UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS

FACULTAD DE INGENIERÍA

PROGRAMA ACADÉMICO DE INGENIERÍA DE TELECOMUNICACIONES Y REDES

Diseño e implementación de un modelo de red neuronal artificial para la predicción del rendimiento académico utilizando información no personal

TESIS

Para optar el título profesional de Ingeniero de Telecomunicaciones y Redes

AUTOR(ES)

Chavez Arias, Jorge Heyul 0000-0001-9415-2782

ASESOR(ES)

Sanchez Huapaya, Alonso Sebastian 0000-0002-4449-786X
Raymundo Ibañez, Carlos Arturo 0000-0001-9615-0991

Lima, 08 de noviembre de 2023

DEDICATORIA

Esta tesis está dedicada a mi madre y a mi padre celestial que siempre confían en mi inclusive cuando yo no lo haga.

AGRADECIMIENTOS

En primer lugar, agradezco a Dios que es mi guía y maestro, y a mi familia que siempre me apoya incondicionalmente.

También agradezco a mi asesor el Doctor Carlos Raymundo Ibañez y la Universidad Peruana de Ciencias Aplicadas (UPC) que gracias a su apoyo permitieron el desarrollo del presente estudio, en conjunto con su publicación.

Diseño e implementación de un modelo de red neuronal artificial para la predicción del rendimiento académico utilizando información no personal

RESUMEN

La presente investigación se basa en la investigación realizada por los autores, denominada "Artificial neural network model to predict student performance using nonpersonal information" [1] que constó de lo siguiente:

- **Análisis y propósito de la investigación:** El bajo rendimiento académico es uno de los principales problemas que enfrenta la educación superior, y es difícil combatirlo ya que es influenciado por diferentes factores. En el Perú, la pandemia del COVID-19 forzó a los estudiantes migrar a entornos virtuales de aprendizaje. Esto introdujo la necesidad de que estudiantes y profesores se adaptaran a un entorno de aprendizaje virtual. Esta investigación propone desarrollar un sistema que pueda predecir el rendimiento académico en base a la interacción de los alumnos con la plataforma virtual de aprendizaje sin usar información personal.
- **Diseño y Metodología:** Para el diseño del sistema predictor es necesario obtener y procesar una base de datos que tengas toda esta información, luego de obtenida, se realiza una limpieza, selección previa de variables, transformación, extracción, combinación, manipulación, etc. para generar las variables deseadas para el análisis del modelado predictivo, a esto también se le conoce como *Feature engineering*. Luego de tener los datos listos se realiza la implementación y optimización de distintos modelos de Aprendizaje Automático y Redes Neuronales Profundas, para luego ser comparadas y seleccionar al que logre el mejor desempeño.
- **Resultados:** El algoritmo escogido es una Red Neuronal Artificial, al realizar la prueba de validación de un total de 291 estudiantes que aprobaban un curso, 274 se predijeron correctamente, y de un total de 210 estudiantes que desaprobaron, 196 se predijeron correctamente. Las métricas que obtuvo el modelo fueron las siguientes: 93.81% accuracy, 94.15% precision, 95.13% recall, y 94.64% F1-score.
- **Principal conclusión:** La predicción del rendimiento académico de los estudiantes permite a las instituciones identificar mejor a los estudiantes en riesgo y tomar medidas de apoyo, garantizando así que los estudiantes completen con éxito sus ciclos académicos, a su vez, ayudando a las instituciones a ganar más prestigio.

Palabras clave: privacidad; datos personales; redes neuronales; rendimiento académico.

Design and implementation of an artificial neural network model for academic performance prediction using non-personal information.

ABSTRACT

The present research is based on the research conducted by the authors, entitled "Artificial neural network model to predict student performance using nonpersonal information" [1] which consisted of the following:

- **Analysis and purpose of the research:** Low academic performance is one of the main problems facing higher education, and it is difficult to combat it as it is influenced by different factors. In Peru, the COVID-19 pandemic forced students to migrate to virtual learning environments. This introduced the need for students and teachers to adapt to a virtual learning environment. This research proposes to develop a system that can predict academic performance based on student interaction with the virtual learning platform without using personal information.
- **Design and Methodology:** For the design of the predictive system it is necessary to obtain and process a database that has all this information, after obtaining it, a cleaning, previous selection of variables, transformation, extraction, combination, manipulation, etc. is performed to generate the desired variables for the analysis of the predictive modeling, this is also known as Feature engineering. After having the data ready, the implementation and optimization of different models of Machine Learning and Deep Neural Networks is performed, to then be compared and select the one that achieves the best performance.
- **Results:** The chosen algorithm is an Artificial Neural Network, when performing the validation test of a total of 291 students who passed a course, 274 were correctly predicted, and of a total of 210 students who failed, 196 were correctly predicted. The metrics obtained by the model were as follows: 93.81% accuracy, 94.15% precision, 95.13% recall, and 94.64% F1-score.
- **Main conclusion:** Predicting students' academic performance allows institutions to better identify at-risk students and take supportive measures, thus ensuring that students successfully complete their academic cycles, in turn, helping institutions gain more prestige.

Keywords: privacy; personal data; neural networks; academic performance.

N°8852_Chavez Arias Jorge Heyul_Diseño e implementación de un modelo de red neuronal artificial para la predicción del rendimiento académico utilizando información no personal

INFORME DE ORIGINALIDAD

15%	13%	11%	8%
INDICE DE SIMILITUD	FUENTES DE INTERNET	PUBLICACIONES	TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	scjournal.ius.edu.ba Fuente de Internet	1%
2	repositorioacademico.upc.edu.pe Fuente de Internet	1%
3	www.dropbox.com Fuente de Internet	1%
4	eprints.whiterose.ac.uk Fuente de Internet	1%
5	Submitted to University of Suffolk Trabajo del estudiante	1%
6	www.frontiersin.org Fuente de Internet	1%
7	www.kluniversity.in Fuente de Internet	1%
8	Submitted to Queensland University of Technology Trabajo del estudiante	1%

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

1. ARTICULO EXTENDIDO (JOURNAL PAPER)

1.1 Introduction

Artificial intelligence has exerted a large impact in most industries, including education. The use of these technologies (Deep Learning, Automatization and Natural Language Processing) provides important knowledge that benefits the industry, such as providing better understanding of the student learning processes or improving their results.

Low academic performance is one of the main problems facing higher education worldwide. One reason why this issue has become difficult to solve is that academic performance is influenced by several factors. F. Tejedor, and A. García-Valcárcel [2] mentions five factors: identification, psychological, academic, pedagogical, and socio-family. Similarly, there can be various consequences of poor academic performance, including academic attrition. According to Vicerrectorado Académico [3], one of the main factors for academic desertion is poor academic performance. Within the same context, Viale [4] argues that the percentage of students who fail classes in the first cycle is usually high, but when students repeat a class and fail it again, many decide to drop out of university. Furthermore, low academic performance is considered a quality indicator for educational institutions, so its assessment is critical for these institutions. In Peru, SINEACE [5] states that following up on students and leveling their deficiencies to prevent low academic performance is considered a quality standard for university programs and to achieve accreditation.

The COVID-19 pandemic resulted in students migrating to virtual learning environments. This introduced the need for students and teachers to adapt to a hybrid learning environment, if necessary, where they face different obstacles. C. Liao and J. Wu [6] mentioned that hybrid learning has provided students with more professional growth opportunities; however, this has also become a challenge because students are exposed to different types of distractions during the learning process.

In their research, J. Blanco, S. Lovelle, R. Fernandez, and E. Perez [7] and E. G. Rincon-Flores, E. López-Camacho, J. Mena and O. O. López [8] forecast academic student performance using different algorithms. J. Blanco, S. Lovelle, R. Fernandez, and E. Perez [7] proposed a deep neural network to predict academic performance, achieving 78% accuracy, whereas E. G. Rincon-Flores, E. López-Camacho, J. Mena and O. O. López [8] mentioned several models, such as K-nearest neighbors (KNN) and Random Forest, to achieve 80% accuracy. However, these works do not justify why they are forecasting academic student performance, the application of these predictions, and who they are going to help.

This research seeks to propose and develop a model for the prediction of student academic performance, due to the large amount of data to be used, it is intended through deep neural networks to respond to the educational problem of low academic performance in universities. Forecasting academic student performance allows institutions to better identify at-risk

Este trabajo originalmente fue publicado en Chavez H, Chavez-Arias B, Contreras-Rosas S, Alvarez-Rodríguez JM and Raymundo C (2023) Artificial neural network model to predict student performance using nonpersonal information. Front. Educ. 8:1106679. https://10.3389/feduc.2023.1106679

students and take supporting measures, thereby ensuring that students successfully complete their academic cycles, in turn, helping institutions gain more prestige.

The model training data used will be provided by J. Kuzilek, M. Hlosta and Z. Zdrahal [9], wherein we can acquire information such as courses, exams, personal student information, and student interaction with the resources of the Open University's online academic platform. First, we apply preprocessing techniques to prepare the data that will be used for training. Second, we sort the students into pass or fail categories. Third, we split the data into training and testing datasets. Finally, we train the model using the corresponding dataset. As part of our results, the accuracy of our prediction model will be compared with other similar works.

This research seeks to answer the following question: How to estimate whether a student will pass a subject before completing 50% of the course without using personal information?

1.2 State of the art

1.2.1 Using the Naïve Bayes algorithm for forecasting students' academic performance

The literature offers various mathematical applications, wherein the Naïve Bayes algorithm stands out due to its large application percentage. In this section, we will discuss the success percentages of each study that used the Naïve Bayes algorithm as well as their characteristics, such as the quantity of information required and under what circumstances each study was conducted.

In their research, V. U. Kumar, A. Krishna, P. Neelakanteswara, and C. Z. Basha [10] assessed student performance in technical exams used for university application, an important process for universities. Forecasting whether students would pass or fail a technical exam based on their performance in certain subjects of their course was the technique used to approach the problem. The authors used data regarding 200 K L University students from 2013 to 2017. Students were sorted and classified using two different algorithms: K-means clustering and hierarchical clustering. Once classified, the authors used the Naïve Bayes algorithm for their predictions at 72% accuracy.

In their study, S. Jayaprakash, S. Krishnan, and V. Jaiganesh [11] discuss student dropout rates in the early stages of the program by calculating the factors that exert the most influence on academic student performance. For these predictions, the authors used 887 instances and 19 variables. Additionally, they used the Naïve Bayes algorithm at 85.7% accuracy.

In their work, Daud *et al.* [12] assessed the reasons why many students drop out of their careers. For this, the study predicted personal, family, and economic factors that most influence student performance to determine whether any given student would complete their studies. For testing, the dataset is cleaned to obtain 50 students who finished their studies and 50 who did not. As part of their results, the Naïve Bayes algorithm achieved 84.8% in the F1-score.

The study reported by Y. Widyaningsih, N. Fitriani, and D. Sarwinda [13] intended to forecast academic performance for freshman students using a semisupervised learning

approach to classify student performance. In this study, the Naïve Bayes classifier was used as a methodology, which achieved 96% accuracy.

T. Devasia, T. P. Vinushree, and V. Hegde [14] addressed student desertion in higher education. The proposed system is a web-based application that employs the Naïve Bayesian mining technique to extract useful information. According to this study, the basic algorithm provides more accuracy in this field than other methods such as regression, decision tree, and neural networks.

M. V. Amazona and A. A. Hernandez [15] forecast student performance to improve decision-making education. Educational data mining was used to model student academic performance through the Naïve Bayes algorithm, assess the dataset, and predict academic performance. Results reveal that even when the Naïve Bayes classifier is outperformed by other algorithms, it achieved an overall prediction rate of 88% in classification accuracy.

In their research, A. Páez and D. Guzmán [16] solved problems affecting several teaching and learning processes. To this end, they designed and automated a predictive model of student academic performance. The study was based on several algorithms, wherein the Naïve Bayes outperformed the others at a 73% prediction rate.

In some cases, clustering is required for further data analysis. The aforementioned studies are aware of this fact because K-means is used in V. U. Kumar, A. Krishna, P. Neelakanteswara, and C. Z. Basha [10] and Y. Widyaningsih, N. Fitriani, and D. Sarwinda [13], hierarchical in V. U. Kumar, A. Krishna, P. Neelakanteswara, and C. Z. Basha [10], Daud *et al.* [12], and Y. Widyaningsih, N. Fitriani, and D. Sarwinda [13], and custom clustering types in A. Páez and D. Guzmán [16] and others depending on the desired level of classification. Nevertheless, the most common and easiest to use is Naïve Bayes, which was used in S. Jayaprakash, S. Krishnan, and V. Jaiganesh [11], Y. Widyaningsih, N. Fitriani and D. Sarwinda [13], and M. V. Amazona and A. A. Hernandez [15]. The type of data available is critical because more specific data will better cover student learning, academic environment issues, and more. Thus, as mentioned in Y. Widyaningsih, N. Fitriani, and D. Sarwinda [13], groups must be as differentiated as possible.

1.2.2 Using decision trees/random forests to predict student academic performance

V. U. Kumar, A. Krishna, P. Neelakanteswara, and C. Z. Basha [10] also proposed using the C5.0 algorithm, a decision tree algorithm, to predict student performance. For this, they used the exact same methodology as with Naïve Bayes. As a result, they obtained 81% accuracy.

Similarly, S. Jayaprakash, S. Krishnan, and V. Jaiganesh [11] proposed the random forest algorithm and an in-house variation, the improved random forest algorithm. These algorithms yielded 91 and 93% accuracy, respectively, when predicting academic performance.

In their work, Daud *et al.* [12] used the C4.5 and CART algorithms to forecast whether students would complete their coursework. Through these algorithms, the authors achieved 76.6 and 71%, respectively, in F1-score.

