Revista Innovación y Software
Vol. 2, No. 2, Mes Septiembre-Febrero, 2021
ISSN: 2708-0935
Pág. 75-91
https://revistas.ulasalle.edu.pe/innosoft

Tipo de artículo: Artículos originales
Temática: Inteligencia artificial
Recibido: 10/08/2021 | Aceptado: 15/09/2021 | Publicado: 30/09/2021

Identificadores persistentes:
ARK: ark:/42411/s6/a46
PURL: 42411/s6/a46

Sistema para proponer la nota final de los estudiantes mediante Redes Neuronales

System to propose the final grade of the students through Neural Networks

Kleber Ernesto Baldarrago Salas [1], Erika Cayllahua Chicaña [2], Fanny Lorena Lorenzo Quilla[3*], Maria Quijia Alvarez [4]

[1] Universidad Nacional de San Agustín, Arequipa, Perú. kbaldarrago@unsa.edu.pe

[2] Universidad Nacional de San Agustín, Arequipa, Perú. ecayllahua@unsa.edu.pe

[3] Universidad Nacional de San Agustín, Arequipa, Perú. florenzo@unsa.edu.pe

[4] Universidad Nacional de San Agustín, Arequipa, Perú. mquijia@unsa.edu.pe

[*] Autor para correspondencia: florenzo@unsa.edu.pe

Resumen

Debido al problema recurrente presentado en los alumnos en lo que se refiere a su desempeño académico, se desarrolló una aplicación de redes neuronales con el objetivo de ayudar al docente, ya que esta es capaz de dar resultados de las notas finales de los alumnos y ayudará al docente a comprender el porqué de los resultados, puesto que esta red neuronal toma en cuenta diferentes factores que conlleva al alumno a tener una nota aprobatoria o desaprobatoria. Para obtener los resultados se trabajó en el entrenamiento de la red neuronal mediante el modelo de clasificación el cual muestra en el resultado la cantidad de alumnos aprobados o desaprobados y el otro modelo de regresión el cual predice la nota de un alumno dadas las características de su encuesta inicial, ambos modelos fueron de gran ayuda para predecir el comportamiento de los datos.

Palabras clave: Clasificación, Redes neuronales, Regresión, Predicción, Compartimiento del alumno.

Revista Innovación y Software
Vol. 2, No. 2, Mes Septiembre-Febrero, 2021
ISSN: 2708-0935
Pág. 75-91
https://revistas.ulasalle.edu.pe/innosoft

Abstract

Due to the recurring problem presented in the students regarding their academic performance, an application of neural networks was developed with the aim of helping the teacher, since it is capable of giving results of the final grades of the students and will help the teacher to understand the reason for the results, since this neural network takes into account different factors that lead the student to have a passing or failing grade. To obtain the results, we worked on the training of the neural network through the classification model which shows in the result the number of approved or disapproved students and the other regression model which predicts a student's grade given the characteristics of their initial survey, both models were of great help in predicting the behavior of the data.

Keywords: *Classification, Neural networks, Regression, Prediction, Student sharing.*

Introducción

En este trabajo se presenta el desarrollo de una aplicación con el uso de Redes Neuronales multicapa el cual busca clasificar a los alumnos de acuerdo a su nota para ver si serán aprobados o desaprobados en un curso, a su vez esta toma en cuenta los diferentes factores que influyen en la nota final del alumno los cuales fueron clasificados en variables independientes y dependientes que posteriormente serán sometidos a una limpieza de datos donde se tratarán datos duplicados y datos anómalos y estarán listos para ser normalizados y almacenados en un *dataset*.

La herramienta que se utilizó para el desarrollo de la aplicación es Anaconda en su entorno de Spyder en el lenguaje de python donde para construir la red neuronal fue necesario instalar diferentes librerías las cuales ayudarán a predecir el comportamiento de los datos y las fallas que influyen en el desempeño de los alumnos los cuales serán reflejados en las notas finales. Esta aplicación tiene como objetivo ayudar a los profesores a proveer la nota final de sus alumnos y verificar las causas y los factores que influyen a que un alumno tenga nota aprobatoria o desaprobatoria al final del curso. Así también el modelo de regresión permitirá predecir la nota final de un alumno dadas las características iniciales de este.

1. Materiales y métodos o Metodología computacional

1.1. Descripción general de la aplicación:

Un Instituto Politécnico de Informática (IPI) quiere desarrollar una aplicación que apoye a los docentes a proveer el resultado final de aprobado o no aprobado para los estudiantes. Este resultado deberá ser propuesto por la

Facultad de Ingeniería
Universidad La Salle, Arequipa, Perú
facin.innosoft@ulasalle.edu.pe

aplicación en función de los datos que los estudiantes brindaron al inicio del curso.

Así mismo, de estas encuestas se obtuvieron 1 044 encuestas recogidas en 2 IPIs de una ciudad durante el curso anterior en las asignaturas de inglés y matemáticas, así como el rendimiento de dichos estudiantes.

1.2. Selección y justificación del modelo

Debido a la capacidad de aprendizaje, las redes neuronales pueden llevar a cabo ciertas tareas basadas en un entrenamiento. Por ello, para abordar esta etapa inicial de la investigación, se pensó en un proceso de entrenamiento de la red neuronal en el cual se expondrá un conjunto de patrones de entrada y se ajustarán los pesos de forma que al final de este proceso se obtengan las salidas deseadas, que en este momento están clasificadas en dos categorías diferentes aprobado o desaprobado.

