDM), and artificial neural networks to predict student performance for the best result (Abulhaija et al.,2023; Preetha and Anitha, 2022). Considering and extracting features have played important roles in developing decision-making models for predicting student performance (Zaffar et al., 2020). Learning management systems generate valuable quantitative insights through reports and learning data, enabling educators to analyze and improve course content and its delivery (Kuppusamy and Joseph, 2021). The emergence of global pandemics, such as COVID-19, has further accelerated the migration of education to digital platforms, alleviating concerns related to homework, tests, and attendance for young students. However, this shift has also reduced face-to-face interactions between educators and pupils, potentially impacting the seriousness with which online courses are taken.

Addressing this challenge is imperative, as we must develop the capability to predict student performance

*Corresponding Author: miniagarwal21@gmail.com

when instruction transitions away from online learning. Reliable predictions are crucial in preventing a significant decline in student achievement. Ultimately, the application of predictive analytics can benefit not only students but also professors, administrators, and the overall reputation of educational institutions.

Students today employ intelligent devices to connect to wireless networks and access digital content, enabling them to engage in customized and uninterrupted learning experiences. This concept of smart education, characterizing learning in the digital era, has garnered increasing interest (Zhu et al., 2016).

Teachers often lack real-time insights into students' actual performance, so they may resort to extrapolating their performance based on statistical data (Hamadneh et al., 2022). Utilizing an evolving composite model can evaluate the interrelation between the learning process, course components, and student performance (Jiao et al., 2022). Through the use of artificial neural networks and data mining models, assessment metrics and key factors affecting student performance can be examined to determine the most effective approaches (Rodríguez-Hernández et al., 2021).

Initially, our focus lies in investigating and implementing an Artificial Neural Network (ANN). Ensuring that our model operates effectively and produces the desired results is crucial. Subsequently, we will compare various machine learning algorithms using our dataset to determine which yields the highest quality results.

This investigation prompts several key questions:

Q.1 How does the performance of a neural network compare to that of other classifiers?

Q.2 What are the factors that influence the performance of an ANN in the context of education?

Q.3 How can manipulating epochs, training, and testing sizes contribute to reducing errors in the ANN?".

Artificial Neural Network (ANN)

ANN is a combination of processing elements that are connected through a wire, and this connection is called neurons. These neurons have two layers: the first layer is the input layer, and the second layer is the output layer. In ANN all the neurons are connected to perform a specific task (Haloi et al., 2023; Venkata and Damodar, 2023). In this, each neuron is called a node and each connection means the neuron-to-neuron connection is called edges. These edges have some weights that are multiplied by the input node. Summation of all inputs after weights and activation function sent to the output layer. Weights provide firmness to each neuron connection. The activation function is a function that helps in providing a goal.

Working of Multilayer Perceptron (MLP)

Multilayer perceptron has been trained using both supervised and unsupervised learning methods. In supervised learning, training is to identify whether the selected object belongs to specified groups of predictors or not. MLP deals with both prediction and classification issues. MLP has three layers, first, A layer represents the input layer in which the predictor applies the input variable. These input variables multiply with weights that are passed to the second layer, i.e., the hidden layer performs some operations and maps with input data and the last layer is the output layer that produces the output. Some activation functions are also applied to the implementation of MLP. The predictor finds errors in the output by comparing the predicted output with the desired output. If a difference comes, it backpropagates the error to the model until we find the desired output.

The Backpropagation Method (BP)

Using Supervised learning in MLP has expanded the implementation of BP. BP occurs in two stages first one is the forward stage and another is the backward stage. In the forward propagation, the predictive weights of the MLP are evaluated and the input signals are sent through the layers until the desired output is achieved. In the second stage, backward propagation, the error signal is produced by comparing the MLP output to the expected output. This signal is propagated among the layers but in the backward direction. Through this, MLP can optimize the predictive weights and minimize the errors in each iteration until a certain accuracy is achieved. In the present study Gradient descent optimization function is used to minimize the error. In our research, we adjusted the learning rate in each cycle for the learning process and the activation function to achieve an accurate result (Rodríguez-Hernandez et al., 2021).

Related work

Previous research studies are important as they are the foundation for new research endeavors. My research idea is rooted in previous studies' findings, which have contributed valuable insights to the field. While there are numerous relevant papers, Table 1 highlights some of the most crucial ones for reference.

In this research, we employ a range of machine learning classifiers, including Support Vector Machines (SVM), Decision Trees, Random Forest, Logistic Regression, Naïve Bayes, and Artificial Neural Networks (ANN). Our objective is to examine previous research that has utilized these classifiers comprehensively. We

DOI: https://doi.org/10.52756/ijerr.2024.v40spl.005

analyze this extensive body of work from various angles, considering factors related to education, psychology, emotions and students' backgrounds.

DOI: https://doi.org/10.52756/ijerr.2024.v40spl.005

Table 1. Summarization of previous work.

Author	Classifiers	Outcome
Musso et al., 2013	ANN	Divided students into two groups (greater than 33% and less than 33%) and achieved 100% accurate categorization.
Triventi, 2014	Binomial Regression	Analyzed the impact of working hours on working students' study methods.
Kyndt et al., 2015	ANN	Predicted the student's end-term performance after the first-year completion based on three approaches cognition, motivation, and learning.
Mesarić, 2016	Decision tree	Classified students into different groups based on first-year results and teacher rankings with 79% accuracy.
Zhu et al., 2016	Framework of Smart Education	Proposed a three-tier framework for Smart Education.
Alves et al., 2017	structural equation model (SEM)	Findings: Explored factors affecting student performance, with family variables contributing significantly (90%).
Ahmad and Shahzadi, 2018	ANN	They were predicted student passing risks with 95% training accuracy and 85% testing accuracy.
Adekitan and Salau, 2019	Decision Tree, Random Forest, Naïve Bayes, PNN, Tree Ensemble, Logistic Regression	Analyzed three years of grading data to predict final year results, with Logistic Regression achieving 89.15% accuracy.
Abu-Zohair, 2019	NB, KNN, LDA, MLP, SVM	Analyzed data for start-up universities and found LDA performed best with 79% accuracy.
Vairachilai and Vamshidharreddy, 2020	Decision Tree, Support Vector Machine (SVM), and Naive Bayes	Analyzed data for start-up universities and found LDA performed best with 79% accuracy.
Zhang et al., 2021	Artificial Intelligence and Educational Data Mining Algorithms	Compared various AI and DM algorithms and identified Decision Tree and Logistic Regression as effective for complex problems.
Ahmad et al., 2021	ANN	Predicted student results based on first semester scores with 93.20% accuracy
Ghosh and Janan, 2021	Random Forest Classifier	Investigated reasons for student dropouts and achieved a 98.66% accuracy rate.
Agarwal and Agarwal, 2022	Data mining Classifiers and ANN	Analyzed different classifiers and ANN models, with Decision Tree and Naïve Bayes achieving the highest prediction accuracy.
Orji and Vassileva, 2022	Decision Tree, K-Nearest Neighbour, Random Forest, Logistic Regression, and Support Vector Machine	Studied student learning patterns and achieved 94.9% accuracy with Random Forest.
Yadav and Deshmukh, 2022	Artificial Intelligence and Data Mining classification algorithms	Explored various classification and ANN algorithms, with accuracy varying based on attributes.

