UNIVERSIDAD NACIONAL AGRARIA DE LA SELVA

FACULTAD DE INGENIERÍA EN INFORMÁTICA Y SISTEMAS

ESCUELA PROFESIONAL DE INGENIERIA EN INFORMÁTICA Y SISTEMAS

TÉCNICAS DE MACHINE LEARNING PARA PREDECIR EL RENDIMIENTO ACADÉMICO EN LOS ESTUDIANTES DE LA UNIVERSIDAD NACIONAL AGRARIA DE LA SELVA

Tesis

Para optar el título de:

INGENIERO EN INFORMÁTICA Y SISTEMAS

PRESENTADO POR:

MARIA LUCI ZAMORA HERNANDEZ

Tingo María – Perú.

2024

Universidad Nacional Agraria de la Selva Facultad de Ingeniería en Informática y Sistemas	**ACTA DE SUSTENTACIÓN DE TESIS N.° 04-2024**	**COMISION DE GRADOS Y TITULOS** **Fecha:** 03/diciembre/2024

PARTE 1. FASE INICIAL

Siendo las 14:00 **horas del día 3 de diciembre de 2024**; en la Sala de Conferencias de la FIIS, se instala el jurado calificador conformado por:

Jurado 1: Mg. Eudolio Gregorio Vásquez Pinedo (presidente)

Jurado 2: Mg. Ronald Eduardo Ibarra Zapata

Jurado 3: Mg. Brian Cesar Pando Soto

Oficializado mediante **RESOLUCIÓN N° 019-2024-D-FIIS-UNAS** del 13 de marzo de 2024, para el proceso de sustentación del informe final de Tesis del bachiller MARIA LUCI ZAMORA HERNÁNDEZ, titulado: **"TÉCNICAS DE MACHINE LEARNING PARA PREDECIR EL RENDIMIENTO ACADÉMICO EN LOS ESTUDIANTES DE LA UNIVERSIDAD NACIONAL AGRARIA DE LA SELVA".** ASESOR: **Mg. *Rannoverng Yanac Montesino.***
Se manifiesta que el bachiller cumple con los requisitos exigidos de Ley y se le invita a disertar su Tesis por espacio de 30 minutos, asimismo se dispondrá de igual tiempo para la absolver preguntas y sugerencias.

PARTE 2. FASE DE PREGUNTAS Y RESULTADO

Culminada la exposición se inicia la fase de preguntas por parte del jurado calificador; también se invita a los asistentes a formular preguntas sobre el tema de Tesis.
Absueltas todas las peticiones, el jurado calificador procede a deliberar en privado la calificación y resultado.
Concluida la deliberación y en presencia del público, el jurado calificador anuncia que el resultado de la Sustentación de Tesis es: APROBADO POR UNANIMIDAD.
(NOTA: consignar una de la siguientes: DESAPROBADO, APROBADO POR MAYORIA o APROBADO POR UNANIMIDAD)
Con calificativo de: BUENO
(NOTA: consignar una de la siguientes: EXCELENTE, MUY BUENO, BUENO, DEFICIENTE, MUY DEFICIENTE)
Por lo que se comunicará a las instancias correspondientes para el trámite respectivo.

PARTE 3. CONFORMIDAD

De todo lo mencionado se firma al pie en señal de conformidad, siendo las 16:15 horas se da por finalizada la ceremonia de Sustentación de Tesis.

Firma:	Firma:	Firma:
Jurado 1: EUDOLIO G. VASQUEZ P.	**Jurado 2:** BRIAN C. PANDO SOTO	**Jurado 3:** RONALD IBARRA ZAPATA
Firma:	Firma:	
Sustentante:	**Asesor:** Rannoverng Yanac M.	

"Año de la recuperación y consolidación de la economía peruana"

CERTIFICADO DE SIMILITUD T.I. N° 263 - 2025 - CS-RIDUNAS

El Jefe de la Unidad de Gestión de Investigación de la Universidad Nacional Agraria de la Selva, quien suscribe,

CERTIFICA QUE:

El Trabajo de Investigación; aprobó el proceso de revisión a través del software TURNITIN, evidenciándose en el informe de originalidad un índice de similitud no mayor del 25% (Art. 3° - Resolución N° 466-2019-CU-R-UNAS).