Siendo así que justificamos el uso de un modelo de clasificación para la predicción de si estará aprobado o no, y el uso de un modelo de regresión para predecir la nota como tal.

1.3. Herramientas

Anaconda

Es una distribución libre y abierta de los lenguajes Python y R, utilizada en ciencia de datos, y aprendizaje automático (machine learning). Este incluye procesamiento de grandes volúmenes de información, análisis predictivo y cómputos científicos. Está orientado a simplificar el despliegue y administración de los paquetes de software [1].

Spyder

Es un entorno científico escrito en Python, para Python, y diseñado por y para científicos, ingenieros y analistas de datos. El cual ofrece una combinación única de funciones avanzadas de edición, análisis, depuración y creación de perfiles de una herramienta de desarrollo integral con la exploración de datos, ejecución interactiva, inspección profunda y hermosas capacidades de visualización de un paquete científico y para aprovechar todas estas funciones es necesario instalar el entorno de anaconda [2].

1.4. Librerías y framework

Para el desarrollo de la aplicación se instalaron librerías que facilitarán el tratamiento de los datos las cuales son:

Theano

Es un compilador para matemáticas en Python. Sabe cómo tomar sus estructuras para convertirlas en código eficiente que utiliza NumPy, el cual fue diseñado específicamente para manejar los tipos de computación requeridos para grandes empresas como los algoritmos de redes neuronales usados en deep learning [3].

Instalación de Theano Línea de comando

pip install Theano

Tensor Flow

Es una librería Python para computación numérica la cual puede utilizarse para crear modelos de Deep Learning directamente o utilizando librerías de envolturas que simplifican el proceso, fue diseñado para su uso tanto en investigación y desarrollo, como por ejemplo en sistemas de producción. Puede funcionar en una sola CPUs, GPUs, así como dispositivos móviles y sistemas distribuidos a gran escala de cientos de máquinas [4].

Instalación de Tensor Flow línea de comando

#pip install tensorflow=1.0.0

Keras

Es un framework de alto nivel para el aprendizaje, escrito en Python y capaz de correr sobre los frameworks TensorFlow, CNTK, o Theano. Fue desarrollado con el objeto de facilitar un proceso de experimentación rápida [5].

Instalación de Keras línea de comando

pip install keras

Numpy

Es una extensión de Python, que le agrega mayor soporte para vectores y matrices, constituyendo una biblioteca de funciones matemáticas de alto nivel para operar con esos vectores o matrices [6].

# Instalación de Numpy línea de comando:
import numpy as np

Matplotlib.pyplop

Es una interfaz basada en estados para matplotlib. Proporciona una forma de trazado similar a MATLAB.Pyplot está diseñado principalmente para gráficos interactivos y casos simples de generación de gráficos programáticos

[7].

# Instalación de Matplotlib.pyplop línea de comando:
import matplotlib.pyplot as plt

Pandas

Es una herramienta de manipulación y análisis de datos de código abierto rápida, potente, flexible y fácil de usar, construida sobre el lenguaje de programación Python [8].

# Instalación de Pandas línea de comando:
import pandas as pd

Sklearn.metrics

Implementa funciones que evalúan el error de predicción para propósitos específicos. Estas métricas se detallan en las secciones en las métricas de clasificación, MULTILABEL métricas de clasificación, las métricas de regresión y las métricas de agrupamiento [9].

# Instalación de Sklearn.metrics línea de comando:
from sklearn.metrics import mean_squared_error

1.5. Selección del lenguaje de desarrollo

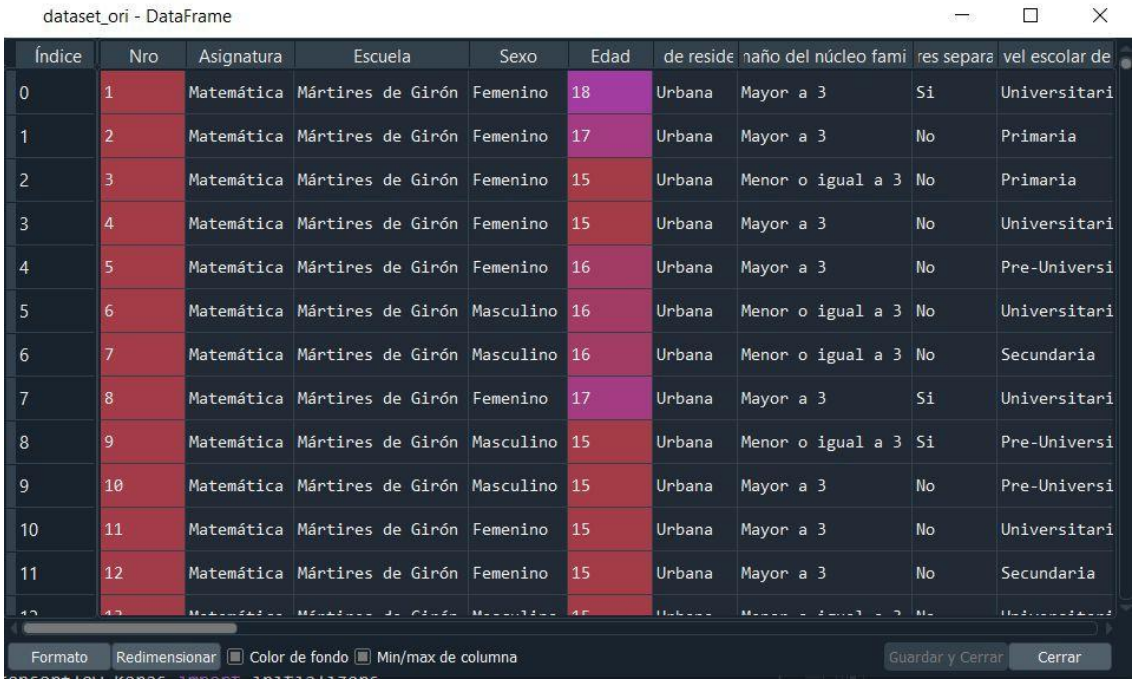
La elección del lenguaje para el desarrollo de esta investigación viene dado gracias a las ventajas de la misma; en este sentido, el trabajar con el lenguaje Python puede brindar facilidades como su sencillez, la usabilidad multiplataforma que permite la obtención de paquetes para la preparación del código en distintas plataformas. Otro punto resaltante, es la flexibilidad del lenguaje al permitir trabajar con programación Orientada a Objetos o scripting. Sin embargo, uno de los aspectos más interesantes de este lenguaje es que permite trabajar el desarrollo de RN con librerías gratuitas propias del lenguaje, las cuales permiten acelerar el proceso de desarrollo e implementación [10].