X. Ma, Y. Yang and Z. Zhou [17] proposed using deep neural networks to predict the passing rate of online students. They used data from Guo Pei Online Education. First, the features that most affect passing rate were filtered through the filter-type feature selection algorithm, which provided 27 representative features. Then, preprocessing and standardization were performed on the remaining dataset (27 features). The decision tree algorithm was used for prediction, and the grid search algorithm was used to optimize the decision tree. The following metrics were used with these algorithms: precision, recall, F1-score, and runtime. Before testing, the data were divided into pass and fail. Finally, using the grid search algorithm at 50% pass / 50% fail data, decision tree achieved 96% accuracy.

Of all the literature in this section, six authors used decision tree algorithms: V. U. Kumar, A. Krishna, P. Neelakanteswara, and C. Z. Basha [10], O. Castrillón, W. Sarache and S. Ruiz [18], Daud *et al*. [12], X. Ma, Y. Yang and Z. Zhou [17], H. M. R. Hasan, A. S. A. Rabby, M. T. Islam and S. A. Hossain [19], and M. V. Amazona and A. A. Hernandez [15], while S. Jayaprakash, S. Krishnan, and V. Jaiganesh [11], E. G. Rincon-Flores, E. López-Camacho, J. Mena and O. O. López [8], and C. I. P. Benablo, E. T. Sarte, J. M. D. Dormido and T. Palaoag [20] used random forest algorithms. Despite random forests being a subset of decision trees, most studies use decision trees for multiple-variable predictions, such as those for academic performance.

In their study, H. M. R. Hasan, A. S. A. Rabby, M. T. Islam and S. A. Hossain [19] present a model that tries to predict final exam results for a given student. For this, they used a dataset of 1,170 students in 3 courses. Then, the authors preprocessed the dataset by removing unnecessary columns such as Student ID. They used the KNN algorithm and a decision tree classifier (the ID3 algorithm) for their predictions, thereby obtaining 94.44% accuracy based on the decision tree classifier algorithm.

In their research, C. I. P. Benablo, E. T. Sarte, J. M. D. Dormido and T. Palaoag [20] forecast student performance based on social network data, such as Facebook, Twitter, Instagram, and YouTube, as well as online games. A single-format input was created to serve as the training dataset following the attribute relationship file format. To finalize the model, the researchers considered using support vector machine (SVM), KNN, and random forest for prediction. Furthermore, to ensure that the model accurately classified a given dataset, a 10-fold cross validation was performed using the WEKA software. For these modes, three metrics were used: precision, recall, and F1-score. For the results, the authors used three types of samples (30, 50, and 100 instances). After testing 100 instances, random forest achieved a result of 100% in precision, 80.6% in recall, and 89.3% in F1-score.

M. V. Amazona and A. A. Hernandez [15] proposed to use a decision tree for prediction following the same methodology as V. U. Kumar, A. Krishna, P. Neelakanteswara, and C. Z. Basha [10] and X. Ma, Y. Yang and Z. Zhou [17]. The results from the decision tree were 93% in precision, 96% in F1-score, and 100% in recall.

From V. U. Kumar, A. Krishna, P. Neelakanteswara, and C. Z. Basha [10], O. Castrillón, W. Sarache and S. Ruiz [18], Daud *et al*. [12], X. Ma, Y. Yang and Z. Zhou [17], H. M. R. Hasan, A. S. A. Rabby, M. T. Islam and S. A. Hossain [19], and M. V. Amazona and A. A. Hernandez [15], E. G. Rincon-Flores, E. López-Camacho, J. Mena and O. O. López [8]

reported the highest accuracy rate using a decision tree algorithm at 96%. Moreover, it considered the second-highest number of forecasting variables (27).

1.2.3 Using neural networks to predict student academic performance

The study by J. Blanco, S. Lovelle, R. Fernandez, and E. Perez [7] focuses on predicting student results for the Data Structures I and II classes because these computer engineering classes present a significant degree of difficulty and require serious dedication and rigor from students, which means that their grades are often lower than desirable. Therefore, the authors designed a model based on the MATLAB system that predicts academic results in these classes after having been trained with specific data from each class. The model's architecture is designed with three layers, wherein an activation function was used for each neuron: for the input and hidden layers, the sigmoidal hyperbolic tangent function was used. For the output layer, the linear function was used because the authors wanted to achieve the largest possible range amplitude in the output interval, which also facilitated result interpretation. In this study, a prediction effectiveness of over 78% was achieved for the first subject and 75% for the second subject.

The research reported by Sekeroglu et al. [22] was developed based on the need to improve AI-based systems in the field of education because in-class and remote students often need help to improve their performance. Therefore, two neural network models were used: backpropagation (BP) and long/short-term memory (LSTM). The former uses a gradient descent algorithm during learning and propagates the error to update weights and minimize error values; the latter memorizes previous neural network inputs to provide more accurate results. The results were as follows: BP (70% variance) and LSTM (77.9% variance).

M. V. Amazona and A. A. Hernandez [15] also proposed using a deep learning neural network model for prediction. The same methodology as above was used. The deep learning results were as follows: 98% precision, 97% F1-score, and 98% recall.

Of the studies carried out by J. Blanco, S. Lovelle, R. Fernandez, and E. Perez [7], M. V. Amazona and A. A. Hernandez [15], and Sekeroglu et al. [22], the neural network that reported the highest accuracy was the one reported by M. V. Amazona and A. A. Hernandez [15] at 98% precision, 98% recall and 97% F1-score. Moreover, M. V. Amazona and A. A. Hernandez [15] used the least number of input variables at only 9.

In this sense, J. Blanco, S. Lovelle, R. Fernandez, and E. Perez [7] used the highest number of input variables, 21. Nevertheless, its effectiveness rate when predicting student performance was considerably low (75% accuracy) compared with the effectiveness reported by M. V. Amazona and A. A. Hernandez [15] (95% accuracy). It may be observed that X. Ma, Y. Yang and Z. Zhou [17] uses fewer layers and, therefore, fewer network neurons, which leads to a more inaccurate process.

1.2.4 Support vector machine (SVM)/support vector regression (SVR) prediction models to forecast student academic performance

Daud *et al*. [12], used a supervised learning model to predict whether students will complete or abandon their study programs. Specifically, they used the SVM model, wherein the best result was obtained with 86% in the F1-score test.

X. Ma, Y. Yang and Z. Zhou [17] also used the SVM supervised learning model to predict online student passing rates. Using the grid search algorithm at 50% pass / 50% fail data, this model achieved 95% accuracy.

Sekeroglu et al. [22], used the SVR model to improve AI-based systems in the field of education. This model provided predictions at 79.7% variance.

C. I. P. Benablo, E. T. Sarte, J. M. D. Dormido and T. Palaoag [20] forecasted student performance based on social network data, such as Facebook, Twitter, Instagram, and YouTube, using the SVM model. In a 100-instance test, this model reported 100% precision, 96.8% recall and 98.4% F1-score, thereby being the most effective model.

In their study, S. N. Liao *et al*. [23] sought to predict which students are at risk of achieving poor performance in a given class. This must be identified early enough to allow instructors to help students before they fall behind. Thus, they designed a model that predicts the student's final exam grade using a binary SVM classifier, which is trained with the radial basis function kernel based on the selected course. Additionally, the model is based on three parameters that indicate the severity of different classification error types. Based on this model, at least 62% of the at-risk students were correctly identified.

From Daud *et al*. [12], X. Ma, Y. Yang and Z. Zhou [17], Sekeroglu et al. [22], C. I. P. Benablo, E. T. Sarte, J. M. D. Dormido and T. Palaoag [20], and S. N. Liao *et al*. [23], the supervised learning model that reported the highest accuracy was the one used by C. I. P. Benablo, E. T. Sarte, J. M. D. Dormido and T. Palaoag [20] at 100% precision, 96.8% recall, and 98.4% F1-score. To reach this accuracy, the authors used cross validation 10 times to avoid overfitting the model. Here, we can see a pattern: Sekeroglu et al. [22] did not use cross validation, and it reported the lowest accuracy rate at 79.7%. Similarly, Daud *et al*. [12] performed cross validation but only five times, which is half as many times as C. I. P. Benablo, E. T. Sarte, J. M. D. Dormido and T. Palaoag [20]. Daud *et al*. [12] reported an accuracy rate of 86%, which is considerably high but not as high as the accuracy rates reported by S. N. Liao *et al*. [23], X. Ma, Y. Yang and Z. Zhou [17], and C. I. P. Benablo, E. T. Sarte, J. M. D. Dormido and T. Palaoag [20], who performed the cross validation 10 times.

1.2.5 Comparison of models

This section will summarize and compare the most relevant models in the literature, which will be shown in Table 1. Data such as the technique used, accuracy, precision, recall, etc. were analyzed. It should be noted that not all models had all validation metrics.

Table 1

Model comparison

Author	Technique	Dataset	Use of Personal Information	Accuracy	Precision	Recall	F1-Score
Proposed Model	R.N.	OU	NO	93.81%	94.15%	95.13%	94.64%
[19]	SVM	Private	YES	-	98.1%	100.0%	99.0%
[19]	KNN	Private	YES	-	98.1%	99.0%	98.6%
[31]	Naïve Bayes	ICFES [28]	YES	96.0%	85.7%	89.0%	87.3%
[14]	Decision Tree	ASCOT [29]	YES	93.0%	93.0%	100.0%	93.0%
[32]	R.N.	Turkish' SIS [25]	YES	86.3%	74.8%	74.6%	72.3%
[33]	R.N.	Student Performance Data Set [30]	YES	90.1%	90.0%	90.0%	90.0%

1.2.6 Professional ethics

According to F. Alaieri and A. Vellino [24], where different ethical regimes are applicable, artificial intelligence decisions must be reliable and duly justified so that users, manufacturers, and legislators can understand how decisions are made and what ethical principles were applied in each case. This study presents a model that breaks down the stages of ethical decision-making into their elementary components. The proposed project references ethical decision-making matters regarding the algorithm used to predict student performance. The ethical issues are detailed in Section 1.3.2.

1.3 Contribution

This section introduces and explains the proposed neural network architecture and machine learning models to predict student academic performance.

1.3.1 Architecture

Figure 1 illustrates the architecture used in the model proposed.

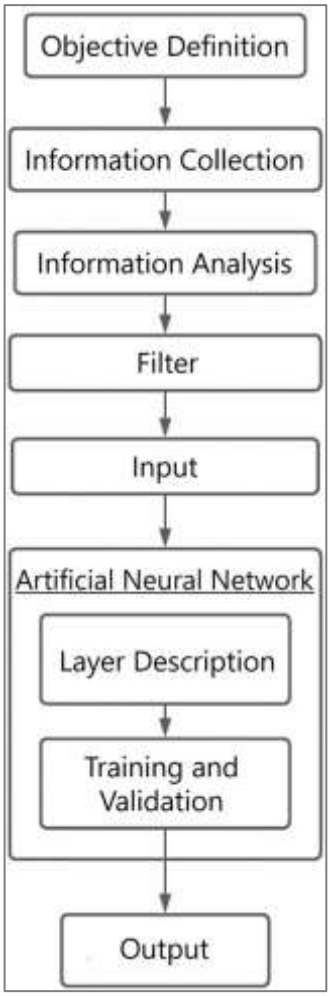

Fig. 1. Proposed architecture.

1.3.1.1 Objective definition

The main objective of this project is to forecast the academic performance of university students throughout their course to help professors and academic institutions better identify students at risk of failing and adopt corresponding supporting measures, thereby ensuring that students successfully complete their academic cycle, which in turn helps the institution garner prestige.

1.3.1.2 Information collection

The dataset used for project development was collected from the Open University Learning Analytics dataset, which provides information about courses, students, and their interactions with the Virtual Learning Environment (VLE). The dataset consists of tables interconnected with a single identifier. All tables are in.csv format. Figure 2 denotes the dataset structure.