Programa de Estudio:

Ingeniería en Informática y Sistemas

Tipo de documento:

Tesis	X	Trabajo de Suficiencia Profesional	

TÍTULO	AUTOR	PORCENTAJE DE SIMILITUD
TÉCNICAS DE MACHINE LEARNING PARA PREDECIR EL RENDIMIENTO ACADÉMICO EN LOS ESTUDIANTES DE LA UNIVERSIDAD NACIONAL AGRARIA DE LA SELVA	MARIA LUCI ZAMORA HERNANDEZ	10 % Diez

Tingo María, 07 de agosto de 2025

UNIVERSIDAD NACIONAL AGRARIA DE LA SELVA
UNIDAD DE GESTIÓN DE LA INVESTIGACIÓN

Dr. Tomas Menacho Mallqui
JEFE

C.C. Archivo

UNIVERSIDAD NACIONAL AGRARIA DE LA SELVA

FACULTAD DE INGENIERÍA EN INFORMÁTICA Y SISTEMAS

ESCUELA PROFESIONAL DE INGENIERIA EN INFORMÁTICA Y SISTEMAS

TÉCNICAS DE MACHINE LEARNING PARA PREDECIR EL RENDIMIENTO ACADÉMICO EN LOS ESTUDIANTES DE LA UNIVERSIDAD NACIONAL AGRARIA DE LA SELVA

Autor : **Maria Luci Zamora Hernandez**

Asesor (es) : **Mg. Rannoverng Yanac Montesino**

Programa de investigación : **Ingeniería de Software**

Línea de investigación : **Computación**

Eje temático : **Inteligencia Artificial**

Lugar de ejecución : **Universidad Nacional Agraria de la Selva**

Duración : **24 meses**

Financiamiento : **propio**

Tingo María – Perú. 2024

DEDICATORIA

A Dios por ser mi guía eterno, por no soltar mi mano en ningún momento. Todo lo logrado es fruto de tu infinita misericordia y amor.

A mi madre, por ser mi luz, por todo el amor y apoyo brindado, por sus enseñanzas de fe, humildad, lucha y constancia, por su fortaleza para superar cualquier obstáculo en la vida con una sonrisa, por demostrarme que el verdadero amor no puede debilitarse por la distancia. Siempre vivirás, porque de mi mente y mi corazón no te has ido.

A mi papá, mi héroe silencioso, su amor y apoyo han sido fundamentales en mi crecimiento personal y académico.

A mis hermanas Nanci, Aurea, Karina y Melisa y a mi hermano Oscar; por creer en mí, por siempre tener una palabra de aliento. Son mi inspiración.

A Walter por su amor, por su apoyo incondicional y por estar a mi lado en todo momento.

A mis amigos y compañeros de estudio, Roy, Luz y Roberto, gracias por su apoyo constante y su amistad sincera que valoro profundamente. Me llena de alegría que continúen formando parte de mi vida.

AGRADECIMIENTOS

A mi asesor el Mg. Yanac Rannoverng Montesino por sus consejos, correcciones y apoyo quien con sus conocimientos permitió el desarrollo de esta investigación.

Al Dr. Alfredo Daza Vergaray por su disposición para ayudarme, por compartir su experiencia las cuales han sido significativas para la realización de esta investigación.

Al Ing. José Victor Lizárraga Trebejo por su motivación y apoyo desinteresado en cualquier consulta respecto a mi tema de estudio.

Al Ing. Jorge Luis Jara Linares por brindarme la información solicitada de los estudiantes de la UNAS.

A mi alma mater, la Universidad Nacional Agraria de la selva, en especial a la Facultad de Ingeniería en Informática y Sistemas por abrirme las puertas y darme la oportunidad de lograr mi formación académica.

A todos mis profesores por impartir sus enseñanzas y haber contribuido con sus conocimientos los cuales han constituido la base de mi vida profesional.

