



POSGRADOS

MAESTRÍA EN SOFTWARE CON MENCIÓN EN DISEÑO DE ARQUITECTURA DE SISTEMAS

RPC-SO-34-NO.778-2021

OPCIÓN DE TITULACIÓN:

PROYECTO DE TITULACIÓN CON
COMPONENTES DE INVESTIGACIÓN
APLICADA Y/O DE DESARROLLO

TEMA:

IMPLEMENTACIÓN Y EVALUACIÓN DE
MODELOS DE APRENDIZAJE AUTOMÁTICO,
UTILIZANDO FRAMEWORK TENSORFLOW
2.13.0 , PARA LA PREDICCIÓN DE
DESEMPEÑO ESTUDIANTIL

AUTOR:

FREDDY EDWIN QUEZADA BERMEO

DIRECTOR:

RODOLFO XAVIER BOJORQUE CHASI

CUENCA – ECUADOR
2024

Autor:



Freddy Edwin Quezada Bermeo.

Ingeniero de Sistemas.

Candidato a Magíster en Software con Mención en Diseño de Arquitectura de Sistemas por la Universidad Politécnica Salesiana – Sede Cuenca.

fquezadab1@est.ups.edu.ec

Dirigido por:



Rodolfo Xavier Bojorque Chasi.

Ingeniero de Sistemas.

Doctor en Ciencias y Tecnologías de la Computación para Smart Cities.

rbojorque@ups.edu.ec

Todos los derechos reservados.

Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución, comunicación pública y transformación de esta obra para fines comerciales, sin contar con autorización de los titulares de propiedad intelectual. La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual. Se permite la libre difusión de este texto con fines académicos investigativos por cualquier medio, con la debida notificación a los autores.

DERECHOS RESERVADOS

©2024 Universidad Politécnica Salesiana.

CUENCA – ECUADOR – SUDAMÉRICA

FREDDY EDWIN QUEZADA BERMEO

Implementación y evaluación de modelos de aprendizaje automático, utilizando Framework Tensorflow 2.13.0 , para la predicción de desempeño estudiantil

Dedicatoria

Dios, por ser mi guía y fuente de fortaleza en cada paso de mi vida. A Él dedico cada logro y todo lo que soy, agradeciendo por la sabiduría y la protección que me ha brindado siempre.

A mi amada esposa, Adry, por ser mi compañera incondicional, mi apoyo en los momentos difíciles y mi inspiración constante. Gracias por tu amor y por estar siempre a mi lado.

A mis tres hijos, Mathi, Santy y Alejandro, quienes son mi mayor motivación para superarme cada día. Ustedes me impulsan a ser mejor en todo lo que hago, y este esfuerzo es también por y para ustedes.

Gracias por ser el motor que me impulsa a seguir adelante y nunca rendirme.

Agradecimiento

Agradezco en primer lugar a Dios, por ser mi fuerza y guía durante todo este proceso, brindándome la sabiduría necesaria para superar cada obstáculo. Sin Su presencia, este logro no habría sido posible.

A mis padres, por su amor incondicional y por inculcarme desde pequeño los valores del esfuerzo, la dedicación y el trabajo duro. Su apoyo constante ha sido una de las mayores bendiciones en mi vida, y sin ellos, no habría llegado hasta aquí.

A mi tutor y profesores, por su dedicación, orientación y tiempo a lo largo de este proyecto. Su guía fue crucial para completar esta investigación y para mi formación académica en general.

A mis amigos y familiares, quienes me brindaron ánimos y apoyo emocional, recordándome la importancia de perseverar y no rendirme. Su compañía fue vital para mantenerme enfocado y motivado durante los momentos difíciles.

Finalmente, agradezco a todos aquellos que, de manera directa o indirecta, contribuyeron a la culminación de este proyecto. Sus palabras de aliento y su ayuda, por más pequeñas que parecieran, fueron un gran impulso para mí. Este trabajo también es un testimonio del impacto positivo que las personas pueden tener en nuestra vida cuando creen en nosotros.

Índice general

Índice de Figuras	VIII
Índice de Tablas	I
Resumen	I
Abstract	I
1. Introducción	1
1.1. Descripción general del problema.....	2
1.1.1. Detalle del problema	3
1.2. Objetivos.....	3
1.2.1. Objetivo general.....	3
1.2.2. Objetivos específicos	3
1.3. Contribuciones.....	3
1.4. Herramientas Conceptuales	4
1.5. Organización del manuscrito.....	5
1.5.1. Capítulo 1: Introducción.....	5
1.5.2. Capítulo 2: Fundamentos Teóricos.....	6
1.5.3. Capítulo 3: Metodología	6
1.5.4. Capítulo 4: Implementación y Evaluación.....	6
1.5.5. Capítulo 5: Desarrollo del Dashboard.....	6
1.5.6. Capítulo 6: Discusión.....	7
1.5.7. Capítulo 7: Conclusiones y Futuras Direcciones	7
2. Estado del Arte	8
2.1. Estado del Arte.....	9

3. Metodología	14
3.1. Introducción a la Metodología CRISP-DM.....	15
3.1.1. Comprensión del Negocio:.....	15
3.1.2. Comprensión de los Datos:	16
3.1.3. Preparación de los Datos.....	16
3.1.4. Modelado:.....	16
3.1.5. Evaluación:	16
3.1.6. Despliegue:	16
3.2. Recolección de Datos:.....	17
3.2.1. Fuente de los Datos:.....	17
3.2.2. Descripción de los Archivos .xls:.....	17
3.2.3. Proceso de Transformación .xls a .csv:.....	17
3.3. Preprocesamiento de Datos	18
3.3.1. Limpieza de Datos.....	18
3.3.2. Selección de Características	18
3.4. Diseño del Modelo de Aprendizaje Automático.....	19
3.4.1. Arquitectura del Modelo	19
3.4.2. Entrenamiento del Modelo.....	20
3.4.3. Evaluación del Modelo	23
3.5. Implementación del Modelo	24
3.5.1. Diseño del Dashboard.....	24
4. Implementación y Evaluación	26
4.1. Implementación del Modelo:	27
4.1.1. Desafíos Técnicos Superados	27
4.1.2. Evaluación de Modelo	28
4.1.3. Resultados Obtenidos.....	29
5. Desarrollo del Dashboard	30
5.1. Diseño e Interfaz:.....	31
5.2. Funcionalidades Clave:.....	35
6. Análisis y Conclusiones	36
6.1. Análisis de Resultados:	37
6.2. Limitaciones del Estudio:.....	37
6.3. Conclusiones:.....	37
7. Anexos	39
7.1. Dataset Obtenido.....	40
7.2. Dashboard Conversión a .Csv.....	41

<i>ÍNDICE GENERAL</i>	VII
7.3. Archivo .Cvs.....	42
7.4. DashBoardVisual.py	44

Índice de Figuras

3.1. Ciclo de vida de minería de datos. Fuente: <i>IBM SPSS Modeler</i> [1].....	15
3.2. Representación Arquitectura Red Deep Neural Network, DNN	22
3.3. Representación Red Deep Neural Network.....	23
3.4. Ejemplo Dashboard : Autor. Freddy Quezada	25
3.5. Ejemplo de Dashboard Matriz de Confusión - Curva ROC : Autor.Freddy Quezada	25
7.1. Datos Obtenidos Institución Educativa	40
7.2. DataSet de Entrada	40
7.3. Dashboard Conversión a .Cvs	41
7.4. Archivo Normalizado Limpiado Filas Y columnas - Caso Matemáticas	42
7.5. Archivo Limpio .CSV - Vista Entorno de Desarrollo.....	43
7.6. DASHBOARD EN EJECUCIÓN - DIAGRAMA.....	44
7.7. DASHBOARD EN EJECUCIÓN - MATRIZ DE CONFUSIÓN	45
7.8. DASHBOARD EN EJECUCIÓN - CURVA DE ROC	46

**IMPLEMENTACIÓN Y
EVALUACIÓN DE MODELOS
DE APRENDIZAJE
AUTOMÁTICO,
UTILIZANDO FRAMEWORK
TENSORFLOW 2.13.0,
PARA LA PREDICCIÓN DE
DESEMPEÑO ESTUDIANTIL**

Autor(es):

FREDDY EDWIN QUEZADA BERMEO

Resumen

Este proyecto estudia la exploración y valoración de técnicas avanzadas de aprendizaje automático. En este trabajo se concretará la Implementación de modelos desarrollados con el soporte de TensorFlow 2.13.0, siendo esta la herramienta de vanguardia en el campo de Inteligencia Artificial. La finalidad se basa en predecir el rendimiento académico de estudiantes. Dicho de otra forma ¿cómo a través de la recolección y análisis de datos sobre el rendimiento académico, comportamiento, actividades de estudio, entre otros, obtener modelos predictivos que permitan anticiparse al rendimiento académico de futuros estudiantes?

Este dicho proceso incluye, entre otros, la escogencia cuidadosa de variables significativas que pudieran modelar el rendimiento estudiantil. Tales como hábitos de estudio, participación en clase, resultado de evaluaciones anteriores, entre otros factores. Con el soporte de TensorFlow 2.13.0 se desarrollarán y entrenarán los modelos.

Abstract

This project studies the exploration and evaluation of advanced machine learning techniques. In this work, the Implementation of models developed with the support of TensorFlow 2.13.0 will be specified, this being the cutting-edge tool in the field of Artificial Intelligence. The purpose is based on predicting the academic performance of students. In other words, how, through the collection and analysis of data on academic performance, behavior, study activities, among others, can we obtain predictive models that allow us to anticipate the academic performance of future students?

This process includes, among others, the careful selection of significant variables that can model student performance. Stories such as study habits, class participation, results of previous evaluations, among other factors. With the support of TensorFlow 2.13.0 the models will be developed and trained.

Capítulo 1

Introducción

1.1. Descripción general del problema

En la Institución TIC'S con vinculo a la UNIVERSIDAD de CUENCA, que es la encargada de analizar los datos que genera las unidades educativas de la provincia del Azuay, se reconoció la necesidad de mejorar el uso de los datos académicos generados, ya que estos tienen el potencial de mejorar la toma de decisiones pedagógicas y la asignación de recursos. Sin embargo, este potencial no se aprovecha debido a que: Los datos académicos existentes no se utilizan de manera eficiente o efectiva. Esto podría deberse a una falta de comprensión sobre cómo estos datos pueden ser aplicados para mejorar los procesos educativos, desde la identificación de las necesidades de aprendizaje de los estudiantes hasta la optimización de los programas y recursos académicos. Las unidades educativas carecen de la infraestructura necesaria para recoger, almacenar, procesar y analizar los datos. Esto implica tanto la falta de herramientas tecnológicas adecuadas como la carencia de capacidades técnicas o especialización en análisis de datos dentro del personal educativo. Este problema es multifacético y afecta a múltiples niveles del sistema educativo.

