Babylonian Journal of Artificial Intelligence
Vol. 2024, **pp**. 83–101
DOI: https://doi.org/10.58496/BJAI/2024/011 ISSN: 2958-6453
https://mesopotamian.press/journals/index.php/BJAI

Research Article

Predictive analytics model for students' grade prediction using machine learning

Muhammed Fareed Flayyih[1*], Hassan TOUT [2]

[1] *Computer Science Dep., Arts, Sciences and Technology University in Lebanon, Beirut, Leonean.*
[2] *Computer Science Dep., Lebanese University in Lebanon, Beirut, Leonean.*

ARTICLE INFO

Article History
Received 19 May 2024
Accepted 21 Jul 2024
Published 05 Aug 2024

Keywords

Predictive analytics model
Machine Learning Model
students' grade
AI

ABSTRACT

This project aims to develop a predictive analytics model using machine learning to forecast student grades, helping educational institutions identify struggling students early for targeted support. By leveraging machine learning, the model can analyze large datasets to detect complex patterns, enhancing prediction accuracy in education. The project employs neural networks due to their ability to capture non-linear relationships in data. Two models were created: one trained with data from low-rated schools and tested on both low- and high-rated schools, achieving 85.7% and 83.3% accuracy, respectively. The second model, trained with high-rated school data, yielded 88.9% accuracy for high-rated schools but only 35.7% for low-rated ones. Results indicate that separate models for different school levels are more effective due to discrepancies in grade reporting accuracy among Iraqi schools.

1. INTRODUCTION

Prediction involves evaluating our understanding of the current state based on limited observations, acknowledging that this knowledge is often uncertain, incomplete, and derived from approximate models. Despite these challenges, prediction plays a crucial role in managing future uncertainties[1].

A "Predictive Analytics Model for Students' Grade Prediction using Machine Learning" leverages machine learning and data analytics to forecast students' academic performance using historical data and various factors from their educational experiences. The main objective is to provide insights into potential academic outcomes, like final grades, by analyzing past performance and relevant attributes[1].

These models offer practical benefits, allowing institutions, educators, and administrators to identify students who may be struggling and provide early, targeted interventions to improve their learning experiences. High-performing students can also be recognized for advanced opportunities. Predictive models assist in resource allocation, curriculum development, and personalized learning strategies, making them indispensable in education.

Predictive analytics enhances personalized learning by evaluating data on student performance, attendance, and demographics, creating individualized profiles that enable educators to adjust teaching strategies and interventions. They facilitate early intervention by identifying students at risk of underperformance, allowing timely support through tutoring or counseling. These models optimize resource allocation by predicting program and service demand, ensuring efficient use of resources. Additionally, they help prevent dropouts by identifying at-risk students early, providing targeted support to improve retention rates. Predictive models also assess the effectiveness of educational programs, empowering institutions to make informed, data-driven improvements.

2. MACHINE LEARNING

*Corresponding author. Email: mfa058@live.aul.edu.lb

The goal of artificial intelligence (AI), especially machine learning, is to create models and algorithms that enable computers to learn from experience and forecast future events using the information they have collected. These algorithms and models are essentially data-driven in their training. There are several subcategories within machine learning, each with distinct features and specific uses [2]. The following section delineates the primary categories of machine learning (Fig 1).

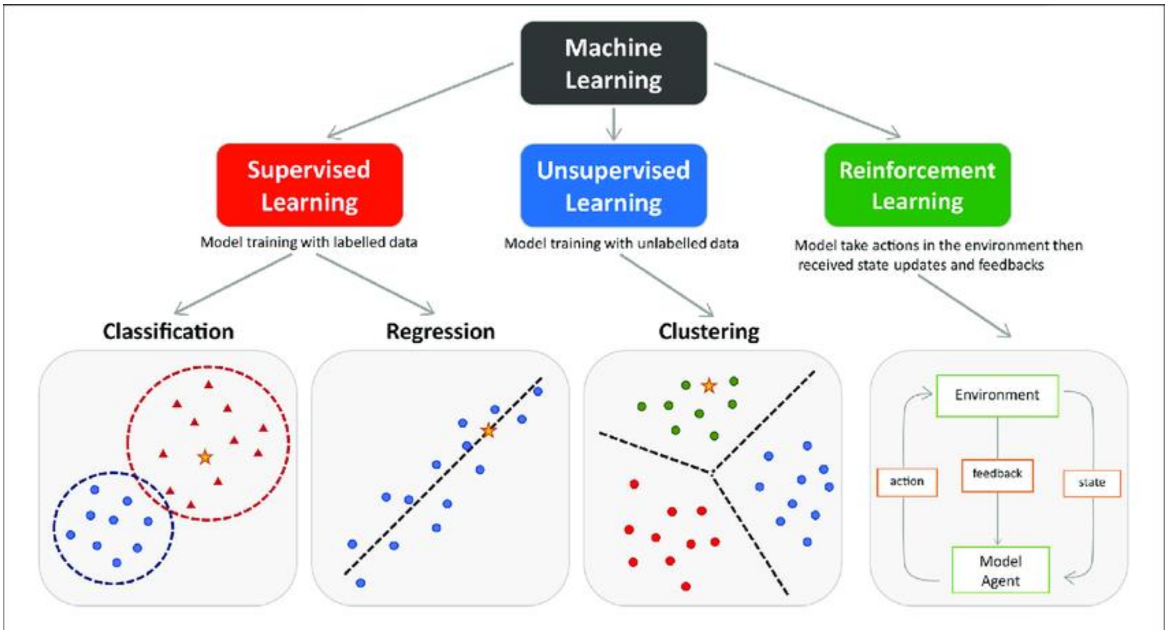

Fig .1. Inflation neural network diagram.

2.1 Supervised Learning

In machine learning, supervised learning is a special technique in which algorithms learn from a labeled dataset. Every input data point in this paradigm has an equivalent output label or target assigned to it. Unsupervised learning, on the other hand, involves algorithms that learn from unlabeled datasets, identifying patterns and relationships to build a mapping or function for using knowledge from labeled training data to make predictions or classifications on new, unseen data. [3].
Supervised learning, one of the predominant branches in machine learning, primarily aims to attain a predictive or classificatory capability. This involves dividing the dataset into two main sections as a common practice:

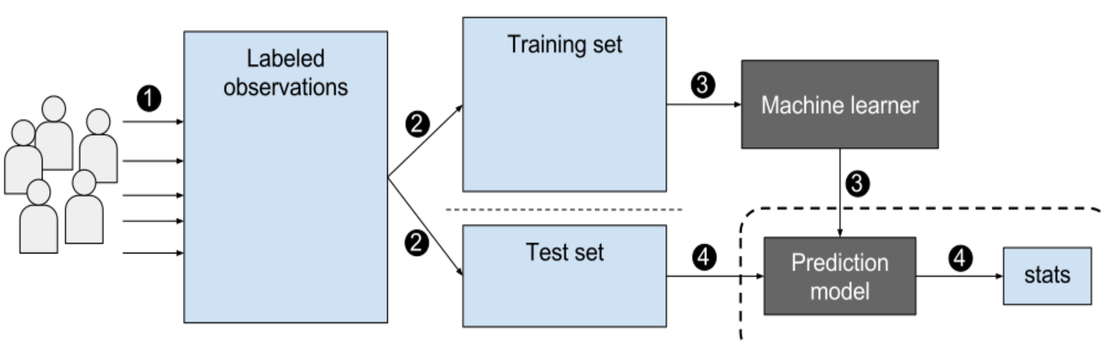

Fig 2. supervised learning.

1. **Input Features (X):**

The attributes or features of the data employed for making forecasts are represented as X, constituting the input features. Each data point in the dataset is characterized by a set of input features. For example, when predicting housing prices, the input features may include square footage, number of bedrooms, location, and other relevant factors.

2. **Output Labels or Targets (y):**

The resulting predicted values, represented as "y," signify the expected outcomes based on a set of input features. In the context of supervised learning, the goal is to guarantee that a model can consistently produce predictions or classifications for these labels. For instance, the specific outputs in email categorization could be labeled as either "spam" or "not spam." In a broader sense, "X" denotes the features of the inputs, and "y" represents targets. The model undergoes training by understanding the associations that associate the input X to the output y, allowing it to make predictions on future, new data.