ÍNDICE

ÍNDICE DE TABLAS

Tabla Página

ÍNDICE DE FIGURAS

Figura Página

RESUMEN

El objetivo de esta investigación es determinar en qué porcentaje las técnicas de machine learning permiten predecir el rendimiento académico de los estudiantes de la UNAS, mediante las métricas de evaluación: exactitud, precisión, sensibilidad, especificidad, puntuación F1 y Curva ROC, con el fin de poder identificar a los alumnos con probabilidad de éxito o fracaso de acuerdo con una escala de calificación. En este estudio se hizo uso de una población de 4584 estudiantes, por lo que se usó la totalidad de la población como muestra. Asimismo, el estudio es de tipo aplicada, enfoque cuantitativo, alcance descriptivo, y diseño de investigación experimental de tipo pre-experimental de un solo grupo, esto porque luego de aplicar las técnicas de machine learning se podrá observar los resultados y realizar la medición. Para la creación de los modelos predictivos, se siguieron los pasos de la metodología KDD, utilizando Python como lenguaje de programación y Jupyter Notebook de la suite Anaconda como interfaz de desarrollo. Para visualizar los resultados, se desarrolló un prototipo en el entorno de desarrollo Spyder. Los resultados confirman la validez de las técnicas de machine learning para predecir el rendimiento académico de los estudiantes de la UNAS. Por lo que se destaca que la técnica con mejor resultado en este contexto fue Redes Neuronales con los siguientes valores: Exactitud= 96.24%, sensibilidad = 96.23%, y puntuación F1 = 96.21%. No obstante Árbol de decisión obtuvo los mejores valores de: Precisión= 95.72%, y especificidad = 95.68%, a la vez que SVM obtuvo el valor óptimo de Curva ROC=99.42%.

Palabras Clave:

Machine learning, Rendimiento académico, KDD, Estudiantes, UNAS.

ABSTRACT

The objective of this research is to determine in what percentage machine learning techniques allow predicting the academic performance of UNAS students using evaluation metrics such as accuracy, precision, recall, specificity, F1 score, and ROC curve, in order be able to identify students with a probability of success or failure according to a grading scale. In this study, a population of 4584 students was used, so the entire population was used as a sample. Likewise, the study is of an applied type, quantitative approach, descriptive scope, and design of experimental research of pre-experimental type of a single group, this because after applying machine learning techniques it will be possible to observe the results and make the measurement. For the creation of predictive models, the KDD methodology was followed, using Python as the programming language and Jupyter Notebook from the Anaconda suite as the development interface. To visualize the results, a prototype was developed in the Spyder development environment. The results confirm the validity of machine learning techniques to predict the academic performance of UNAS students. Therefore, it is highlighted that the technique with the best result in this context was Neural Networks with the following values: Accuracy = 96.24%, sensitivity = 96.23%, and F1 score = 96.21%. However, Decision Tree obtained the best values of: Precision = 95.72%, and specificity = 95.68%, while SVM obtained the most optimal value of ROC Curve = 99.42%.

Keywords:

Machine learning, Academic performance, KDD, Students, UNAS.

I. INTRODUCCIÓN

Hoy en día, las técnicas de machine learning son objeto de estudio por diversos investigadores en el ámbito educativo para diferentes tareas, entre ellos la resolución de problemas como la deserción de estudiantes, la selección de cursos, apoyo en actividades relacionadas con pasantías y muchos otros problemas similares (Henriquez, Salcedo y Sanchez, 2022).

La educación es un tema de estudio de relevancia mundial dado su rol fundamental en el desarrollo de un país, por lo tanto, se debe prestar vital atención a los resultados del desempeño de los estudiantes y los factores que tienen influencia en esos resultados, la introducción de nuevas tecnologías está generando mayor interés en el sector educativo puesto que está conllevando a la búsqueda de herramientas que posibiliten la predicción del rendimiento académico estudiantil. (Olortegui, 2024).