1.6. Descripción de los procesos

Tratamiento y limpieza de datos

Revista Innovación y Software
Vol. 2, No. 2, Mes Septiembre-Febrero, 2021
ISSN: 2708-0935
Pág. 75-91
https://revistas.ulasalle.edu.pe/innosoft

Lo primero es tratar el *dataset* proporcionado, en éste, se corregirán registros(encuestas) erróneas. Este proceso permitirá identificar registros incompletos, incorrectos, inexactos o no pertinentes, posterior a ello y con el *dataset* limpio, se puede comenzar a trabajar en el objetivo de esta investigación.



Índice	Nro	Asignatura	Escuela	Sexo	Edad	de reside	ıaño del núcleo fami	res separa	vel escolar de
0	1	Matemática	Mártires de Girón	Femenino	18	Urbana	Mayor a 3	Si	Universitari
1	2	Matemática	Mártires de Girón	Femenino	17	Urbana	Mayor a 3	No	Primaria
2	3	Matemática	Mártires de Girón	Femenino	15	Urbana	Menor o igual a 3	No	Primaria
3	4	Matemática	Mártires de Girón	Femenino	15	Urbana	Mayor a 3	No	Universitari
4	5	Matemática	Mártires de Girón	Femenino	16	Urbana	Mayor a 3	No	Pre-Universi
5	6	Matemática	Mártires de Girón	Masculino	16	Urbana	Menor o igual a 3	No	Universitari
6	7	Matemática	Mártires de Girón	Masculino	16	Urbana	Menor o igual a 3	No	Secundaria
7	8	Matemática	Mártires de Girón	Femenino	17	Urbana	Mayor a 3	Si	Universitari
8	9	Matemática	Mártires de Girón	Masculino	15	Urbana	Menor o igual a 3	Si	Pre-Universi
9	10	Matemática	Mártires de Girón	Masculino	15	Urbana	Mayor a 3	No	Pre-Universi
10	11	Matemática	Mártires de Girón	Femenino	15	Urbana	Mayor a 3	No	Universitari
11	12	Matemática	Mártires de Girón	Femenino	15	Urbana	Mayor a 3	No	Secundaria

Figura 1. Dataset limpio.

Transformación de Datos

Lo primero es el reconocimiento de las variables dependientes e independientes sobre las cuales se va a trabajar y modelar la red neuronal.

Variables Independientes:

Y	Nota final de curso

Dependientes:

X_0	Asignatura.
X_1	Escuela.
X_2	Sexo del estudiante.
X_3	Edad del estudiante.
X_4	Tipo de residencia del estudiante (Urbana o Rural).
X_5	Tamaño del núcleo familiar del estudiante.
X_6	Si el estudiante tiene padres separados.

Facultad de Ingeniería
Universidad La Salle, Arequipa, Perú
facin.innosoft@ulasalle.edu.pe

Revista Innovación y Software
Vol. 2, No. 2, Mes Septiembre-Febrero, 2021
ISSN: 2708-0935
Pág. 75-91
https://revistas.ulasalle.edu.pe/innosoft

X_7	Nivel escolar de la madre.
X_8	Nivel escolar del padre.
X_9	Trabajo de la madre.
X_10	Trabajo del padre.
X_11	Razón por la que el estudiante escogió esa escuela en la secundaria.
X_12	Tutor legal.
X_13	Tiempo de viaje desde la casa del estudiante a la escuela.
X_14	Tiempo de estudio semanal en horas.
X_15	Número de asignaturas desaprobadas en la secundaria.
X_16	Si la escuela le brinda al estudiante atención diferenciada.
X_17	Si la familia del estudiante le presta atención diferenciada.
X_18	Si el estudiante ha contratado a terceras personas para que lo ayuden en las asignaturas.
X_19	Si el estudiante participa en actividades extracurriculares de la escuela.
X_20	Si el estudiante ha sido atendido en la enfermería de la escuela.
X_21	Si el estudiante pretende estudiar en la Universidad.
X_22	Si el estudiante tiene internet en su hogar.
X_23	Si el estudiante está en una relación de pareja.
X_24	Calidad de la relación del estudiante con su familia.
X_25	Tiempo libre después de la escuela.
X_26	Si el estudiante sale con sus amigos.
X_27	Cantidad de alcohol consumido entre semana.
X_28	Cantidad de alcohol consumido en el fin de semana.
X_29	Estado de salud
X_30	Cantidad de ausencias a la escuela durante el curso.
X_31	Nota del primer trabajo de control parcial.
X_32	Nota del segundo trabajo de control parcial.