Within the domain of supervised learning, there exist diverse subtypes, each designed to address specific types of prediction or classification tasks. The primary subtypes encompass classification and regression:

- **Classification:** A notable subtype within supervised learning, centers on the accurate prediction of distinct class labels or categories. For instance, it can be applied to tasks such as identifying the sentiment of customer reviews (positive, negative, neutral) or classifying different types of vehicles based on images, showcasing its versatility across various practical applications.
- **Regression:** In regression tasks, the objective is to estimate continuous values, such as numerical ratings or continuous measurements. For instance, this could involve predicting an individual's age based on demographic information or estimating the price of a home using features like square footage and location. These instances fall under the category of regression tasks.

2.2 Unsupervised learning

Another facet of machine learning is unsupervised learning, where algorithms can discern patterns and structure information that hasn't been pre-labeled. Unlike supervised learning, unsupervised learning operates without target labels for the training set. In this domain, algorithms seek hidden structures, sets of data, or patterns within datasets [4]. Unsupervised learning can take on various forms, including:

- **Clustering:** Clustering, a frequently employed technique in unsupervised learning, involves grouping points into clusters based on their similarity. Clustering algorithms like K-Means, hierarchical clustering, and DBSCAN are used in many real-world applications for image segmentation, anomaly detection, and customer segmentation. K-Means clustering is one particular example of this methodology.
- **Dimensionality Reduction:** Dimensionality reduction methods aim to decrease the number of attributes or dimensions in a dataset without sacrificing crucial information. Widely used techniques for dimensionality reduction include principal component analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE). These methods provide tools to visualize the complexity of data and identify pertinent characteristics, simplifying the handling of the data.

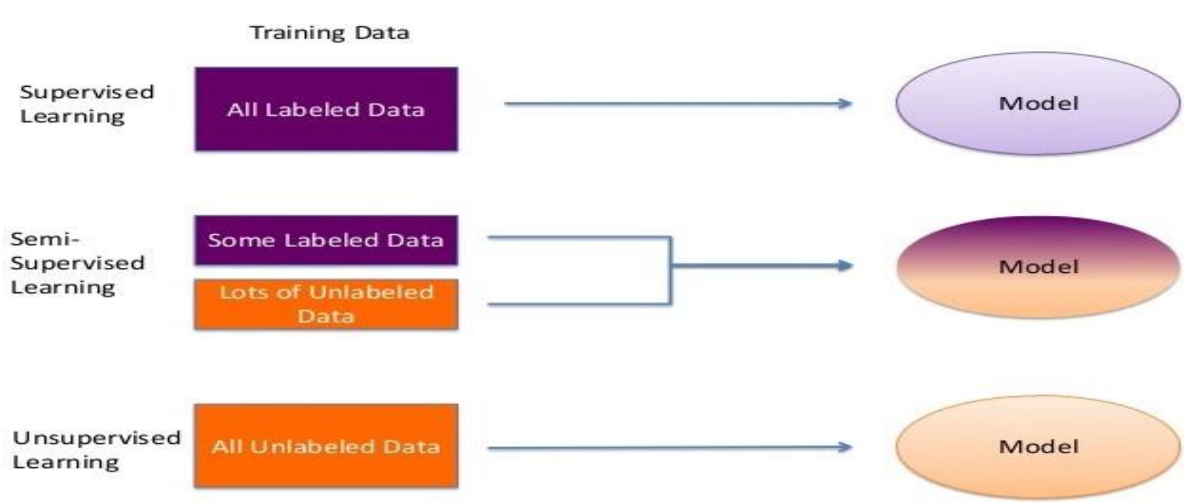

Fig .3. Unsupervised Learning

2.3 Reinforcement Learning

Reinforcement Learning (RL) constitutes a dimension within Machine Learning (ML) where an agent learns to navigate through an environment by taking successive actions. Unlike supervised learning, which relies on labeled data for model training, and unsupervised learning, which uncovers patterns in unlabeled data, reinforcement learning distinguishes itself by concentrating on developing an optimal decision-making strategy through trial and error [5]. This approach finds applications in diverse fields such as robotics, recommendation systems, self-driving cars, gaming (as demonstrated by AlphaGo), and natural language processing. Prominent reinforcement learning algorithms comprise DQN, Policy Gradient Methods, Q-Learning, and more sophisticated techniques like PPO and TRPO.

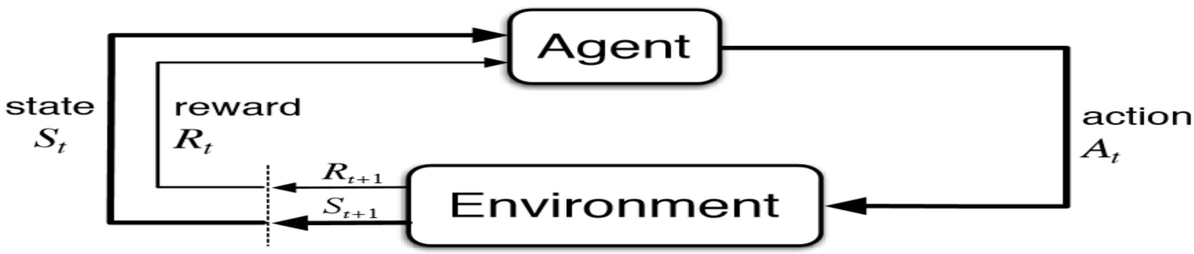

Fig .4. Reinforcement Learning

Reinforcement learning grapples with a significant dilemma, navigating the delicate balance between exploration (attempting to discover more productive activities) and exploitation (selecting effective procedures). Investigating various approaches, such as Thompson sampling, Monte Carlo techniques, and Epsilon-greedy policies, is necessary to overcome this difficulty. The goal of current reinforcement learning research is to improve theoretical understanding as well as practical applications in real-world scenarios.

2.4 Types of layers in forecasting applications

Commonly utilized in approximation, classification, and forecasting applications are layers such as the perceptron, probabilistic layer, long-short-term memory (LSTM), scaling layer, unscaling layer, and bounding layer [6].

1. **Perceptron layer**

In the neural network, dense or "perceptron" layers act as the cognitive center, driving the learning process. The provided image portrays a perceptron layer receiving numerical inputs, undergoing operations with biases and weights, ultimately producing the final output.

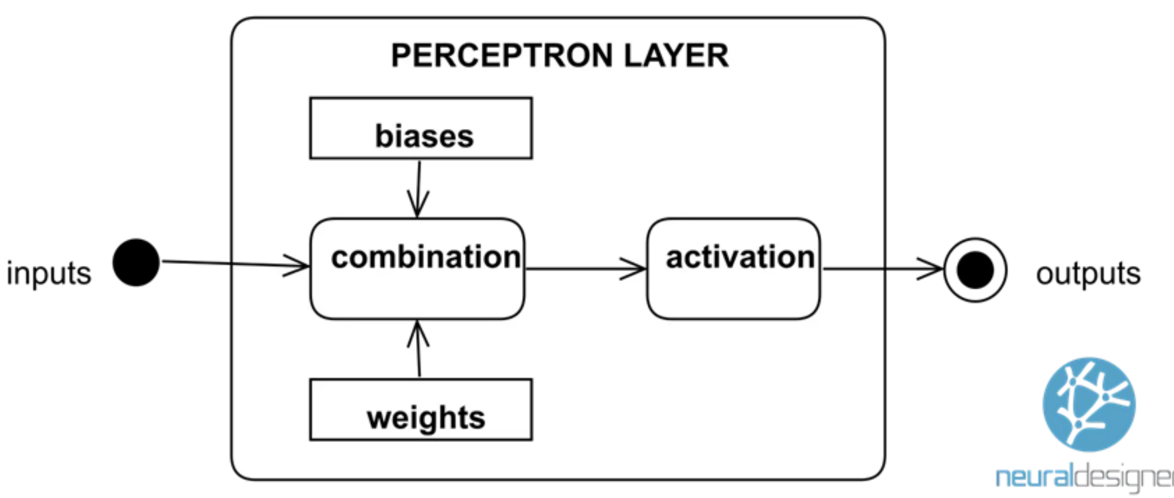

Fig .5. Perceptron layers

A layer's behavior is shaped by the activation function. The following activation functions are commonly used:

- hyperbolic tangent activation function;
- linear activation function.
- The function of logistic activation.
- The Rectified Linear Activation Function (ReLU).

Linear activation function

A perceptron that uses a linear activation function multiplies the total of the input weights by the bias associated with those weights to determine the output. Said another way, the weight and bias parameters selected determine how the output behaves, which is a linear function of the input. Nevertheless, in this case, the linear activation function is devoid of non-linearities, which restricts the neural network's capacity to understand intricate patterns.