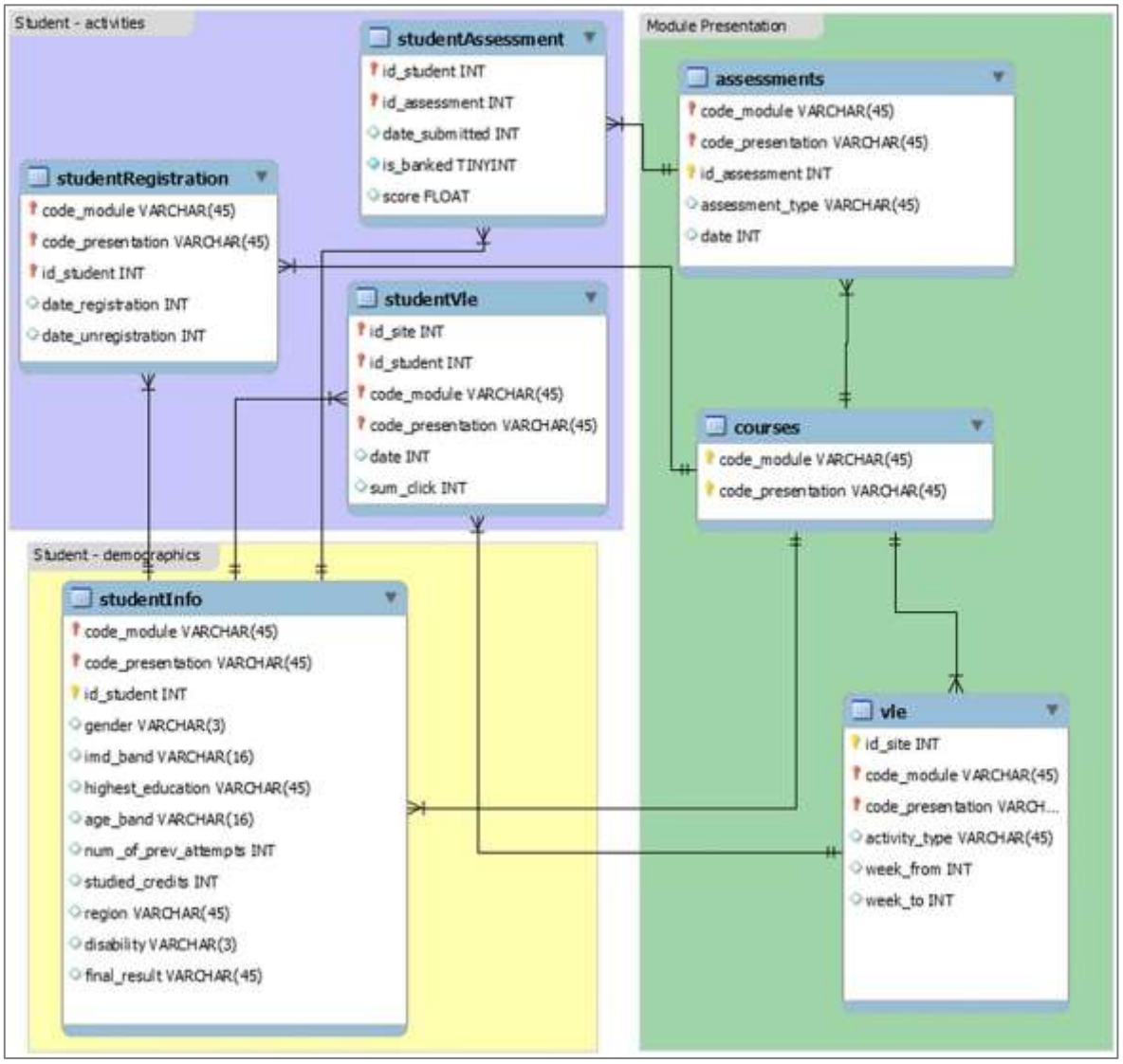

Fig. 2. Dataset structure J. Kuzilek, M. Hlosta and Z. Zdrahal [9].

The dataset obtained from J. Kuzilek, M. Hlosta and Z. Zdrahal [9] contains information about 22 courses, 32,593 students, their evaluation results, and records of their interactions with the VLE represented by daily summaries of student clicks (10,655,280 entries). This information is arranged in seven tables.

1.3.1.3 Information analysis

After the information has been collected, it must be analyzed. This analysis will help us segment the information and facilitate its manipulation. The dataset collected is detailed below, each set represents a table of the entire database.

- Courses.csv stores course information.
- Assessments.csv stores information regarding course assessments.
- Vle.csv contains information about the materials available on the VLE.
- StudentInfo.csv stores general information about the students, such as demographics and final grade.
- StudentRegistration.csv stores information about student registration to a course module.
- StudentAssessment.csv contains information about student assessment results.
- StudentVle.csv contains information about the student's interaction with the materials on the VLE.

The tables have a total of 41 attributes, of which 20 are identifiers (Primary Key and Foreign Key); of the remaining 21 attributes, 9 were discarded, since they did not have a direct relationship with the student's grade according to the documentation of the database itself.

Of the remaining 12 attributes, 3 contained personal information: age, gender and location, and another 3 attributes contained information not relevant to the application. This left only 6 attributes from the entire database that were useful for our purpose.

The target variable is "final_result" found in the "studentInfo" table as shown in Figure 2, this variable is a "string," representing whether a student passed or failed the course, which was transformed to "bulean" so that it can then be entered into the model.

1.3.1.4 Filters and preprocessing

As can be seen in Figure 3, shows the correlation of the variables using the spearman method, and Figure 4 shows the correlation of the variables but using the pearson method. After observing a correlation between the pass_rate and weighted_grade variables in the two figures, we applied a sklearn method (feature_importances) to see which variable had the greatest impact on both the machine learning models and the neural network.

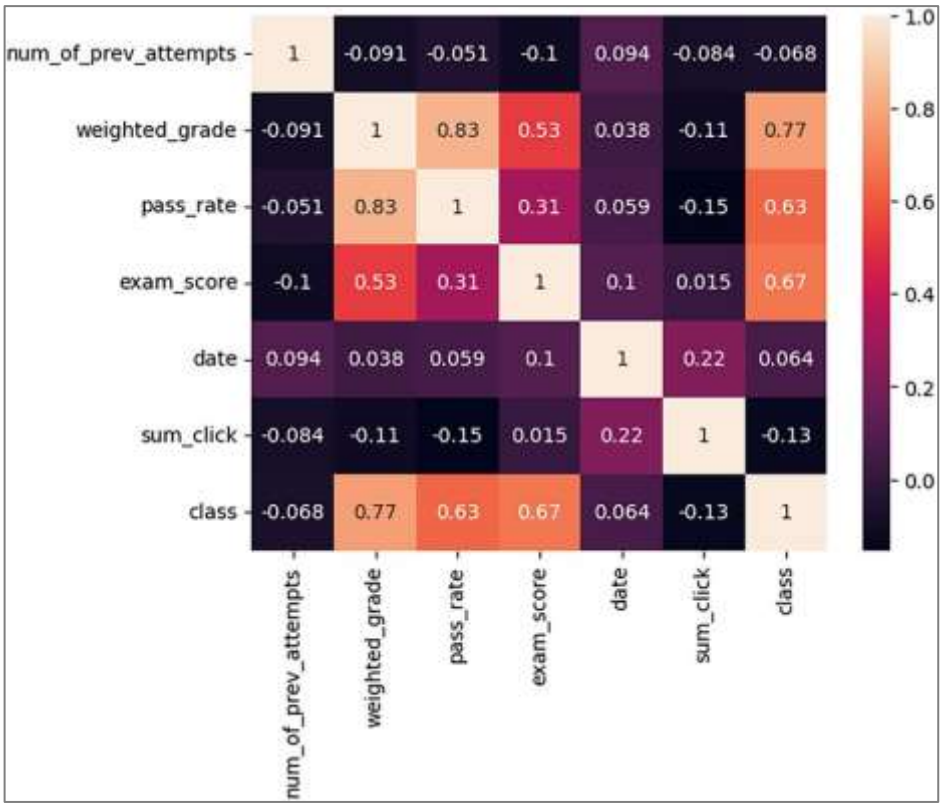

Fig. 3. Spearman's correlation coefficient

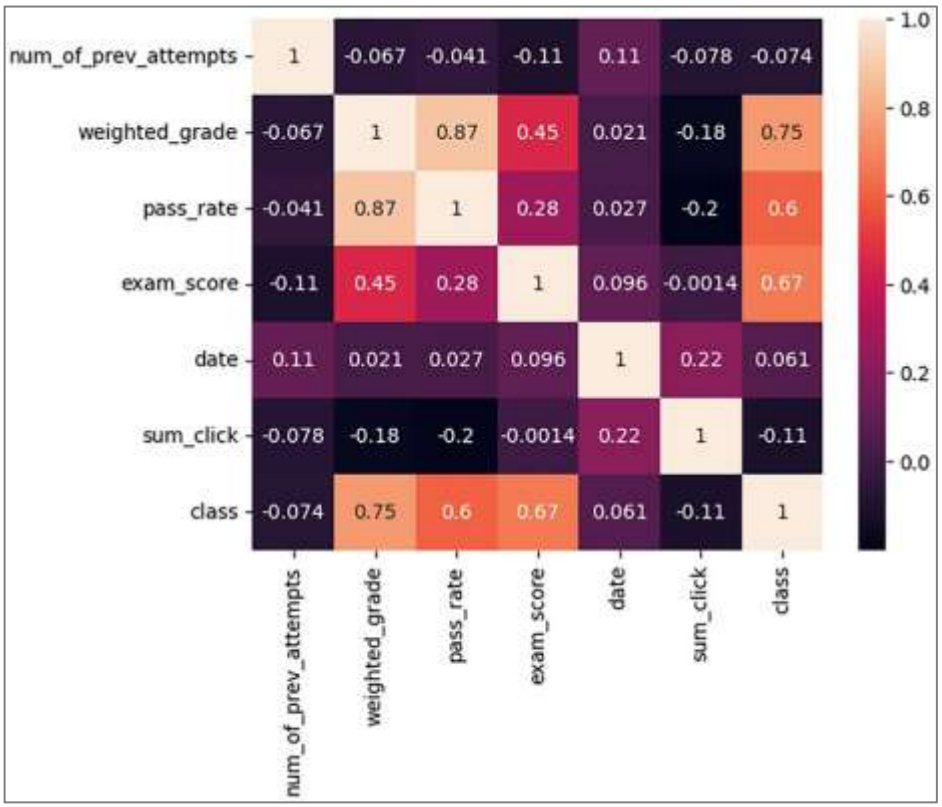

Fig. 4. Pearson's correlation coefficient

Using the Feature_importances function, we observed that weighted_grade has more impact on the models results, so we tested all the models using pass_rate and removing it as input in the models, the results are shown in Table 2.

Table 2

Accuracy comparison using the "pass_rate" variable.

Models	Accuracy	
	With pass_rate	Without pass_rate
ANN (Proposed model)	92.22%	93.81%
Random Forest	92.61%	92.81%
SVM	80.03%	80.03%
Naïve Bayes	90.01%	92.21%

On Random Forest and SVM, the change is not as significant as on ANN and Naive Bayes, for these results the pass_rate column was removed as an input.

Once the dataset has been filtered and sorted, it is ready for input into the artificial intelligence engine.

1.3.1.5 Input

The inputs used to predict student academic performance in final exams are as follows:

- Number of attempts (num_of_prev_attempts): Number of times a student has taken or repeated the course.
- Average grade of the evaluation (weighted_grade): Weighted grade obtained by the student in all the previous evaluations of the course. Not to be confused with the final exam.
- Test score (exam_score): The student's score on this evaluation. The range is from 0 to 100. A score below 40 is interpreted as a Fail. Grades are in the range of 0 to 100.
- Average date of use of materials (date): average of the dates on which the learner interacted with the materials in the EVE measured as the number of days since the presentation of the module.
- Average number of clicks (sum_click): Average number of interactions the learner had with the course materials.

It should be noted that the inputs have a uniform distribution.

1.3.1.6 Output

The outputs will be used by professors to easily identify students with the highest risk of failing the course so they can decide, based on their expertise, what type of specific methodology can help these students pass the course. To predict academic performance, we defined two categories:

- Pass: The student will pass the course.
- Fail: The student could fail the course.

The proposed artificial intelligence will use one of these categories to indicate the results.

1.3.1.7 Models and parameters

Three machine learning modes and one artificial neural network were proposed and developed; the architecture and characteristics of each model will be detailed below.

1.3.1.7.1 Artificial neural network

The proposed neural network contains three layers (Figure 5):

- The input layer contains five neurons representing the number of variables in the clean dataset. Its activation function is "ReLU," since the computational step and the gradient calculation, compared to other activation functions, is faster.
- A hidden layer wherein the number of neurons will be selected using the following empirical method [27]: "The number of hidden neurons must be less than twice the size of the input layer." Therefore, this hidden layer will have eight neurons, with an activation function of "ReLU".
- The output layer is comprised of one neuron, which represents the result from the model prediction. Its activation function is "sigmoid," this is because we have a boolean output.

For training, the following hyper parameters were used:

- Epochs: 100
- Activation Function: ReLU & Sigmoid
    - ReLU:

$$max(0, x) \tag{1}$$

    - Sigmoid:

$$S(x) = \frac{1}{1 + e^{-x}} \tag{2}$$

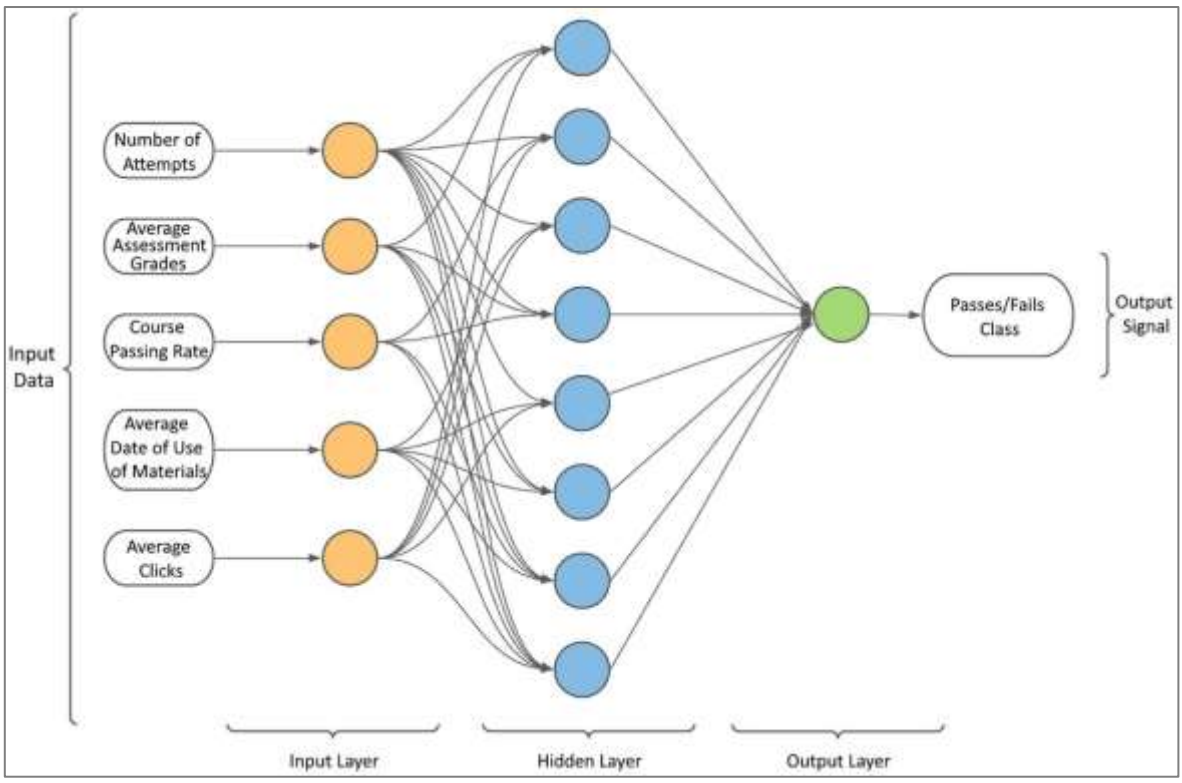

Fig. 5. Neural network architecture chart.