De esta manera, se debe identificar aquellas variables categóricas y no categóricas. Sin embargo, se ha considerado trabajar con Y_1 para la clasificación (desaprobado/aprobado) y Y_2 para la regresión (campo de notas finales).

Ahora, a las variables mostradas en el cuadro anterior se les aplicó un preprocesamiento, es decir la asignación de un equivalente numérico, teniendo en cuenta lo que representaba cada campo, para su posterior normalización, como resultado del mismo tenemos las siguientes equivalencias:

Tabla 1. Normalización de variables parte I.

X_0	X_1	X_2	X_3	X_4	X_5	X_6	X_7
Asignatura	Escuela	Sexo	Edad	Tipo de residencia	Tamaño del núcleo familiar	¿Padres separados?	Nivel escolar de la madre
Matemática	0 Mártires de Girón	0 Femenino	0 15	Urbana	0 Menor o igual	0 Si	1 Ninguno 0
Inglés	1 Camilo Cienfuegos	1 Masculino	1 16	Rural	1 Mayor a 3	1 No	0 Primaria 1
			17				Secundaria 2
			18				Pre-Universitario 3
			19				Universitario 4
			20				
			21				
			22				

Revista Innovación y Software
Vol. 2, No. 2, Mes Septiembre-Febrero, 2021
ISSN: 2708-0935
Pág. 75-91
https://revistas.ulasalle.edu.pe/innosoft

Tabla 2. Normalización de variables parte II.

X_8		X_9		X_10		X_11		X_12		X_13		X_14		X_15		X_16	
Atención difereciada por la escuela		Atención diferenciada por la familia		Contratos a terceros para ayuda con asignaturas		¿Participa en actividades extracurricular es?		¿Atendido en la enfermería de la escuela?		¿Quiere estudiar en la Universidad?		¿Tiene internet en la casa?		¿Está en una relación de pareja?		Calidad de las relación con su familia	
Si	1	Si	1	Si	1	Si	1	Si	1	Si	1	Si	1	Si	1	Muy mala	0
No	0	No	0	No	0	No	0	No	0	No	0	No	0	No	0	Mala	1
																Regular	2
																Buena	3
																Muy Buena	4

Tabla 3. Normalización de variables parte III.

X_17		X_18		X_19		X_20		X_21		X_22		X_23		X_24
Nivel escolar del padre		Trabajo de la madre		Trabajo del padre		Razón por la que el estudiante escogió esa escuela		Tutor legal		Tiempo de viaje a la escuela		Tiempo de estudio semanal		Número de asignaturas desaprobadas en la secundaria
Ninguno	0	En casa	0	En casa	0	Asignaturas que imparte	0	Madre	0	Menos de 15 mir	0	Menos de 2 hora	0	0
Primaria	1	Salud	1	Salud	1	Otro	1	Padre	1	De 15 a 30 minu	1	De 2 a 5 horas	2	1
Secundaria	2	Otro	0	Otro	0	Distancia a su casa	0	Otro	-1	De 30 a 60 minu	2	De 5 a 10 horas	5	2
Pre-Universitario	3	Profesor	1	Profesor	1	Reputación	1			Más de 1 hora	4	Más de 10 horas	10	3
Universitario	4	Servicios	0	Servicios	0									
		En casa	0	En casa	0	Asignaturas que imparte	0	Madre	1					
		Salud	1	Salud	1	Otro	1	Padre	2					
		Otro	2	Otro	2	Distancia a su casa	2	Otro	0					
		Profesor	3	Profesor	3	Reputación	3							
		Servicios	4	Servicios	4									

Tabla 4. Normalización de variables parte IV.

X_25		X_26		X_27		X_28		X_29		X_30	X_31	X_32
Tiempo libre después de la escuela		¿Sale con amigos?		Cantidad de alcohol consumido entre semana		Cantidad de alcohol consumido en el fin de semana		Estado de salud		Cantidad de ausencias	Nota del primer trabajo de control parcial	Nota del segundo trabajo de control parcial
Muy poco	0	Muy poco	0	Muy poco	0	Muy poco	0	Mal	0	0	0	0
Poco	1	Poco	1	Poco	1	Poco	1	Muy mal	1	1	15	20
Más o menos	2	Regularmente	2	Regular	2	Regular	2	Regular	2	2	20	25
Mucho	3	Frecuentemente	3	Demasiado	3	Demasiado	3	Bueno	3	3	25	30
Demasiado	4	Muy frecuente	4	Bastante	4	Bastante	4	Muy bueno	4	4	30	35
										5	35	40
										6	40	45
										7	45	50
										8	50	55
										9	55	60
										10	60	65
										11	65	70
										12	70	75
										13	75	80
										14	80	85
										15	85	90
										16	90	95
										17	95	
										18		
										19		
										20		
										21		