This feature makes linear activation functions more suitable for simpler tasks where there is no need for non-linear transformations.

When a perceptron uses a linear activation function, it operates by multiplying the sum of the input weights with the associated bias. Simply put, the output is determined by the chosen weight and bias parameters, making it a linear function of the input. However, because this activation function lacks non-linear components, it limits the neural network's ability to capture complex patterns. As a result, linear activation functions are more appropriate for simpler tasks that do not require non-linear transformations..

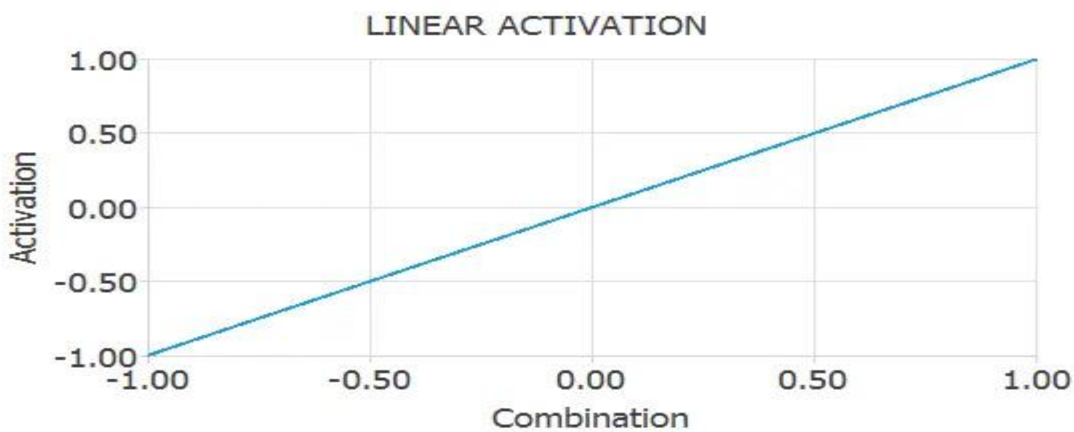

Fig .6. Linear activation function

Hyperbolic tangent activation function

The hyperbolic tangent, or Tanh, is a widely used activation function in neural network architectures. Tanh maps input values to a range between -1 and +1, functioning similarly to the sigmoid function. This property allows the model to recognize both positive and negative relationships in the data. Tanh is particularly beneficial in helping the neural network understand and capture complex patterns, especially when the data exhibits distinct positive or negative trends.

$$activation = tanh(combination)$$

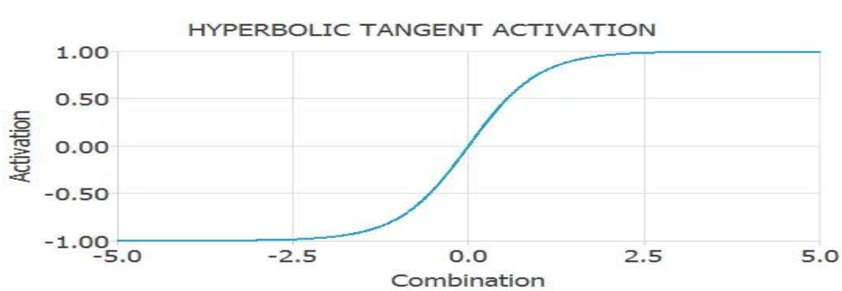

Fig .7. Hyperbolic tangent activation function

Logistic activation function

Similar to logistic activation and the hyperbolic tangent, the logistic activation function ensures that the output remains within a defined range, typically between 0 and 1. This feature is particularly useful in binary classification tasks, where outputs need to fall within a true-or-false spectrum of 0 to 1. The function is well-suited for classification because it interprets outputs as probability values within this range, enabling the neural network to make decisions based on the likelihood of certain outcomes.

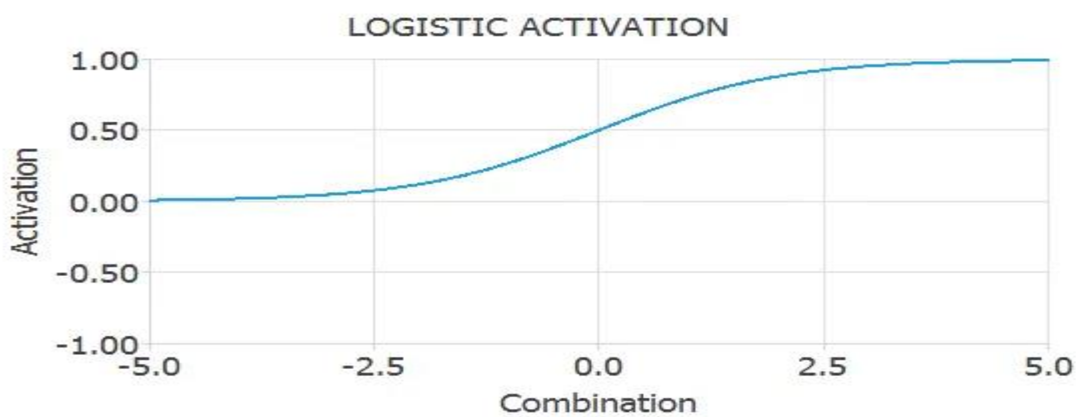

Fig .8. Logistic activation function

Rectified linear (ReLU) activation function

The Rectified Linear Activation Function (ReLU) is one of the most popular activation functions currently used in neural networks. It works by outputting 0 when the input is negative, while maintaining the input value if it is zero or positive. Mathematically, ReLU dictates that if the input 'x' is negative, the output is zero; otherwise, it remains 'x'. The simplicity and efficiency of ReLU have made it widely embraced. Additionally, unlike the linear activation function, which restricts the neural network to learning simple patterns and relationships in the data, ReLU allows for more complex learning.

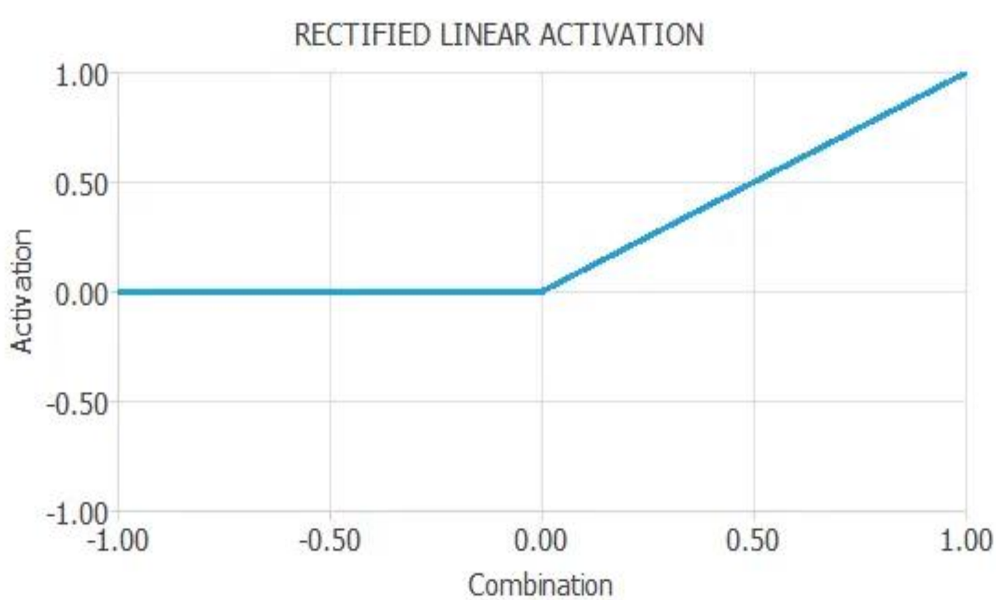

Fig .9. Rectified Linear Activation Function

2. Probabilistic layer

In these instances, the outputs are typically construed as probabilities signifying class affiliation. Therefore, the probabilistic layer produces outcomes between 0 and 1, guaranteeing that the combined class probabilities sum up to one. This attribute enables the outcomes to be regarded as a probabilistic distribution across classes.
A visual representation of a probabilistic layer resembles a perceptron layer, as depicted below. Notably, the activation functions must exhibit a probabilistic nature. This limitation is consistent with the need to produce comprehensible output probabilities, which qualifies the layer for probabilistic modeling—particularly in classification tasks.