- Optimizer: Adam

$$m_t = \beta m_t + (1 - \beta)[\frac{\partial \mathrm{L}}{\partial w_t}] \tag{3}$$

- Loss Function: Binary_CrossEntropy

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \ \log\big(p(y_i)\big) + (1 - y_i) + \log\big(1 - p(y_i)\big) \tag{4}$$

- Metrics: Accuracy and Loss

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \tag{5}$$

$$log\ loss = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij} \log(p_{ij}) \tag{6}$$

- Overfitting: EarlyStopping

$$\int_p (x) = \int_Y y \, dp(y|x) \, , x \epsilon X \tag{7}$$

1.3.1.7.1.1 Training

Because overfitting is a common problem in neural networks, we implement the following strategy to prevent this issue. We divide the dataset 70/30 such that 70% is directed toward the training of the neural network and the remaining 30% is used for neural network validation.

The number of training epochs will be 100, and in each epoch, the backpropagation algorithm will be used to improve weights and input biases and obtain a more accurate result or "minimize loss."

1.3.1.7.2 SVM

To know exactly which parameters are necessary for the good performance of our model, we apply the sklearn function GrigSearchCV, which receives a set of parameters and returns the best ones for the model according to the training data, it is worth noting that this function was used for the following machine learning models.

As input for the GrigSearchCV function, we gave it the following parameters:

- C = [0.1, 1, 10, 100]
- Gamma = [1, 0.1, 0.01, 0001]
- Kernel = ['rbf ', 'poly', 'sigmoid']

The parameters used for the SVM model were as follows:

- C = 100
- Kernel = rbf
- Gamma = 0.1
- Probability = True

The percentage for training was 70% of the dataset.

1.3.1.7.3 Naïve Bayes

In the case of Naive Baye no parameters were specified, due to the same architecture, only the percentage for training had to be defined, which was the same as in the case of SVM, 70% of the dataset.

1.3.1.7.4 Random Forest

In the case of random fores, we used the GridSearchCV function mentioned above, with the following input parameters:

- n_estimators = [200, 700]

- max_features = [‘auto’, ‘sqrt’, ‘log2’]
- max_depth = [2, 3, 4, 5, 6, 7, 8, 9, 10]

The parameters used for the Random Forest model were as follows:

- n_estimators = 700
- max_features = log2
- max_depth = 5

It is worth mentioning that in this model we used the feature_importances function which allows us to see which features are more relevant for the model according to the dataset, as a result we obtained weighted_grade as the most important feature, this can be seen in Figure 6.

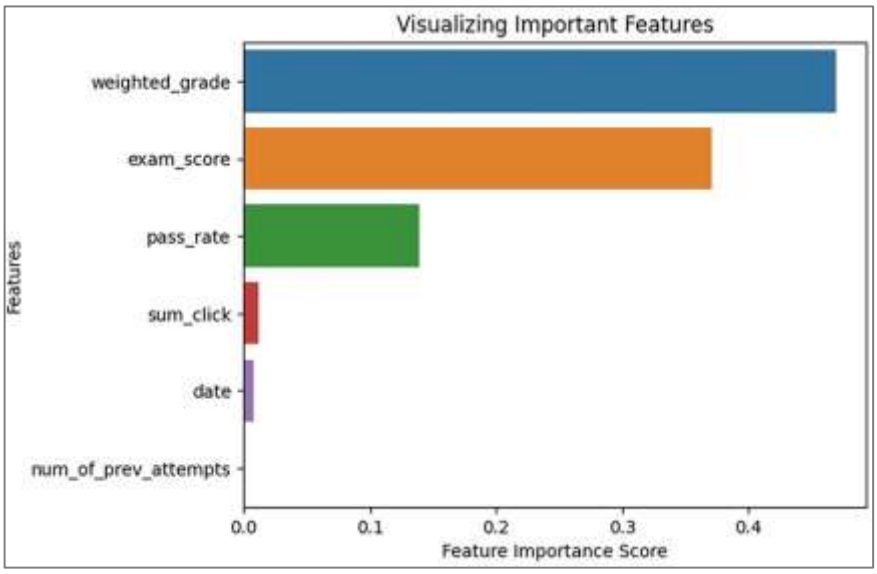

Fig. 6. Important features.

1.3.2 Ethics statement

1.3.2.1.1 Discrimination criteria

Given the dataset variables used in the project, we handle some student personal information, such as gender. We use this variable neither for training the neural network nor for predictions. We believe that using this information will be unethical because it can generate discrimination regarding which gender is more “intelligent” or has greater capacities. We do not consider this relevant because a small section of the population does not appropriately represent all women or men worldwide. Furthermore, we do not want to create controversy in the classroom by implying that one group has better abilities than another; we are only trying to predict student performance in the course and not the abilities that a person may have.

1.3.2.2 Data exclusion

In Section 3.1.4, we mentioned some filters that will be applied to the dataset. One of these filters removes data from students who did not take some of the course assessments because they may be absent due to several reasons, such as illnesses, economic issues, and personal issues, which may have influenced the student's performance in the course. We believe that it would be unethical to deem a student inefficient when they may be experiencing circumstances that may prevent them from attending class and affect their academic performance.

1.3.2.3 Data manipulation for unethical purposes

The ethics model used in this project secures information provided by users because this information shall not be retained or used for unethical purposes. Given the number of cases where information is leaked with malicious intent, personal data was not required for the development of the project and was not used at any stage of the project.

1.4 Results

This section presents the results from the proposed process as explained in the Contribution section. We must remember that the proposed algorithm is an artificial neural network, which predicts student academic performance in a given course.

The confusion matrix (Figure 7) of the model after training is shown below, where it is observed that out of a total of 291 passing students, 274 were predicted correctly, and out of a total of 210 failing students, 196 were predicted correctly.

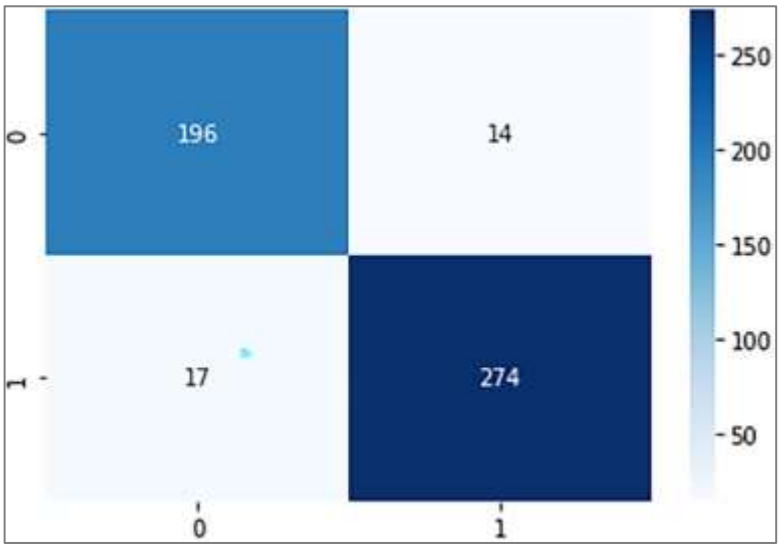

Fig. 7. Confusion matrix

Table 3 shows the results of the confusion matrices and the accuracy of the proposed and developed models, the compared models were trained and validated with the same data set.

Table 3

Results of the confusion matrices

Models	True Positives	False Positive	True Negatives	False Negatives	Accuracy
ANN (Proposed model)	274	17	196	14	93.81%
Random Forest	288	21	177	15	92.81%
SVM	294	15	107	85	80.03%
Naïve Bayes	284	25	178	14	92.21%

1.4.1 Result interpretation

To track the evaluation of the training and validation process of the artificial neural network, the Accuracy and Loss metrics were used in each epoch.

Figure 8 illustrates the Accuracy and Loss recorded for each epoch. On one hand, Training Accuracy, in orange, reaches a certain percentage where it remains stable, and Validation Accuracy, in red, follows the same behavior as Validation Accuracy. On the other hand, Loss Training and Validation follow a similar decreasing pattern until reaching a minimum percentage, where they stabilize.

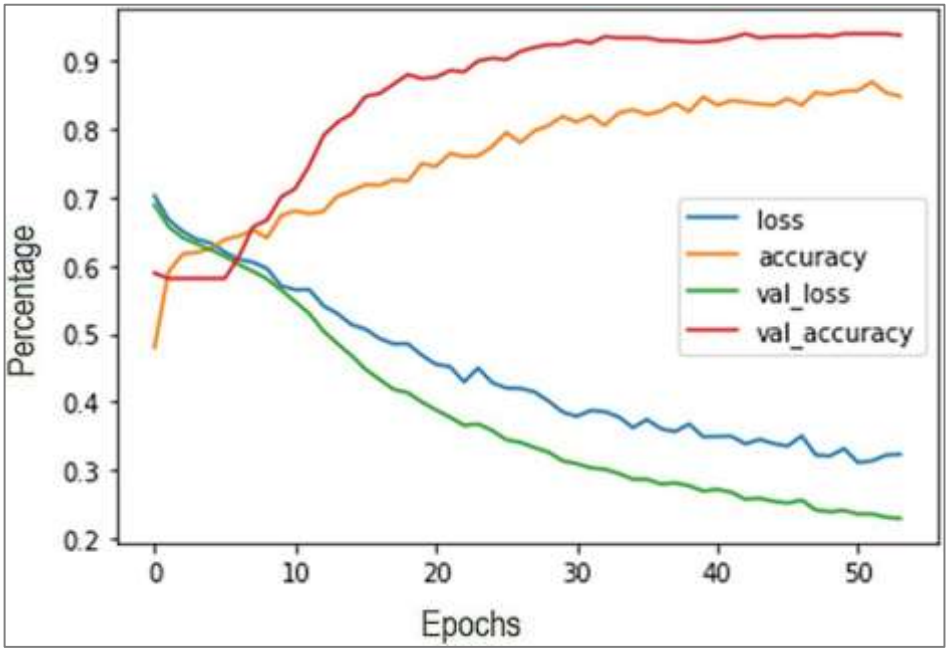

Fig. 8. Accuracy and loss recorded for each epoch.

A total of 100 epochs were required for training in the hyperparameters. However, to avoid overfitting, EarlyStopping was used, thereby cutting off at 54 epochs upon detecting that the accuracy percentage remained constant.

The model achieved 93.81% accuracy, which indicates that the prediction was correct for the percentage of cases validated. Likewise, 94.15% precision was obtained. This measures the quality of the model, indicates the number of students who actually passed the course, and predicts that they would pass the course. In other words, it is the percentage of students correctly identified by the model as passing from the total number of students identified by the model as passing. The model achieved 95.13% recall, which indicates the number of approved students that the model identified, that is, the percentage of students correctly identified as approved from the total number of approved students. Finally, the model obtained an 94.64% F1-score, which is the combination of the accuracy and recall metrics as a single value.

1.4.2 Discussion

This section compares the performance of the proposed model with the models that had the best performance according to the literature. The most relevant studies are specified in Table 1. Three machine learning models were implemented, which were trained with the same set of datasets with which the neural network was trained in order to compare them.

According to C. I. P. Benablo, E. T. Sarte, J. M. D. Dormido and T. Palaoag [20], who evaluated the SVM and KNN techniques, the SVM model provides 98% accuracy due to the type of model used to predict performance as a function of the input data. C. I. P. Benablo, E. T. Sarte, J. M. D. Dormido and T. Palaoag [20] also details three other models used in which a variation in the instances of the dataset is seen. The higher the number of instances, the higher the accuracy, sometimes reaching 100%. After the implementation of our own SVM model, and the comparison with the proposed neural network model, we can observe which neural network is better both in accuracy and in the test results shown in the Table 3.

After comparing with the M. V. Amazona and A. A. Hernandez [15] proposal, three different models were used: Naïve Bayes, deep learning and decision tree, with an accuracy of 96, 98 and 93%, respectively. We opted to implement our own Naïve Bayes model to train it on our dataset for comparative purposes. The results of the Naïve Bayes model were the lowest in the comparisons performed, as can be seen in Table 3.

In his research, C. Rodrígiez-Hernández, M. Musso, E. Kyndt and E. Cascallar [21] implemented an artificial neural network to predict the academic performance of senior students. The research notes that they used information from a sample of 162,030 students and that the trained model provided an accuracy of 82%. However, they also used personal information about the students (e.g., their socioeconomic status, household characteristics, personal background, the types of schools they had attended, and the salary of working students) to train the model. In this case, to perform the comparison of the models with the same data set, the C. Rodrígiez-Hernández, M. Musso, E. Kyndt and E. Cascallar [21] model

would have to be modified, so a correct comparison cannot be performed; however, we can observe that 8 of the 10 attributes used as input data, are personal data, this means that if we remove the personal data from their model, the model would suffer a decay in performance. M. Yağcı [25] implemented several models to predict students' academic performance.