DOI: https://doi.org/10.52756/ijerr.2024.v40spl.005

Wojciukc et al., 2022	CNN	The research assesses the significance of hyperparameters, determines the most effective ranges for these hyperparameters, and evaluates various optimization techniques.
Honghe Jin, 2022	Supervised Learning Machine Algorithms	The paper introduces a concept of hyperparameter importance by analyzing the variance of the risk function across different hyperparameter values. Additionally, it outlines a technique for estimating this importance through subsampling procedures.
Liu et al., 2023	Reinforcement Learning	The paper introduces an innovative approach to accelerate the training process of hyperparameter optimization (HPO) for machine learning algorithms, addressing the challenge of time and resource-intensive procedures.
Chavez et al., 2023.	ANN	They have predicted student exam outcomes without revealing student information, achieving 93.81% accuracy.

Methods and Materials

In our research, Figure 1 illustrates the framework we employed, comprising various stages. In a study by Carlos Felipe Rodríguez-Hernández et al., they tested different parameters such as learning rate values (0.001, 0.0005, 0.0001, 0.00005, 0.00001) and transfer functions for hidden and output layers (hyperbolic tangent, Linear sigmoid, Sigmoid and SoftMax), resulting in a high accuracy of 82% for the model. To further enhance the model's accuracy, we individually applied each of the three transfer functions (Sigmoid, ReLU. and Softmax) to both the hidden and output layers. We chose these functions because they are suitable for different types of tasks: Sigmoid for binary classification, ReLU for efficient processing in hidden layers, and Softmax for multi-class classification. Additionally, we adjusted the learning rate values by multiplying them by 5 (0.001, 0.0001, 0.005, 0.0005) and the momentum value (ranging from 0.1 to 0.9) due to the achieved accuracy of our model. These adjustments enabled the model to make small weight updates, which is beneficial for fine-tuning the model or handling complex data patterns. Further details on these adjustments are provided below.

Data Collection

The data collection process unfolded in two distinct phases. Initially, we conducted a questionnaire survey involving 150 stakeholders to pinpoint pertinent attributes. Subsequently, we gathered data from 689 B.Tech students at IP University via Google Forms. These attributes were subsequently grouped into three categories: psychological, educational, and background traits, which exhibit interconnectedness. Background attributes encompass familial elements such as the number of siblings, parental income, educational achievement, and caste. Educational traits encompass data related to prior educational experiences, attendance, admission methods, scholarships, assignments, and language proficiency. Health factors are important for the physical and mental well-being of students. Parental relations signify whether the parents share a blood relation or not. Lastly, travel time indicates the duration of a student's commute. All the Attributes and their ranges are shown in Table 2.

Statistical Analysis

Table 2 represents the statistical examination of the attributes processed in this study. We computed each attribute's valid frequency, cumulative frequency, mean, standard deviation, variance, and p-value. It's noteworthy that no outliers were identified during the analysis.

Data Preparation and Initialization

In this, we prepare the data for data processing. We converted each attribute name from A1 to A19, as shown in Table 2. We apply Formula 1 (multiplying each attribute with the certain weight w_n and domain range f_n) to calculate the attribute's new domain range (A_n).

$$A_n = f_n * w_n \quad \text{--------(Formula 1)}$$

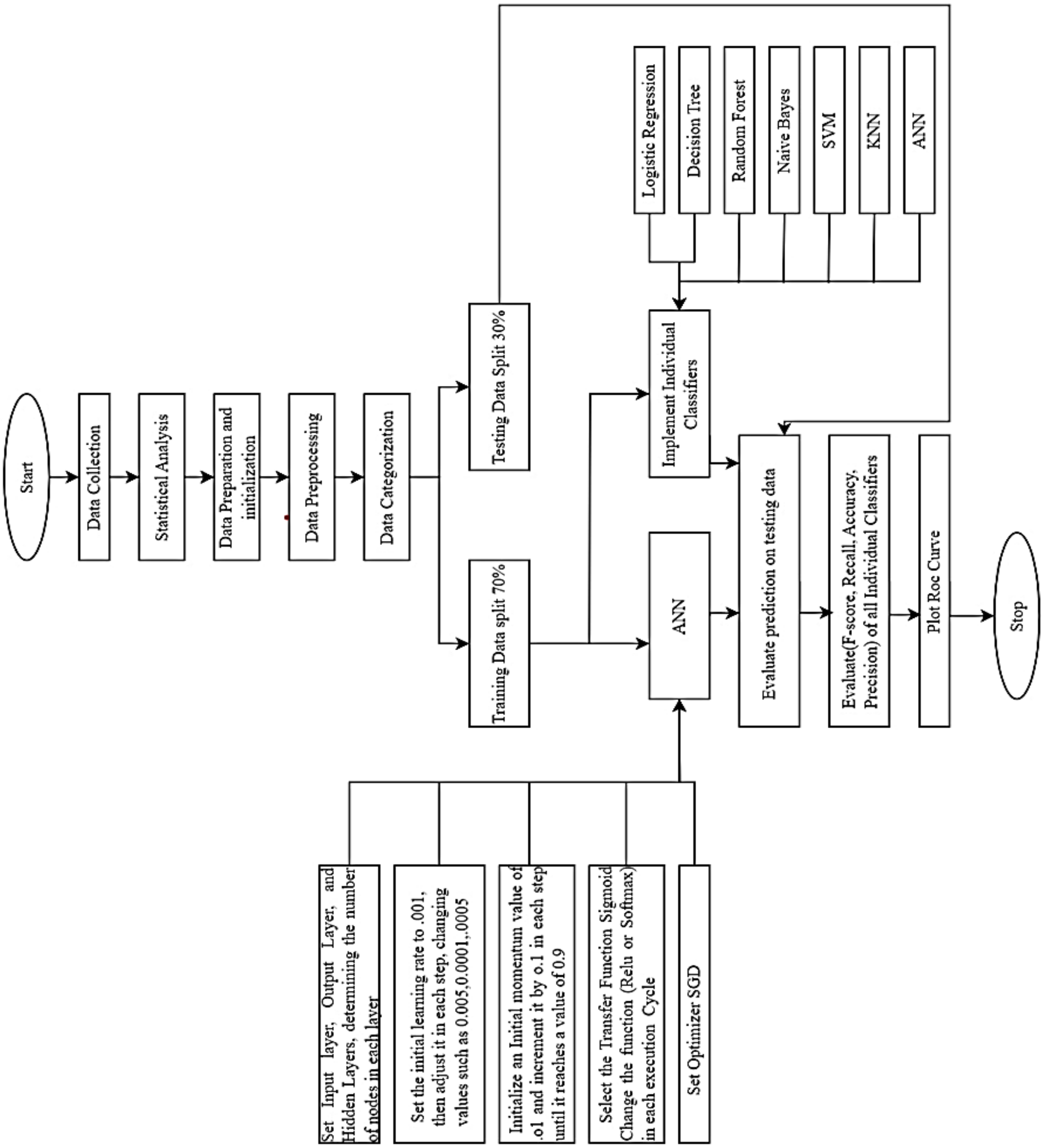

Figure 1. Framework of the research.

DOI: https://doi.org/10.52756/ijerr.2024.v40spl.005

Table 2. Attribute Statistical Description.