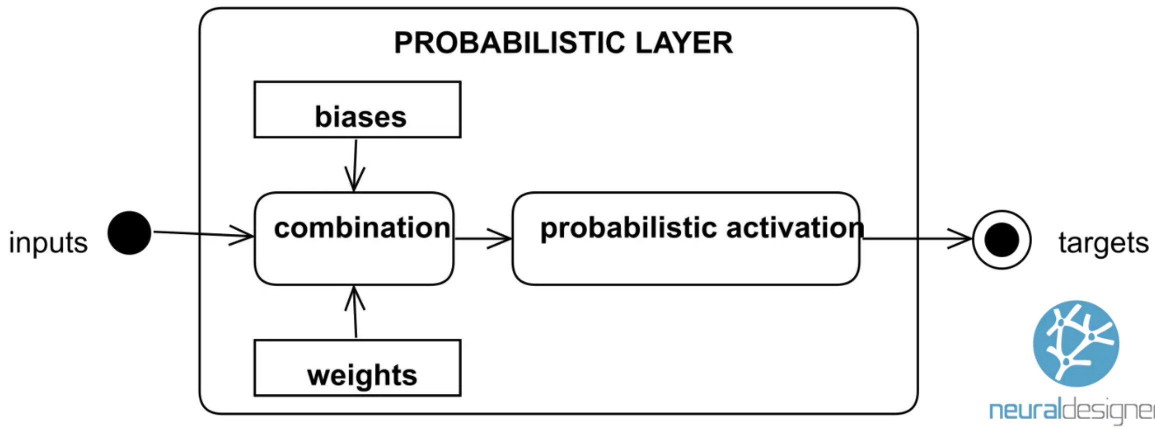

Fig .10. Probabilistic activation

The logistic probabilistic activation and softmax are two examples of popular probabilistic activations.

Logistic probabilistic activation
Sorting observations into one of two classes—positive and negative, for example, or yes and no—is known as binary classification. Because the logistic activation function is sigmoidal and thus limited to a range of values [0, 1], it is therefore a good fit for this task.
This characteristic proves useful, allowing the model to generate predictions that align with probabilities. In the case of binary classification, these probabilities can be thresholded to make final decisions. For instance, an observation might be assigned to a specific class based on whether its chance is greater compared to another category.
SoftMax probabilistic activation
The prevalent utilization of the probabilistic SoftMax activation is observed in classification challenges. Smoothly functioning as a Probability - based Function, it ensures that every output is limited to 0 and 1 (inclusive), adding up to 1. When dealing with classification problems, SoftMax activation is essential, particularly when an input can belong to multiple classes. SoftMax is particularly notable for converting logits or raw scores into a probability distribution across multiple categories. The output of each SoftMax layer represents the probability that the input belongs to a specific class. This normalization is especially useful for accurately estimating the probability distribution of all classes. Therefore, SoftMax is essential in scenarios involving multi-class problems with mutually exclusive classes, as it helps predict the likelihood of each possible class.

3. Long-short-term memory (LSTM) layer

LSTM layers, a specialized type of recurrent layer, are extensively used in forecasting tasks and play a key role in sequence-to-sequence modeling, as shown in the figure. In forecasting, where the goal is to predict future values based on past data, LSTM layers excel at capturing long-term dependencies. These layers consist of essential components, including the forget gate, input gate, cell state, and output gate, which collectively enhance the modeling of sequences to improve prediction accuracy.
When working with sequential data like time-series, numeric input sequences are typically fed into the LSTM layer to process and learn from the data patterns.
Gate Mechanisms: Different gates within the layer process the information as follows:

- Forget Gate: Decides what information from the previous state should be excluded.
- Input Gate: It controls the flow of information into the current time.
- State Gate: Updates the internal state by using the input and forget gate outputs.

- Output Gate: Produces a final output based on the handled data.

Memory Storage: To help with the retention of data from long sequences, the LSTM layer keeps track of both a cell state and a hidden status.

Output Generation: In the end, this layer generates final outputs that can be used as input features or predictions for further predictive tasks.

The distinctive architecture of LSTM enables it to adeptly manage and preserve long-term information, rendering it particularly well-suited for applications involving forecasting and scenarios where accuracy in predictions concerning temporal relationships is paramount.

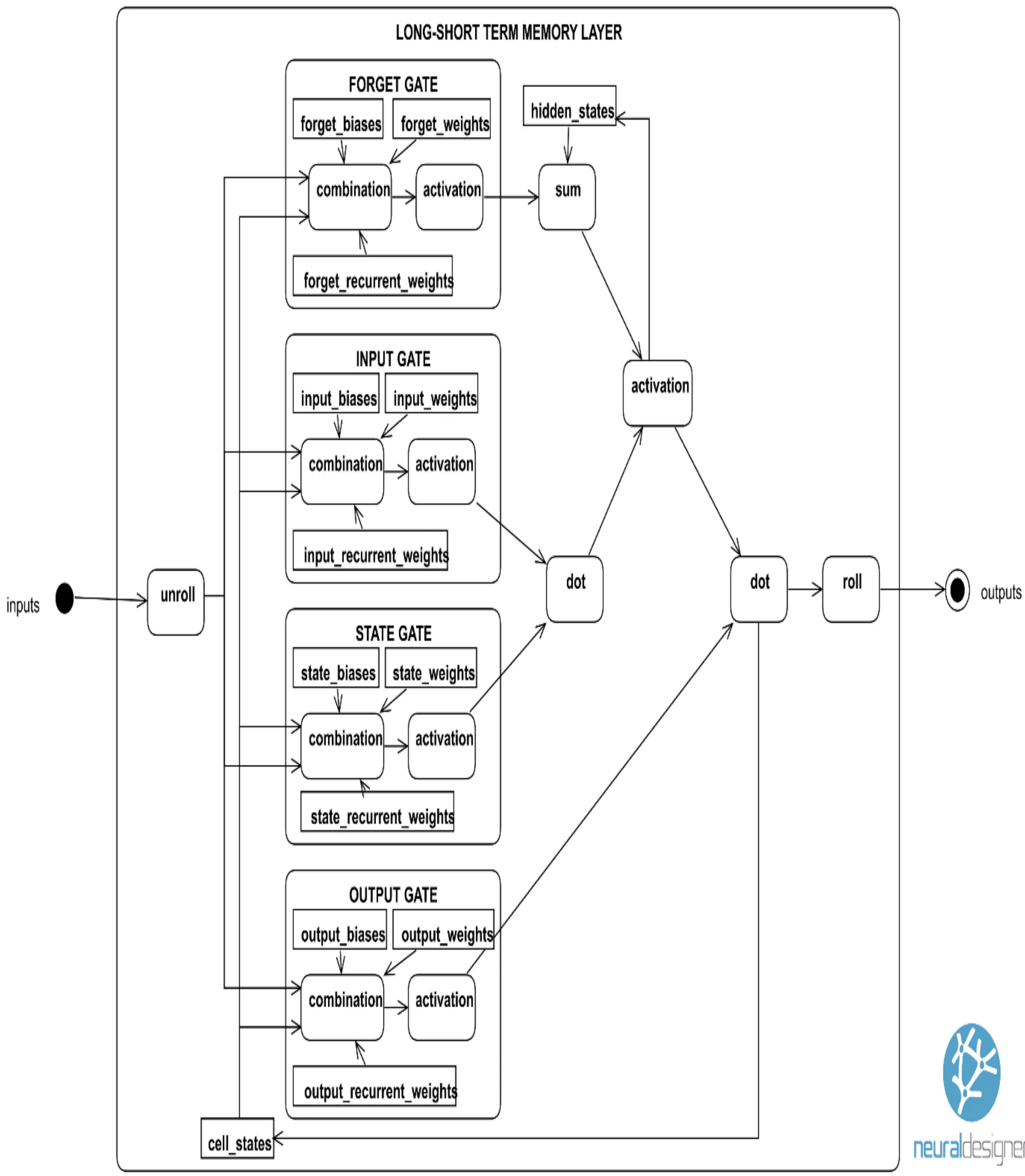

Fig .11. long-short-term memory (LSTM) layers

As observed, long-short-term memory (LSTM) layers exhibit complexity with a multitude of parameters. This intricate structure renders them well-suited to learning dependencies from time-series data because of their complex structure.

4. Scaling layer

In practical applications, the scaling layer adjusts the input range of neural networks to ensure all inputs are normalized within specified boundaries. Normalizing input values to a standard scale is often advantageous, and this task is performed by the scaling layer, which utilizes basic statistical information derived from the inputs, such as mean, standard deviation, minimum, and maximum values.