- A nivel estudiantil, la falta de análisis de datos significa que no se pueden identificar de manera proactiva estudiantes en riesgo de bajo rendimiento o deserción escolar, lo que a su vez impide la implementación de intervenciones tempranas y personalizadas.
- Para los educadores y administrativos, la ausencia de una infraestructura de datos adecuada limita su capacidad para evaluar la efectividad de las metodologías pedagógicas y programas educativos, y por lo tanto, dificulta la mejora continua de la calidad educativa.
- En cuanto a la gestión de recursos, la incapacidad para analizar y entender los datos académicos conduce a una asignación subóptima de recursos, lo que puede resultar en inversiones ineficientes y la perpetuación de desigualdades en el acceso y la calidad de la educación.

Estas afectaciones, en conjunto, impiden que las unidades educativas alcancen su máximo potencial en términos de ofrecer una educación que responda eficazmente a las necesidades de los estudiantes y la sociedad. La necesidad de abordar estos problemas se hace cada vez más crítica a medida que el volumen de datos generados por las instituciones educativas continúa creciendo.

1.1.1. Detalle del problema

El uso ineficiente de los datos académicos y la falta de infraestructura adecuada para su tratamiento constituyen un problema significativo por las siguientes razones.

- **Impacto en los Estudiantes:** La inadecuada gestión de datos impide la identificación oportuna y el apoyo a estudiantes que podrían estar enfrentando dificultades académicas, limitando las posibilidades de intervención personalizada para fomentar su éxito educativo.
- **Retroceso en el Ámbito Docente y Administrativo:** La falta de un sistema de datos robusto obstaculiza la habilidad del personal docente y administrativo para evaluar y mejorar la calidad de la enseñanza, ya que no cuentan con la información necesaria para tomar decisiones fundamentadas.

1.2. Objetivos

1.2.1. Objetivo general

Utilizar el framework TensorFlow 2.13.0 para diseñar, poner en práctica y valorar modelos de aprendizaje automático para predecir el rendimiento académico de los estudiantes.

1.2.2. Objetivos específicos

- Explorar investigaciones recientes en el ámbito de la predicción del rendimiento académico mediante aprendizaje automático, con un enfoque en aquellos estudios que utilicen TensorFlow.
- Definir los parámetros para la estructuración del modelo, tomando en cuenta las particularidades de los datos educativos.
- Implementar dashboard interactivo utilizando herramientas como Plotly Dash de Python, que muestre el resultado de la predicción.

1.3. Contribuciones

El presente trabajo, titulado Implementación y evaluación de modelos de aprendizaje automático para la predicción del desempeño estudiantil,

aporta mejoras significativas tanto al ámbito académico como a la práctica educativa. Estas contribuciones se detallan a continuación:

- **Avance en la Pedagogía Personalizada:** Este estudio propone un enfoque innovador para personalizar el proceso educativo. Gracias a modelos predictivos precisos, es posible prever el desempeño de los estudiantes, lo que permite a los educadores e instituciones implementar estrategias educativas adaptadas a las necesidades individuales. Esto fomenta un aprendizaje más inclusivo y equitativo.
- **Desarrollo de Herramientas Prácticas:** Se diseñó un dashboard interactivo que integra los resultados de los modelos predictivos, brindando a los educadores una herramienta eficiente y accesible. Este dashboard no solo permite monitorear el progreso estudiantil en tiempo real, sino que también proporciona información clave para diseñar intervenciones tempranas y mejorar los resultados educativos.
- **Contribuciones al Ámbito Académico:** Este trabajo integra la metodología CRISP-DM y aprovecha el potencial de TensorFlow 2.13.0, sentando las bases para investigaciones futuras en la intersección entre inteligencia artificial y educación. Además, permite explorar nuevas aplicaciones prácticas de los modelos de aprendizaje automático en el entorno educativo.
- **Potencial de Escalabilidad:** Aunque el estudio se centra en una institución educativa específica, las técnicas empleadas tienen el potencial de aplicarse en otros contextos y a escalas mayores, contribuyendo al desarrollo de sistemas educativos más eficientes y personalizados.

1.4. Herramientas Conceptuales

- **Redes Neuronales:** "Las redes neuronales son modelos predictivos, lo que significa que, dados algunos datos por adelantado, son capaces de generar una predicción cuando se les presentan nuevos datos. Otros modelos predictivos más conocidos son la regresión lineal simple, la regresión lineal múltiple y la regresión logística. Las redes neuronales producen mejores resultados que esos modelos cuando se enfrentan a problemas más complejos"[2].
- **Red Neuronal Convolutiva (CNN):** Una Red Neuronal Convolutiva (CNN) en el contexto de la predicción del rendimiento estudiantil

se adapta para analizar patrones dentro de los datos educativos que tengan una estructura espacial o que puedan ser representados de manera que imiten las características de los datos visuales, su aplicación en este contexto estaría orientada a reconocer y aprender las características complejas de los datos educativos que están dispuestos de forma matricial o gráfica.

La CNN funcionaría extrayendo características relevantes de los datos estructurados de manera que se asemejen a una imagen (resultado Dashboard, tales como la progresión del rendimiento a lo largo del tiempo representada en una matriz donde las filas podrían ser el tiempo y las columnas diferentes métricas de rendimiento. La CNN podría identificar patrones como aumentos o disminuciones en el rendimiento, correlaciones entre diferentes tipos de evaluaciones y asignaturas, o tendencias grupales dentro de la población estudiantil.

- TensorFlow
- Keras

1.5. Organización del manuscrito

El manuscrito que ha preparado para compartir los hallazgos y contribuciones del presente trabajo sobre la predicción del desempeño estudiantil utilizando aprendizaje automático y TensorFlow 2.13.0 está estructurado meticulosamente para guiar al lector a través de nuestra investigación, desde los cimientos teóricos hasta las aplicaciones prácticas.

1.5.1. Capítulo 1: Introducción

- Contexto y Motivación: Se establece el escenario, explicando por qué es vital predecir el desempeño estudiantil y cómo la tecnología puede ser una herramienta transformadora en la educación.
- Objetivos del Estudio: Se centra en los objetivos específicos que se propone alcanzar con este trabajo de investigación.
- Estructura del Manuscrito: Se ofrece una visión general de cómo está organizado el documento, guiando al lector sobre qué esperar en cada sección.

1.5.2. Capítulo 2: Fundamentos Teóricos

- Aprendizaje Automático y TensorFlow: Introducimos los conceptos básicos del aprendizaje automático, con un enfoque particular en las características y ventajas de TensorFlow 2.13.0.
- Revisión de Literatura: Se analizará los estudios previos relevantes, estableciendo el estado del arte y cómo la investigación se construye sobre este.

1.5.3. Capítulo 3: Metodología

- Recopilación de Datos: Se describe cómo y de donde se recopilaron los datos utilizados para entrenar y probar el modelo.
- Preprocesamiento de Datos: Explicación de los pasos tomados para limpiar y preparar los datos para el análisis, asegurando su calidad y relevancia.
- Diseño del Modelo: Se detalla el proceso de selección y diseño de los modelos de aprendizaje automático, incluyendo la justificación de las decisiones tomadas.

1.5.4. Capítulo 4: Implementación y Evaluación

- Implementación de Modelos: Se describe cómo se implementaron el modelo utilizando TensorFlow 2.13.0, incluyendo desafíos técnicos superados.
- Evaluación de Modelos: Se presenta los métodos y métricas utilizados para evaluar el desempeño de los modelos, junto con una discusión detallada de los resultados obtenidos.

1.5.5. Capítulo 5: Desarrollo del Dashboard

- Diseño e Interfaz: Se explica cómo se diseñó el dashboard para visualizar los resultados de las predicciones y su interacción con el usuario final.
- Funcionalidades Clave: Se detalla las principales características del dashboard, incluyendo cómo facilita la interpretación y aplicación de los resultados del modelo.

1.5.6. Capítulo 6: Discusión

- **Análisis de Resultados:** Reflexión sobre los resultados obtenidos, considerando su impacto en la educación y la predicción del desempeño estudiantil.
- **Limitaciones del Estudio:** Se reconoce las limitaciones de la investigación, ofreciendo una perspectiva equilibrada.

1.5.7. Capítulo 7: Conclusiones y Futuras Direcciones

- **Contribuciones Principales:** Se resume las principales contribuciones de nuestro trabajo y su relevancia para el campo.
- **Recomendaciones para Futuras Investigaciones:** Se propone áreas para investigaciones futuras, basándonos en los hallazgos y experiencias.

Cada capítulo ha sido diseñado para construir sobre el anterior, guiando al lector desde los conceptos básicos hasta las aplicaciones más complejas y avanzadas, asegurando una comprensión profunda y completa del presente trabajo y sus implicaciones en el ámbito educativo.

Capítulo 2

Estado del Arte

Este capítulo proporcionar una base sólida en los conceptos clave y el contexto histórico y teórico que sustentan el estudio sobre la predicción del desempeño estudiantil mediante el uso de aprendizaje automático y TensorFlow 2.13.0.

2.1. Estado del Arte

En el análisis del estado del arte, basado en la revisión de artículos relevantes, se elaboró una tabla comparativa cuyo objetivo principal es presentar de manera estructurada y detallada las diferentes técnicas y métodos utilizados en la literatura reciente para la predicción del desempeño estudiantil mediante modelos de aprendizaje automático.

Esta tabla busca destacar las estrategias aplicadas en diversos estudios, incluyendo metodologías como el procesamiento de lenguaje natural, redes neuronales (convolucionales, recurrentes y densas), y modelos secuenciales, entre otros. Asimismo, se incluyen los enfoques empleados para la validación de resultados y los métodos de análisis utilizados para evaluar la efectividad de estas técnicas. Entre las métricas y herramientas más frecuentes se encuentran la precisión, el F1-score, la curva ROC y el área bajo la curva (AUC), las cuales son fundamentales para medir la calidad de los modelos.

La finalidad de este análisis es proporcionar un panorama claro y comprensivo de las tendencias actuales en el ámbito del aprendizaje automático aplicado al sector educativo. La tabla no solo facilita la identificación de patrones recurrentes y estrategias efectivas, sino que también permite detectar brechas en la investigación actual, lo que abre nuevas oportunidades para futuras exploraciones.

Además, al organizar la información de manera accesible y comparativa, se pretende ofrecer una herramienta práctica tanto para investigadores como para profesionales del sector educativo, que puedan aprovechar estos conocimientos para implementar mejoras tecnológicas en los modelos predictivos educativos. De esta manera, se subraya la importancia de actualizar y adaptar las técnicas existentes para maximizar su impacto en el ámbito académico.