Attribute	marks %	Domain Range	Frequency	Valid Percentage	Cumulative Percentage	Mean	Standard Deviation	Variance	P-Value
10th Marks (A1)	> 33%	0	0	0	0	0.85	0.13	0.02	3.84E-04
	33% - 40%	0.4	26	4	3.77				
	41% - 50%	0.5	18	3	6.38				
	51% - 60%	0.6	18	3	8.99				
	61% - 70%	0.7	28	4	13.06				
	71% - 80%	0.8	43	6	19.3				
	81% - 100%	0.9	556	81	100				
	Total		**689**	**100**					
12th Marks (A2)	> 33%	0	0	0	0	0.83	0.13	0.018	2.42E-07
	33% - 40%	0.4	22	3	3.193				
	41% - 50%	0.5	33	5	7.98				
	51% - 60%	0.6	29	4	12.19				
	61% - 70%	0.7	26	4	15.96				
	71% - 80%	0.8	50	7	23.22				
	81% - 100%	0.9	529	77	100				
	Total		**689**	**100**					
B.Tech Iyr Marks (A3)	> 33%	0	0	0	0	0.81	0.13	0.016	1.93E-16
	33% - 40%	0.4	24	3	3.48				
	41% - 50%	0.5	14	2	5.51				
	51% - 60%	0.6	31	4	10.01				
	61% - 70%	0.7	41	6	15.96				
	71% - 80%	0.8	224	33	48.48				
	81% - 100%	0.9	355	52	100				
	Total		**689**	**100**					
Parents Annual Salary (A4)	below 199999	0.4	269	39	39.04	0.59	0.14	0.019	3.73E-10
	200000<=599999	0.6	221	32	71.11				
	600000<=1099999,	0.7	162	24	94.63				
	1100000<=1599999	0.8	24	3	98.11				
	greater than1600000	0.9	13	2	100				
	Total		**689**	**100**					
Language (A5)	Others	0.4	268	39	38.89	0.52	0.09	0.009	1.00E-18
	English	0.6	421	61	100				
	Total		**689**	**100**					
Category (Caste) (A6)	General	0.4	324	47	47.02	0.53	0.13	0.016	4.64E-03
	OBC	0.6	189	27	74.45				
	SC & ST	0.7	176	26	100				
	Total		**689**	**100**					
Admission Mode (A7)	Management Quota	0.4	268	39	38.89	0.52	0.09	0.009	1.22E-09
	Enterance	0.6	421	61	100				
	Total		**689**	**100**					
Attendance (A8)	> 30%	0	0	0	0	0.76	0.17	0.03	2.09E-26
	30% - 40%	0.4	89	13	12.91				
	41% - 50%	0.6	111	16	29.02				
	51% - 60%	0.7	71	10	39.33				
	61% - 70%	0.8	30	4	43.68				
	Above 70%	0.9	388	56	100				
	Total		**689**	**100**					
Scholarship (A9)	No	0.4	97	14	14.07	0.57	0.07	0.005	6.70E-03
	Yes	0.6	592	86	100				
	Total		**689**	**100**					
Gender (A10)	Female	0.6	96	14	100	0.43	0.07	0.005	1.12E-06
	Male	0.4	593	86	86.06				
	Total		**689**	**100**					
Mother Education (A11)	below 10	0	0	0	0	0.75	0.14	0.018	1.06E-26
	10^{th}	0.4	41	6	5.95				
	12^{th}	0.6	144	21	26.85				
	Graduation	0.8	337	49	75.76				
	Post Graduation	0.9	167	24	100				
	Total		**689**	**100**					

DOI: https://doi.org/10.52756/ijerr.2024.v40spl.005

Father Education (A12)	below 10	0	0	0	0	0.57	0.07	0.004	1.20E-05
	10^{th}	0.4	33	5	4.78				
	12^{th}	0.6	105	15	20.02				
	Graduation	0.8	417	61	80.55				
	Post Graduation	0.9	134	19	100				
	Total		**689**	**100**					
Siblings (A13)	Yes	0.4	297	43	43.1	0.57	0.07	0.005	6.12E-16
	No	0.6	392	57	100				
	Total		**689**	**100**					
Assignment (A14)	No	0.4	318	46	46.15	0.57	0.07	0.005	4.20E-22
	Yes	0.6	371	54	100				
	Total		**689**	**100**					
Mother's Job (A15)	Other	0.4	49	7	7.11	0.57	0.07	0.01	2.45E-06
	Home Maker	0.5	16	2	9.43				
	Civil services	0.6	10	1	10.88				
	Health care	0.7	28	4	14.94				
	Business	0.8	365	53	67.92				
	Teacher	0.9	221	32	100				
	Total		**689**	**100**					
Father's Job (A16)	Other	0.4	46	7	6.67	0.57	0.07	0.01	6.20E-05
	Home Maker	0.5	18	3	9.28				
	Civil services	0.6	9	1	10.59				
	Health care	0.7	27	4	14.51				
	Business	0.8	374	54	68.79				
	Teacher	0.9	215	31	100				
	Total		**689**	**100**					
Travel Time (A17)	15 mins - 30 mins	0.4	320	46	46.44	0.57	0.07	0.004	6.31E-12
	1 hour	0.6	199	29	75.32				
	<1hour	0.7	170	25	100				
	Total		**689**	**100**					
Health Issue	Yes	0.4	435	63	63.13	0.47	0.09	0.009	8.35E-13
	No	0.6	254	37	100				
	Total		**689**	**100**					
Parents Status	Divorced	0.4	322	47	46.73	0.50	0.10	0.009	3.69E-04
	Living Together	0.6	367	53	100				
	Total		**689**	**100**					

Data Preparation and Initialization

In this, we prepare the data for data processing. We converted each attribute name from A1 to A19, as shown in Table 2. We apply Formula 1 (multiplying each attribute with the certain weight w_n and domain range f_n) to calculate the attribute's new domain range (A_n).

$$A_n = f_n * w_n \quad \text{--------(Formula 1)}$$

After applying the formula on the attribute, we calculate the attribute range according to the below formulas: -

A1=A1*1.5, A2 = A2*2, A3 = A3*4, A4 = A4*1.5, A5 = A5*2.5, A6 = A6*1, A7 = A7*2.5, A8 = A8*3, A9 = A9*1.5, A10 = A10*3,

A11 = A11*3, A12 = A12*1.5, A13=A13*3, A14=A14*3.5, A15=A15*1.5, A16 = A16*1, A17 = A17*2.5, A18 = A18*2, A19=A19*1

These weights were finalized according to the importance of each attribute, which was calculated based on the stakeholders' answers. Then, we submit all these attributes and calculate the total, i.e., as shown in Formula 2. We analyzed the total and calculated the final performance into four categories (Poor, Sufficient, Good, and Excellent) shown in Rule 1. In Rule 1 we divide the total into ranges, and according to the range, students divide into four categories. After applying all the formulas and rules, the dataset is shown in Figure 2.

$$Total = \sum_n^1 A_n \quad \text{-----------------}\textbf{Formula 2}$$

[(data.total =24) & (data.total<= 27), 'FinalGrade'] = 'Good'

[(data.total = 22) & (data.total <24), 'FinalGrade'] = 'Satisfactory'

[(data.total > 19)& (data.total < 22) , 'FinalGrade'] = 'POOR'

Rule 1

Data Processing

After data preparation and initialization, we evaluate the data for processing. Remove anomalies and fill or remove empty value rows. After this, we correlate each attribute to another attribute using the attribute elevating algorithm. We also calculated the feature correlation and feature importance score of the attributes shown in Fig 3.