The models used were random forest, neural network, SVM, logistic regression, Naïve Bayes and KNN, where the neural network provided the best results with an accuracy of 86.30%. M. Yağcı [25] did not use students' personal data. They only used their midterm and final exam grades, career path and academic department as information. The 86% accuracy obtained by this research study is the highest accuracy rate. Thus, a proprietary Random Forest model was implemented, trained with the same data set with which our proposed model was trained. The result of the Random Forest model was the best performing among the implemented models, this can be seen in Table 3.

The research of B. K. Yousafzai *et al.* [26] compares the existing models, such as RNN, CNN, LSTM, SVM and BiLSTM, with the model proposed by the authors, which achieved a better accuracy of 90.16%. B. K. Yousafzai *et al.* [26] uses personal data, such as age, gender, address, parents' occupation, type of school and parents' educational level, to train the model. The dataset used for training is mainly composed of personal data. Similar case to C. Rodrígiez-Hernández, M. Musso, E. Kyndt and E. Cascallar [21], due to the architecture developed by the authors and the database used, it is not possible to perform an equivalent comparison; however, we can observe that from the model proposed by B. K. Yousafzai *et al.* [26], 15 of the 33 entries for his model are personal information of the students, such as: age, location, love relationship, father's jobs, mother's jobs, etc. which means that if we were to remove those 15 entries, the model would suffer a loss of performance.

Comparing our model with the literature, we can observe that personal data significantly affect the model results, regardless of the type of model used. For this reason, and due to the increasing vulnerability of personal data evidenced by data breaches and leaks, we propose a student performance prediction tool that does not expose sensitive student and teacher information and yields better results than the observed literature.

1.5 Conclusion

In this research, an artificial neural network model was proposed to predict students' academic performance in their courses, which obtained an accuracy of 93.81%, without exposing their personal data.

Low student performance such as student dropout in courses is a problem, which affects both educational institutions and students. Because of this, we developed a model superior to all the proposed literature that can predict whether a student will pass or fail a subject in the middle of the course. This information will be useful for the competent authorities of student achievement to take measures to avoid withdrawal and underachievement.

It was necessary to propose some ethical metrics for the development of our model. Unlike the literature, we proposed a neural network model that did not use personal input data and

preserved students' privacy. This gives rise to a new area of study in which personal data is avoided, as many industries today constantly report personal data breach and leakage.

1.6 Future work

In future work, more data regarding student interaction with digital platforms can be used, since as a consequence of the COVID-19 pandemic, academic learning has become largely virtualized and much more information will be available.

It is worth noting that this process of predicting student performance under ethical standards can be automated on university platforms, thus facilitating access to information for student affairs officers.

Data availability statement

Publicly available datasets were analyzed in this study. This data can be found at: https://analyse.kmi.open.ac.uk/open_dataset.

Author contributions

HC, BC-A, and SC-R performed the predictive analysis. CR, HC, and BC-A wrote the first draft of the manuscript. CR, HC, BC-A, SC-R, and JA-R wrote sections of the manuscript. All authors contributed to the manuscript revision, read, and approved the submitted version.

Acknowledgments

The authors would like to thank the Research Directorate of the Universidad Peruana de Ciencias Aplicadas for the support provided to carry out this research work through the UPC-EXPOST-2023-1 incentive.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

2. CONCLUSIONES

En esta investigación se logró predecir la aprobación de una materia universitaria, mediante un modelo de red neuronal artificial y los datos extraídos de la interacción del alumno con el aula virtual sin usar datos personales; se logró una precisión del 93,81%.

Esta información será de utilidad para las autoridades competentes en materia de rendimiento estudiantil, ya que podrán evitar el bajo rendimiento académico y el abandono de los estudios de los alumnos.

Fue necesario proponer algunas métricas éticas para el desarrollo del modelo, ya que, a diferencia de la literatura, se ha desarrollado un modelo de red neuronal que no requiere datos personales y preserva la privacidad de los estudiantes. Esto da lugar a una nueva área de estudio en la que se evita la exposición de datos personales, debido a que muchas industrias informan constantemente sobre violaciones y filtraciones de estos.

3. REFERENCIAS BIBLIOGRAFICAS

[1] Chavez H, Chavez-Arias B, Contreras-Rosas S, Alvarez-Rodríguez JM and Raymundo C. "Artificial neural network model to predict student performance using nonpersonal information". *Front. Educ.* 8:1106679, 2023. doi: 10.3389/feduc.2023.1106679

[2] Tejedor, F., and García-Valcárcel, A. (2007). Causas del bajo rendimiento del estudiante universitario (en opinión de los profesores y alumnos). Propuestas de mejora en el marco del EEES. Revista de Educación 342, 443–473. Available at: https://dialnet.unirioja.es/servlet/articulo?codigo=2254218 (Accessed May 6, 2021).

[3] Vicerrectorado Académico (2017). Cuando lo que se sabe nos dice cuánto no se sabe— Vicerrectorado Académico. Available at: https://vicerrectorado.pucp.edu.pe/academico/noticias/cuando-lo-que-se-sabe-nos-dice-cuanto-no-se-sabe/ (Accessed May 3, 2021).

[4] Viale, H. (2014). Una aproximación teórica a la deserción estudiantil. Revista Digital de Investigación en Docencia Universitaria. 8, 59–76. doi: 10.19083/ridu.8.366

[5] SINEACE (2016). Modelo de Acreditación para Programas de Estudios de Educación Superior Universitaria, Lima, Perú.

[6] Liao, C., and Wu, J. (2022). Deploying multimodal learning analytics models to explore the impact of digital distraction and peer learning on student performance. Comput. Educ. 190:104599. doi: 10.1016/j.compedu.2022.104599

[7] Blanco, J., Lovelle, S., Fernandez, R., and Perez, E. (2016). Predicción de resultados académicos de estudiantes de informática mediante el uso de redes neuronales. Ingeniare. Revista chilena de ingeniería 24, 715–727. doi: 10.4067/S0718-33052016000400015

[8] Rincon-Flores, E. G., López-Camacho, E., Mena, J., and López, O. O. (2020). "Predicting academic performance with artificial intelligence (AI), a new tool for teachers and students." in 2020 IEEE Global Engineering Education Conference (EDUCON). Porto, Portugal. 1049–1054. doi: 10.1109/EDUCON45650.2020.9125141

[9] Kuzilek, J., Hlosta, M., and Zdrahal, Z. (2017). Open University learning analytics dataset. Sci. Data 4:170171. doi: 10.1038/sdata.2017.171

[10] Kumar, V. U., Krishna, A., Neelakanteswara, P., and Basha, C. Z. (2020). "Advanced

prediction of performance of a student in an university using machine learning techniques." in 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC). Coimbatore, India. 121–126. doi: 10.1109/ICESC48915.2020.9155557

[11] Jayaprakash, S., Krishnan, S., and Jaiganesh, J. (2020). "Predicting students academic performance using an improved random Forest classifier." in 2020 International Conference on Emerging Smart Computing and Informatics (ESCI). Pune, India. 238–243. doi: 10.1109/ESCI48226.2020.9167547

[12] Daud, A., Aljohani, N. R., Abbasi, R. A., Lytras, M. D., Abbas, F., and Alowibdi, J. S. (2017). "Predicting student performance using advanced learning analytics." in Proceedings of the 26th International Conference on World Wide Web Companion—WWW '17 Companion. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee. 415–421. doi: 10.1145/3041021.3054164

[13] Widyaningsih, Y., Fitriani, N., and Sarwinda, D. (2019). "A semi-supervised learning approach for predicting Student's performance: first-year students case study." in 2019 12th International Conference on Information & Communication Technology and System (ICTS). Surabaya, Indonesia. 291–295. doi: 10.1109/ICTS.2019.8850950

[14] Devasia, T., Vinushree, T. P., and Hegde, V. (2016). "Prediction of students' performance using educational data mining." in 2016 International Conference on Data Mining and Advanced Computing (SAPIENCE). Ernakulam, India. 91–95. doi: 10.1109/SAPIENCE.2016.7684167

[15] Amazona, M. V., and Hernandez, A. A. (2019). "Modelling student performance using data mining techniques." in Proceedings of the 2019 5th International Conference on Computing and Data Engineering—ICCDE' 19. New York, NY, USA: Association for Computing Machinery. 36–40. doi: 10.1145/3330530.3330544

[16] Páez, A., and Guzmán, D. (2018). Diseño de un modelo para automatizar la predicción del rendimiento académico en estudiantes del IPN. Revista Iberoamericana para la Investigación y el Desarrollo Educativo 8, 246–266. doi: 10.23913/ride.v8i16.340

[17] Ma, X., Yang, Y., and Zhou, Z. (2018). "Using machine learning algorithm to predict student pass rates in online education." in Proceedings of the 3rd International Conference on Multimedia Systems and Signal Processing - ICMSSP '18. New York, NY, USA: Association for Computing Machinery. 203–207. doi: 10.1145/3220162.3220188

[18] Castrillón, O., Sarache, W., and Ruiz, S. (2020). Prediction of academic performance using artificial intelligence techniques. From Univ. 13, 93–102. doi: 10.4067/S0718-50062020000100093

[19] Hasan, H. M. R., Rabby, A. S. A., Islam, M. T., and Hossain, S. A. (2019). "Machine learning algorithm for Student's performance prediction." 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT). Kanpur, India. 1–7. doi: 10.1109/ICCCNT45670.2019.8944629

[20] Benablo, C. I. P., Sarte, E. T., Dormido, J. M. D., and Palaoag, T. (2018). Higher education Student's academic performance analysis through predictive analytics in Proceedings of the 2018 7th International Conference on Software and Computer Applications—ICSCA 2018. New York, NY, USA: Association for Computing Machinery. 238–242. doi: 10.1145/3185089.3185102

[21] Rodrígiez-Hernández, C., Musso, M., Kyndt, E., and Cascallar, E. (2021). Artificial neural networks in academic performance prediction: systematic implementation and predictor evaluation. Comput. Educ. Artif. Intell. 2:100018. doi: 10.1016/j.caeai.2021.100018

[22] Sekeroglu, B., Dimililer, K., and Tuncal, K. (2019). "Student performance prediction and classification using machine learning algorithms." in Proceedings of the 2019 8th International Conference on Educational and Information Technology—ICEIT 2019. New York, NY, USA: Association for Computing Machinery. 7–11. doi: 10.1145/3318396.3318419

[23] Liao, S. N., Zingaro, D., Thai, K., Alvarado, C., Griswold, W. G., and Porter, L. (2019). A robust machine learning technique to predict low-performing students. ACM Trans. Comput. Educ. 19, 1–19. doi: 10.1145/3277569

[24] Alaieri, F., and Vellino, A. (2017). “A decision making model for ethical (Ro)bots.” in IEE International Symposium on Robotics and Intelligent Sensors (IRIS), Ottawa, ON, Canada. 203–207. doi: 10.1109/IRIS.2017.8250122

[25] Yağcı, M. (2022). Educational data mining: prediction of students’ academic performance using machine learning algorithms. Smart Learn. Environ. 9, 9:11. doi: 10.1186/s40561-022-00192-z

[26] Yousafzai, B. K., Khan, S. A., Rahman, T., Khan, I., Ullah, I., Ur Rehman, A., et al. (2021). Student-Performulator: student academic performance using hybrid deep neural network. Sustainability 13:9775. doi: 10.3390/su13179775

[27] Heaton, J. (2009). Introduction to Neural Networks with Java, 2nd. Chesterfield, MO: Heaton Research, 2009, p. 129.

[28] Instituto Colombiano para la Evaluación de la Educación – ICFES. DataIcfes. Available: https://www2.icfes.gov.co/data-icfes (Accessed Jun 6, 2021).

[29] Department of Information Technology. Aurora State College of Technology. Available: https://www.ascot.edu.ph/departments/department-of-information-technology/ (Accessed Jun 6, 2021).

[30] CI Machine Learning Repository. Student Performance Data Set. Available: https://archive.ics.uci.edu/ml/datasets/student+performance (Accessed Jun 6, 2021).

4. ANEXOS

ANEXO 1: Información de gestión del proyecto

A11. Objetivo general

Diseñar e implementar un modelo de red neuronal artificial en un ambiente de cloud computing que permita predecir la aprobación de un curso antes de completar el 50% del mismo, sin utilizar información personal de los estudiantes.

A12. Objetivos específicos

- OE1: Analizar, evaluar y comparar modelos de inteligencia artificial propuestos en la literatura para la predicción del rendimiento académico, haciendo énfasis en los implementados en un entorno de cloud computing y considerando las ventajas en términos de manejo y análisis de big data.
- OE2: Diseñar un modelo de inteligencia artificial que prediga el rendimiento académico sin utilizar información personal de los estudiantes, optimizado para su alojamiento y operación en plataformas de cloud computing.
- OE3: Validar la funcionalidad del modelo usando métricas y métodos estadísticos en un ambiente de cloud computing.
- OE4: Presentar los hallazgos y limitaciones encontrados en la presente investigación, con un enfoque en infraestructura de cloud multizona para un acceso global y rápido.
- OE5: Publicar la investigación en formato paper en un journal indexado por Scopus.