Año	Autor	Solución	Referencia
2023	Biqian Cheng, Yuping Liu y Yunjian Jia	El artículo se centra en la evaluación del rendimiento estudiantil durante el período académico utilizando un modelo híbrido que combina el Clasificador XG-Boost con una optimización mejorada basada en el ecosistema artificial. Explora diversas técnicas de aprendizaje automático y redes neuronales para predecir el éxito académico, comparando su eficacia con métricas de rendimiento estándar y aplicando técnicas como SVM-SMOTE para mejorar la precisión de las predicciones. La principal herramienta fue el uso del algoritmo XG-Boost para la evaluación del desempeño académico de los estudiantes.	[3]
2024	Fatema Alshaikh y Nabil Hewahi	La solución propuesta en el estudio es el uso de una Red Neuronal Convolucional (CNN) para predecir el rendimiento académico de los estudiantes en un Sistema de Tutoría Inteligente. El modelo CNN mejorado logró una precisión de hasta el 94.36 % en la predicción del desempeño estudiantil en las primeras etapas utilizando un conjunto de datos y mejorando al 84.83 % con otro conjunto. El estudio subraya la eficacia de combinar técnicas de aprendizaje automático con algoritmos metaheurísticos para mejorar la precisión en la clasificación y predicción del rendimiento de los estudiantes. Aplicaron el algoritmo con CNN a través de Árboles de Decisión para preparar la data obtenida y alimentar la red neuronal.	[4]
2024	E. Golden Julie, Y. Harold Robinson, J. Jesu Vedha Nayahi y Thavavel Vaiyapuri	La solución o temática central que aborda el libro es la aplicación de la tecnología blockchain en sistemas inteligentes, explorando su potencial para revolucionar la computación de próxima generación a través de diferentes protocolos y enfoques. El libro detalla tanto las oportunidades como los retos que la integración de blockchain presenta en diversos sectores, incluyendo la medicina y el Internet de las Cosas (IoT). Las redes neuronales podrían analizar y predecir patrones complejos de datos, optimizar procesos de decisión y reforzar la integridad y confiabilidad en las transacciones blockchain, contribuyendo así a sistemas más inteligentes y autónomos. En este libro aplican el algoritmo de CNN a través de una arquitectura de preprocesamiento, configuración, entrenamiento e interpretación, demostrando que la intervención de las CNN en el ámbito educativo es cada vez mejor.	[5]

Año	Autor	Solución	Referencia
2024	Qionghao Huang y Yan Zeng	<p>Introduce un modelo innovador llamado APP-DGNN para predecir el rendimiento académico usando redes neuronales de grafos duales. Este modelo analiza tanto las actividades de interacción como los atributos de los estudiantes para predecir su rendimiento, logrando una precisión significativa. Utiliza técnicas avanzadas como la convolución de grafos y aprendizaje de representación de local a global para manejar eficientemente la información estructurada y dinámica de los datos educativos. Los algoritmos específicos empleados en este modelo son: Red Neuronal de Grafos Basada en Interacciones (IGCN) y Red Neuronal de Grafos Basada en Atributos (AGCN)</p>	[6]
2024	S. Lakshmi y C. P. Maheswaran	<p>Propone un sistema de predicción de calificaciones utilizando la Unidad Recurrente de Compuerta (GRU) y optimización de características mediante Análisis de Varianza (ANOVA). La solución presentada mejora la predicción de las calificaciones finales de los estudiantes mediante el análisis de registros académicos y la optimización de características para resaltar los factores más influyentes en el rendimiento académico. El artículo utiliza Python y TensorFlow para implementar y evaluar el modelo propuesto. Cabe señalar que ANOVA tomando como variación de una red neuronal recurrente dio buenos resultados para este modelo y aplicado como el algoritmo principal en este estudio optimizo el análisis y las predicciones.</p>	[7]
2023	Didi Supriyadi, Purwanto Purwanto y Budi Warsito	<p>Presenta una solución denominada HyFeS-ROS-ANN, que combina selección de características híbridas y técnicas de remuestreo para clasificar el rendimiento académico utilizando redes neuronales artificiales de perceptrón multicapa (MLP). El estudio mejora la precisión de la clasificación del rendimiento académico seleccionando características esenciales y manejando datos de clases desequilibradas alcanzando una precisión del 100 %. El lenguaje de programación utilizado en el estudio es Python, específicamente empleando TensorFlow 2.14.0 para implementar el modelo de redes neuronales multicapa (MLP). Además, se utilizaron bibliotecas como Scikit-Learn, Numpy, pandas y matplotlib para el procesamiento de datos, selección de características y visualización. El algoritmo principal que se utilizó es el perceptrón a multicapa (MLP),</p>	[8]

Año	Autor	Solución	Referencia
2023	Pallavi Asthana, Sumita Mishra, Nishu Gupta, Mohammad Derawi, y Anil Kumar	<p>Presentan una solución para la predicción del rendimiento estudiantil mediante coeficientes de aprendizaje evaluados a través de la regresión en modelos de aprendizaje automático. El estudio destacó que la regresión lineal obtuvo la mayor precisión (97 %) en comparación con otros modelos. El lenguaje de programación utilizado en su investigación es Python, especialmente en herramientas como Google Colaboratory (Colab) para el análisis y soporte de la sintaxis completa de Python aplican modelos de regresión como la regresión lineal y algoritmos como árboles de decisión y bosques aleatorios, que no emplean neuronas en el sentido tradicional como lo hacen las redes neuronales. Claramente en este estudio el algoritmo dominante es Regresión Lineal siendo el más efectivo.</p>	[9]
2022	Marina Segura, Jorge Mello, y Adolfo Hernández	<p>En este estudio se aplicaron métodos de aprendizaje automático, como Redes Neuronales Artificiales (ANN), para predecir la deserción universitaria. El enfoque principal fue analizar cómo la preferencia del estudiante por su carrera influye en este fenómeno. Aunque el lenguaje de programación específico utilizado no se menciona directamente, es probable que se haya empleado un lenguaje común en ciencia de datos, como Python o R, debido a la naturaleza del estudio. Además, se destaca el uso de la regresión logística como algoritmo que, si bien ofrece resultados ligeramente inferiores a los modelos más complejos, proporciona una base razonable y más accesible para realizar comparaciones.</p>	[10]
2022	Nawaf N. Hamadneh, Samer Atawneh, Waqar A.Khan, Khaled A. Almejalliy Adeeb Alhomoud	<p>La solución presentada es un modelo de Red Neuronal Artificial (ANN) que utiliza el Algoritmo Firefly (FFA) para el entrenamiento, enfocado en predecir el rendimiento académico de los estudiantes en un entorno de aprendizaje mixto. El estudio empleó herramientas de inteligencia artificial, específicamente redes neuronales pero no especifica un lenguaje de programación particular en el fragmento analizado. El artículo menciona el uso de una Red Neuronal Artificial (ANN) en su solución para predecir el rendimiento académico de los estudiantes. Sin embargo, no especifica el tipo. El algoritmo principal utilizado en el artículo es el Perceptrón multicapa (MLP), que es un tipo de red neuronal artificial</p>	[11]

Año	Autor	Solución	Referencia
2024	Qionghao Huang y Yan Zeng	<p>El estudio presenta un modelo llamado APP-DGNN para predecir el rendimiento académico, utilizando redes neuronales de grafos duales que aprovechan tanto las actividades de interacción como las características de atributos de los estudiantes. El modelo propuesto, APP-DGNN, demostró una precisión significativa en la predicción del rendimiento académico, superando a los métodos existentes con una precisión del 83.96 % para predecir estudiantes que pasan o fallan, y un 90.18 % para predecir estudiantes que pasan o se retiran.</p> <p>El algoritmo principal utilizado en el artículo es una red neuronal de picos adelantada (feedforward spiking neural network).</p>	[12]

Capítulo 3

Metodología

3.1. Introducción a la Metodología CRISP-DM

La metodología CRISP-DM (Cross-Industry Standard Process for Data Mining) fue adoptada en este proyecto para estructurar las fases del trabajo. Esta metodología es ampliamente reconocida en la minería de datos y consta de seis fases independientes:

3.1.1. Comprensión del Negocio:

Se definieron los objetivos y requisitos del proyecto desde una perspectiva educativa. El objetivo principal fue diseñar un modelo predictivo para identificar estudiantes que podrían necesitar apoyo académico, guiando la intervención oportuna.

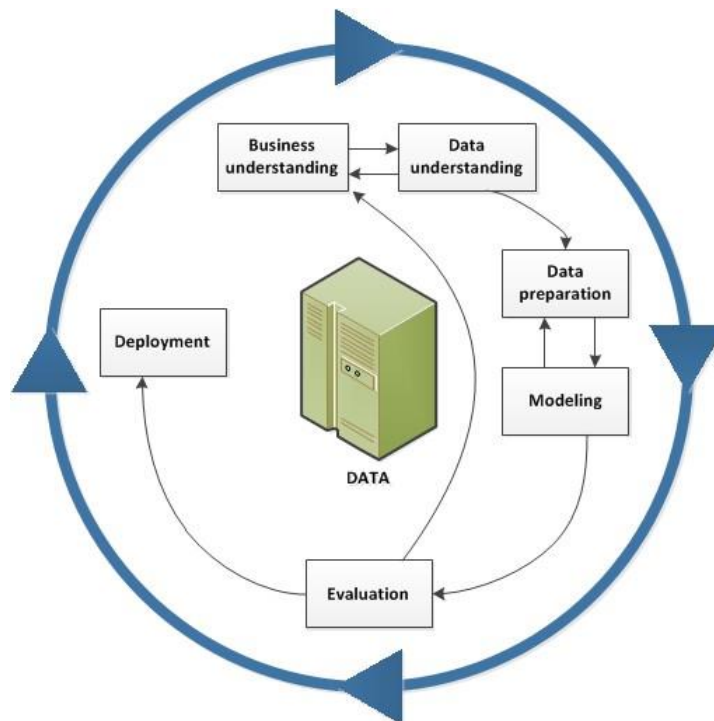


Figura 3.1: Ciclo de vida de minería de datos. Fuente: *IBM SPSS Modeler* [1].

3.1.2. Comprensión de los Datos:

En esta etapa se analizaron los datos disponibles para evaluar su calidad, identificar problemas y generar hipótesis sobre patrones relevantes. Esto incluyó una inspección preliminar de las variables disponibles y una detección de valores atípicos.