DOI: https://doi.org/10.52756/ijerr.2024.v40spl.005

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	FinalGrade
0	1.35	1.8	3.6	0.60	1.00	0.4	1.0	1.2	0.9	1.2	2.7	1.20	1.8	1.4	1.35	0.8	1.00	0.8	0.4	Good
1	1.35	1.8	3.6	0.90	1.00	0.6	1.0	2.1	0.9	1.2	2.7	1.20	1.8	1.4	1.35	0.8	1.50	0.8	0.6	Good
2	1.35	1.8	3.2	0.90	1.00	0.6	1.0	2.1	0.9	1.2	2.7	1.20	1.8	1.4	1.20	0.9	1.50	1.2	0.6	Good
3	1.20	1.6	3.2	1.05	1.00	0.6	1.0	2.7	0.9	1.2	2.7	1.20	1.8	1.4	1.20	0.8	1.75	1.2	0.6	Excellent
4	1.35	1.8	3.2	1.35	1.50	0.6	1.0	2.7	0.9	1.2	2.7	1.20	1.2	2.1	1.20	0.8	1.75	1.2	0.4	Excellent
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
684	1.35	1.8	3.2	0.60	1.00	0.4	1.0	2.7	0.9	1.2	2.4	1.20	1.8	2.1	1.20	0.8	1.00	1.2	0.6	Good
685	0.90	1.2	2.4	0.60	1.75	0.4	1.0	2.7	0.9	1.8	2.4	1.35	1.8	2.1	1.35	0.9	1.50	1.2	0.6	Good
686	0.60	1.4	2.8	0.60	1.75	0.4	1.0	2.7	0.9	1.2	2.4	1.35	1.8	2.1	0.60	0.8	1.75	1.2	0.6	Good
687	1.20	1.6	2.8	0.60	1.00	0.6	1.0	2.7	0.9	1.2	2.4	1.35	1.8	2.1	0.60	0.4	1.00	1.2	0.6	Good
688	1.05	1.2	2.4	0.60	1.00	0.6	1.0	2.1	0.9	1.2	2.4	1.35	1.8	2.1	0.90	0.8	1.00	1.2	0.6	Good

689 rows × 20 columns

Figure 2. Dataset after applying the rules and formula.

Model Implementation:-

The implementation and analysis of the model were carried out using Python tools. The implementation of artificial neural networks (ANNs) aimed to predict students' academic performance through systematic training and data testing. The dataset was divided into distinct training and testing sets. Accuracy was computed for both the training and testing datasets. The pseudocode for the tuning process of the ANN model is given below-

Pseudocode for the Tuning Process:-

```
Procedure NeuralNetworkConfiguration():
// Neural Network Architecture Parameters
 InputLayerNodes <- 482
HiddenLayerNodes <- 241
OutputLayerNodes <- 2
NumberOfHiddenLayers <- 1
// Training Parameters
TotalEpochs <- 100
 OutputTransferFunctions <- [Sigmoid, Relu, Softmax]
 LearningRates <- [0.001, 0.005, 0.0001, 0.0005]
MomentumRange <- [0.1 to 0.9]
OptimizationAlgorithms <- SGD
// Neural Network Configuration Steps
InitializeNeuralNetwork(InputLayerNodes, HiddenLayerNodes, OutputLayerNodes)
SetHiddenLayers(NumberOfHiddenLayers)
ConfigureTrainingAndTesting(TotalEpochs)
ConfigureOutputTransferFunctions(OutputTransferFunctions)
SetLearningRates(LearningRates)
SetMomentumRange(MomentumRange)
ChooseOptimizationAlgorithm(OptimizationAlgorithms)
End Procedure
```

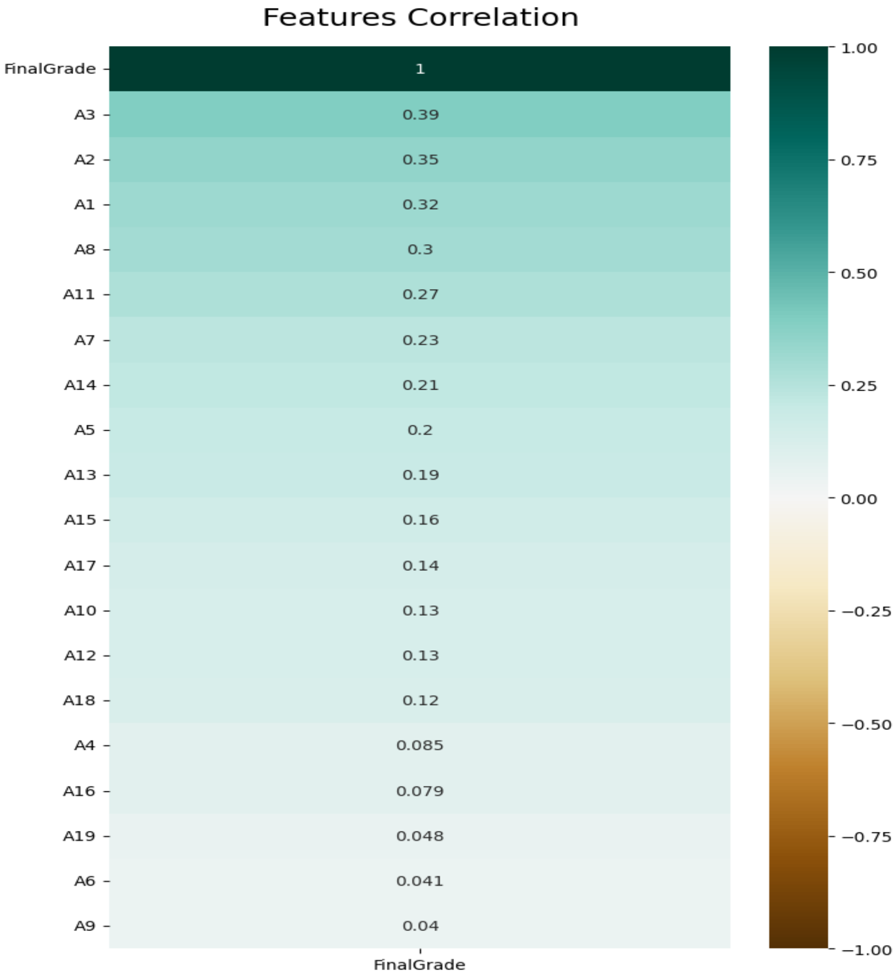

Figure 3. Feature correlation.

Pseudocode of the ANN Model

Class NeuralNetwork:

Constructor(X, y, size_hidden, eta, my, epochs, optimizer, verbose):

Initialize samples, labels, w01, w12, v01, v12, g01, g12, b1, b2, eta, epochs, my, optimizer, and verbose

Function sigmoid(x, deriv):

If deriv is true, return x * (1 - x)

Else, return 1 / (1 + exp(-x))

Function softmax(x, deriv):

If deriv is true, calculate and return the partial derivative

Else, calculate and return the softmax function

Function relu(x, deriv):

If deriv is true, return 1. * (x > 0)

Else, return x * (x > 0)

Function fit():

Initialize accuracy and no_epochs lists

Initialize sample_no to 0

If optimizer is "SGD", initialize gti_01 and gti_12 matrices

For each epoch in range(epochs):

For i in range(len(samples)):

Increment sample_no by 1

Set l0 to the i-th sample

Set y to the i-th label

// Feed Forward Pass:

Calculate l1 and l2 using relu and softmax activation functions

Calculate l2_error and l2_error_total

If l2_error_total is 1.0, return with an "Overflow" message

// Backpropagation:

Calculate l2_delta

Calculate l1_delta

// Update weights using SGD if the optimizer is "SGD"

If optimizer is "SGD":