A13. Cronograma proyectado

Tabla 4

Plan de actividades

Fase del proyecto	Fecha de cierre Estimada	Entregables incluidos	Prioridad
Inicio y Planificación	Semana 3	• Project Charter. • Cronograma. • Plan de trabajo.	Alta
Ejecución: Toma de información	Semana 6	• Investigación y revisión de la literatura.	Alta
	Semana 9	• Investigación de modelos implementados.	Alta
Ejecución: Diseño del modelo	Semana 13	• Implementación de diferentes modelos	Alta
	Semana 15	• Comparación y optimización del modelo seleccionado.	Alta
Ejecución: Presentación y exposición	Semana 19	• Exposición de la memoria parcial del proyecto.	Alta
Ejecución: Validación del modelo	Semana 23	• Validación del modelo	Alta
Cierre: Presentación y exposición	Semana 30	• Memoria del proyecto	Alta
Cierre: Publicación de journal paper	Semana 34	• Redacción del paper en su versión journal.	Alta
	Semana 39	• Redacción del journal paper y corrección de observaciones indicadas.	Alta
	Semana 45	• Carta de aprobación del journal por parte de la revista.	Alta

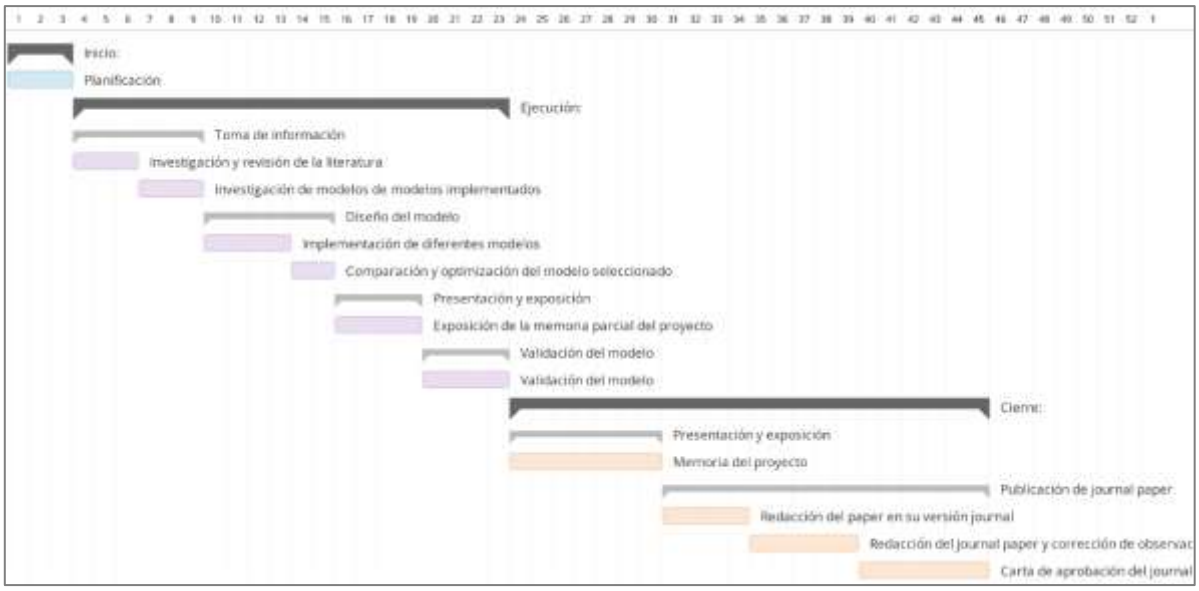

Fig. 9. Cronograma de proyecto

A14. Alcance

El presente trabajo incluye el desarrollo y optimización de un modelo de aprendizaje automático implementado en el lenguaje Python capaz de pronosticar la aprobación de un curso sin usar información personal del alumno. Esta predicción se logra por medio de la interacción del alumno con el aula virtual y sus antecedentes.

ANEXO 2: Despliegue del modelo

En el presente anexo, se detalla el proceso de despliegue de la red neuronal propuesta, articulándolo a través de diferentes componentes y servicios. La Figura 10 muestra la arquitectura general, destacando la interconexión de servidores alojados en la nube (cloud-based servers) y las bases de datos esenciales para el funcionamiento del sistema. El termino en inglés "cloud-based servers" hace referencia a los servidores que operan en centros de datos administrados por proveedores de servicios en la nube, como Google Cloud, Amazon Web Services (AWS), Microsoft Azure, entre otros. En este caso se usará Google Cloud.

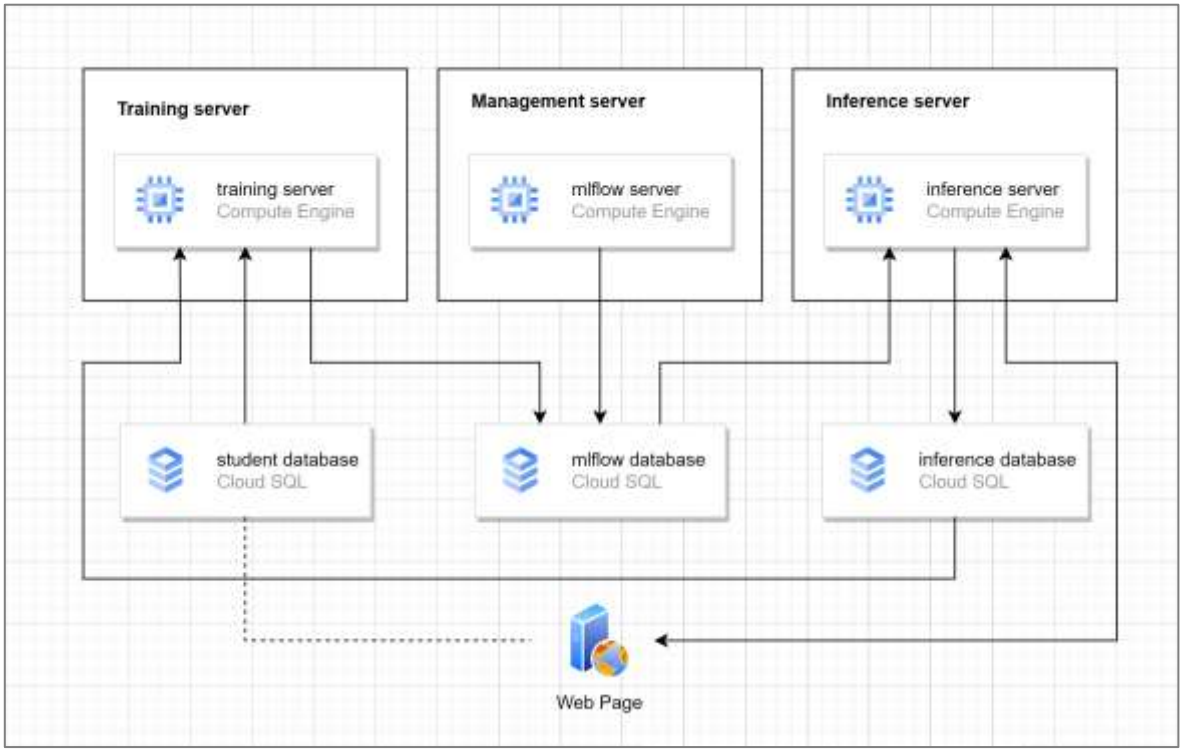

Fig. 10. Arquitectura propuesta

El uso de cloud computing brinda una plataforma ágil y escalable, lo que facilita el entrenamiento y la implementación de redes neuronales a gran escala. Además, optar por soluciones como las ofrecidas por Google asegura una alta disponibilidad y según su Acuerdo de Nivel de Servicio (SLA por sus siglas en ingles), Compute Engine garantiza un 99.99% de tiempo de actividad mensual para instancias en múltiples zonas [1]. Esta alta disponibilidad, junto con la capacidad de crear multizonas, garantiza la estabilidad y la accesibilidad mundial de la operación de los servicios.

Un aspecto crucial en este despliegue es la incorporación de MLflow [2], una herramienta que permite el seguimiento y gestión óptima de los modelos de machine learning ya que ofrece un registro centralizado de modelos, garantiza la reproducibilidad mediante la encapsulación del entorno, permite integrarse con plataformas líderes en machine learning y facilita el despliegue de los modelos. Su diseño intuitivo y escalable permite una gestión coherente a lo largo del ciclo de vida del modelo, desde la experimentación hasta la producción, haciendo que la gestión y comparación de modelos sea eficiente y visual.

Para la implementación de MLflow (como se ve en la Figura 10), se instaló una base de datos para gestionar y almacenar los modelos y experimentos. Esta base de datos puede ser local o en la nube, dependiendo de las preferencias y necesidades de los clientes.

Adicionalmente, se cuenta con un servidor dedicado exclusivamente para gestionar las inferencias y, en este caso, se instaló un servidor de base de datos para las inferencias de forma separada por diferentes razones. Una de las ventajas de estar alojado de forma independiente, es que ofrece una accesibilidad y escalabilidad optimizada, permitiendo adaptarse a la creciente demanda y reentrenar el modelo desde diferentes ubicaciones sin restricciones. Además, esta separación actúa como una barrera de seguridad adicional en caso de que el servidor de inferencias sea comprometido, ya que los datos esenciales quedan protegidos en un ambiente aislado. El liberar al servidor de inferencias de consulta de base de datos, permite que pueda enfocarse y optimizar su tarea principal: la realización de inferencias rápidas y precisas. Finalmente, la integridad y resiliencia de los datos se fortalece dado que es posible incorporar respaldos y recuperaciones eficientes protegiendo la información valiosa.

En paralelo, otro servidor se encarga del alojamiento de la interfaz web. Esta plataforma proporciona a los usuarios un medio intuitivo para ingresar datos y, en consecuencia, recibir las predicciones pertinentes.

Para culminar, es imperativo destacar que todas las inferencias generadas son registradas en un servidor SQL ya que facilita análisis subsecuentes y, además, contribuye a una retroalimentación constante del modelo, potenciando su precisión y efectividad.

En el siguiente segmento, se detalla la funcionalidad de cada servidor, tal como se ilustra en la Figura 10.

A21. Training server

Descripción Funcional:

El servidor de entrenamiento cumple un papel esencial en el ciclo de vida de modelos de inteligencia artificial desde su desarrollo hasta su perfeccionamiento, garantizando que los modelos sean precisos y efectivos. Las tareas principales ejecutadas por este servidor incluyen:

- Ingeniería de Datos: El servidor se encarga de la extracción y procesamiento de la información de la base de datos denominada "student database". Su objetivo principal es la preparación de datos relacionados con los alumnos para entrenar el modelo. Este proceso engloba actividades esenciales como la integración, limpieza y transformación de los datos, la estandarización de formatos, así como un exhaustivo análisis orientado a la identificación y selección de las características más relevantes para el entrenamiento del modelo.
- Desarrollo y mejora del modelo: En esta fase, se lleva a cabo la implementación de diversos algoritmos orientados al desarrollo de modelos, ya sean redes neuronales o técnicas de aprendizaje automático. Se realiza una optimización de los parámetros de los modelos con el propósito de alcanzar métricas óptimas en las predicciones.

Descripción Técnica:

El training server no opera de manera continua. Se activa específicamente durante las fases de entrenamiento y optimización, lo cual maximiza la eficiencia y aprovechamiento de recursos. Aunque es una herramienta poderosa, no es la única opción para el entrenamiento de modelos:

- Configuración: El servidor no está en funcionamiento permanente. Esta configuración puntual ayuda a economizar y gestionar eficientemente los recursos.
- Alternativas de entrenamiento: Los desarrolladores y científicos de datos pueden optar por entrenar modelos en sus computadoras personales, conectándose directamente al servidor SQL. Esta opción brinda flexibilidad, permitiendo que el entrenamiento se adapte a diferentes contextos y necesidades.

A22. Management server

Descripción Funcional:

El servidor de gestión incorpora MLflow, una plataforma de código abierto reconocida por su capacidad en el seguimiento y administración de experimentos relacionados con el aprendizaje automático y redes neuronales. La Figura 11 ilustra la manera en que los modelos son administrados a través de MLflow.

MLflow permite el registro de modelos provenientes de diversos entornos de ejecución, incluyendo Jupyter Notebooks [3], Google Colab [4], y otros entornos de desarrollo basados en interfaces de desarrollo integradas (IDE). Es relevante destacar su versatilidad, pues es compatible con una amplia gama de frameworks de aprendizaje automático, tales como PyTorch [5], TensorFlow [6], XGBoost [7], scikit-learn [8], entre otros.

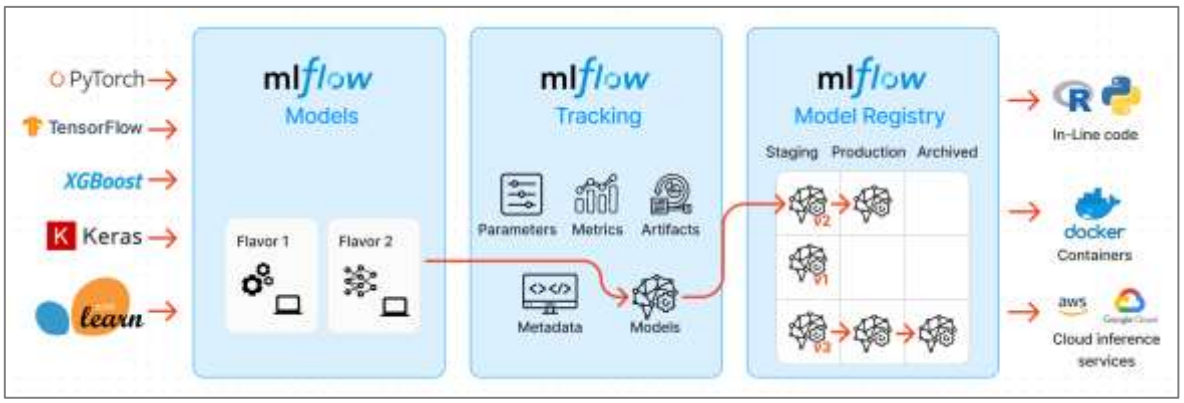

Fig. 11. Arquitectura de MLflow

Descripción Técnica:

- Infraestructura y Especificaciones: La instalación de MLflow se llevó a cabo en un servidor alojado en Google Cloud [9], específicamente en Compute Engine [10]. Tal como se detalla en la Figura 12, este servidor posee un CPU virtual y está equipado con 4GB de memoria RAM y 20GB de capacidad de almacenamiento. La elección de la capacidad de almacenamiento y procesamiento para el despliegue de MLflow fue una decisión basada en una combinación de factores. Primeramente, MLflow, siendo una herramienta de seguimiento y administración, no requiere una gran cantidad de recursos para su funcionamiento. Por lo general, MLflow utiliza recursos para registrar, rastrear y administrar experimentos, en comparación con los procesos computacionalmente intensos como el entrenamiento de modelos. Además, el

sistema operativo y las dependencias básicas de MLflow no requieren una gran cantidad de memoria y espacio de almacenamiento. De hecho, los requerimientos de MLflow son mínimos, y 4GB de RAM es más que suficiente para manejar múltiples experimentos, modelos y métricas simultáneamente.