3.1.3. Preparación de los Datos

Esta fase consistió en transformar los datos brutos en un formato adecuado para el modelado. Las actividades principales incluyeron:

1. **Transformación de Archivos .xls a .csv:** Utilizando bibliotecas como Pandas en Python.
2. **Limpieza de Datos:** Eliminación de duplicados, manejo de valores faltantes y corrección de errores.
3. **Selección de Características:** Uso de análisis de correlación y métodos como RFE (*Recursive Feature Elimination*) para identificar variables clave.

3.1.4. Modelado:

El modelo predictivo se implementó utilizando una arquitectura de red neuronal profunda (DNN) en TensorFlow 2.13.0. Este proceso incluyó la configuración de capas, selección de hiperparámetros y validación cruzada para garantizar la robustez del modelo.

3.1.5. Evaluación:

Se evaluó la calidad del modelo mediante métricas estándar como precisión, recall, F1-score y área bajo la curva ROC (AUC). Estas métricas aseguraron que el modelo cumpliera los objetivos de predicción.

3.1.6. Despliegue:

Los resultados del modelo se integraron en un dashboard interactivo desarrollado con Dash y Plotly. Este dashboard permite cargar nuevos datos, realizar predicciones y visualizar resultados clave, como la matriz de confusión y la curva ROC.

3.2. Recolección de Datos:

3.2.1. Fuente de los Datos:

La recopilación de datos es una etapa crucial para el desarrollo del modelo de aprendizaje automático. En este estudio, los datos académicos se obtuvieron de una Institución Educativa de la provincia del Azuay en el cantón Cuenca. Se tuvo acceso a los registros académicos de los estudiantes del año lectivo 2023-2024 específicamente. Estos datos incorporan variables relevantes como asistencia, calificaciones y participación en actividades extracurriculares, entre otros.

3.2.2. Descripción de los Archivos .xls:

Los datos académicos están almacenados en archivos con formato .xls. Estos archivos contienen múltiples variables que registran aspectos cuantitativos y cualitativos del rendimiento académico de los estudiantes. Las principales columnas presentes en estos archivos son:

- ID Estudiante : Identificación única para cada estudiante.
- Asignaturas: Nombres de las asignaturas cursadas.
- Calificaciones: Calificaciones obtenidas en cada asignatura.
- Asistencia: Porcentaje de asistencia a clases.
- CantidadHoras: Indicador de horas asistidas.
- Participación: Indicador de participación en actividades clasificadas por asignatura.

3.2.3. Proceso de Transformación .xls a .csv:

Dado que los archivos .xls requieren de una conversión para ser manipulados más eficientemente en entornos de programación modernos, se transformó a formato .csv. Este proceso no solo facilita la manipulación de los datos utilizando herramientas de programación como Python y bibliotecas especializadas en análisis de datos como Pandas, sino que también ayuda a asegurar la compatibilidad con TensorFlow para el desarrollo y entrenamiento de modelos de aprendizaje automático.

3.3. Preprocesamiento de Datos

El preprocesamiento de datos es crucial para asegurar la calidad del modelo de aprendizaje automático. Este proceso incluye limpiar los datos de posibles errores y seleccionar las características más relevantes que influirán en la precisión del modelo predictivo.

3.3.1. Limpieza de Datos

Antes de entrenar el modelo, es esencial limpiar los datos para eliminar registros erróneos o incompletos que puedan distorsionar los resultados del análisis. Los pasos ejecutados en la limpieza de datos fueron:

- **Eliminación de Duplicados:** Revisar los datos para eliminar entradas duplicadas que puedan afectar la representatividad del modelo. Esto se realizó utilizando la función `drop_duplicates` de `pandas`.
- **Manejo de Valores Faltantes:** Aplicar técnicas para gestionar los valores ausentes es conocido como imputación. Dependiendo del contexto, esto puede implicar completar valores usando la media, mediana o moda de la columna; o eliminar filas o columnas con una alta proporción de valores faltantes. Se utilizó funciones como `fillna` y `dropna` de `pandas` para este propósito.
- **Corrección de Errores de Formato:** Normalizar los formatos de datos, especialmente en fechas y categorías nominales, para asegurar consistencia a lo largo del dataset. Esto incluyó convertir fechas a un formato uniforme y asegurar que las categorías estén correctamente codificadas.
- **Validación de Datos:** Verificar la plausibilidad y coherencia de los datos, corrigiendo valores atípicos o errores introducidos durante la recolección de datos. Se utilizó técnicas de detección de valores atípicos y se realizó una validación cruzada para asegurar la integridad de los datos.

3.3.2. Selección de Características

La selección de características implica identificar las variables más relevantes para el modelo predictivo. Esto no solo mejora la eficiencia del modelo al reducir la dimensionalidad de los datos, sino que también puede

incrementar su precisión al eliminar el ruido. Las técnicas utilizadas pueden incluir:

- **Análisis de Correlación:** Para identificar relaciones significativas entre variables. Se utilizará la función `corr` de `pandas` para calcular las correlaciones entre las variables y seleccionar aquellas con mayor impacto en el rendimiento académico.
- **Métodos de Selección de Características:** Como Selección de Características Basada en Importancia (Feature Importance) y Eliminación Recursiva de Características (Recursive Feature Elimination, RFE). De hecho esta técnica se implementó utilizando bibliotecas como `Scikit-learn`.

3.4. Diseño del Modelo de Aprendizaje Automático

3.4.1. Arquitectura del Modelo

Como ya se menciona, se utilizó una arquitectura de red neuronal profunda (Deep Neural Network, DNN) implementada en `TensorFlow 2.13.0`. Esta arquitectura es adecuada debido a su capacidad para manejar grandes cantidades de datos y extraer patrones complejos.

Capas de Entrada

Incluye las características seleccionadas como entrada. Cada característica se representará como una neurona en la capa de entrada. Recibe las características de los estudiantes (notas, promedios, evaluaciones).

Capas Ocultas

Varias capas ocultas con neuronas que utilizan funciones de activación `ReLU` (Rectified Linear Unit). El número de capas y neuronas por capa se determinará mediante la optimización de hiperparámetros.

Capa de Salida

Se utilizó una función de activación sigmoide para la predicción binaria del rendimiento (aprobado/no aprobado) o una función de activación `softmax` para clasificación multicategoría. Genera la predicción final, en este caso, una probabilidad de que el estudiante necesite apoyo.

3.4.2. Entrenamiento del Modelo

Arquitectura del Modelo

La implementación de la predicción del desempeño estudiantil se basa principalmente en una red neuronal profunda (DNN).

Estructura de la red neuronal

Capa de Entrada (Input Layer):

- Descripción: La capa de entrada recibe los datos que alimentan la red neuronal. En este caso, los datos de entrada tienen 5 características. La capa de entrada no realiza ninguna transformación en los datos, simplemente los pasa a la primera capa oculta.
- Tamaño del Input : (None, 5), lo que indica que hay 5 características de entrada y "None" significa que la cantidad de datos puede variar según el lote.

Capa Oculta 1 (Capa Oculta 1) :

- Descripción: Esta es la primera capa oculta de la red neuronal. Está compuesta por 128 neuronas completamente conectadas. Cada neurona aplica una transformación lineal seguida de una función de activación no lineal ReLU (Rectified Linear Unit) , que introduce no linealidad al modelo y ayuda a aprender características complejas.
- Tamaño del Input : (None, 5)proviene de la capa de entrada.
- Tamaño del Output : (None, 128)corresponde a las 128 neuronas de la capa oculta.

Capa Oculta 2 (Capa Oculta 2) :

- Descripción: La segunda capa oculta tiene 64 neuronas. Similar a la primera capa oculta, aplica la activación ReLU después de realizar una combinación lineal de las entradas.
- Tamaño del Input : (None, 128), que proviene de la salida de la Capa Oculta 1.
- Tamaño del Output : (None, 64), correspondiente al número de neuronas en esta capa.

Capa Oculta 3 (Capa Oculta 3) :

- Descripción: La tercera capa oculta tiene 32 neuronas. Esta capa también utiliza la activación ReLU para introducir no linealidad.
- Tamaño del Input : (None, 64), proveniente de la salida de la Capa Oculta 2.
- Tamaño del Output : (None, 32), correspondiente a las 32 neuronas de esta capa

Capa de Salida (Capa Salida) :

- Descripción: La capa de salida tiene una única neurona y utiliza la función de activación Sigmoide , que es adecuada para problemas de clasificación binaria, ya que convierte el valor de salida en una probabilidad entre 0 y 1.
- Tamaño del Input : (None, 32), proveniente de la salida de la Capa Oculta 3.
- Tamaño del Output : (None, 1), lo que indica que la red produce una sola predicción para cada conjunto de características de entrada.

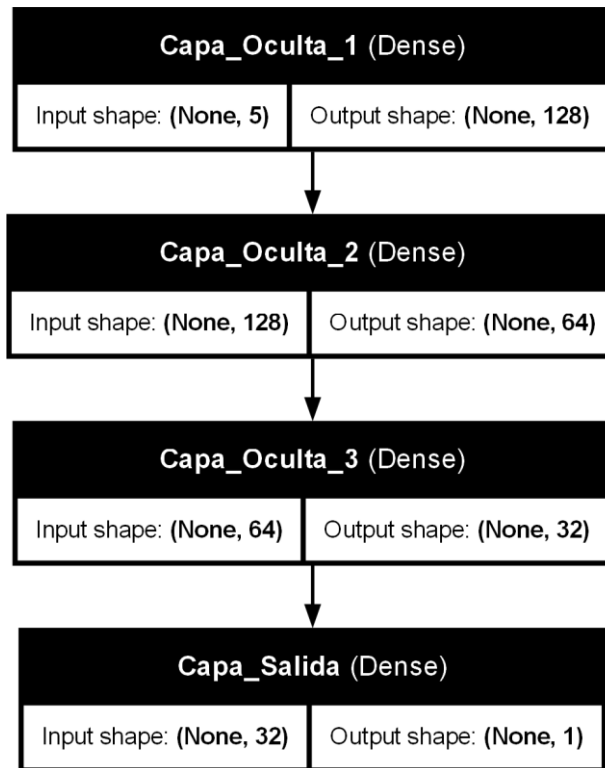


Figura 3.2: Representación Arquitectura Red Deep Neural Network, DNN

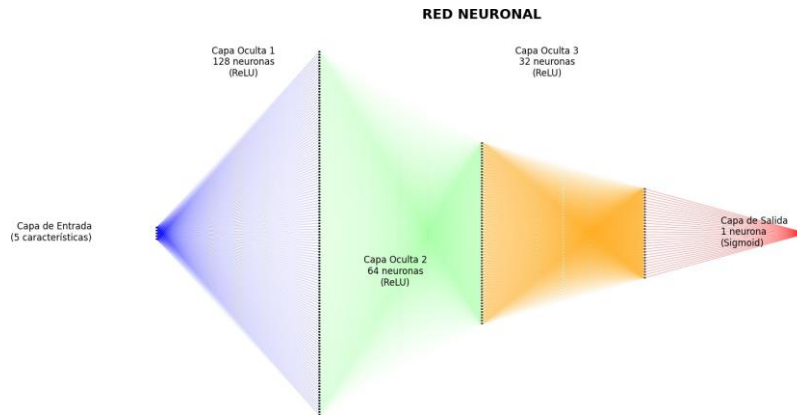


Figura 3.3: Representación Red Deep Neural Network

3.4.3. Evaluación del Modelo

El modelo se evaluará utilizando un conjunto de datos independiente y se medirán métricas de rendimiento como la precisión, recall, y F1-score para asegurar su efectividad.