Several scaling methods are commonly used:

1. Minimum and Maximum Scaling Method: Standardizes the dataset between -1 and 1, ideal for variables with an even distribution.
2. Mean and Standard Deviation Scaling Method: Frequently applied to normally distributed variables, this approach scales inputs to have a zero mean and unit variance.
3. Standard Deviation Scaling Method: Often used for variables that are non-negative, this technique generates inputs with a standard deviation of one, creating a half-normal distribution with a zero mean. Linear scaling methods can achieve similar results.

It's essential to align the scaling of input and output elements with that of the neural network. Neural Designer manages this integration automatically, without the need for manual adjustments.

By ensuring normalized inputs, the scaling layer enhances the overall performance and coherence of neural networks, supporting smooth learning and improved generalization.

5. Unscaling layer

The unscaling layer is vital for converting the scaled outputs of a neural network back into their original units, allowing the model's predictions to be accurately interpreted in the context of observed data. Like the scaling layer, the unscaling layer uses key statistics derived from the outputs, including mean, standard deviation, minimum, and maximum values.

Several unscaling methods are commonly employed:

1. Minimum and Maximum Unscaling Method: Restores variables to their original range after being scaled between -1 and 1.
2. Mean and Standard Deviation Unscaling Method: Reverts variables to their initial units after scaling them to a mean of 0 and a standard deviation of 1.
3. Standard Deviation Unscaling Method: Returns variables to their original range after being scaled to a standard deviation of 1.
4. Logarithmic Unscaling Method: Reverses the logarithmic transformation to restore variables to their original scale.

Unscaling models can be linear or nonlinear, depending on the transformation used, and the unscaling method chosen should correspond to the initial scaling technique.

It is crucial to synchronize the unscaling of neural network outputs with the scaling of the target data, which Neural Designer handles automatically without requiring manual adjustments. The unscaling layer ensures that neural network predictions are consistent with the original data, facilitating a clearer understanding of the modeled information.

6. Bounding layer

In many scenarios, it is crucial to restrict the output of a neural network within specific limits. For instance, when evaluating product quality on a rating scale, the output may need to be confined to a set range, such as 1-5 stars. A "Bounding Layer" is used to manage the model's predictions, ensuring the output stays within the desired range.

The main function of the bounding layer is to constrain the neural network's output to fall within defined boundaries. This is particularly important when outputs must remain within specified limits.

Application in Quality Assessment: Consider a star rating system ranging from 1 to 5 used for evaluating product quality. The bounding layer ensures that the neural network's predictions are confined to this range.

Flexible Boundary Definition: The bounding layer allows boundaries to be adjusted according to the needs of the specific problem, whether involving a discrete set, numerical range, or other predefined limits.

Ensuring Output Consistency: By integrating a bounding layer, the neural network can limit its outputs according to expected constraints, enhancing the interpretability of results and ensuring the practical relevance of predictions.

Diverse Applications: While the quality rating example demonstrates one application, the bounding layer can be applied wherever outputs must adhere to predefined limits, such as predicting financial metrics, assessing risks, or any situation requiring outputs within set boundaries.
The bounding layer is a key component in neural networks, regulating output parameters. Its flexibility allows it to be used in various contexts, ensuring neural network predictions align with the constraints and requirements of the task at hand.

2.3 Unveiling Exemplary Projects

In the field of education, forecasting student performance has become essential for improving learning outcomes and teaching strategies. This detailed analysis explores the complexities of predicting student performance using machine learning methods [7].
There are many different types of data involved, but since we want to make predictions about student performance it would be interesting to have:

- A measurement of the students' commitment to the course throughout the period
- A measurement of their performance during the period
- Their final grades, as they are a huge part of the final grade composition.

2.3.1 OU Analyse

Ou Analyse is a system powered by machine learning methods for early identification of students at risk of failing. The data set used in this system [8] is contained under the following database:

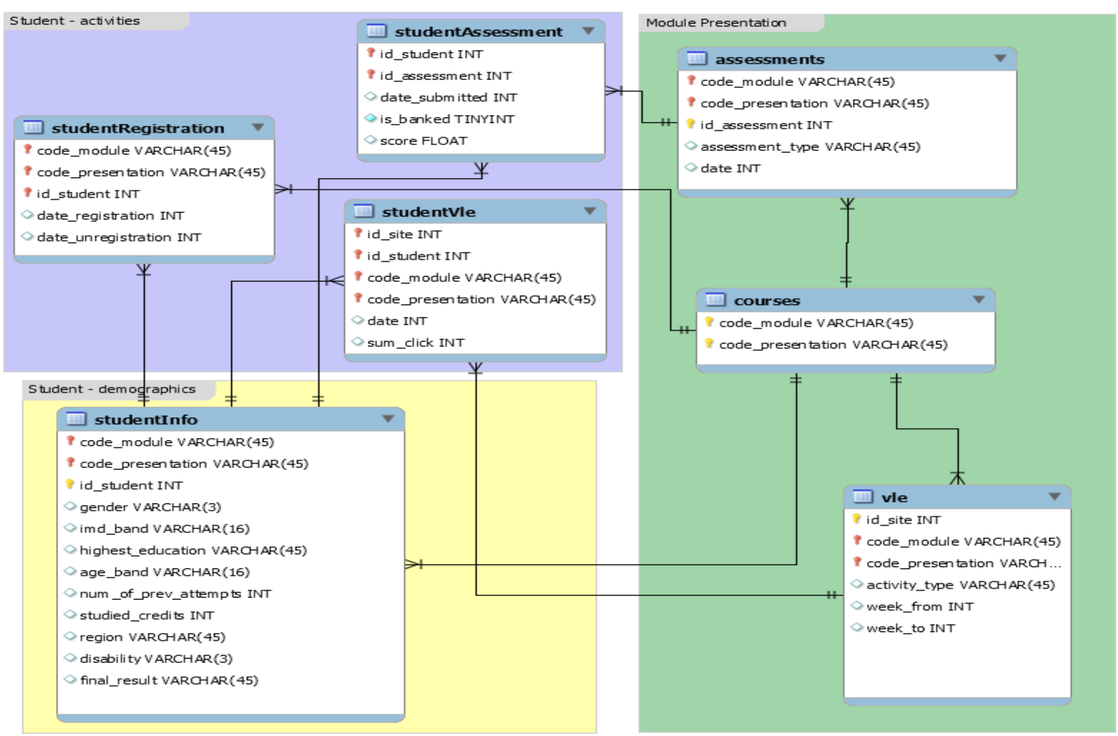

Fig .12. Open University Learning Analytics dataset

All information is downloadable and comprise into 7 csv files:

courses.csv

Contained within this document is a comprehensive list detailing all accessible modules along with their respective presentations, with the columns consisting of:

- code_module – code name of the module, which serves as the identifier.
- code_presentation – code name of the presentation. It consists of the year and “B” for the presentation starting in February and “J” for the presentation starting in October.
- length - length of the module-presentation in days.

The arrangement of B and J presentations may vary, making it advisable to analyze B and J presentations independently. However, for certain presentations, there is no corresponding previous B/J presentation, necessitating the use of the J presentation to inform the B presentation, or vice versa. This situation is observed in the dataset for CCC, EEE, and GGG modules (three of the seven modules or courses selected in this study and named AAA, BBB, CCC, DDD, EEE, FFF and GGG).

assessments.csv

Contained within this file is information pertaining to assessments in module-presentations, where each presentation is accompanied by a set of assessments followed by a final exam. The CSV file encompasses columns that provide details on this information:

- code_module – the code that identifies a module, to which the assessment belongs.
- code_presentation – a code that identifies code the presentation, to which the assessment belongs.
- id_assessment – identification number of the assessment.
- assessment_type – type of assessment. This is one of the three types of assessments: Tutor Marked Assessment (TMA), Computer Marked Assessment (CMA) and Final Exam (Exam).
- date – this is the date of submission of the assessment calculated as the number of days since the start of the module-presentation. The starting date of the presentation has number 0 (zero).
- weight - the assessment weight in %. The sum of all assessments is 100%.