Update weights using SGD

If epoch is divisible by 1:

If verbose is true, print epoch, error, and accuracy on the test and training sets

Append accuracy to the accuracy list

Append sample_no to the no_epochs list

Function predict(test_samples, test_labels):

Calculate l1 and l2 using relu and softmax activation functions

Convert the predicted labels using argmax and checkEqual1 functions

Return the predicted labels and true labels

// For each eta in etas:

For each eta in etas:

// Create an instance of NeuralNetwork

neural_net = NeuralNetwork(X, y, size_hidden, eta, my, epochs, optimizer, verbose)

// Fit the model to the dataset

neural_net.fit()

// Plot accuracy and error

Plot accuracy and erro

// Predict and print accuracy

predicted_labels, true_labels = neural_net.predict(test_samples, test_labels)

Print accuracy

// Print classification report

Print classification report

Pseudocode for Training the Individual Classifiers:-

Input: Preprocessed data, X_train, y_train

Output: Performance metrics of individual models, metrics

def train_individual_models(X_train, y_train):

Define a list of models with their names

models = [('KNN', KNeighborsClassifier()),

('MLP', MLPClassifier()),

('SVC', SVM Classifier()),

('GNB', GaussianNB()),

('DT', DecisionTreeClassifier()),

('LR', LogisticRegressionClassifier()),

('Random Forest', RandomForestClassifier())]

Create an empty list to store the performance metrics of each model

metrics = []

Loop through each model in the collection of models

for name, model in models:

Train the model using the preprocessed training data

model.fit(X_train, y_train)

Make predictions on the preprocessed test data

y_pred = model.predict(X_test)

Calculate various performance metrics

precision = precision_score(y_test, y_pred)

recall = recall_score(y_test, y_pred)

accuracy = accuracy_score(y_test, y_pred)

f1 = f1_score(y_test, y_pred)

auc = roc_auc_score(y_test, y_pred)

Store the model name and its associated performance metrics in a dictionary

model_metrics = {

'name': name,

'precision': precision,

'recall': recall,

'accuracy': accuracy,

'f1': f1,

'auc': auc

}

Append the model's metrics dictionary to the list of metrics

metrics.append(model_metrics)

Return the list of metrics containing performance information for each model

return metrics

In our research model implementation, the neural network featured one input layer with 482 nodes, one hidden layer with 241 nodes, and one output layer with 2 nodes. The dataset was split into training and testing sets in a 70% to 30% ratio. . Each training and testing session consisted of 100 epochs.

Additionally, we implemented other machine learning models, including Decision Tree, Naive Bayes, Support Vector Machine, K-Nearest Neighbor, Random Forest, and Logistic Regression, with the same 70% to 30% training-to-testing set ratio.

The formulas for accuracy, F-score, recall, precision, and ROC curve are provided in Formula 3, where TP (True Positive), TN (True Negative), FN (False Negative), and FP (False Positive) are defined.

Please note that the specific details of Formula 3 and other technical details would need to be included if they are relevant to the context.

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

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

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

$$F1\_score = \frac{2 * (\text{precision} * \text{recall})}{(\text{precision} + \text{recall})}$$

Formula 3

Model Evaluation

Model evaluation is segmented into two components. The initial segment presents the outcomes derived from assessing the ANN model through various combinations of learning rates, momentum values, transfer functions, and optimization algorithms, as detailed in *section F.1*. The subsequent segment F.2.

involves comparing the results of various algorithms (Decision Tree, Naive Bayes, Logistic Regression, SVM, Random Forest, KNN, and ANN).

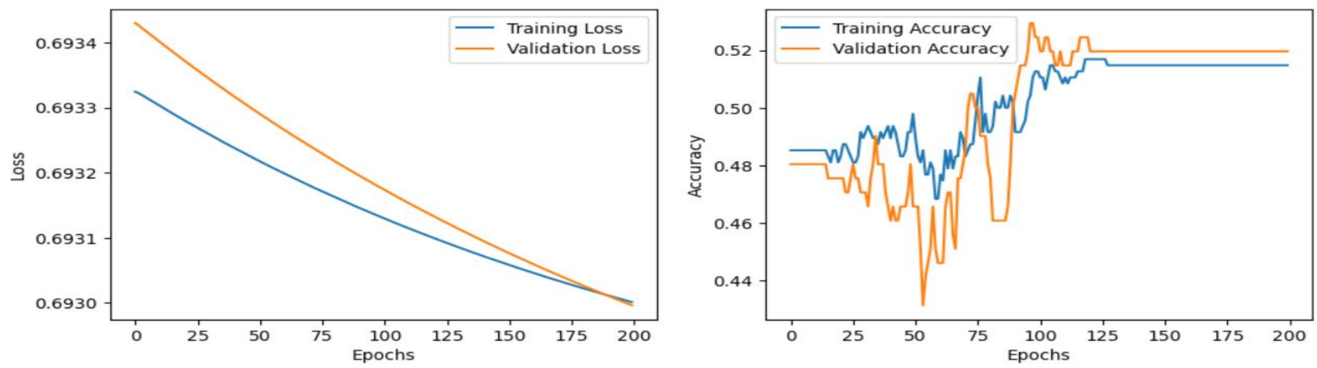

Figure 4. Traning and validation graph of accuracy and loss When (Function=Softmax, lr=0.4)

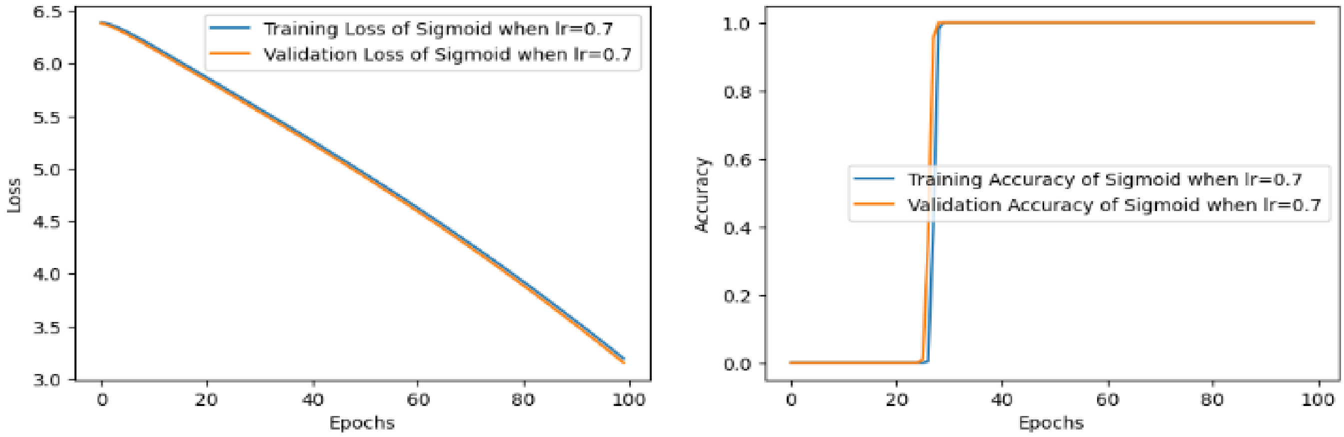

Figure 5. Training and validation graph of accuracy and loss When (Function=Sigmoid, lr=0.7)

Results of the testing and training

During this stage, it was noted that attaining a lower error didn't necessarily lead to the best overall performance, as evidenced by this analysis. After adjusting the learning rate and momentum values, we conducted a comprehensive assessment of 107 outcomes. In the case of the softmax function, utilizing a learning rate of 0.005 and a momentum value of 0.4 resulted in a lower accuracy, specifically 52%. Training and validation accuracy is shown in Fig 4. Conversely, when experimenting with the sigmoid, softmax and relu functions using different learning rates and momentum values, significantly higher accuracy was achieved with a momentum value of 0.7 and a learning rate of 0.005 for the sigmoid function, reaching 98%. Training and validation accuracy is shown in Fig 5. These findings collectively indicate that the sigmoid function excels in terms of achieving a lower error curve, higher accuracy, and quicker training and testing speeds compared to the other functions utilized in fitting the model. All averaged weighted classification metrics are shown in Table 3.