Métricas de Evaluación

- **Precisión (Accuracy):** Proporción de predicciones correctas.
- **Recall:** Proporción de verdaderos positivos sobre el total de verdaderos positivos y falsos negativos.
- **F1-score:** Media armónica de la precisión y el recall.
- **Área bajo la curva ROC (AUC-ROC):** Medida de la capacidad del modelo para distinguir entre clases.

Validación Cruzada

Se realizó para asegurar que el modelo generalice bien a datos no vistos. Se implementó utilizando la función `cross_val_score` de Scikit-learn.

3.5. Implementación del Modelo

3.5.1. Diseño del Dashboard

El dashboard se diseñó utilizando Plotly Dash de Python para visualizar los resultados de las predicciones y facilitar la interpretación por parte del usuario con las siguientes funcionalidades:

- Subir Archivo.xls con formato predefinido
- Visualizar predicción con gráficos de apoyo según parámetros de seguimiento
- Visualizar que estudiantes necesitan apoyo con margen de calificación parametrizada de 1-10 siendo como 7 el punto de arranque de análisis de cada caso.
- Exportar Gráficos en entregables en formato .pdf para su posterior análisis y archivo.
- Visualización Matriz de Confusión.
- Visualización Curva de ROC.

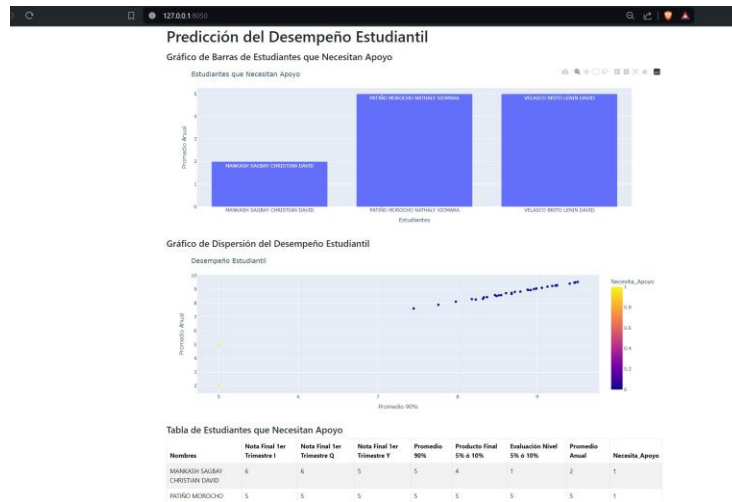


Figura 3.4: Ejemplo Dashboard : Autor. Freddy Quezada

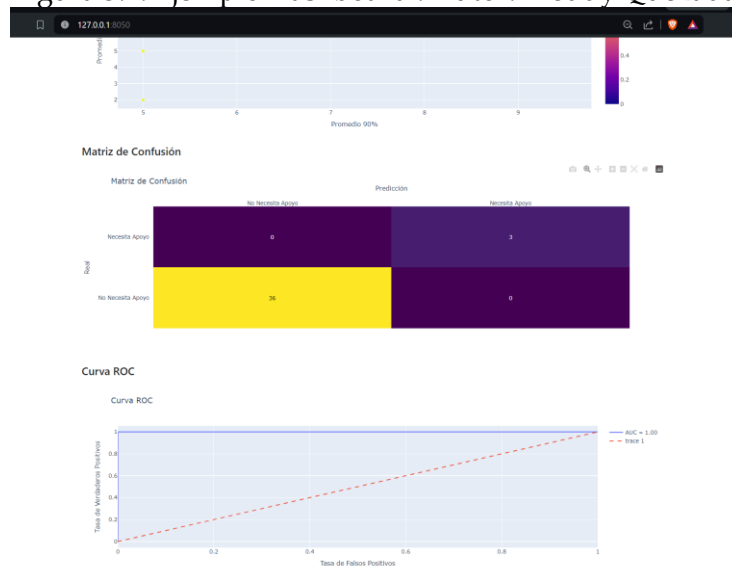


Figura 3.5: Ejemplo de Dashboard Matriz de Confusión - Curva ROC : Autor.Freddy Quezada

Capítulo 4

Implementación y Evaluación

4.1. Implementación del Modelo:

La implementación del modelo se realizó utilizando TensorFlow 2.13.0, una biblioteca de código abierto para aprendizaje automático. La arquitectura del modelo elegido fue una red neuronal densa (feedforward neural network), adecuada para el tipo de datos tabulares como son las calificaciones. El objetivo era predecir si un estudiante necesita apoyo basado en su rendimiento académico.

Código Fuente del Modelo

```
1 from sklearn.metrics import confusion_matrix, roc_curve, auc
2 import matplotlib.pyplot as plt
3
4 # Realizar predicciones
5 y_pred = (model.predict(X_test) > 0.5).astype("int32")
6
7 # Matriz de Confusión
8 cm = confusion_matrix(y_test, y_pred)
9 print('Matriz de Confusión:')
10 print(cm)
11
12 # Curva ROC
13 fpr, tpr, thresholds = roc_curve(y_test, model.predict(X_test))
14 auc_value = auc(fpr, tpr)
15 print(f'AUC: {auc_value:.2f}')
16
17 # Graficar Curva ROC
18 plt.figure()
19 plt.plot(fpr, tpr, label=f'ROC curve (area = {auc_value:.2f})')
20 plt.plot([0, 1], [0, 1], 'k--')
21 plt.xlim([0.0, 1.0])
22 plt.ylim([0.0, 1.05])
23 plt.xlabel('False Positive Rate')
24 plt.ylabel('True Positive Rate')
25 plt.title('Receiver Operating Characteristic')
26 plt.legend(loc="lower right")
27 plt.show()
```

4.1.1. Desafíos Técnicos Superados

- Preprocesamiento de Datos: Un desafío importante fue la normalización de los datos, asegurando que todas las características

tuvieran una escala similar. Se utilizó StandardScaler de scikit-learn para normalizar los datos.

- Manejo de Datos Faltantes: Algunos archivos CSV contenían datos faltantes. Se implementó técnicas de imputación y limpieza de datos para manejar estos casos y evitar que afectaran el entrenamiento del modelo.
- Ajuste de Hiperparámetros: La selección de hiperparámetros óptimos fue crucial para mejorar el rendimiento del modelo. Se realizaron experimentos con diferentes configuraciones para identificar la combinación que ofreciera el mejor desempeño.

4.1.2. Evaluación de Modelo

La evaluación del modelo se realizó utilizando métricas estándar de clasificación, como la precisión (accuracy), la matriz de confusión, la curva ROC y el AUC (Area Under the Curve).

Código Fuente de la Evaluación del Modelo

```
1 from sklearn.metrics import confusion_matrix, roc_curve, auc
2 import matplotlib.pyplot as plt
3
4 # Realizar predicciones
5 y_pred = (model.predict(X_test) > 0.5).astype("int32")
6
7 # Matriz de Confusi n
8 cm = confusion_matrix(y_test, y_pred)
9 print('Matriz de Confusi n:')
10 print(cm)
11
12 # Curva ROC
13 fpr, tpr, thresholds = roc_curve(y_test, model.predict(X_test))
14 auc_value = auc(fpr, tpr)
15 print(f'AUC: {auc_value:.2f}')
16
17 # Graficar Curva ROC
18 plt.figure()
19 plt.plot(fpr, tpr, label=f'ROC curve (area = {auc_value:.2f})')
20 plt.plot([0, 1], [0, 1], 'k--')
21 plt.xlim([0.0, 1.0])
22 plt.ylim([0.0, 1.05])
23 plt.xlabel('False Positive Rate')
24 plt.ylabel('True Positive Rate')
```

```
25 plt.title('Receiver Operating Characteristic')
26 plt.legend(loc="lower right")
27 plt.show()
```

4.1.3. Resultados Obtenidos

Para entender los resultados obtenidos se debe considerar que se trabajó sobre un dataset reducido.

- **Precisión del Modelo:** La precisión alcanzada fue del 96.88 % en el conjunto de entrenamiento y del 100 % en el conjunto de prueba (*se debe considerar que los casos de prueba fueron limitados*), lo que indica un modelo preciso. Sin embargo, no se debe sobre valorar las métricas de prueba, el prototipo podrá demostrar su verdadera efectividad cuando pase a producción en caso que la unidad educativa lo requiera.
- **Matriz de Confusión:** La matriz de confusión mostró que el modelo era capaz de distinguir correctamente entre estudiantes que necesitan apoyo y aquellos que no, con muy pocas falsas alarmas.
- **Curva ROC:** La curva ROC y el valor del AUC demostraron que el modelo tiene una excelente capacidad de discriminación entre las clases.

Capítulo 5

Desarrollo del Dashboard

5.1. Diseño e Interfaz:

El dashboard fue diseñado para ser intuitivo y fácil de usar, facilitando la carga de archivos de datos, la visualización de resultados de predicción y la interpretación de estos resultados. Utilizamos Dash y Dash Bootstrap Components para el diseño y la interfaz de usuario.