The final exam date is missing, it is by default the end of the last presentation week.

vle.csv

The vle.csv file covers data regarding the resources accessible in the VLE (Virtual Learning Environment), which typically include HTML pages, PDF files, and similar content. Students can access these resources online, and their interactions with the resources are systematically recorded. The columns featured in the vle.csv file include:

- id_site – a number that identifies the resource.
- code_module – the code for a module.
- code_presentation - code of a presentation.
- activity_type – the role associated with the module resource.
- week_from – the week from which the resource is premeditated to be used.
- week_to – week until which the material is projected to be used.

studentInfo.csv

Within this document are details relating to the demographic information of the students and their corresponding academic results. studentInfo file contains the following columns:

- code_module – a code for a module on which the student is displayed.
- code_presentation - the code that identifies the presentation during which the student is enrolled on the module.
- id_student – the student identification.
- gender – the student's gender.
- region – identifies the region where the student live.
- highest_education – highest student education level on entry to the module presentation.
- imd_band – specifies the Index of Multiple Depravity band of the place where the student lived during the module-presentation.
- age_band – group of the student's age.
- num_of_prev_attempts – the number times the student has attempted this module.
- studied_credits – the sum of credits for all the modules the student is currently studying.
- disability – indicates whether the student has declared a incapacity.
- final_result – student's final result in the module.

studentRegistration.csv

This document provides details regarding the date when students registered for the module presentation, with additional recording of the dates when students unregistered for a module.

File contains five columns:

- code_module – a code for a module.
- code_presentation - the code of the presentation.
- id_student – a student's id.

- date_registration – the number of days of student's registration on the module relative to the start of the presentation (e.g. the negative value -30 means that the student registered to module presentation 30 days before it started).
- date_unregistration – date of student unregistration from the module presentation. Also, it is the number of days measured relative to the start of the module-presentation.

studentAssessment.csv

This document encompasses the outcomes of assessments submitted by students, noting that students who did not submit an assessment have no recorded result for that specific assessment. The columns featured in this file include:

- id_assessment – the id of the assessment.
- id_student – a unique ID number for the student.
- date_submitted – the date of submission, calculated as the number of days since the start of the presentation.
- is_banked – a status label indicating that the assessment result has been transmitted from a previous presentation.
- score – the student's score in the assessment. The score's possible value is between 0 and 100. The student is considered as failed if the mark is lower than 40.

studentVle.csv

The studentVle.csv file contains information about students' interactions with the resources in the VLE. It contains the following data:

- code_module – the code for a module.
- code_presentation - the code of the presentation.
- id_student – a unique ID number for the student.
- id_site - the VLE material identification number.
- date – the date of the interaction of the student with the resource. It is measured as the number of days since the start of the module-presentation.
- sum_click – the number of times a student interacts with the material in that day.

For the machine learning algorithm, they used four algorithms to compare between.

Logistic Regression: The "Logistic Regression" algorithm was used to predict student performance by considering various features. The dataset was split into three subsets, and separate Logistic Regression models (lr1, lr2, and lr3) were trained on each subset with different sets of input features. The main goal was to evaluate how well these models could classify student performance into three categories: Distinction, Fail, and Pass.

Linear Discriminant Analysis: The "Linear Discriminant Analysis" (LDA) classification technique was also utilized to predict student performance. LDA identifies a linear combination of features that best separates the classes. Three LDA experiments, labeled lda1, lda2, and lda3, were conducted using different sets of input features. Each LDA model was trained on a designated training dataset and tested on an independent test set. Confusion matrices provided insights into the models' accuracy in classifying instances into the categories: Distinction, Fail, and Pass.

Random Forest: The study employed the Random Forest Classifier, a powerful ensemble learning algorithm, to predict student performance based on various input features. The Random Forest constructs multiple decision trees during training and uses the combined output to make accurate classifications. Three Random Forest models, labeled rf1, rf2, and rf3, were tested using different feature sets. Each model was trained on a specific training dataset and evaluated on a separate test dataset, with confusion matrices used to illustrate the models' ability to classify instances into Distinction, Fail, and Pass.

Neural Network: The analysis used three Neural Network Classifier models (model1, model2, and model3) to predict student outcomes using distinct input feature sets. Each model went through individual training and testing phases. Model 1, featuring a deep neural architecture with densely connected layers, achieved 95% accuracy after 199 epochs, assisted by early stopping, and showed a strong precision-recall balance in the confusion matrix. Model 2, with a similar design, ended training at 125 epochs and also reached 95% accuracy. However, Model 3 struggled with classifying the 'Fail' category, attaining zero precision for this class even with early stopping at 71 epochs, leading to an overall accuracy of 86%. Model 3 requires further optimization to improve its ability to predict instances in the 'Fail' category.

2.3.2 Predict students performance in by building a neural network with Tensorflow with DNNRegressor

In this project the authors aim to force the power of neural networks and TensorFlow to predict students' performance in the topics of Portuguese and Math. By building and training neural network models on a dataset containing relevant features and corresponding grades, the authors aim to create accurate predictive models that can aid in identifying students' academic results.[9]

This dataset focuses on the academic performance of students in secondary education from two Portuguese schools. It encompasses attributes such as student grades, demographic details, social factors, and school-related features.

The data is acquired through school reports and questionnaires, with two distinct datasets available for the subjects of Mathematics (mat) and Portuguese language (por). In a previous study [10], the two datasets were structured to facilitate binary/five-level classification and regression tasks.

Neural Network Algorithm:
In this project, the neural network consists of three hidden layers, each containing 32 neurons. These hidden layers are responsible for learning and capturing complex patterns within the data. The activation function applied in these layers is ReLU (Rectified Linear Unit), a commonly used function in artificial neural networks, especially in deep learning models. ReLU introduces non-linearity to the network, enabling it to learn and model intricate relationships in the data. The output layer is defined implicitly according to the regression task handled by DNNRegressor, a class in the TensorFlow library designed specifically for building and training Deep Neural Network (DNN) models for regression tasks.

2.3.3 Students' performance and difficulties prediction

This academic project focuses on employing machine learning algorithms to forecast the likelihood of a student passing the final exam. The widespread disruption caused by the 2020 coronavirus outbreak has had a substantial impact on global educational systems.
Studies indicate a decline in student performance since then, emphasizing the imperative to address this issue earnestly and explore effective solutions along with the contributing factors. [11]
The dataset [12] contains school, sex, age, address, famsize, Pstatus, Medu, Fedu, Mjob, Fjob, reason, guardian, traveltime,studytime, failures, schoolsup, famsup, paid, activities, nursery, higher, internet, romantic, famrel, freetime, goout, Dalc, Walc, health, absences, passe
Dataset visualization:
After processing the dataset, the next step is dataset visualization; the objective is to detect the patterns, trends and correlations that might not otherwise be detected.
This field will give some perceptions into the data and help authors to understand the dataset by placing it in a visual context using python libraries such as: matplotlib and seaborn.
In this project the visualization methods chosen are:

1. plotting distribution histograms to display the number of samples that occur in each specific category.
2. plotting "Boxplots" to show how the student status is distributed according to each variable.
3. plotting the correlation output that list all the features and their correlations to the target variable. So that authors may detect the most impactful elements on the student's status.

Machine learning algorithm:
The ML algorithm used in this project is K-Nearest Neighbour, which is one of the simplest Machine Learning algorithms based on Supervised Learning technique. It assumes the similarity between the new case data and available cases and puts the new case into the category that is most like. (The available categories K-NN algorithm stores all the available data and classifies a new data point based on the similarity). This means when new data appears then it can be easily classified into a well suite category by using the K- NN algorithm.

2.4 Comparative Analysis: importance of Neural Network

The first project centers around predicting student performance using a variety of machine learning algorithms. Logistic Regression, Linear Discriminant Analysis, Random Forest, and Neural Network classifiers were employed to predict student outcomes based on distinct sets of input features. The algorithms were valued based on their capability to classify student performance into categories of Distinction, Fail, and Pass. Each algorithm was exposed to a rigorous judgment, leading to a universal evaluation of their predictive efficiency. The neural network classifiers had difficulties predicting the cases of failure, but overall had a better performance comparing with the other models. Other models could be used in headhunting programs, developed to select students who are very likely to graduate with distinction and offering them scholarships, jobs, etc.
In the second project, the focus turns to leveraging the power of neural networks and TensorFlow for predicting student performance in Portuguese and Math. The dataset contains essential features and corresponding grades. By constructing and training neural network models, the aim is to create accurate predictive models capable of discerning students' academic results. The neural network architecture, presenting hidden layers with 32 neurons and ReLU activation functions, proposes

a platform for learning complex patterns within the data. This regression-based approach seeks to predict continuous numeric values for student performance.