En España, el rendimiento académico en la educación superior es preocupante, casi el 30% de los estudiantes no logra terminar la carrera universitaria en la que se matriculó; esto se debe a diferentes factores como son: falta de orientación y preparación previa de los estudiantes; diseño inadecuado de los planes de estudios, escaso acompañamiento en el proceso formativo; bajo rendimiento académico de los estudiantes debido a limitaciones en sus habilidades, dedicación o desmotivación. (RRHHDigital, 2021)

En América Latina, los expertos han venido señalando que la educación tiene serias deficiencias, y esto ha sido comprobado por muchos años en los informes que la Organización para la Cooperación y el Desarrollo Económicos (OCDE) ha elaborado sobre este tema, en dichos informes se puede constatar que la región está muy por debajo de los estándares mundiales de rendimiento académico. En el país de Argentina se observa elevadas tasas de bajo desempeño académico (entre el 60% y el 80% en los últimos años), considerando que el bajo rendimiento académico tiene asociación con elevados índices de deserción estudiantil, así como también es afectado por una variedad de factores heterogéneos, de naturaleza interna y externa, que inciden en el desempeño del estudiante.

A nivel nacional en una investigación realizada en una universidad del Perú, muestra que aproximadamente el 75% logra aprobar el curso en el que se matriculan, sin embargo, hay un conjunto de programas académicos, donde la proporción de estudiantes que aprueban el curso es notablemente inferior, cuyas escuelas vinculadas a las ciencias básicas e ingenierías destacan con bajos rendimientos académicos, con tasas de desaprobados (No se presentaron, insatisfactorio y en proceso) que oscilan entre el 30% al 39% . (PISA, 2018)

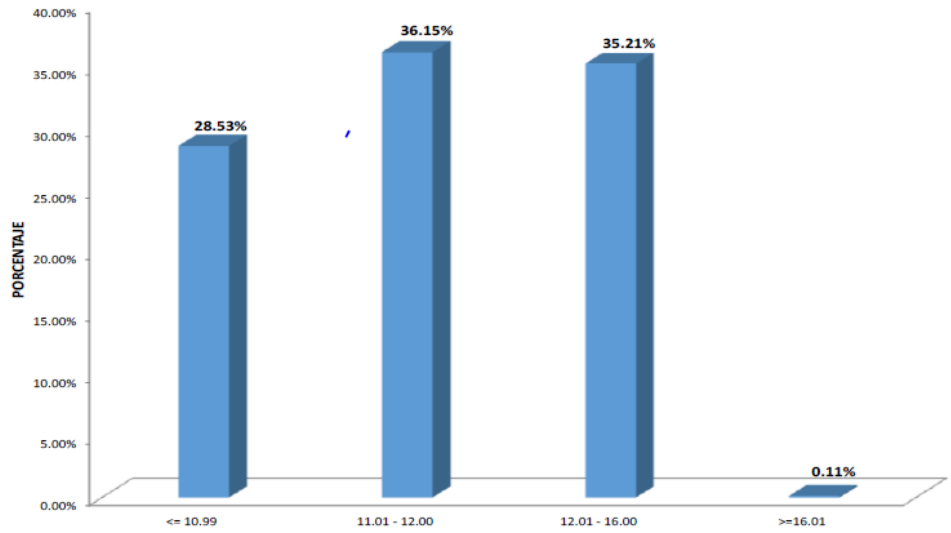
En la Universidad Nacional Agraria de la Selva (UNAS), el rendimiento académico se evalúa mediante criterios establecidos en los sílabos de cada asignatura; de acuerdo con el (Reglamento General de la UNAS, 2017) las calificaciones se expresan en una escala vigesimal, que va de cero a veinte puntos, siendo 11 la nota mínima necesaria para aprobar (Ver Tabla 1)

Tabla 1. *Escala de Calificaciones*

Calificación	Nota
Excelente	20, 19 y 18
Muy Bueno	17,16 y 15
Bueno	14 y 13
Regular	12 y 11
Desaprobado	10 o menos

Fuente: Reglamento General de la UNAS, 2017

Figura 1 *Rendimiento académico por nota promedio acumulado de los estudiantes de la UNAS, año 2014*



Fuente: Oficina de Coordinación y Desarrollo Académico (OCDA). Extraído de (Arrascue, 2015)