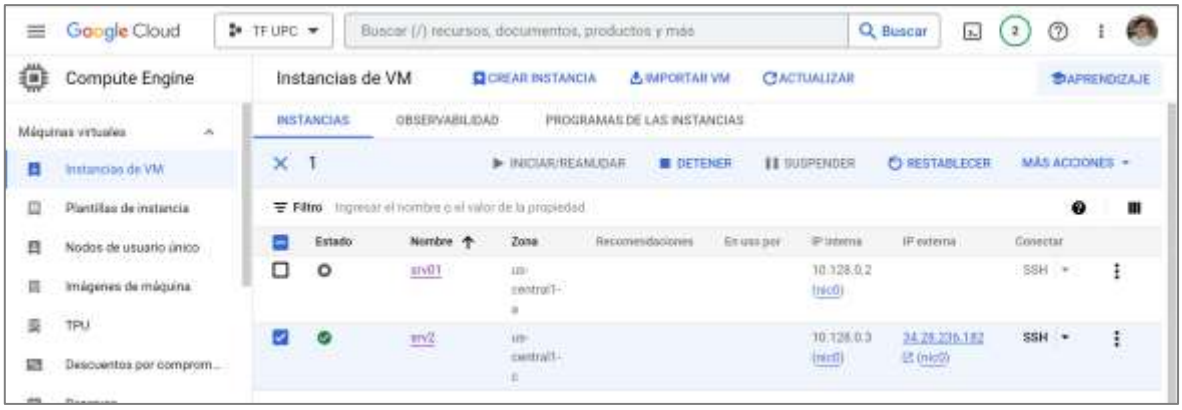

Fig. 12. Compute Engine – Google Cloud

- Acceso y Operación: El acceso al mismo se realiza a través de una conexión SSH. Esta infraestructura facilita la puesta en línea de la plataforma MLflow, la cual desempeña un papel importante en la gestión y versionado de los modelos. En la Figura 13 se muestra la interfaz web de MLflow con los diversos experimentos realizados (DP3, DP2, SVM, Random Forest, etc.), los nombres se repiten ya que cada uno de ellos tienen diferentes configuraciones (versiones) en la cual, el administrador de la plataforma se encarga de comparar las métricas y escoger el modelo más adecuado para llevarlo a la siguiente etapa (MLflow Model Registry, ver Figura 11).

  Luego de una evaluación de las métricas y el rendimiento de los modelos mediante MLflow Tracking, se avanza a la siguiente fase: el MLflow Model Registry, en este proceso se administra las diferentes versiones y el modelo seleccionado es llevado a producción para que posteriormente, sea accesible para su consumo desde el servidor de inferencia como se ilustra en la Figura 14, en este caso se puso a producción NN y N2, que hacen referencia a “Neuronal Network”.

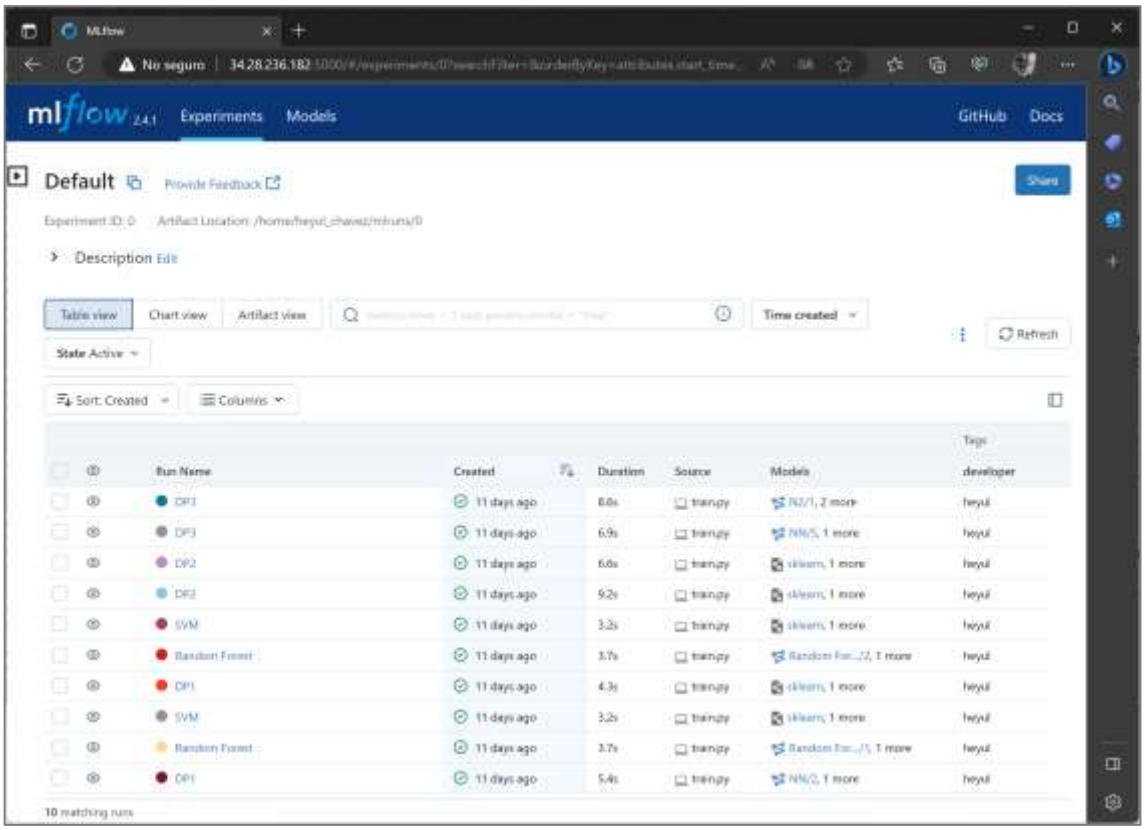

Fig. 13. Plataforma de MLflow

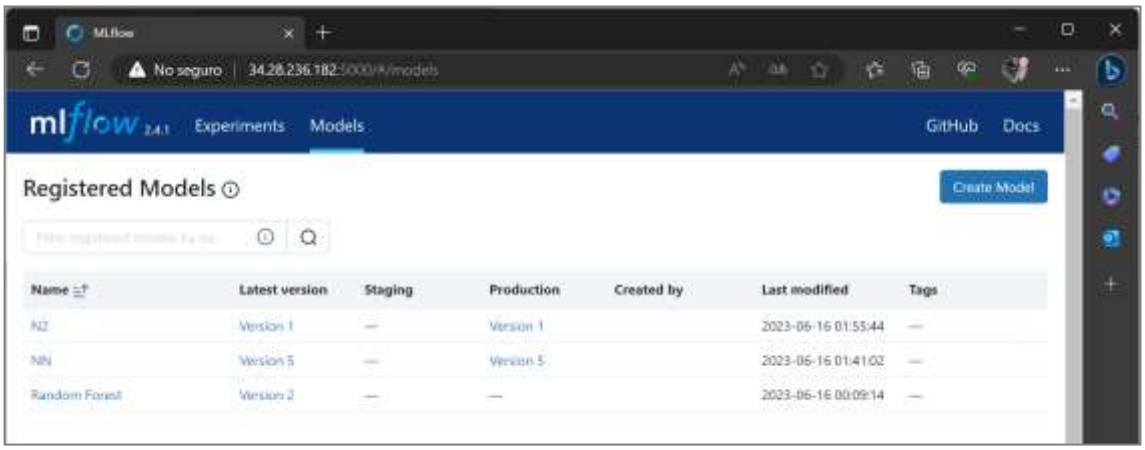

Fig. 14. MLflow Model Registry

A23. Inference Server

Descripción Funcional:

El servidor de inferencias actúa como el puente entre el usuario y los modelos de predicción. Su función es transformar los datos suministrados por el usuario en predicciones, y además proporciona los datos para la retroalimentación de los modelos entrenados. Este servidor permite que las complejas operaciones de inteligencia artificial se traduzcan en respuestas claras y directas para el usuario final.

Descripción Técnica:

El servidor de inferencias se aloja en Compute Engine, una herramienta de Google Cloud. Esta infraestructura está específicamente diseñada para administrar inferencias a través de Flask [11]. Su función principal es recibir datos del usuario de la página web (recuadro rojo en Figura 15) y seguidamente ejecutar la inferencia utilizando el modelo en producción alojado en el servidor de base de datos de mlflow, para obtener la predicción (recuadro azul) en este caso se obtiene "0" lo cual significa que el alumno desaprobará el curso. Este proceso tuvo una duración aproximada de 0.03 segundos como se muestra en el recuadro amarillo. Posteriormente, la predicción es enviada y almacenada en el servidor de base de datos de inferencias (inference database), como se ve en el recuadro verde, y se envía la respuesta al cliente en la página web.

Es relevante mencionar que los resultados derivados de estas inferencias son almacenados en una base de datos para evaluar y optimizar continuamente el rendimiento de los modelos.

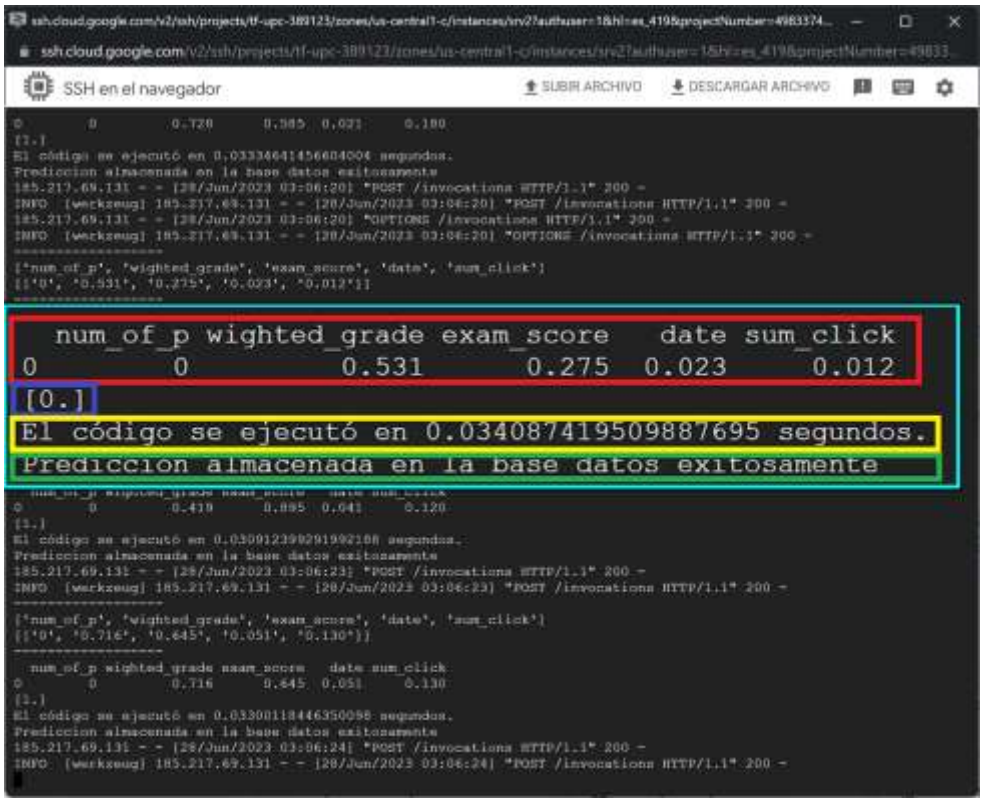

Fig. 15. Salidas del Inference server

A24. Web server

Descripción Funcional:

El servidor web proporciona una plataforma interactiva que conecta a los usuarios con los modelos de predicción a través de una interfaz de página web. Esta conexión facilita la obtención de predicciones en tiempo real, brindando a los usuarios una experiencia práctica y directa del modelo de red neuronal. Como se muestra en la Figura 16, la página principal del sitio web está estructurada en tres segmentos distintos, cada uno de ellos presentando un ejemplo práctico de la implementación y utilidad del modelo propuesto en este estudio.

Descripción Técnica:

Para asegurar la disponibilidad y accesibilidad en línea de la herramienta, se optó por un alojamiento web en forma de hosting compartido. Esta decisión fue tomada considerando las expectativas actuales de tráfico, ya que no se prevé un alto volumen de visitantes para

esta fase de prueba. El servicio técnico subyacente se apoya en un servidor IIS 10 y opera bajo el sistema operativo Windows Server 2019.