Código Fuente del Dashboard

```

1 import pandas as pd
2 import dash
3 from dash import dcc, html, Input, Output, State
4 import dash_bootstrap_components as dbc
5 import plotly.express as px
6 import plotly.figure_factory as ff
7 import plotly.graph_objects as go
8 from tensorflow.keras.models import load_model
9 from sklearn.metrics import confusion_matrix, roc_curve, auc
10 from sklearn.preprocessing import StandardScaler
11 import base64
12 import io
13
14 # Inicializar la aplicación Dash
15 app = dash.Dash(__name__, external_stylesheets=[dbc.themes.BOOTSTRAP])
16
17 # Cargar el modelo
18 model = load_model('my_model.keras')
19
20 # Layout de la aplicación
21 app.layout = dbc.Container([
22     dbc.Row([
23         dbc.Col([
24             dbc.Nav(
25                 [
26                     dbc.NavLink("Inicio", href="#", active="exact",
27                                 className="btn btn-primary mb-1"),
28                     dbc.NavLink("Buscar Archivos", href="#", active="exact",
29                                 className="btn btn-secondary mb-1"),
30                     dbc.NavLink("Pronosticar", href="#", active="exact",
31                                 className="btn btn-success mb-1"),
32                     dbc.NavLink("Borrar", href="#", active="exact",
33                                 className="btn btn-danger mb-1")
34                 ],
35                 vertical=True,
36                 pills=True,
37             ),
38             ], width=2),

```

```

39 dbc.Col([
40     html.H1("Dashboard para Procesamiento de Datos",
41             className="text-center mb-4"),
42     dcc.Upload(
43         id='upload-data',
44         children=html.Div(['Arrastra y suelta o ',
45                             html.A('Selecciona Archivos')]),
46         style={
47             'width': '100%',
48             'height': '60px',
49             'lineHeight': '60px',
50             'borderWidth': '1px',
51             'borderStyle': 'dashed',
52             'borderRadius': '5px',
53             'textAlign': 'center',
54             'margin': '10px'
55         },
56         multiple=True
57     ),
58     dcc.Dropdown(id='file-dropdown',
59                 placeholder='Selecciona un archivo'),
60     dcc.Graph(id='graph_necesita_apoyo'),
61     dcc.Graph(id='scatter_desempeno'),
62     dcc.Graph(id='confusion_matrix'),
63     dcc.Graph(id='roc_curve'),
64     html.Div(id='tabla_necesita_apoyo')
65 ], width=10)
66 ])
67 ], fluid=True)
68
69 @app.callback(
70     Output('file-dropdown', 'options'),
71     Input('upload-data', 'contents'),
72     State('upload-data', 'filename')
73 )
74 def update_file_dropdown(list_of_contents,
75                           list_of_names):
76     if list_of_contents is not None:
77         # Generar una lista de diccionarios que
78         # tienen los nombres de los archivos
79         return [{'label': name, 'value': name}
80                 for name in list_of_names]
81     return []
82
83 @app.callback(
84     [Output('graph_necesita_apoyo', 'figure'),
85     Output('scatter_desempeno', 'figure'),
86     Output('confusion_matrix', 'figure'),
87     Output('roc_curve', 'figure')],

```

```

88 Output('tabla_necesita_apoyo', 'children')],
89 [Input('file-dropdown', 'value'),
90 State('upload-data', 'contents'),
91 State('upload-data', 'filename')]
92 )
93 def update_dashboard(selected_filename ,
94                      list_of_contents , list_of_names):
95 if selected_filename is not None:
96 # Encontrar el contenido del archivo seleccionado
97 content = list_of_contents[list_of_names.
98                             index(selected_filename)]
99 content_type, content_string = content.split(',')
100 decoded = base64.b64decode(content_string)
101 df = pd.read_csv(io.StringIO(decoded.decode('utf-8')))
102 # Preprocesar y visualizar los datos
103 df['Necesita_Apoyo'] = df['Promedio Anual'] < 7
104 scaler = StandardScaler()
105 X = df[['Nota Final 1er Trimestre I',
106        'Nota Final 1er Trimestre Q',
107        'Nota Final 1er Trimestre Y',
108        'Promedio 90', 'Promedio Anual']]
109 X_scaled = scaler.fit_transform(X)
110 df['Probabilidades'] = model.predict(X_scaled).flatten()
111
112 # Actualizar figuras
113 fig_bar = px.bar(df[df[
114     'Necesita_Apoyo']],
115     x='Nombres', y='Promedio Anual',
116     title="Estudiantes que Necesitan Apoyo")
117 fig_scatter = px.scatter(
118     df, x='Promedio 90',
119     y='Promedio Anual',
120     color='Necesita_Apoyo',
121     title="Desempeño Estudiantil")
122 y_true = df['Necesita_Apoyo']
123 y_pred = (df['Probabilidades']
124           > 0.5)
125 cm = confusion_matrix(y_true, y_pred)
126 cm_labels = ['No Necesita Apoyo',
127             'Necesita Apoyo']
128 if cm.shape == (2, 2):
129 fig_cm = ff.create_annotated_heatmap(
130     z=cm, x=cm_labels, y=cm_labels,
131     annotation_text=cm, colorscale='Viridis')
132 fig_cm.update_layout(title="Matriz de Confusión")
133 else:
134 fig_cm = go.Figure()
135
136 fpr, tpr, thresholds = roc_curve(

```

```

137     y_true, df['Probabilidades'])
138 auc_value = auc(fpr, tpr)
139 fig_roc = go.Figure(
140     data=go.Scatter(
141         x=fpr, y=tpr, mode='lines',
142         name='ROC Curve'))
143 fig_roc.add_shape(type='line',
144                 line=dict(dash='dash'),
145                 x0=0, x1=1, y0=0, y1=1)
146 fig_roc.update_layout(
147     title=f"Curva ROC (AUC = {auc_value:.2f})",
148     xaxis_title='False Positive Rate',
149     yaxis_title='True Positive Rate')
150
151 # Generar una tabla con los estudiantes que necesitan apoyo
152 necesitan_apoyo_df = df[df['Necesita_Apoyo']]
153 tabla_html = html.Table([
154     html.Thead(html.Tr([
155         html.Th(col) for col in necesitan_apoyo_df.columns])),
156     html.Tbody([
157         html.Tr([
158             html.Td(
159                 necesitan_apoyo_df.iloc[i][col])
160                 for col in necesitan_apoyo_df.columns
161             ]) for i in range(min(len(necesitan_apoyo_df), 10))
162         ])
163     ])
164
165 return fig_bar, fig_scatter, fig_cm, fig_roc, tabla_html
166
167 return {}, {}, {}, {}
168 "Por favor, seleccione un archivo para visualizar."
169
170 # Ejecutar la aplicaci n
171 if __name__ == '__main__':
172     app.run_server(debug=True, port=8051)

```

5.2. Funcionalidades Clave:

El dashboard ofrece varias funcionalidades clave para facilitar la interacción del usuario y la interpretación de los resultados del modelo:

- Carga de Archivos: Permite al usuario cargar múltiples archivos CSV que contienen los datos de rendimiento estudiantil.
- Selección de Archivos: Un menú desplegable permite al usuario seleccionar el archivo que desea analizar y visualizar.
- Selección de Archivos: Un menú desplegable permite al usuario seleccionar el archivo que desea analizar y visualizar.
- Tabla de Estudiantes que Necesitan Apoyo: Presenta una lista de estudiantes identificados como necesitados de apoyo, facilitando la identificación de estos casos críticos.
- Tabla de Estudiantes que Necesitan Apoyo: Presenta una lista de estudiantes identificados como necesitados de apoyo, facilitando la identificación de estos casos críticos.

Capítulo 6

Análisis y Conclusiones

6.1. Análisis de Resultados:

Los resultados obtenidos indican que el modelo de predicción basado en una red neuronal densa es altamente preciso para identificar a los estudiantes que necesitan apoyo. La alta precisión y el valor del área bajo la curva ROC (AUC) demuestran la efectividad del modelo en este contexto educativo. Estos resultados tienen un impacto significativo, ya que permiten a los educadores identificar y apoyar a los estudiantes en riesgo de manera más efectiva.

6.2. Limitaciones del Estudio:

Aunque los resultados son prometedores, existen ciertas limitaciones en este estudio:

1. **Tamaño de Muestra:** El modelo fue entrenado y evaluado con una cantidad limitada de datos. Un mayor conjunto de datos podría presentar nuevos desafíos de entrenamiento y permitir la mejora de la precisión y la generalización del modelo.
2. **Diversidad de Datos:** Los datos utilizados provienen de una sola institución educativa. Datos de diferentes instituciones podrían ofrecer una perspectiva más amplia y validar la robustez del modelo en distintos contextos.
3. **Factores Externos:** No se consideraron factores externos que podrían influir en el rendimiento estudiantil, como el entorno familiar, el estado socioeconómico o la salud mental. Esto abre nuevas brechas investigativas en la predicción de apoyo académico, en la era de los grandes volúmenes de datos esto puede conducir a aportes mucho más valiosos.

6.3. Conclusiones:

En el desarrollo de este proyecto, se logró seleccionar una arquitectura de modelo de aprendizaje automático que optimizó el desempeño del sistema basado en métricas de evaluación como la matriz de confusión y la curva ROC. Estas herramientas permitieron analizar la capacidad del modelo para clasificar correctamente los casos, identificando la proporción de falsos positivos y negativos, y garantizando un desempeño balanceado. El

análisis mediante la curva ROC proporcionó información detallada sobre el trade-off entre sensibilidad y especificidad, aspectos críticos en problemas de clasificación.

Posteriormente, el modelo fue integrado en un entorno interactivo mediante un dashboard implementado con la tecnología Dash en Python. Este dashboard permitió a los usuarios cargar y procesar datos, así como visualizar gráficamente el desempeño del modelo a través de herramientas como gráficos de barras y dispersión. Adicionalmente, incluyó visualizaciones avanzadas como la curva ROC y la matriz de confusión, que facilitaron una interpretación clara de los resultados y respaldaron la toma de decisiones informadas.

La arquitectura seleccionada, junto con el diseño modular del sistema, aseguró una alta portabilidad y escalabilidad del modelo. Este despliegue no solo cumplió con los objetivos iniciales del proyecto, sino que también sentó las bases para futuras extensiones. Dichas mejoras podrían incluir la incorporación de métricas adicionales como el F1-score y el recall, ampliando así la robustez y el alcance del análisis del sistema.