Según datos proporcionados por OCDA, se observa que el 28.53% de los alumnos de la UNAS presenta un rendimiento académico inferior a 11; el 36.15% obtiene un promedio entre 11 y 12 en la nota promedio acumulada; el 35.21% registra un promedio alrededor del 12.01 y 16 en la nota promedio acumulada, mientras que solo el 0.11% se sitúa en el quinto superior (de 0 a 20), indicando un promedio superior a 16 (Ver Figura 1). Esto altos porcentajes de bajo

rendimiento académico demuestra la grave situación que atraviesan los estudiantes, esto se debe a diferentes factores o aspectos relacionados al rendimiento académico, estos resultados muchas veces tienen efectos que pueden ser: la pérdida de beneficios como el acceso a becas dentro de la institución, dentro del estado peruano (PRONABEC) o becas internacionales, etc.; así como también, puede traer consigo el retraso en el tiempo de culminación de la carrera, la desaprobación de los cursos y finalmente abandonar completamente sus estudios.

Debido al panorama actual referente al rendimiento académico se formuló la siguiente pregunta general: ¿Existe diferencia en las métricas de evaluación de las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS?

Así como también las siguientes preguntas específicas:

¿Existe diferencia en la exactitud de las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS?

¿Existe diferencia en la precisión de las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS?,

¿Existe diferencia en la sensibilidad de las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS?

¿Existe diferencia en la especificidad de las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS?

¿Existe diferencia en la puntuación F1 las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS?

¿Existe diferencia en la curva ROC de las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS?

Este estudio se justifica en bases teóricas sólidas, al analizar comparativamente el desempeño predictivo de las técnicas de machine learning en el rendimiento académico, como medio para identificar a estudiantes que presentan riesgo de fracaso, y los resultados se contrastan con estudios previos. Sumado a esto, contribuye al conocimiento del comportamiento de diversas variables relacionadas con el rendimiento académico y busca profundizar en este tema a través de teorías, además de sugerir recomendaciones para futuros estudios.

De igual modo, este estudio tiene una relevancia práctica significativa, al abordar un problema real relacionado con la predicción del rendimiento académico de los estudiantes de la UNAS, por lo cual el desarrollo de técnicas de machine learning no solo permite una identificación más precisa de los estudiantes con riesgo de bajo rendimiento académico, sino que también permite implementar estrategias preventivas que mejoren su desempeño académico. De este modo, este tipo de estudio contribuye al fortalecimiento de la calidad educativa y al desarrollo académico, generando un efecto positivo tanto al bienestar personal de los estudiantes como en el contexto educativo que los rodea.

Además, este estudio tiene relevancia social, ya que contribuye al bienestar de la población estudiantil de la UNAS al ayudar a la identificación del rendimiento académico, esto, a su vez, permite a la universidad ofrecer diversas alternativas para dar soporte al desarrollo profesional de los estudiantes, buscando elevar su calidad de vida. Por último, desde una perspectiva tecnológica, este estudio se justifica al crear un modelo predictivo que podría ser útil como base para la realización de un software de apoyo para la universidad, destinado a identificar el rendimiento académico de manera efectiva

Este estudio buscó lograr predecir el rendimiento académico de estudiantes de la UNAS, donde para alcanzar este objetivo se tomaron en cuenta los promedios ponderados de los alumnos que se encuentran registrados en el período académico 2021-II, extraídos de la base de datos de la universidad. A estos datos se les aplicó las técnicas de machine learning, las cuales fueron comparadas para identificar la más efectiva mediante diversas métricas de evaluación.

1.1. Objetivos

1.1.1. Objetivo General

Comparar las técnicas de machine learning para predecir el rendimiento académico de los estudiantes de la UNAS.

1.1.2. Objetivos Específicos

Evaluar la diferencia en la exactitud de las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS

Evaluar la diferencia en la precisión de las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS

Evaluar la diferencia en la sensibilidad de las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS

Evaluar la diferencia en la especificidad de las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS

Evaluar la diferencia en la puntuación F1 de las técnicas de Machine Learning en la en la predicción del rendimiento académico de los estudiantes de la UNAS

Evaluar la diferencia en la curva ROC de las técnicas de Machine Learning en la en la predicción del rendimiento académico de los estudiantes de la UNAS.

1.2. Hipótesis

1.2.1. Hipótesis General

Existe diferencia significativa en las métricas de evaluación de las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS.