Revista Innovación y Software
Vol. 2, No. 2, Mes Septiembre-Febrero, 2021
ISSN: 2708-0935
Pág. 75-91
https://revistas.ulasalle.edu.pe/innosoft

Como resultado final de esta fase, se tuvo el *dataset* a entrenar:

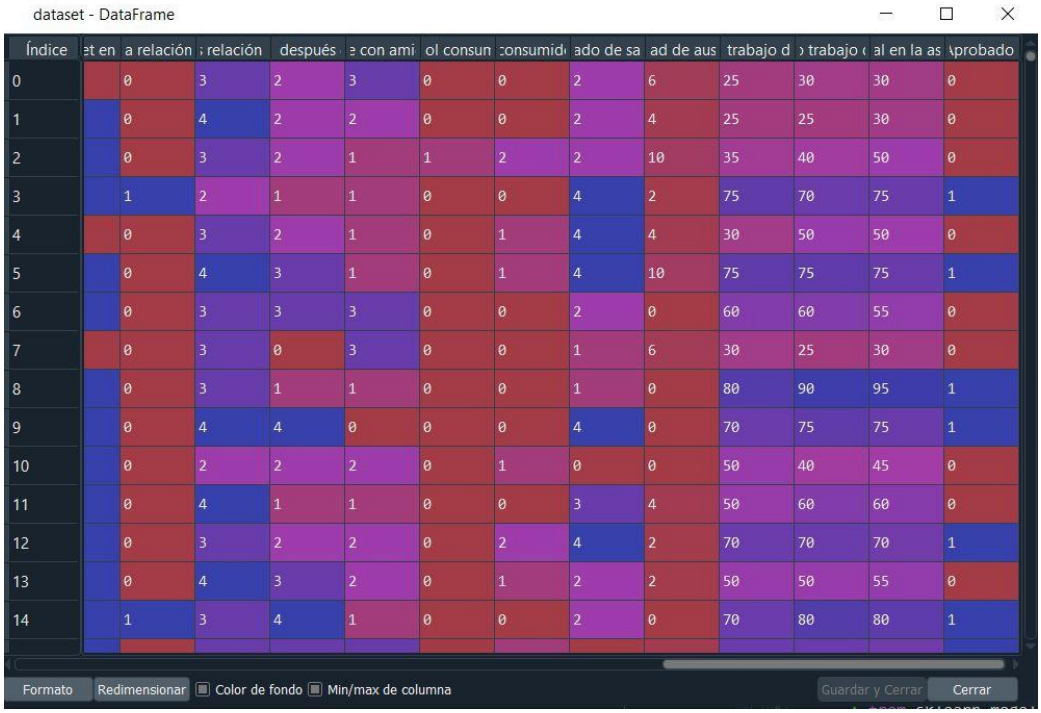


Figura 2. *Dataset* preprocesado - Encuestas de inicio de curso y notas.

Entrenamiento y validación

Fase de preprocesamiento de datos

```
# Parte 1 - Pre procesamiento de datos
# Importando las librerias
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

Figura 3. Implementación del importe de librerías para el preprocesamiento.

```
# Importando los datasets
dataset = pd.read_csv('data_trabajada.csv')
X = dataset.iloc[:, 1:34].values #solo aquellos campos relevantes
y = dataset.iloc[:, 35].values
```

Figura 4. Implementación de la división de variables dependientes e independientes.

Facultad de Ingeniería
Universidad La Salle, Arequipa, Perú
facin.innosoft@ulasalle.edu.pe

Revista Innovación y Software
Vol. 2, No. 2, Mes Septiembre-Febrero, 2021
ISSN: 2708-0935
Pág. 75-91
https://revistas.ulasalle.edu.pe/innosoft

Fase de entrenamiento

```
# Separando sets de datos en Entrenamiento y Prueba
from sklearn.model_selection import train_test_split
#variables
test_percent = 0.25
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = test_percent, random_state = 0)
```

Figura 5. Implementación de la división de la data de test y entrenamiento.

```
# Escalado de Caracteristicas
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

Figura 6. Implementación para el escalamiento de datos.

Fase de Creación de la red neuronal

Sequential es utilizada para que la red neuronal sea creada en secuencias, capa por capa, manualmente

```
# Parte 2 - Creando Red neuronal

# Importando librerias de Keras
import keras
from keras.models import Sequential
from keras.layers import Dense
from tensorflow.keras import initializers
```

Figura 7. Importe de librerías para la creación de la red neuronal.

```
# Iniciando Red Neuronal
classifier = Sequential()

i = 33 #nro de neuronas capa de entrada
o = 1    #nro de neuronas capa de salida
r = (i/o)**(1/3)
h1 = round(o*(r**2) )+1
h2 = round(o*r)+1
```

Figura 8. Implementación para el escalamiento de datos.