Advantages of Using Neural Networks and TensorFlow for Predicting Student Performance in Portuguese and Math:

Complex Pattern Recognition: Neural networks are highly effective at identifying complex and nonlinear patterns in data. In educational settings, where student performance is influenced by intricate interactions, neural networks are particularly well-suited for uncovering indirect relationships that traditional algorithms might overlook.

Feature Extraction: TensorFlow's versatility allows for efficient feature extraction and transformation. Neural networks can automatically identify and extract important features from raw data, reducing the need for manual feature engineering and improving predictive accuracy.

End-to-End Learning: Neural networks support end-to-end learning, where raw data is fed directly into the model, which then learns to perform tasks like predicting student performance. This approach minimizes the need for manual preprocessing and additional steps.

Adaptability to Various Data Types: Neural networks can handle different types of data, such as numerical, categorical, and textual information. This adaptability is essential in educational datasets that often include a wide range of attributes.

Disadvantages:

Data Quality and Quantity: Neural networks require large volumes of high-quality data to train effectively. Insufficient or noisy data can significantly reduce the model's performance.

Overfitting: Complex neural network structures are prone to overfitting, where the model learns noise from the training data instead of general patterns. Regularization techniques are necessary to mitigate this issue.

Model Complexity: Although neural networks can detect complex patterns, their intricate nature often makes them hard to interpret. This "black box" characteristic can hinder educators from understanding the reasoning behind the model's predictions.

The third project focuses on predicting whether a student will pass the final exam, a critical task given the disruptions caused by the global pandemic. The dataset encompasses various attributes related to students' backgrounds, activities, health, and more. The K-Nearest Neighbors (K-NN) algorithm is used, which classifies new data points based on their similarity to known categories. K-NN's straightforward and supervised learning approach makes it well-suited for this task.

Advantages and Disadvantages of Knn algorithm:

Advantages:

- It is simple to implement.
- It is robust to the noisy training data
- It can be more effective if the training data is large.

Disadvantages:

- Always needs to determine the value of K which may be complex sometimes.
- The computation cost is high because of calculating the distance between all training set

A comparison of these projects reveals that various machine learning algorithms have been used to predict student performance, with each project selecting a different approach tailored to the specific needs of the task. While Logistic Regression, Linear Discriminant Analysis, and Random Forest focus on classification techniques, Neural Networks and K-NN highlight regression and similarity-based classifications, respectively.

Additionally, the attributes and structure of the datasets significantly influence the choice of algorithms. Despite their drawbacks, neural networks are often the preferred choice for prediction models dealing with complex and nonlinear patterns, particularly when ample high-quality data is available for training.

2.5 Concluding Insights

As we conclude our exploration of predicting student academic performance through machine learning, a clear pattern emerges that highlights the core of our comparative analysis. Among the various machine learning algorithms employed across these projects, one stands out for its effectiveness in the educational field: the Neural Network.

The Success of Neural Networks:

Among the different algorithms, Neural Networks have proven to be the cornerstone of predictive accuracy and adaptability in forecasting student outcomes. In educational data analysis, Neural Networks have shown exceptional flexibility, making them a powerful tool for tackling the complex challenges of academic prediction.

Precision and Predictive Efficacy:

The strength of Neural Networks lies in their impressive predictive accuracy. Projects utilizing Neural Networks, like the second project, consistently achieve high precision in predicting student performance. By identifying intricate patterns and relationships within complex datasets, Neural Networks effectively capture the diverse variables that influence academic outcomes.

Flexibility and Adaptability:

The adaptability of Neural Networks has been key to their prominence. Whether used for regression tasks, as in the second project, or for classification tasks, as seen in the first project, Neural Networks have seamlessly handled a wide range of prediction scenarios. Their ability to model nonlinear relationships, learn hierarchical representations, and account for complex interactions enables them to navigate the complexities of educational data.

Broader Implications and Future Impact:

The dominance of Neural Networks in predicting student performance has significant implications for the education sector. Beyond grade prediction, Neural Networks have the potential to revolutionize personalized learning, optimize teaching strategies, and improve educational outcomes. They are valuable for a variety of applications, from providing targeted interventions for struggling students to refining educational policies based on data-driven insights.

In summary, the combination of precision, adaptability, and predictive power has positioned Neural Networks at the forefront of educational analytics. Neural Networks are the most prominently used machine learning algorithm for shaping the future of education through data-driven insights and informed decision-making. Consequently, we have chosen to implement the Neural Network algorithm in our project to predict student performance.

3. METHODOLOGY

Predicting student outcomes plays a crucial role in personalized learning and timely interventions within the dynamic field of educational data analysis. Neural networks, a sophisticated approach in modern machine learning, are widely employed for their ability to deliver high accuracy and reliability in forecasting academic success or failure among students. This project provides a detailed analysis of our advanced model, which utilizes neural networks for binary classification to predict student performance.

3.1 Significance of Predictive Modeling in Education

Knowing what lies ahead for students is critical for both teachers and administrators. Through predictive modeling, schools can identify students who are most likely to face academic difficulties in good time, hence providing focused support mechanisms. We are applying advanced machine learning here with the goal of making this prediction process more accurate and efficient thereby giving it more relevance to education professionals.

3.1.1 The Role of Neural Networks in Educational Predictive Modeling

They are adept at capturing complex patterns and relationships in complicated data sets, inspired by the structure and function of the human brain. Neural networks are excellent at identifying subtle indicators that contribute to a student’s academic path when it comes to educational data. This model follows a binary classification technique where the neural network distinguishes between two possible outcomes: success and failure. This particular binary classification paradigm is appropriate because it matches with the need for comprehensible predictions that can be used in educational establishments.

3.1.2 Model Architecture

Input Layer

We cannot talk about input layer without describing the features selected in our model. The following table presents 7 course that enter into the official baccalaureate exams of the Iraqi education system and that we consider as feature of our model.

مادة الاختصاص	كيمياء	فيزياء	رياضات	انكليزي	عربي	اسلامية

The first layer of the neural network is the input layer which can host seven different grade points. These indicators will be the foundation for predicting student outcomes, encompassing various aspects of academic performance.

Hidden Layers

We utilized three hidden layers in our model, which are designed to capture relationships within the data that are too subtle for other models. The architecture enables the network to extract hierarchical features, so that it can identify even small patterns indicating a learner’s chances of success.

Output Layer

In our final layer, we have chosen sigmoid as an activation function that forms the basis for binary classification tasks. For instance, at output level it produces a binary value where 0 means failure and 1 means success.

3.2 Evaluation Metrics for Model Performance

Evaluating the model is crucial to assess its effectiveness. Metrics like accuracy, precision, recall, confusion matrix analysis, and the ROC curve with AUC offer a thorough insight into the model's predictive performance.
The practical application of this predictive model extends beyond theoretical analysis. By accurately forecasting student outcomes, educational institutions can implement timely interventions, offer personalized support, and allocate resources more efficiently, ultimately enhancing student success rates.
However, developing and deploying such predictive models is not without its challenges. These include handling imbalanced datasets, ensuring model interpretability and addressing scalability concerns among others. Hence, continuous refinement and adaptation are necessary for it to be effective amidst changing educational or machine learning landscapes.
In conclusion, our model for predicting student outcomes through binary classification with a neural network represents a significant advancement in the realm of educational data analysis. We see neural networks as a way of giving educational institutions a device that does more than predict whether students will pass or fail; instead assists in what proactive measures they can take so as to nurture each learner along their academic path. As we navigate the intersection of education and machine learning, the potential for positive impact remains boundless.

4 RESULTS

As shown in the figure below, our predictive model of student achievement yielded 600 correct predictions out of 700 students (we show here a part of the result with 12 correct predictions out of 14 students). It is important to note that our model was trained and tested using only official data, which ensures that the results are reliable and valid. The data was splitted randomly on training data (70% of the 700 students) and test data (30%).
It is also to note that in the data we used for high level schools the data is unbalanced because the majority of students passed the official exam and then the number of students in the “passed” class is significantly greater than (>95%) the number of students in the “failed” class. On the other hand, the number of students of the class “failed” in the low level schools represents over 70% of the total number of students in these schools.
Before putting data into a neural network, there is a very detailed pre-processing phase to go through. It involves cleaning up data in order to deal with missing values as well as outliers, normalizing data so that all features have same scale and encoding categorical variables into format that can be read by the ANN.
The training set includes historical student cases labeled with their binary results so that students’ input factors could subsequently be linked with corresponding binary results by training them on these datasets. Training iteratively adjusts weights of models or programs during this process., minimizing a binary cross-entropy loss function, and optimizing through algorithms like stochastic gradient descent.