1.2.2. Hipótesis Específicas

Existe diferencia estadística significativa en la exactitud de las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS.

Existe diferencia estadística significativa en la precisión de las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS.

Existe diferencia estadística significativa en la sensibilidad de las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS.

Existe diferencia estadística significativa en la especificidad de las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS.

Existe diferencia estadística significativa en la puntuación F1 de las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS.

Existe diferencia estadística significativa en la curva ROC de las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS.

II. REVISIÓN DE LITERATURA

En esta sección se procederá a definir las teorías relacionadas que engloba las variables de estudio, asimismo se conceptualizará las dimensiones, los cuales serán recopilados de bases de datos indexadas, tales como Scopus, ScienceDirect y Web of Science, así como de libros, con relación a los 8 últimos años.

2.1. Marco teórico

2.1.1. Machine learning

Según Geron (2019), es aquella ciencia (y el arte) de programar computadoras capaces de aprender automáticamente a partir de la información proporcionada.

Por otra parte, Veliz (2020), afirma que el machine learning comprende diversas técnicas orientadas a crear un modelo basado en un conjunto de datos, sin la necesidad de una programación personalizada para cada problema. Esto facilita la resolución de tareas como la predicción de casos nuevos y la explicación de resultados.

2.1.2. Tipos de Machine Learning

Las técnicas de machine learning suelen clasificarse en dos principales categorías, denominadas aprendizaje supervisado y no supervisado y dos secundarias que son el aprendizaje semi supervisado y el aprendizaje por refuerzo (Subasi, 2020).

2.1.2.1.Aprendizaje supervisado

Para Watt et al (2020), este tipo de aprendizaje implica relaciones de entrada/salida. Aplicable a una amplia gama de situaciones y tipos de datos, este tipo de problema se dividen en: regresión y clasificación, esto dependerá de la forma numérica general de la salida.

Regresión

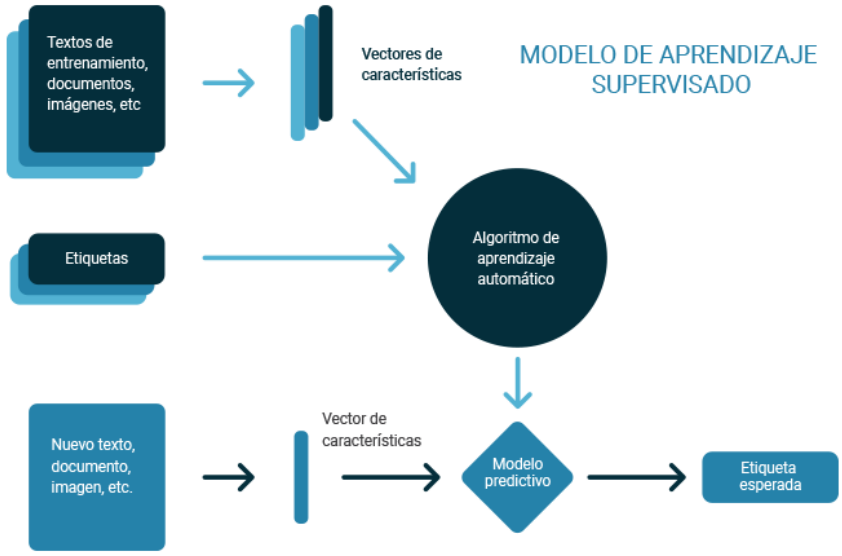
Implica ajustar un modelo utilizando un conjunto de datos de entrenamiento, permite realizar predicciones sobre un resultado de valor continuo. Esto significa predecir un valor numérico que se tiene de objetivo, como por ejemplo el valor de un vehículo puede estimarse a partir de variables predictoras como la antigüedad, la marca o el kilometraje recorrido (Watt et al., 2020).

Clasificación

En principio, es similar a la de la regresión, con la diferencia clave de que, en lugar de predecir un valor continuo, la clasificación predice valores discretos o clases. (Watt et al., 2020)

Según Geron (2019), las técnicas de aprendizaje supervisado de mayor importancia son: Redes neuronales, K Vecinos más cercanos, Regresión Logística, Regresión lineal, SVM (Máquinas de vectores de soporte), bosques aleatorios y Árboles de decisión. Se debe tener en cuenta que algunas técnicas de regresión pueden utilizarse también para la clasificación, y viceversa.