Facultad de Ingeniería
Universidad La Salle, Arequipa, Perú
facin.innosoft@ulasalle.edu.pe

Revista Innovación y Software
Vol. 2, No. 2, Mes Septiembre-Febrero, 2021
ISSN: 2708-0935
Pág. 75-91
https://revistas.ulasalle.edu.pe/innosoft

```
# Agregando capa Input y primera capa oculta
classifier.add(Dense(units = h1, kernel_initializer = initializers.TruncatedNormal(mean=0.0, stddev=0.05, seed=None),
                     bias_initializer=initializers.RandomNormal(), activation = 'relu', input_dim = i))

# Agregando segunda capa oculta
classifier.add(Dense(units = h2 , kernel_initializer =initializers.TruncatedNormal(mean=0.0, stddev=0.05, seed=None),
                     bias_initializer=initializers.RandomNormal(), activation = 'relu'))

# Agregando la capa output
classifier.add(Dense(units = 1, kernel_initializer = initializers.TruncatedNormal(mean=0.0, stddev=0.05, seed=None),
                     bias_initializer=initializers.RandomNormal(), activation = 'sigmoid'))

# Compilando la Red Neuronal
classifier.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])
```

Figura 9. Implementación del modelo (capas).

```
# Encajando Red Neuronal con set de Entrenamiento
epochs_hist = classifier.fit(X_train, y_train, batch_size = 27, epochs = 300)
# peso/bias  de h1
weights = classifier.layers[0].get_weights()[0]
biases = classifier.layers[0].get_weights()[1]
# peso/bias  de h2
weights2 = classifier.layers[1].get_weights()[0]
biases2 = classifier.layers[1].get_weights()[1]
# peso/bias  de h2
weights3 = classifier.layers[2].get_weights()[0]
biases3 = classifier.layers[2].get_weights()[1]
```

Figura 10. Implementación del encajamiento de la RN con el entrenamiento.

Fase de predicción y comprobación

```
# Parte 3 - Haciendo Predicciones y Evaluando el Modelo

# Prediccion con uso de los registros de prueba separados
y_pred = classifier.predict(X_test)
# metricas
from sklearn.metrics import mean_squared_error
mse = mean_squared_error(y_test, y_pred)
mse
```

Figura 11. Implementación de la predicción.

```
#Midiendo data de entrenamiento
scores = classifier.evaluate(X_train, y_train, verbose=0)
print("%s: %.2f%%" % (classifier.metrics_names[1], scores[1]*100))

#Midiendo data de Prueba
scores = classifier.evaluate(X_test, y_test, verbose=0)
print("%s: %.2f%%" % (classifier.metrics_names[1], scores[1]*100))
```

Figura 12. Implementación de la medición del entrenamiento y la prueba.

Revista Innovación y Software
Vol. 2, No. 2, Mes Septiembre-Febrero, 2021
ISSN: 2708-0935
Pág. 75-91
https://revistas.ulasalle.edu.pe/innosoft

```
def saveModel(model, nameModel):
    # serializa el modelo para JSON
    model_json1 = loaded_model.to_json()
    with open("model1_2.json", "w") as json_file:
    json_file.write(model_json1)
    #serializan los pesos (weights) para HDF5
    loaded_model.save_weights("model1_2.h5")
    print("Modelo guardado en el PC")
```

Figura 13. Implementación para la serialización del modelo para json.

```
def getModel():
    json_file = open('model1_2.json', 'r')
    loaded_model_json = json_file.read()
    json_file.close()
    loaded_model = model_from_json(loaded_model_json)
    # se cargan los pesos (weights) en el nuevo modelo
    loaded_model_weights = loaded_model.load_weights("model1_2.h5")
    print("Modelo cargado desde el PC")

   # se evalua el modelo cargado con los datos de los test
    loaded_model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```

Figura 14. Implementación de la de pesos en el nuevo modelo.

```
# Gráficos
#Historial de epocs
epochs_hist.history.keys()

#Grafico-  Progreso de entrenamiento del modelo según epocs
plt.plot(epochs_hist.history['loss'])
#plt.plot(epochs_hist.history['val_loss'])
plt.title('Progreso de entrenamiento del modelo')
plt.xlabel('Epoch')
plt.ylabel('Training Loss')
plt.legend(['Training Loss'])
```

Figura 15. Implementación para mostrar los resultados gráficamente.

```
#Prediccion - gráfico de resultados de predicción
y_predict = classifier.predict(X_test) # predicción del modelo
plt.plot(y_test, y_predict, "^", color = 'g')
plt.title('Resultados de predicción del modelo')
plt.xlabel('Prediccion del Modelo')
plt.ylabel('Valores Verdaderos')
```

Figura 16. Implementación para mostrar la predicción gráficamente.

Revista Innovación y Software
Vol. 2, No. 2, Mes Septiembre-Febrero, 2021
ISSN: 2708-0935
Pág. 75-91
https://revistas.ulasalle.edu.pe/innosoft

```
#Prediciendo resultados de estudiantes nuevos
##Nro,Asignatura,Escuela,Sexo,Edad,Tipo de residencia,Tamaño del núcleo familiar,¿Padres separados?,Nivel escolar de la madre,Nivel escol
#469,Inglés,Mártires de Girón,Masculino,16,Urbana,Mayor a 3,No,Pre-Universitario,Primaria,Otro,Otro,Reputación,Madre,Menos de 15 minutos,
#469,1,0,1,16,0,1,0,3,1,0,0,1,0,0,0,0,0,0,0,0,1,1,1,0,0,4,2,1,1,1,4,0,65,65,70,1

nuevo_estudiante=loaded_model.predict(sc.transform(np.array([[1,0,1,16,0,1,0,3,1,0,0,1,0,0,0,0,0,0,0,0,1,1,1,0,0,4,2,1,1,1,4,0,65,65]])))
nuevo_estudiante = (nuevo_estudiante > 0.5)