Capítulo 7

Anexos

7.1. Dataset.Obtenido

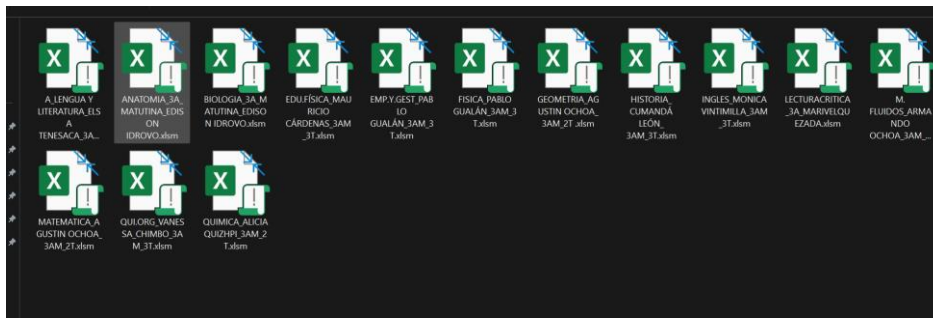


Figura 7.1: Datos Obtenidos Institución Educativa

N°	NOMINA	PROFESOR	MATERIA	NOTA	FECHA	ESTADO	FECHA DE ENTREGA
1	ANGEL MULLA ENRIQUE SUJANA	9.00	2.00	8.00	8.80	DA	2.00 3.00 4.00
2	ALBERTO ALONSO FERNANDA MERIZZI	9.00	8.00	8.20	8.90	DA	8.00 9.00 10.00
3	JAVIER MONCADA JORGE GUANDUPE	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
4	IRISOLVA FERNANDEZ NADIA BELLEN	8.00	8.00	8.20	8.30	DA	8.00 9.00 10.00
5	BERNARDO GUERRA ESTEBAN WOLFE	7.00	5.00	4.00	4.00	DA	8.00 9.00 10.00
6	BERNARDO GUERRA ESTEBAN WOLFE	8.00	4.00	4.50	8.50	DA	2.00 3.00 4.00
7	BRUNO ANTONIO PARRA BARRERA	8.00	8.00	8.00	8.00	DA	8.00 9.00 10.00
8	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
9	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
10	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
11	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
12	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
13	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
14	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
15	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
16	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
17	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
18	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
19	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
20	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
21	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
22	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
23	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
24	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
25	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
26	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
27	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
28	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
29	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
30	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
31	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
32	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
33	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
34	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
35	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
36	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
37	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
38	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
39	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
40	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
41	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
42	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
43	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
44	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
45	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
46	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
47	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
48	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
49	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00
50	BRUNO ANTONIO PARRA BARRERA	7.00	7.00	7.00	7.00	DA	8.00 9.00 10.00

Figura 7.2: DataSet de Entrada

7.2. Dashboard Conversión a .Cvs

Cargar Datos

Arrastra y suelta o Selecciona Archivos

A_LENGUA Y LITERATURA_ELSA TENESACA_3AM_2T.xlsm

Nº	Nombres	Nota Final 1er Trimestre I	Nota Final 1er Trimestre Q	Nota Final 1er Trimestre Y	Promedio 90%	Producto Final 5% ó 10%	Evaluación Ni
1	AGUILAR NAULA JENNIFER JULIANA	8.8	7.37	9.48	8.55	10	
2	ALVAREZ ALARCON FERNANDA MERCEDES	9.29	8.3	9.3	8.96	10	
3	APOLO MENDOZA JORDY LEANDRO	7			7		
4	BACULIÑA FERNANDEZ YADIRA BELEN	8.3	8.24	9.64	8.72	10	
5	BERNEO GUAYAN KERLY NICOLE	6.89	7.2	9.21	7.76	10	

lengua

ANATOMIA_3A_MATUTINA_EDISON IDROVO.xlsm

Nº	Nombres	Nota Final 1er Trimestre I	Nota Final 1er Trimestre Q	Nota Final 1er Trimestre Y	Promedio 90%	Producto Final 5% ó 10%	Evaluación Ni
1	AGUILAR NAULA JENNIFER JULIANA	9.3	7	9.9	8.73	10	
2	ALVAREZ ALARCON FERNANDA MERCEDES	9.5	7.3	9.65	8.81	10	
3	APOLO MENDOZA JORDY LEANDRO	8.5			8.5		
4	BACULIÑA FERNANDEZ YADIRA BELEN	7.4	7.4	9	7.93	10	
5	BERNEO GUAYAN KERLY NICOLE	7.25	7	9.85	8.03	10	

anatomía

MATEMATICA_AGUSTIN OCHOA_3AM_2T.xlsm

Nº	Nombres	Nota Final 1er Trimestre I	Nota Final 1er Trimestre Q	Nota Final 1er Trimestre Y	Promedio 90%	Producto Final 5% ó 10%	Evaluación Ni
1	AGUILAR NAULA JENNIFER JULIANA	8.6	9.18	8.2	8.66	10	
2	ALVAREZ ALARCON FERNANDA MERCEDES	7.67	7.94	8.4	8	10	
3	APOLO MENDOZA JORDY LEANDRO						

Figura 7.3: Dashboard Conversión a .Cvs

7.3. Archivo Cvs

	A	B	C	D	E	F	G	H
4	3	APOLO MENDOZA JORDY LEANDRO						
5	4	BACULIMA FERNANDEZ YADIRA BELEN	7.8	8.01	6.0	7.27	10.0	6.0
6	5	BERMEO GUAMAN KERLY NICOLE	7.55	7.14	7.5	7.39	10.0	7.5
7	6	BERMEO GUAMAN MARIA FERNANDA	7.12	7.81	7.75	7.56	10.0	5.5
8	7	BRAVO IZQUIERDO PAULA ALEJANDRA	9.72	9.94	6.65	9.43	10.0	10.0
9	8	BRAVO JADAN JAHIRA JAMILETH	7.2	9.28	6.0	7.16	10.0	6.0
10	9	CABRERA BORJA ESTEFANIA VALENTINA	5.15	7.02	4.65	5.6	10.0	6.0
11	10	CALE ASTUDILLO NURIA DEL ROCIO	7.87	8.04	8.8	8.23	10.0	7.0
12	11	CHICA ARIAS JUAN ESTEBAN	9.45	9.94	8.65	9.34	10.0	10.0
13	12	CHILQUINGA CURILLO ANGIE MONSERRATH	7.4	8.74	7.35	7.83	10.0	6.0
14	13	CHIMBO REINO VIVIANA PATRICIA	8.22	7.94	8.25	8.13	10.0	6.0
15	14	CONTRERAS ARBITO MATHEW LEONARDO	8.42	9.54	8.9	8.95	10.0	8.0
16	15	CUMBE FEJOO JUAN SEBASTIAN	8.77	8.33	8.25	8.45	10.0	6.0
17	16	DELEG FAJARDO KATHERINE ANDREA	8.22	7.35	7.3	7.62	10.0	5.5
18	17	GUAMAN COCHANELA DOMENICA ALEXANDRA	8.5	9.43	8.5	8.81	10.0	8.5
19	18	GUANUCHI TENESACA ANDRES SEBASTIAN	6.15	7.18	6.85	6.72	10.0	5.5
20	19	JADAN ALVARADO SANTIAGO RAFAEL	5.97	7.42	8.2	7.19	10.0	5.5
21	20	LEON BRAVO DANIA ABIGAIL	8.32	9.04	8.75	8.7	10.0	6.5
22	21	LOOR LEON MERCEDES NICOLE	8.42	8.49	8.8	8.57	10.0	7.0
23	22	LUNA PAREDES PAULA SARAHÍ	9.32	9.99	9.3	9.53	10.0	7.5
24	23	MANKASH SAGBAY CHRISTIAN DAVID	8.22	8.73	8.25	8.4	10.0	6.0
25	24	MONTALEZA CASTILLO JOHN MATEO	10.0	9.39	8.65	9.34	10.0	10.0
26	25	MOROCHO BACULIMA PEDRO FERNANDO	6.15	7.47	7.9	7.17	10.0	7.0
27	26	MUCOZ CUENCA VALENTINA NAYELI	9.38	9.88	8.3	9.18	10.0	6.5
28	27	OCHOA FAICAN SAMUEL DAVID	8.97	9.37	8.3	8.88	10.0	6.5
29	28	ORDOCEZ LEON EDGAR PAUL	7.3	8.27	8.2	7.92	10.0	5.5
30	29	ORTIZ GARATE KAREN ROCIO	7.23	7.36	7.85	7.48	10.0	6.5
31	30	PARRA CARABAJÓ MIGUEL JOSUE	5.05	5.71	7.85	6.2	10.0	6.5
32	31	PATICO MOROCHO NATHALY XIOMARA	8.82	9.94	8.95	9.23	10.0	8.5
33	32	RIVERA GALLEGOS ELIZABETH FERNANDA	7.42	7.32	6.45	7.06	10.0	6.0
34	33	SAREZ YUNGA MELANY ALEXANDRA	8.47	8.28	8.2	8.31	10.0	5.5
35	34	URGILES SUQUI JAIRO FABIAN	7.39	8.07	8.3	7.92	10.0	6.5
36	35	VASQUEZ AVILA KEVIN ANDRES	9.9	9.54	9.55	9.66	10.0	10.0
37	36	VELASCO BRITO LENIN DAVID	7.78	7.42	8.5	7.9	10.0	8.5
38	37	VELECELA SUPACELA GISELLA ANAHI	8.42	7.94	6.5	7.62	10.0	6.5
39	38	YELA VELEZ JOYCE BETSABE	8.72	8.99	7.95	8.55	10.0	7.5
40	39	YUNGA LEON ERICK RAFAEL	8.7	8.42	8.7	8.6	10.0	6.0
41	40	ZHIZHPON FLORES JUAN PABLO	5.3	7.2	6.4	6.3	10.0	5.5