Results Comparison of the Model Evaluation

Figures 6, 7, and 8 present a comprehensive comparative analysis of several algorithms, including Decision Tree, Naive Bayes, Logistic Regression, SVM, Random Forest, KNN, and ANN. Figure 6,7, and 8, focus on micro-averaged, macro-averaged, and weighted-averaged metrics, with a training-to-testing ratio set at 70% to 30%. Among these algorithms, MLP achieved the highest accuracy at 96%, while Decision Tree demonstrated the lowest accuracy at 89%. MLP exhibited the highest recall, precision, and F1-score, establishing it as the most effective predictor in this category. Figure 9 provides insights into the ROC curve for all classifiers. In this representation, LR displayed a superior AUC value of 0.97. In contrast, the Decision Tree exhibited the lowest AUC value at 0.70.

DOI: https://doi.org/10.52756/ijerr.2024.v40spl.005

Table 3. Weighted average classification report(Accuracy, F-score, Recall, Precision)

Hyperparameters of the model			Evaluation Metric			
			Accuracy	Precision	Recall	F1 Score
Output function	Learning rate	Momentum	ACC	PREC	REC	F1
SOFTMAX	0.001	0.1	0.92	0.86	0.93	0.89
		0.2	0.71	0.88	0.71	0.78
		0.3	0.96	0.93	0.96	0.95
		0.4	0.91	0.84	0.92	0.88
		0.5	0.91	0.83	0.91	0.87
		0.6	0.95	0.91	0.96	0.93
		0.7	0.94	0.9	0.95	0.92
		0.8	0.88	0.79	0.89	0.84
		0.9	0.93	0.87	0.93	0.9
	0.005	0.1	0.78	0.83	0.78	0.74
		0.2	0.90	0.95	0.9	0.92
		0.3	0.70	0.91	0.7	0.79
		0.4	0.52	0.75	0.52	0.36
		0.5	0.89	0.8	0.9	0.85
		0.6	0.72	0.92	0.73	0.81
		0.7	0.94	0.89	0.94	0.91
		0.8	0.85	0.73	0.85	0.79
		0.9	0.90	0.82	0.9	0.87
	0.0001	0.1	0.94	0.88	0.94	0.91
		0.2	0.90	0.82	0.91	0.86
		0.3	0.93	0.88	0.94	0.9
		0.4	0.92	0.86	0.93	0.89
		0.5	0.94	0.89	0.94	0.91
		0.6	0.90	0.82	0.91	0.86
		0.7	0.95	0.9	0.95	0.92
		0.8	0.92	0.85	0.92	0.88
		0.9	0.91	0.83	0.91	0.87
	0.0005	0.1	0.90	0.82	0.9	0.86
		0.2	0.92	0.85	0.92	0.88
		0.3	0.95	0.9	0.95	0.93
		0.4	0.90	0.81	0.9	0.86
		0.5	0.92	0.86	0.92	0.89
		0.6	0.93	0.96	0.94	0.95
		0.7	0.91	0.84	0.92	0.88
		0.8	0.94	0.9	0.95	0.92
		0.9	0.93	0.88	0.94	0.91
RELU	0.001	0.1	0.92	0.85	0.92	0.88
		0.2	0.95	0.9	0.95	0.93
		0.3	0.94	0.87	0.93	0.9
		0.4	0.93	0.86	0.93	0.89
		0.5	0.95	0.9	0.95	0.92
		0.6	0.96	0.92	0.96	0.94
		0.7	0.91	0.84	0.91	0.88
		0.8	0.93	0.86	0.93	0.89
		0.9	0.95	0.9	0.95	0.92
	0.005	0.1	0.93	0.87	0.93	0.9
		0.2	0.90	0.82	0.9	0.87
		0.3	0.94	0.89	0.94	0.91
		0.4	0.93	0.84	0.92	0.88
		0.5	0.92	0.86	0.92	0.89
		0.6	0.95	0.9	0.95	0.93
		0.7	0.93	0.87	0.93	0.9
		0.8	0.94	0.9	0.94	0.92
		0.9	0.92	0.88	0.92	0.9

DOI: https://doi.org/10.52756/ijerr.2024.v40spl.005

	0.0001	0.1	0.95	0.9	0.95	0.92
		0.2	0.94	0.89	0.94	0.91
		0.3	0.93	0.87	0.93	0.9
		0.4	0.94	0.89	0.94	0.91
		0.5	0.95	0.9	0.95	0.92
		0.6	0.91	0.84	0.91	0.88
		0.7	0.93	0.88	0.93	0.91
		0.8	0.90	0.82	0.9	0.86
		0.9	0.92	0.85	0.92	0.88
	0.0005	0.1	0.93	0.88	0.93	0.91
		0.2	0.91	0.83	0.91	0.87
		0.3	0.94	0.89	0.94	0.91
		0.4	0.88	0.79	0.88	0.83
		0.5	0.92	0.86	0.92	0.88
		0.6	0.94	0.89	0.94	0.91
		0.7	0.93	0.88	0.93	0.9
		0.8	0.95	0.9	0.95	0.93
		0.9	0.92	0.85	0.92	0.88
Sigmoid	0.001	0.1	0.95	0.9	0.95	0.93
		0.2	0.91	0.84	0.92	0.88
		0.3	0.90	0.82	0.9	0.86
		0.4	0.93	0.88	0.93	0.91
		0.5	0.90	0.82	0.9	0.86
		0.6	0.91	0.87	0.91	0.89
		0.7	0.93	0.88	0.93	0.91
		0.8	0.92	0.85	0.92	0.88
		0.9	0.91	0.84	0.92	0.88
	0.005	0.1	0.94	0.89	0.94	0.9
		0.2	0.93	0.88	0.93	0.91
		0.3	0.94	0.94	0.93	0.94
		0.4	0.95	0.9	0.95	0.93
		0.5	0.91	0.84	0.92	0.88
		0.6	0.89	0.8	0.9	0.85
		0.7	0.98	0.94	0.98	0.95
		0.8	0.93	0.88	0.93	0.91
		0.9	0.94	0.92	0.94	0.93
	0.0001	0.1	0.95	0.9	0.95	0.93
		0.2	0.94	0.92	0.94	0.93
		0.3	0.90	0.82	0.9	0.86
		0.4	0.93	0.88	0.93	0.91
		0.5	0.94	0.92	0.94	0.93
		0.6	0.92	0.85	0.92	0.88
		0.7	0.96	0.92	0.96	0.94
		0.8	0.92	0.85	0.92	0.88
		0.9	0.91	0.84	0.92	0.88
	0.0005	0.1	0.90	0.82	0.91	0.86
		0.2	0.94	0.9	0.95	0.92
		0.3	0.95	0.9	0.95	0.93
		0.4	0.95	0.92	0.96	0.94
		0.5	0.95	0.9	0.95	0.93
		0.6	0.93	0.88	0.93	0.91
		0.7	0.91	0.83	0.91	0.87
		0.8	0.91	0.83	0.91	0.87
		0.9	0.94	0.9	0.95	0.92