		نتائج المدرسة						
نتيجه برنامج الذكاء الاصطناعي	نتيجه الشهادة الرسميه	مادة الإختصاص	الكيمياء	الفيزياء	الرياضيات	انكليزي	عربي	اسلاميه
راسب	راسب	30	78	63	66	55	51	64
راسب	راسب	34	50	46	50	33	50	70
راسب	راسب	54	71	50	57	62	70	80
راسب	راسب	56	54	30	55	35	54	70
ناجح	ناجح	66	64	40	58	45	55	63
راسب	راسب	30	53	42	51	54	50	70
راسب	راسب	45	57	50	25	44	50	55
ناجح	ناجح	50	50	50	45	50	38	56
ناجح	راسب	52	59	50	59	37	58	64
راسب	ناجح	50	50	50	45	42	42	61
راسب	راسب	58	45	58	40	53	57	68
راسب	راسب	50	40	40	35	28	51	65
ناجح	ناجح	51	44	40	50	31	56	56
راسب	راسب	50	20	23	37	54	50	55

Fig .13. Our model prediction

Unbalanced data posed challenges for our machine learning algorithms because the model may become biased toward the majority class, leading to poor performance in predicting the minority class.
It's important to implement strategies to address the imbalance to ensure effective learning and model performance. That will be one of our objectives for the future.
Despite achieving a commendable accuracy of 85.7% in our student outcome prediction model, we are committed to further enhancing its performance. To achieve this, we are integrating additional indicators, particularly considering the academic level at which the student is enrolled.
Obtaining student grades and academic results from schools in Iraq is notoriously challenging, often considered a daunting task. Despite these difficulties, our commitment to enhancing our machine learning model remains strong as we explore additional factors that could lead to more precise predictions.
We believe that incorporating additional indicators into our prediction model will improve its performance and boost accuracy. In our continuous pursuit of improvement, we actively seek detailed feedback from various schools across different regions in Iraq, aiming to refine and enhance our model through ongoing iterations.
It is a well-known principle in machine learning that the abundance and diversity of data significantly enhance predictive models. In this context, increasing the volume and variety of data can greatly benefit our model, potentially leading to substantial accuracy improvements. We are dedicated to this iterative process as we strive to consistently refine and optimize our model.

5. DISCUSSION

Through a comprehensive examination of our research on predicting student outcomes, a significant distinction becomes apparent when comparing our results with similar studies. Consistent with trends in the field, the use of neural networks consistently yields more accurate predictions than other approaches, particularly in identifying subtle patterns within large and complex student performance datasets.
This success is not coincidental but rather a result of neural networks' inherent ability to detect complex relationships within diverse student data. By dynamically adjusting their internal parameters to align with specific data distributions, neural networks prove to be the optimal choice. Our model not only achieves an impressive 85.7% accuracy but also consistently outperforms other methods, demonstrating its robustness across various student datasets and scenarios.
In this section, we critically analyze and compare our work with similar research in our field. Our approach is grounded in a detailed analysis that clarifies our study’s findings. I also provide an honest assessment of how well my thesis objectives have been met, supporting my conclusions with well-reasoned arguments.
Furthermore, I discuss the broader significance of the thesis within the academic field, highlighting the potential impact and implications of our findings. I conclude by exploring potential future directions, emphasizing the innovative contributions of my thesis and suggesting areas for further research that could expand the field’s understanding.
Throughout this project, we made a deliberate decision to use data exclusively from official sources, such as educational authorities or the Ministry of Education, driven by several compelling reasons:

1. Data Quality and Reliability: To ensure high reliability and good data quality for forecasting student outcomes, information from official sources is subjected to stringent collection, validation, and maintenance procedures.
2. Educational Indicators: Official sources provide access to a wide range of educational indicators and performance data, crucial for accurate predictions.
3. Policy Insights: Data from educational authorities can offer insights into policies impacting student outcomes, instrumental for forecasting future trends.
4. Historical Data: Official sources often maintain extensive historical datasets, essential for building predictive models by identifying patterns in student performance over time.
5. Data Granularity: Official sources offer data at different levels, enabling a detailed examination of factors contributing to student performance.

Moreover, utilizing data from official educational sources ensures:

1. Data Security and Compliance: loyalty to strict security and compliance guidelines to protect the integrity and privacy of sensitive student data.
2. Data validation and quality assurance procedures: These procedures lower the possibility of errors, anomalies, or missing data.
3. Transparency and Accountability: Transparency about data collection methods and sources, aiding researchers in understanding the origin and potential limitations of the data.
4. Collaboration and Expertise: Opportunities for collaboration and guidance from experts within educational authorities, offering insights and assistance in working with complex educational data.

This method strengthens the validity and resilience of our study on student outcome prediction. It emphasizes our dedication to using reliable, accurate, and pertinent data, which strengthens the credibility and impact of the report.
Retraining the predictive model with fresh data over time enables it to adjust to evolving patterns, guaranteeing continued relevance and accuracy.

6. CONCLUSION

The objective of our project is to create a model to predict the result of students using machine learning model.
After studying several possible algorithms, we have chosen the neural network algorithm because its many advantages and the success of this type of algorithms in the domain of ML and specially in the prediction models.
Our first intention was to train our model using data from different schools of distinct levels.
The first difficulties we encountered were collecting the information. While the official baccalaureate results are available on the official website of the Ministry of Education, it was very difficult for us to find school results.
In fact, many schools that we contacted categorically refused to provide us with information on the academic results of their students.
The reason may be that we asked for the correct results so that we could use them to train our model and obtain results close to reality.
We have finally obtained data from some schools. We hoped to obtain much more information and data sets, but despite our diligent efforts, we were only able to obtain a small amount of this data through personal relationships with some school managers.
The difficulty in obtaining results from schools pushed us to change our initial strategy and to ask ourselves questions about what is hidden behind this refusal to provide us with the data.
Clearly, we have doubts about the reported results of some schools and the official results they send to the Ministry of Education.
To be as close as possible to reality, we therefore decided to create two distinct models for schools of different levels.
The results obtained after testing these two models using data from schools at different levels showed clearly and clearly the accuracy of our choice.
After this project and the results obtained, it is clear that it is not credible to create a prediction model for all schools in Iraq.
Instead, the creation of prediction models for each level of schools gives more credible results and closer to reality.
This is the choice that we adopted after several trials and attempts.
This conclusion will guide our future work in the prediction of Iraqi student results in the future.

Conflicts Of Interest

The author's paper emphasizes that there are no conflicts of interest, either perceived or actual, that could impact the research integrity.

Funding

The paper does not disclose any collaborations with funded projects or institutions, indicating the author had no external financial support.

Acknowledgment

The author would like to thank the administrative staff at the institution for their assistance and logistical support throughout the duration of this research.

References

[1] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 4th ed. University Book, 2021.
[2] I. H. Witten, Data Mining: Practical Machine Learning Tools and Techniques, 2020.
[3] V. Nasteski, "An overview of the supervised machine learning methods," Horizons, vol. 4, pp. 51-62, 2017.
[4] C. Donalek, "Supervised and unsupervised learning," presented at Astronomy Colloquia, USA, vol. 27, p. 8, Apr. 2011.
[5] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction. MIT Press, 2018.
[6] Neural Network Tutorial, Neural Designer. [Online]. Available: https://www.neuraldesigner.com/machine-learning/neural-network

[7] "Student Performance Prediction - Complete Analysis," Kaggle. [Online]. Available: https://www.kaggle.com/code/devassaxd/student-performance-prediction-complete-analysis
[8] "Analyse," Knowledge Media Institute. [Online]. Available: https://analyse.kmi.open.ac.uk/
[9] O. Kim, "Predict students performance by building a neural network with TensorFlow," GitHub. [Online]. Available: https://github.com/oleksandrkim/Predict-students-performance-by-building-a-neural-network-with-Tensorflow/blob/master/ann_school_regression.ipynb
[10] "Student Performance," UCI Machine Learning Repository. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/student+performance
[11] M. Aljadd, "Students Performance Prediction," GitHub. [Online]. Available: https://github.com/mohammedAljadd/students-performance-prediction/tree/main
[12] M. Aljadd, "Student Data," GitHub. [Online]. Available: https://github.com/mohammedAljadd/students-performance-prediction/blob/main/student-data.csv