Figura 2 *Aprendizaje supervisado*



Fuente: (Gonzalez, 2020)

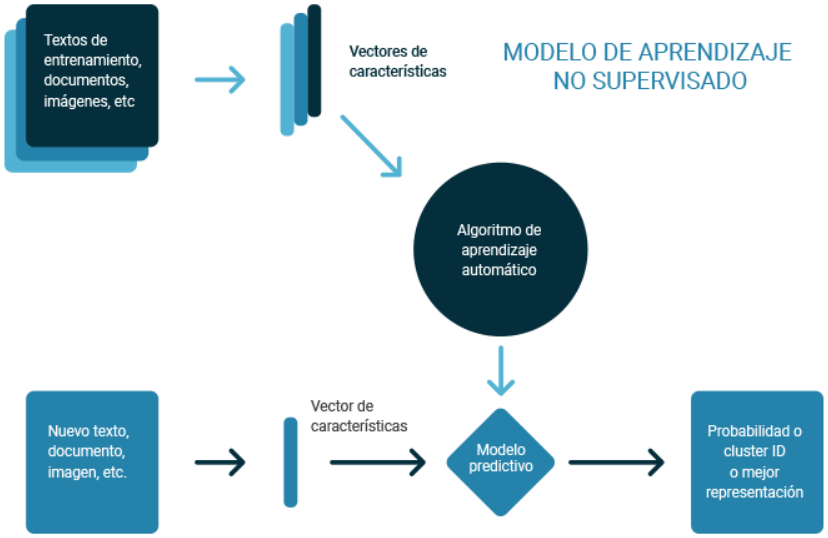
En la Figura 2, se presenta el diagrama de flujo del aprendizaje supervisado, Este proceso inicia con un conjunto de datos de entrenamiento (documentos o imágenes) que ya ha sido etiquetado y organizado previamente. Al ingresar nueva información (texto, imagen o documento), el algoritmo adquiera experiencia al entrenar el modelo para clasificar las muestras de entrada mediante la comparación entre las predicciones generadas y las etiquetas reales correspondientes a cada muestra. Finalmente, el modelo realiza ajustes en función de los errores en la estimación de los resultados para obtener la etiqueta esperada.

2.1.2.2.Aprendizaje no supervisado

Según Contreras, Fuentes y Rodríguez (2020), es un método que aprende de ejemplos simples sin una respuesta y su asociación, en otras palabras, organizando información de grupos de rasgos similares a datos iniciales, no siendo este un pronóstico.

Por otro lado, Veliz (2020), menciona que contrariamente a las técnicas supervisados, en este modelo no existen supervisión de los resultados ni variable dependientes; estos modelos tienen como objetivo encontrar patrones que describan las interacciones entre variables.

Figura 3 *Aprendizaje sin supervisión*



Fuente: Medium (Gonzalez, 2020)

En la Figura 3, se presenta el diagrama de flujo del aprendizaje sin supervisión, este proceso comienza con un conjunto de textos de entrenamiento (documentos o imágenes), los cuales no están clasificados ni etiquetados. Luego, se ingresan los vectores de características similares entre sí, y finalmente, proceder a realizar ajustes al modelo en función a las observaciones para obtener los resultados esperados.

2.1.3. Técnicas de machine learning

Los métodos del aprendizaje automático (como regresión, clasificación, agrupamiento, detección de anomalías, etc.) se utilizan para construir datos de entrenamiento o modelos matemáticos utilizando ciertos algoritmos basados en las estadísticas de los cálculos para hacer predicciones sin programación, porque las técnicas son influyentes. Al hacer que el sistema sea futurista, modela e impulsa la automatización de todo con mano de obra y costos reducidos. (Pedamkar, 2022)

Para llevar a cabo esta investigación se emplearon cinco técnicas de machine learning, entre ellos: Máquinas de Vector de Soporte (SVM), Arboles de decisión, Vecinos más Cercanos (KNN), Redes Neuronales y Redes Bayesianas. La selección de estas técnicas se basó en varios criterios fundamentales. En primer lugar, se consideraron estudios previos que abordaron problemas similares. Tras la revisión exhaustiva de diferentes trabajos de investigación (ver detalle en el **Anexo 4**). En segundo lugar, se tomaron en cuenta las características del conjunto de datos. En tercer lugar, se consideró la familiaridad con estas técnicas, lo que facilitó un

análisis detallado y comprensible de los resultados. Se aseguró que las técnicas elegidas fueran manejables dentro del marco temporal y de recursos del trabajo de investigación.