DOI: https://doi.org/10.52756/ijerr.2024.v40spl.005

Micro-Averaged Metrics:

Model	Precision (Micro)	Recall (Micro)	F1 Score (Micro)	Accuracy
KNN	0.9166666666666666	0.9166666666666666	0.9166666666666666	0.9166666666666666
MLP	0.9656862745098039	0.9656862745098039	0.9656862745098039	0.9656862745098039
SVC	0.9166666666666666	0.9166666666666666	0.9166666666666666	0.9166666666666666
GNB	0.9166666666666666	0.9166666666666666	0.9166666666666666	0.9166666666666666
DT	0.8970588235294118	0.8970588235294118	0.8970588235294118	0.8970588235294118
LR	0.9509803921568627	0.9509803921568627	0.9509803921568627	0.9509803921568627
Random Forest	0.9411764705882353	0.9411764705882353	0.9411764705882353	0.9411764705882353

Figure 6. Micro averaged metrics of all classifiers

Macro-Averaged Metrics:

Model	Precision (Macro)	Recall (Macro)	F1 Score (Macro)	Accuracy
KNN	0.9583333333333333	0.5	0.4782608695652174	0.9166666666666666
MLP	0.9154135338345865	0.8475935828877006	0.8778129545649012	0.9656862745098039
SVC	0.9583333333333333	0.5	0.4782608695652174	0.9166666666666666
GNB	0.7361111111111112	0.820855614973262	0.7695221638864891	0.9166666666666666
DT	0.6675627240143369	0.6764705882352942	0.6718498659517427	0.8970588235294118
LR	0.9746192893401016	0.7058823529411764	0.7786458333333333	0.9509803921568627
Random Forest	0.9698492462311558	0.6470588235294118	0.71172868582195	0.9411764705882353

Figure 7. Macro averaged metrics of all classifiers

Weighted-Averaged Metrics:

Model	Precision (Weighted)	Recall (Weighted)	F1 Score (Weighted)	Accuracy
KNN	0.923611111111111	0.9166666666666666	0.8768115942028986	0.9166666666666666
MLP	0.9639724310776944	0.9656862745098039	0.9641624597130715	0.9656862745098039
SVC	0.923611111111111	0.9166666666666666	0.8768115942028986	0.9166666666666666
GNB	0.9328703703703703	0.9166666666666666	0.922985755743116	0.9166666666666666
DT	0.8997909199522104	0.8970588235294118	0.8983914209115282	0.8970588235294118
LR	0.9534686971235194	0.9509803921568627	0.9414062499999999	0.9509803921568627
Random Forest	0.9447236180904522	0.9411764705882353	0.9260480452190296	0.9411764705882353

Figure 8. Weighted averaged metrics of all classifiers

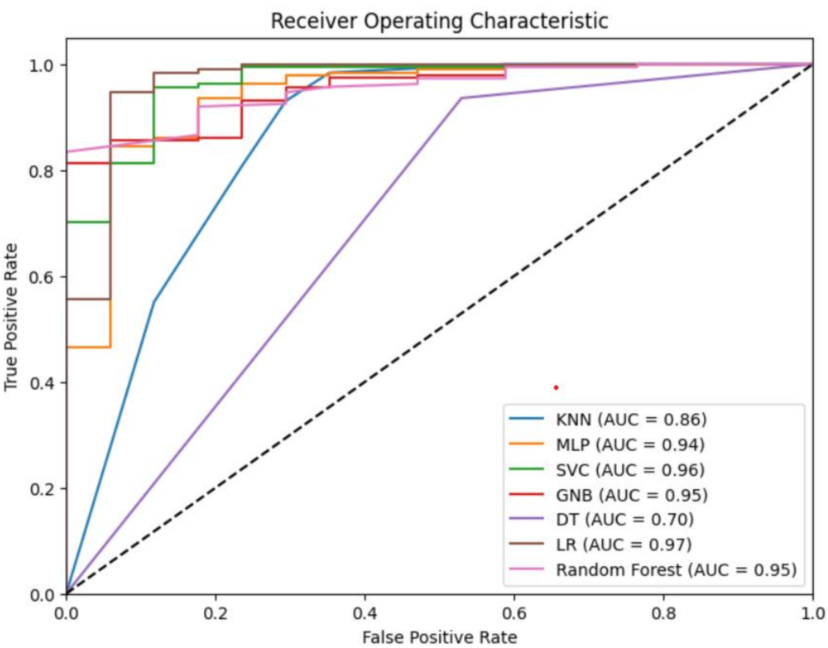

Figure 9. AUC values of all Classifiers

DOI: https://doi.org/10.52756/ijerr.2024.v40spl.005

Conclusion

To understand the factors that influence Artificial Neural Networks (ANN) in the context of smart education, our first objective involved categorizing elements into three distinct groups: background qualities, educational attributes, and psychological traits. Background characteristics, encompassing familial details such as the number of siblings, parents' educational levels, income, employment status, and gender, were identified as influential factors impacting ANN performance, particularly for higher-performance groups benefiting from enhanced educational support. Educational attributes, including academic performance in the 10th, 12th, and B.Tech. First-year examinations, attendance, and assignment performance were found to have the most substantial influence on student outcomes. Concurrently, psychological attributes, considering students' mental and physical health, were recognized as pivotal, acknowledging the correlation between overall success and good health. These factors collectively contributed to the discernible impact on the ANN's performance in the realm of smart education, leading to the categorization of students based on these influential factors.

Moving on to our second objective, which centered on minimizing the ANN error curve, we focused on the careful selection of hyperparameters. Adjusting parameters such as epoch size, training size, testing size, momentum value, and learning rate within the appropriate range was deemed crucial to avoid local minima, reduce training and testing times, and optimize performance. Modifying hyperparameter values was essential for achieving the best performance and the shortest error curve in the smart education context.For our third objective, which involved the performance comparison of classifiers, we divided all classifiers into training and testing sets, allocating 70% and 30% of the data, respectively, as per the specified model evaluation section. Our findings unveiled that the ANN exhibited a remarkable accuracy rate of 97% in predicting student achievement, surpassing the performance of the Decision Tree classifier, which achieved an accuracy of 89%. Notably, the Multilayer Perceptron (MLP) outperformed all other classifiers in terms of recall, precision, and F-score values, reinforcing its efficacy in the smart education domain.

Limitations

This research paper has provided insights into artificial neural networks (ANNs), diverse classifiers, transfer functions, and optimization techniques. Nonetheless, certain limitations are evident, such as the relatively small dataset comprising only 689 students. Additionally, certain factors like students' social interactions, academic engagement, and interpersonal skills have been omitted despite their potential influence on academic performance. These limitations will be thoroughly investigated and addressed in future research endeavors aimed at enhancing the accuracy of student performance prediction.

Future Work

Future research will prioritize including education-related variables and utilizing all relevant factors to enhance prediction accuracy. Our forthcoming models will consider all constraints outlined in the preceding section. We will explore diverse topologies, network configurations, parameters, transfer functions, and optimization techniques to refine our predictive capabilities further.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

Abulhaija, S., Hattab, S., & Etaiwi, W. (2023). Predicting students' performance using machine learning. In *2023 International Conference on Information Technology (ICIT),* Amman, Jordan. *pp. 1-6.* https://doi.org/10.1109/ICIT58056.2023.10225950

Abu-Zohair, L. M. (2019). Prediction of student's performance by modeling small dataset size. *International Journal of Educational Technology in Higher Education, 16*, Article number: 27. https://doi.org/10.1186/s41239-019-0160-3.