Figura 7.4: Archivo Normalizado Limpiado Filas Y columnas - Caso Matemáticas

```

Run Terminal Help
dashboardVisual.py 7 MATEMATICAS.csv x my_model.keras requirements.txt
Datos > MATEMATICAS.csv > data
1 N, Nombres, Nota Final 1er Trimestre I, Nota Final 1er Trimestre Q, Nota Final 1er Trimestre Y, Promedio 90%, Producto Final 5% ó 10%
2 1, AGUILAR NAULA JENNIFER JULIANA, 8.6, 9.18, 8.2, 8.66, 10.0, 5.5, 8.56
3 2, ALVAREZ ALARCON FERNANDA MERCEDES, 7.67, 7.94, 8.4, 8.0, 10.0, 7.5, 8.07
4 3, APOLO MENDOZA JORDY LEANDRO, , , , , ,
5 4, BACULIMA FERNANDEZ YADIRA BELEN, 7.8, 8.01, 6.0, 7.27, 10.0, 6.0, 7.34
6 5, BERMEO GUAMAN KERLY NICOLE, 7.55, 7.14, 7.5, 7.39, 10.0, 7.5, 7.52
7 6, BERMEO GUAMAN MARIA FERNANDA, 7.12, 7.81, 7.75, 7.56, 10.0, 5.5, 7.57
8 7, BRAVO IZQUIERDO PAULA ALEJANDRA, 9.72, 9.94, 8.65, 9.43, 10.0, 10.0, 9.48
9 8, BRAVO JADAN JAHAIRA JAMILETH, 7.2, 8.28, 6.0, 7.16, 10.0, 6.0, 7.24
10 9, CABRERA BORJA ESTEFANIA VALENTINA, 5.15, 7.02, 4.65, 5.6, 10.0, 6.0, 5.84
11 10, CALE ASTUDILLO NURIA DEL ROCIO, 7.87, 8.04, 8.8, 8.23, 10.0, 7.0, 8.25
12 11, CHICA ARIAS JUAN ESTEBAN, 9.45, 9.94, 8.65, 9.34, 10.0, 10.0, 9.4
13 12, CHILIQUINGA CURILLO ANGIE MONSERRATH, 7.4, 8.74, 7.95, 7.83, 10.0, 6.0, 7.84
14 13, CHIMBO REINO VIVIANA PATRICIA, 8.22, 7.94, 8.25, 8.13, 10.0, 6.0, 8.11
15 14, CONTRERAS ARBITO MATHEW LEONARDO, , 8.42, 9.54, 8.9, 8.95, 10.0, 8.0, 8.95
16 15, CUMBE FEIJOO JUAN SEBASTIAN, 8.77, 8.33, 8.25, 8.45, 10.0, 6.0, 8.4
17 16, DELEG FAJARDO KATHERINE ANDREA, , 8.22, 7.35, 7.3, 7.62, 10.0, 5.5, 7.63
18 17, GUAMAN COCHANCELA DOMENICA ALEXANDRA, 8.5, 9.43, 8.5, 8.81, 10.0, 8.5, 8.85
19 18, GUANUCHI TENESACA ANDRES SEBASTIAN, 6.15, 7.18, 6.85, 6.72, 10.0, 5.5, 6.82
20 19, JADAN ALVARADO SANTIAGO RAFAEL, 5.97, 7.42, 8.2, 7.19, 10.0, 5.5, 7.24
21 20, LEON BRAVO DANNA ABIGAIL, 8.32, 9.04, 8.75, 8.7, 10.0, 6.5, 8.65
22 21, LOOR LEON MERCEDES NICOLE, 8.42, 8.49, 8.8, 8.57, 10.0, 7.0, 8.56
23 22, LUNA PAREDES PAULA SARAHÍ, 9.32, 9.99, 9.3, 9.53, 10.0, 7.5, 9.45
24 23, MANKASH SAGBAY CHRISTIAN DAVID, 8.22, 8.73, 8.25, 8.4, 10.0, 6.0, 8.36
25 24, MONTALEZA CASTILLO JOHN MATEO, 10.0, 9.39, 8.65, 9.34, 10.0, 10.0, 9.4
26 25, MOROCHO BACULIMA PEDRO FERNANDO, 6.15, 7.47, 7.9, 7.17, 10.0, 7.0, 7.3
27 26, MUÑOZ CUENCA VALENTINA NAYELI, 9.38, 9.88, 8.3, 9.18, 10.0, 6.5, 9.08
28 27, OCHOA FAICAN SAMUEL DAVID, 8.97, 9.37, 8.3, 8.88, 10.0, 6.5, 8.81
29 28, ORDOÑEZ LEON EDGAR PAUL, 7.3, 8.27, 8.2, 7.92, 10.0, 5.5, 7.9
30 29, ORTIZ GARATE KAREN ROCIO, 7.23, 7.36, 7.85, 7.48, 10.0, 6.5, 7.55
31 30, PARRA CARABAJÓ MIGUEL JOSUE, 5.05, 5.71, 7.85, 6.2, 10.0, 6.5, 6.4
32 31, PATIÑO MOROCHO NATHALY XIOMARA, 8.82, 9.94, 8.95, 9.23, 10.0, 8.5, 9.23
33 32, RIVERA GALLEGOS ELIZABETH FERNANDA, 7.42, 7.32, 6.45, 7.06, 10.0, 6.0, 7.15
34 33, SAREZ YUNGA MELANY ALEXANDRA, , 8.47, 8.28, 8.2, 8.31, 10.0, 5.5, 8.25
35 34, URGILES SUQUI JAIRO FABIAN, 7.39, 8.07, 8.3, 7.92, 10.0, 6.5, 7.95
36 35, VASQUEZ AVILA KEVIN ANDRES, 9.9, 9.54, 9.55, 9.66, 10.0, 10.0, 9.69
37 36, VELASCO BRITO LENIN DAVID, 7.78, 7.42, 8.5, 7.9, 10.0, 8.5, 8.03
38 37, VELECELA SUPACELA GISELLA ANAHI, , 8.42, 7.94, 6.5, 7.62, 10.0, 6.5, 7.68
39 38, YELA VELEZ JOYCE BETSABE, 8.72, 8.99, 7.95, 8.55, 10.0, 7.5, 8.57
40 39, YUNGA LEON ERICK RAFAEL, 8.7, 8.42, 8.7, 8.6, 10.0, 6.0, 8.54
41 40, ZHIZHPON FLORES JUAN PABLO, 5.3, 7.2, 6.4, 6.3, 10.0, 5.5, 6.44
42

```

Figura 7.5: Archivo Limpio .CSV - Vista Entorno de Desarrollo

7.4. DashBoardVisual.py



Figura 7.6: DASHBOARD EN EJECUCIÓN - DIAGRAMA

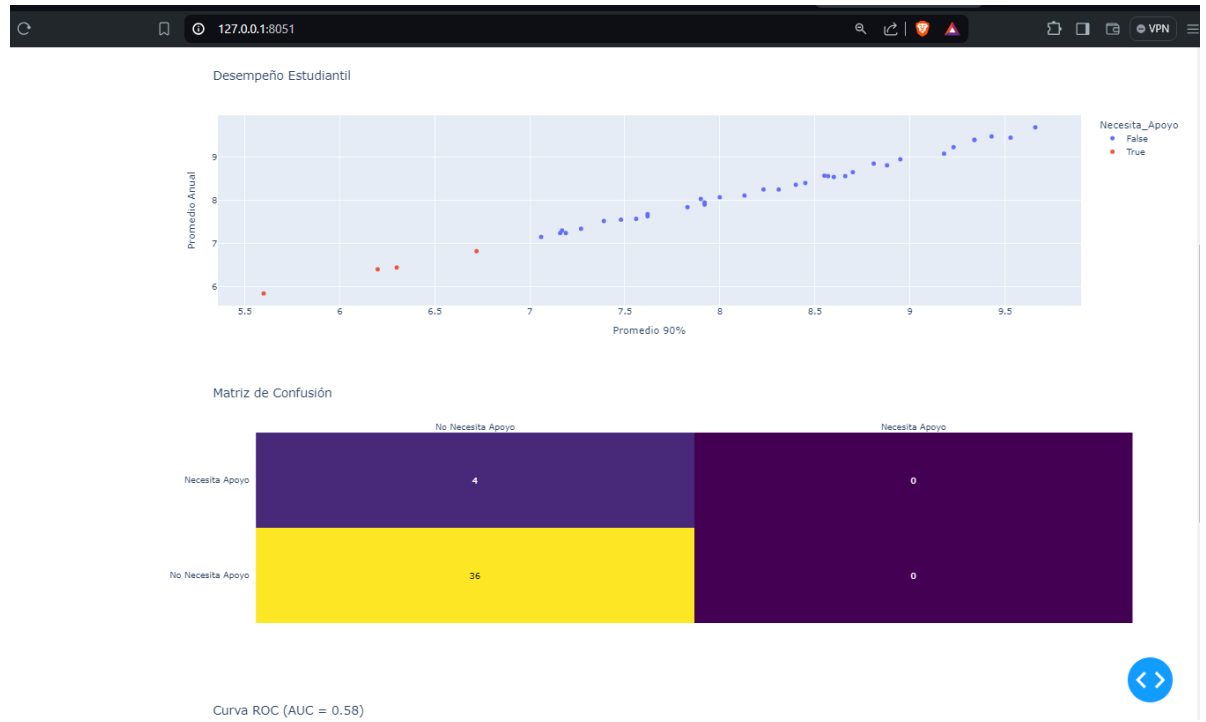


Figura 7.7: DASHBOARD EN EJECUCIÓN - MATRIZ DE CONFUSIÓN



Figura 7.8: DASHBOARD EN EJECUCIÓN - CURVA DE ROC

Bibliografía

- IBM. Crisp-dm help overview. <https://www.ibm.com/docs/es/spss-modeler/saas?topic=dm-crisp-help-overview>, 2021.
- A. Pérez del Barrio, P. Menéndez Fernández-Miranda, P. Sanz Bellón, L. Lloret Iglesias, and D. Rodríguez González. Inteligencia artificial en radiología: introducción a los conceptos más importantes. *Radiología*, 64(3):228–236, 2022.
- Biqian Cheng, Yuping Liu, and Yunjian Jia. Evaluation of students' performance during the academic period using the xg-boost classifier-enhanced aeo hybrid model. *Expert Systems with Applications*, 238:122136, 2024.
- Fatema Alshaikh and Nabil Hewahi. Convolutional neural network for predicting student academic performance in intelligent tutoring system. *International Journal of Computing and Digital Systems*, 15(1):239 – 258, 2024. Cited by: 0; All Open Access, Gold Open Access.
- Parvathy S. Kurup and S.V. Annlin Jeba. *A Secure and Efficient Student Performance Prediction System Based on Blockchain and Data Mining Technologies*. 2024. Cited by: 0.
- Qionghao Huang and Yan Zeng. Improving academic performance predictions with dual graph neural networks. *Complex and Intelligent Systems*, 2024. All Open Access, Gold Open Access.
- S. Lakshmi and C.P. Maheswaran. Effective deep learning based grade prediction system using gated recurrent unit (gru) with feature optimization using analysis of variance (anova). *Automatika*, 65(2):425 – 440, 2024. Cited by: 0; All Open Access, Gold Open Access.
- Didi Supriyadi, Purwanto Purwanto, and Budi Warsito. Optimizing neural networks for academic performance classification using feature selection

and resampling approach. *Mendel*, 29(2):261 – 272, 2023. Cited by: 0; All Open Access, Gold Open Access.

Pallavi Asthana, Sumita Mishra, Nishu Gupta, Mohammad Derawi, and Anil Kumar. Prediction of student's performance with learning coefficients using regression based machine learning models. *IEEE Access*, 11:72732 – 72742, 2023. Cited by: 1; All Open Access, Gold Open Access.

Marina Segura, Jorge Mello, and Adolfo Hernández. Machine learning prediction of university student dropout: Does preference play a key role? *Mathematics*, 10(18), 2022. Cited by: 18; All Open Access, Gold Open Access, Green Open Access.

Nawaf N. Hamadneh, Samer Atawneh, Waqar A. Khan, Khaled A. Almejalli, and Adeeb Alhomoud. Using artificial intelligence to predict students' academic performance in blended learning. *Sustainability (Switzerland)*, 14(18), 2022. Cited by: 9; All Open Access, Gold Open Access.

Chuang Liu, Haojie Wang, and Zhonghu Yuan. A method for predicting the academic performances of college students based on education system data. *Mathematics*, 10(20), 2022. Cited by: 6; All Open Access, Gold Open Access.