Fig. 16. Página web principal

- Vista Estudiantil: Tal como se refleja en la Figura 17, se ha ideado un modelo enfocado en los alumnos. En esta representación de un aula virtual, se ha añadido una función de la predicción de aprobar el curso en la esquina superior derecha. Esta característica puede servir como un impulso motivacional para aquellos estudiantes que muestran un desempeño destacado, y también, para aquellos con pronósticos menos favorables puede actuar como un recordatorio constructivo, incentivándolos a esforzarse más.

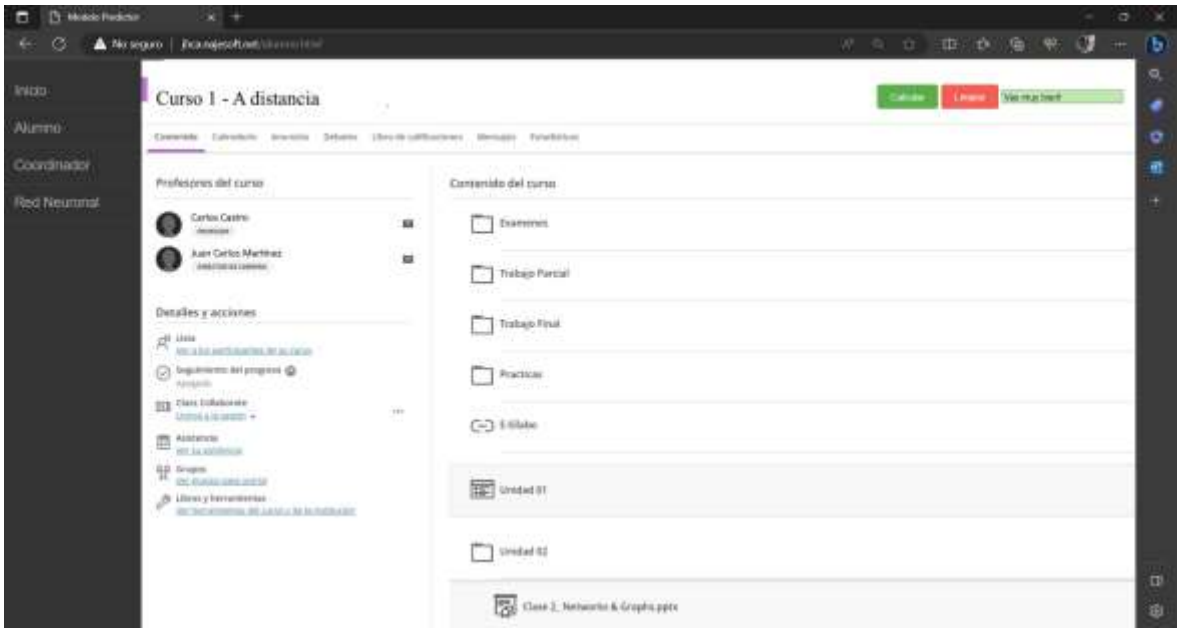

Fig. 17. Vista de alumno

- Vista Coordinador: Esta perspectiva (Figura 18) ofrece herramientas analíticas para un coordinador de curso o cualquier autoridad pertinente de la institución. Desde esta aplicación, es posible seleccionar una sección específica, evaluar el desempeño individual o colectivo, y obtener una representación gráfica que desglosa las predicciones entre estudiantes que aprobarán y los que potencialmente no lo harán. Al elegir una sección, se consulta automáticamente la base de datos para recopilar la lista de estudiantes, y con esos datos obtenidos se generan las predicciones pertinentes.

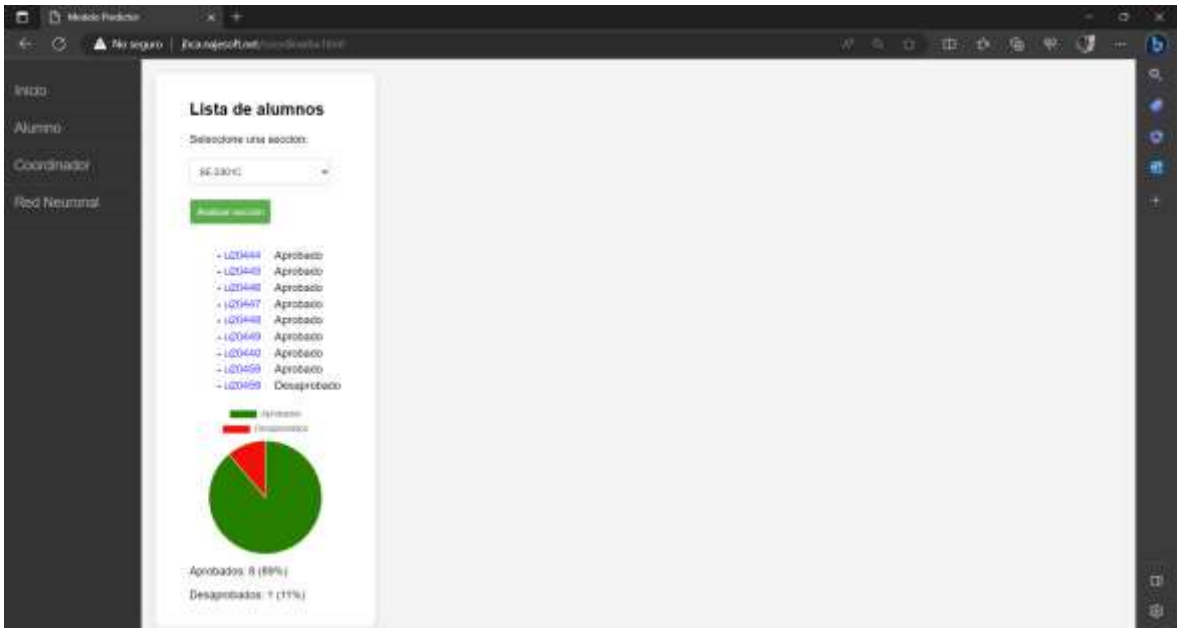

Fig. 18. Vista de coordinador

- Interacción con la Red Neuronal: La última sección (Figura 19) está dedicada a mostrar los parámetros de entrada de la red neuronal. Esto permite a los usuarios experimentar al ingresar diferentes valores y, a cambio, recibir una predicción basada en esos parámetros.

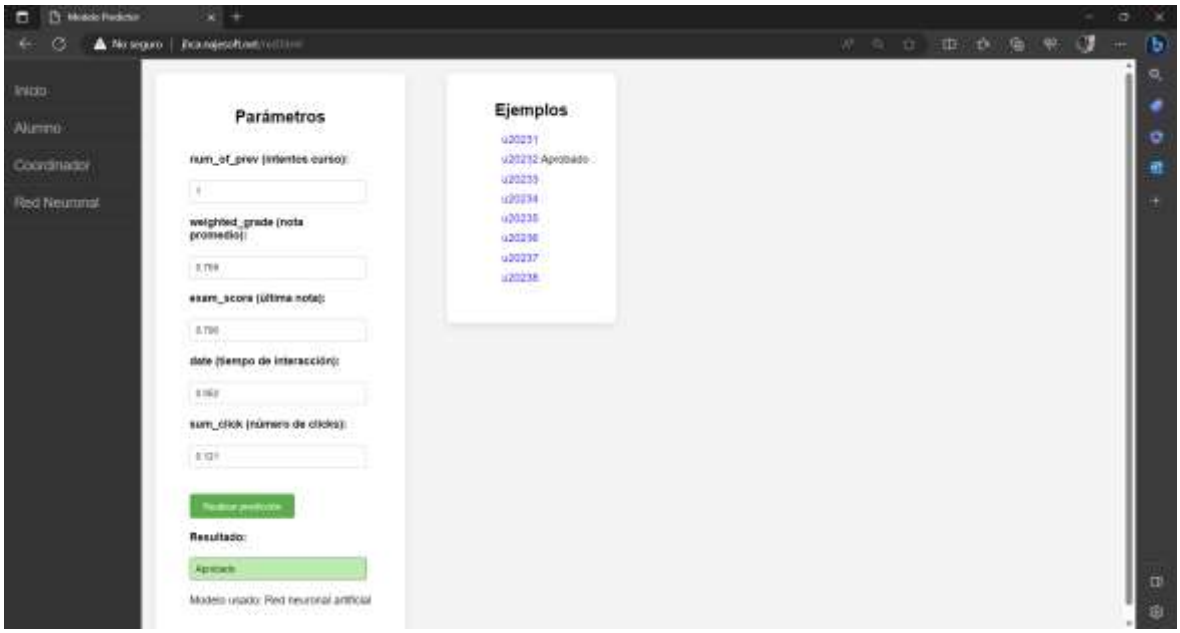

Fig. 19. Interacción con la Red Neuronal

A25. Inference database

Descripción Funcional:

El servidor de base de datos de inferencias es esencial para almacenar y rastrear las inferencias ejecutadas en tiempo real. En la Figura 20, se puede observar que no sólo se almacenan las inferencias, sino que también se registra la fecha y los datos de entrada correspondientes. Este proceso de registro permite análisis posteriores, los cuales son vitales para la optimización y mejora del modelo en uso.

Descripción Técnica:

Para este propósito, se seleccionó el servicio SQL de Google Cloud. La instancia elegida está equipada con PostgreSQL 14, configurada específicamente para el almacenamiento de inferencias. Este diseño de despliegue busca asegurar un sistema que sea escalable y confiable, capaz de gestionar inferencias en tiempo real, registrar datos de entrada y servir como punto de retroalimentación para mejorar el rendimiento del modelo de manera continua.

	fecha	num_of_p	wighted_grade	exam_score	date	sum_click	prediccion
1	2023-06-27 21:04:13.520 -0500	0	0,716	0,645	0,051	0,13	1
2	2023-06-27 21:56:27.976 -0500	1	0,759	0,79	0,052	0,121	1
3	2023-06-27 22:02:09.307 -0500	0	0,613	0,58	0,041	0,131	1
4	2023-06-27 22:02:10.321 -0500	2	0,341	0,52	0,014	0,015	0
5	2023-06-27 22:02:11.123 -0500	2	0,62	0,56	0,074	0,125	1
6	2023-06-27 22:02:11.601 -0500	0	0,728	0,585	0,021	0,18	1
7	2023-06-27 22:02:12.103 -0500	0	0,531	0,275	0,023	0,012	0
8	2023-06-27 22:02:12.857 -0500	0	0,419	0,895	0,041	0,12	1
9	2023-06-27 22:02:13.852 -0500	2	0,62	0,56	0,074	0,125	1
10	2023-06-27 22:02:40.972 -0500	0	0,613	0,58	0,041	0,131	1
11	2023-06-27 22:02:43.425 -0500	0	0,419	0,895	0,041	0,12	1
12	2023-06-27 22:06:15.404 -0500	0	0,613	0,58	0,041	0,131	1
13	2023-06-27 22:06:16.222 -0500	1	0,759	0,79	0,052	0,121	1
14	2023-06-27 22:06:17.707 -0500	2	0,341	0,52	0,014	0,015	0
15	2023-06-27 22:06:19.153 -0500	1	0,62	0,56	0,074	0,125	1
16	2023-06-27 22:06:20.168 -0500	0	0,728	0,585	0,021	0,18	1
17	2023-06-27 22:06:21.723 -0500	0	0,531	0,275	0,023	0,012	0
18	2023-06-27 22:06:23.031 -0500	0	0,419	0,895	0,041	0,12	1
19	2023-06-27 22:06:24.867 -0500	0	0,716	0,645	0,051	0,13	1

Fig. 20. Inference database

Referencias:

[1] Compute Engine Service Level Agreement (SLA) – Google Cloud. Disponible en https://cloud.google.com/compute/sla

[2] MLflow Developers. An Open Source Platform for the Machine Learning Lifecycle [Software]. Disponible en https://mlflow.org/

[3] Project Jupyter. Jupyter Notebooks. Disponible en http://jupyter.org

[4] Google LLC. Google Colaboratory. Disponible en https://colab.research.google.com

[5] PyTorch. [Software]. Disponible en https://pytorch.org

[6] TensorFlow: An end-to-end open source platform for machine learning. [Software]. Disponible en https://www.tensorflow.org

[7] XGBoost Development Team. XGBoost: eXtreme Gradient Boosting. Disponible en https://github.com/dmlc/xgboost

[8] Scikit-learn development team. scikit-learn: Machine Learning in Python. Disponible en https://scikit-learn.org

[9] Google Cloud. Disponible en https://cloud.google.com

[10] Google Cloud. Google Compute Engine. Disponible en https://cloud.google.com/compute

[11] Pallets Projects. Flask. Disponible en https://flask.palletsprojects.com/

ANEXO 3: Formato de bibliometría del artículo de investigación extraído de Scopus

Scopus

EXPORT DATE: 30 October 2023

AUTHORS: Chavez, Heyul (58113365900); Chavez-Arias, Bill (58113366000); Contreras-Rosas, Sebastian (58113366100); Alvarez-Rodríguez, Jose María (55799289900); Raymundo, Carlos (58112192000)

Title: Artificial neural network model to predict student performance using nonpersonal information

(2023) Frontiers in Education, 8, art. no. 1106679, Cited 1 times.

DOI: 10.3389/feduc.2023.1106679

https://www.scopus.com/inward/record.uri?eid=2-s2.0-85148600255&doi=10.3389%2ffeduc.2023.1106679&partnerID=40&md5=4d0a6c170858e0d45590b3c8fbe244c3

PUBLISHER: Frontiers Media S.A.

ISSN: 2504284X

LANGUAGE OF ORIGINAL DOCUMENT: English

ABBREVIATED SOURCE TITLE: Front. Educ.

DOCUMENT TYPE: Article

PUBLICATION STAGE: Final

OPEN ACCESS: All Open Access; Gold Open Access; Green Open Access

SOURCE: Scopus