#449,Inglés,Mártires de Girón,Femenino,15,Urbana,Mayor a 3,No,Universitario,Universitario,Servicios,Servicios,Asignaturas que imparte,Mad
#449,1,0,0,15,0,1,0,4,4,0,0,0,0,0,0,0,1,1,0,0,1,1,1,0,2,2,3,1,2,4,0,65,60,60,0
nuevo_estudiante2=loaded_model.predict(sc.transform(np.array([[1,0,0,15,0,1,0,4,4,0,0,0,0,0,0,0,1,1,0,0,1,1,1,0,2,2,3,1,2,4,0,65,60]])))
nuevo_estudiante2 = (nuevo_estudiante > 0.5)
```

Figura 17. Implementación para predecir resultados de estudiantes nuevos.

Resultados y discusión

En esta parte del documento, se presentan los resultados de la construcción de la red neuronal para el tema de investigación.

Modelo de regresión

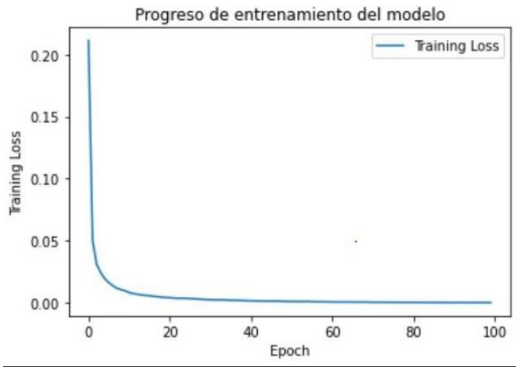


Figura 18. Progreso de entrenamiento - Modelo de regresión.

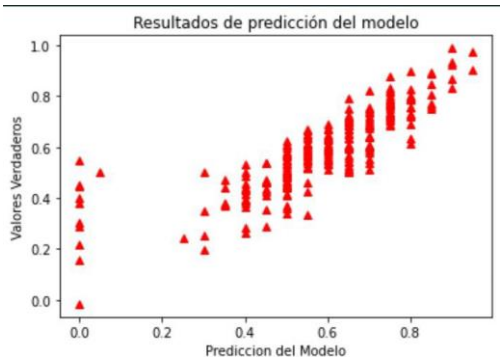


Figura 19. Resultados de predicción - Modelo de regresión.

Las métricas del modelo creado fueron las siguientes:

Revista Innovación y Software
Vol. 2, No. 2, Mes Septiembre-Febrero, 2021
ISSN: 2708-0935
Pág. 75-91
https://revistas.ulasalle.edu.pe/innosoft

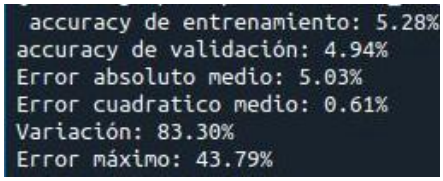


Figura 20. Métricas- Modelo de regresión.

Modelo de clasificación

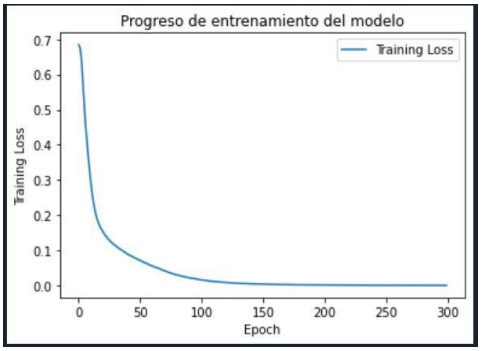


Figura 21. Progreso de entrenamiento - Modelo de clasificación.

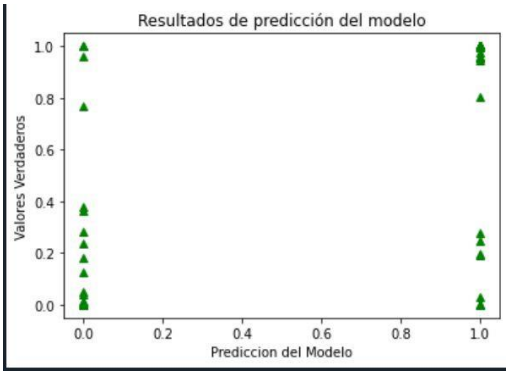


Figura 22. Resultados de predicción - Modelo de clasificación.

		predicción	
		0	1
realidad	0	TN	FP
	1	FN	TP

Figura 23. Lectura de una matriz de confusión.

Revista Innovación y Software
Vol. 2, No. 2, Mes Septiembre-Febrero, 2021
ISSN: 2708-0935
Pág. 75-91
https://revistas.ulasalle.edu.pe/innosoft

La lectura de la matriz de confusión es la siguiente: True Negative [TN], True Positive [TP], False Positive [FP], False Negative [FN]. En nuestro caso, tenemos el siguiente resultado:

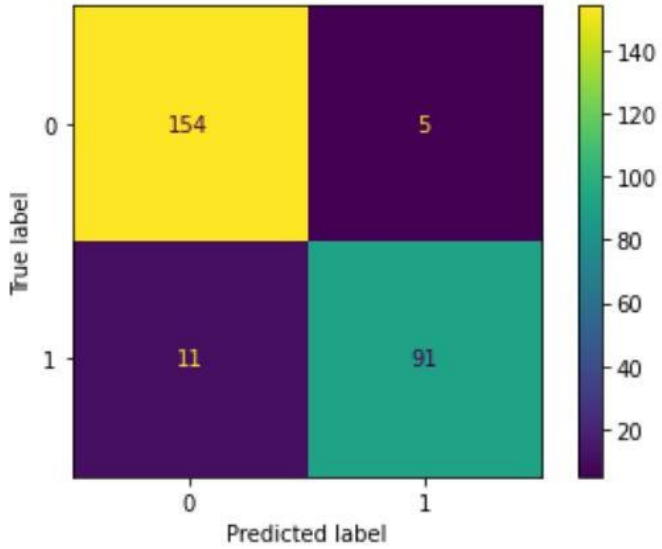


Figura 24. Gráfica de matriz de confusión - Modelo de clasificación.

Tabla 5. Resultados de métricas - Modelo de clasificación.

Nro	Métrica	Resultado
1	Error Cuadrático Medio	0.054983007933077
2	Presicion (Precisión)	0.947916666666666
3	Recall (Exhaustividad)	0.892156862745098
4	Accuracy (Exactitud)	0.938697318007662
5	F1	0.919191919191919

Comparativa

Para poder evaluar los dos métodos, adecuamos los resultados y_predictB = (y_predict>0.6) para tener el número de aprobados versus el número de desaprobados como se muestra en la figura 26. La misma la comparamos con la Figura 25, podemos ver que la predicción del modelo de clasificación con respecto al regresión es más acertada.

Facultad de Ingeniería
Universidad La Salle, Arequipa, Perú
facin.innosoft@ulasalle.edu.pe

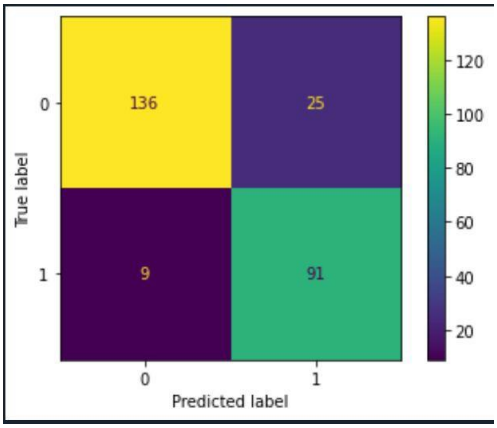