Adekitan, A. I., & Salau, O. (2019). The impact of engineering students' performance in the first three years on their graduation result using educational data mining. *Heliyon*, *5*, e01250. https://doi.org/10.1016/j.heliyon.2019.e01250.

Agarwal, M., & Agarwal, B. B. (2021). Towards Prediction of Students Educational Accomplishments Using Data Mining. *Micro-Electronics and Telecommunication Engineering, Lecture Notes in Networks and Systems, 617*. https://doi.org/10.1007/978-981-19-9512-5_2.

Ahmad, N., Hassan, N., Jaafar, H., & Enzai, N. I. M. (2021). Students' performance prediction using artificial neural network. *IOP Conf. Series: Materials Science and Engineering, 1176*, 012020. https://doi.org/10.1088/1757-899X/1176/1/012020.

Ahmad, Z., & Shahzadi, E. (2018). Prediction of students' academic performance using artificial neural network. *Bulletin of Education and Research, 40*(3), 157–164.

Alves, A. F., Gomes, C. M. A., Martins, A., & Almeida, L. d. S. (2017). Cognitive performance and academic achievement: How do family and school converge? *European Journal of Education and Psychology, 10*(2), 49–56. https://doi.org/10.1016/j.ejeps.2017.07.001

Chavez, H., Chavez-Arias, B., Contreras-Rosas, S., Alvarez-Rodríguez, J. M., & Raymundo, C. (2023). Artificial neural network model to predict student performance using nonpersonal information. *Front. Educ., Sec. Assessment, Testing and Applied Measurement, 8*(202). https://doi.org/10.3389/feduc.2023.1106679

Ghosh, S. K., & Janan, F. (2021). Prediction of student's performance using random forest classifier. In *Proceedings of the 11th Annual International Conference on Industrial Engineering and Operations Management Singapore, March 7-11, 2021*.

Haloi, R., Chanda, D., Hazarika, J., & Barman, A. (2023). Statistical feature-based EEG signals classification using ANN and SVM classifiers for Parkinson's disease detection. *Int. J. Exp. Res. Rev., 31*(Spl Volume), 141-149. https://doi.org/10.52756/10.52756/ijerr.2023.v31spl.014

Hamadneh, N. N., Atawneh, S., Khan, W. A., Almejalli, K. A., & Alhomoud, A. (2022). Using artificial intelligence to predict students' academic performance in blended learning. *Sustainability, 14*, 11642. https://doi.org/10.3390/su141811642

Jiao, P., Ouyang, F., Zhang, Q., & Alavi, A. H. (2022). Artificial intelligence-enabled prediction model of student academic performance in online engineering education. *Artificial Intelligence Review, 55*, 6321–6344. https://doi.org/10.1007/s10462-022-10155-y

Jin, H. (2022). Hyperparameter Importance for Machine Learning Algorithms. arXiv:2201.05132. https://doi.org/10.48550/arXiv.2201.05132

Kuppusamy, P., & Joseph, S. K. (2021). A deep learning model to smart education system. In *e-Conference on Artificial Intelligence and Machine Learning*. Mumbai.

Kyndt, E., Musso, M., Cascallar, E., & Dochy, F. (2015). Predicting academic performance: The role of cognition, motivation, and learning approaches. A neural network analysis. In *Methodological Challenges in Research on Student Learning, 1*, 55–76.

Liu, X., Wu, J., & Chen, S. (2023). Efficient hyperparameters optimization through model-based reinforcement learning with experience exploiting and meta-learning. *Data Analytics and Machine Learning, 27*, 8661–8678. https://doi.org/10.1007/s00500-023-08050-x

Mesaric, J. (2016). Decision trees for predicting the academic success of students. *Croatian Operational Research Review, 7*(2). https://doi.org/10.17535/crorr.2016.0025.

Musso, M. F., Kyndt, E., Cascallar, E. C., & Dochy, F. (2013). Predicting general academic performance and identifying the differential contribution of participating variables using artificial neural networks. *Frontline Learning Research, 1*(1), 42–71. https://doi.org/10.14786/flr.v1i1.13

Orji, F. A., & Vassileva, J. (2021). Machine learning approach for predicting students academic performance and study strategies based on their motivation. *Machine Learning (cs.LG)*, arXiv:2210.08186 [cs.LG]. https://doi.org/10.48550/arXiv.2210.0818

Preetha, S., & Anitha, D. (2022). Prediction of academic performance of students using machine learning. *International Journal of Health Science*. https://doi.org/10.53730/ijhs.v6nS1.7868

Rodríguez-Hernandez, C. F., Musso, M., Kyndt, E., & Cascallar, E. (2021). Artificial neural networks in academic performance prediction: Systematic implementation and predictor evaluation. *Computers and Education: Artificial Intelligence, 2*, 100018. https://doi.org/10.1016/j.caeai.2021.100018.

Triventi, M. (2014). Does working during higher education affect students' academic progression? *Economics of Education Review, 41*, 1–13. https://doi.org/10.1016/j.econedurev.2014.03.006

Vairachilai, S., & Vamshidharreddy. (2020). Student's academic performance prediction using machine learning approach. *International Journal of Advanced Science and Technology, 29*(9s), 6731–6737.

Venkata, G., & Damodar, M. (2023). TLBO-Trained ANN-Based Shunt Active Power Filter for Mitigation of Current Harmonics. *International Journal of Experimental Research and Review, 34*, (Special Vol.), 11-21. https://doi.org/10.52756/ijerr.2023.v34spl.002.

Wojciuk, M., Swiderska-Chadaj, Z., Siwek, K., & Gertych, A. (2022). The role of hyperparameter optimization in fine-tuning of CNN models. *ArXiv*. http://dx.doi.org/10.2139/ssrn.4087642

Yadav, N. R., & Deshmukh, S. S. (2022). Prediction of student performance using machine learning techniques: A review. *ICAMIDA 2022, ACSR105*, pp. 735–741. https://doi.org/10.2991/978-94-6463-136-4_63

Zaffar, M., Hashmani, M. A., Savita, K. S., Rizvi, S. S. H., & Rehman, M. (2020). Role of FCBF feature selection in educational data mining. *Mehran University Research Journal of Engineering and Technology*, *39*(4), 772-778. https://doi.org/10.22581/muet1982.2004.09.

Zhang, Y., Yun, Y., An, R., Cui, J., Dai, H., & Shang, X. (2021). Educational data mining techniques for student performance prediction: Method review and comparison analysis. *Front. Psychol., Sec. Educational Psychology, 12*. https://doi.org/10.3389/fpsyg.2021.698490.

Zhu, Z.T., Yu, M.H., & Riezebos, P. (2016). A research framework of smart education. *Smart Learning Environments, 3*(4). https://doi.org/10.1186/s40561-016-0026-2

How to cite this Article:

Mini Agarwal and Bharat Bhushan Agarwal (2024). Predicting Student Academic Performance Using Neural Networks: Analyzing the Impact of Transfer Functions, Momentum, and Learning Rate. *International Journal of Experimental Research and Review*, *40*(spl.), 56-72.

DOI: https://doi.org/10.52756/ijerr.2024.v40spl.005