Figura 26. Gráfica de matriz de confusión - Modelo de regresión

Conclusiones

- En conclusión, una red neuronal de regresión nos permite predecir valores numéricos es decir la nota del alumno, sin embargo, al ser continuos los datos que calcula, hay un mayor ruido en los indicadores como la exactitud. Podemos medir la calidad de la predicción con métricas como el error absoluto medio, error cuadrático medio, varianza, entre otros. La red neuronal de clasificación nos permite predecir si el estudiante estará o no aprobado, la misma se puede medir empleando métricas tales como el cuadro de confusión, la precisión, la exactitud, entre otros. Los datos que poseemos tenemos que pre procesarlos de acuerdo al modelo que utilicemos y normalizarlos de ser necesario. entendiendo la relación de cada componente con el modelo predictivo.

Referencias

[1] Anaconda su kit de herramientas de ciencia de datos. [Online]. Available: https://www.anaconda.com/products/individual

[2] Spyder vision general. [Online]. Available: https://www.spyder-ide.org/

[3] Theano Project description. [Online]. Available: https://pypi.org/project/Theano/

[4] J. Buhigas“Todo lo que necesitas saber sobre Tensor Flow, la plataforma para Inteligencia Artificial de Google”,febrero, 2018 [Online]. Available: https://puentesdigitales.com/2018/02/14/todo-lo-que-necesitas-saber-sobre-tensorflow-la-plataforma-para-inteligencia-artificial-de-google/#:~:text=TensorFlow%20es%20una%20biblioteca%20de,gr%C3%A1ficos%20de%20flujo%20de%20datos.&text=La%20arquitectura%20flexible%20de%20TensorFlow,m%C3%B3viles%20con%20una%20sola%20API.

[5] Desarrollo de redes neuronales con keras. [Online]. Available: https://unipython.com/desarrolla-primera-red-neural-python-keras-paso-paso/#:~:text=Keras%20es%20una%20librer%C3%ADa%20Python,unas%20pocas%20l%C3%ADneas%20de%20c%C3%B3digo.

[6] L.Gonzales"Introducción a la librería NumPy de Python",setiembre, 2018. [Online]. Available: https://ligdigonzalez.com/introduccion-a-numpy-python-1/#:~:text=NumPy%20es%20un%20paquete%20de,garantizan%20c%C3%A1lculos%20eficientes%20con%20matrices.

[7] Matplotlib: funciones principales. [Online].Available: https://unipython.com/matplotlib-funciones-principales/

[8] Introducción a Pandas. [Online]. Available: https://programacion.net/articulo/introduccion_a_pandas_1632

[9] Metrics and scoring: quantifying the quality of predictions [Online]. Available: https://scikit-learn.org/stable/modules/model_evaluation.html

[10] Ventajas del uso de Python [Online]. Available https://blog.enzymeadvisinggroup.com/redes-neuronales-python#:~:text=La%20principal%20caracter%C3%ADstica%20de%20Python,multiplataforma%20y%20de%20tipado%20fuerte.