2.1.3.1.Árboles de decisión

Es una técnica que busca la creación de un árbol inverso que permite separar los datos en dos partes en base al valor de la diferencia más significante de las variables de entrada. Además, puede generar nodos adicionales, donde la cantidad de condiciones del árbol dependerá de las variables; hay varios criterios de selección de atributos que se pueden utilizar para determinar los atributos que caracterizan los nodos internos del árbol de decisión. Entre los criterios más comunes se encuentra la ganancia de información la cual se basa en el concepto de entropía. Dada una colección S de objetos, con c clases (Radhwan, Abbas y Ali, 2017), la entropía de S utiliza la fórmula que se presenta a continuación:

$$Entropy(S) = \sum_{i=1}^{c} -\text{Pi}\ log_2\ Pi,$$

Donde: Pi es la representación de la proporción de ejemplos en S, que pertenecen a la clase i.

Utilizando la entropía, la ganancia de información de un atributo A en un conjunto de datos S se define como:

$$Gain\ (S, A) = \text{Entropy (S)} - \sum_{\text{v}\in\ \text{Values(A)}} \frac{|\text{Sv}|}{|\text{S}|} Entropy\ (\text{Sv})$$

Donde: Valores (A) denota el conjunto de posibles valores que puede asumir el atributo A, y Sv representa el subconjunto de S que contiene las instancias en las cuales el atributo A toma el valor v..

2.1.3.2.Redes neuronales

Son un modelo simple que se componen de una o más capas de entrada y salida, que va variando según la problemática que tenga y solucionarlo, interconectadas en cada neurona. (Durdevic, 2017). Así mismo, Sánchez y García (2017), indican que una red neuronal, es aquella técnica basada en conjunto de neuronales simples (artificiales), que tiene la capacidad de mostrar cada una de las probables interacciones entre variables de predicción, imitando el comportamiento de las neuronas del cerebro.

El uso de redes neuronales en machine learning, ayuda a la utilización de considerables volúmenes de información y relación no lineal entre predictores, demostrándose en la

enseñanza que es muy beneficioso para la clasificación de resultados. (Lau, Sun y Yang, 2019), que utiliza la siguiente fórmula:

$$Z = Sesgo + W_1X_1 + W_2X_2 + \cdots + W_nX_n$$

Donde: Z es el símbolo para la denotación de la representación gráfica anterior de ANN, Wis, son los pesos o los coeficientes beta, Xis, son las variables independientes o las entradas, y Sesgo o intercepción = W_0.

2.1.3.3.Máquinas de Vector de Soporte (SVM)

Es una técnica usada en regresión, donde la línea recta es trazada con datos de la recta, es decir los puntos cercanos al hiperplano, y las técnicas buscan disminuir el error del monto real y el previsto, y se encuentran: núcleo lineal, función de base radial (RBF) y polinomio. (Dabhade et al, 2021)

Así mismo, es un modelo de machine learning para predecir, clasificar los algoritmos, analizando datos de gran magnitud con características de predicción, este modelo es considerado para margen máximo y labores de clasificación de información de clases N-dimensional hiperplano y gran distanciamiento con vectores de soporte (Sana et al, 2020). Se representa de la siguiente manera:

Si Y_i= +1; wx_i+b≥0 (i)

Si Y_i= -1; wx_i+b≤0 (ii)

Para todos i; Y_i (<w,x_i>+b)≥0 (iii)

Donde: x representa un punto vectorial y w es el peso. Para que los datos sean considerados válidos, deben cumplir con los siguientes requisitos: (i) presentar valores mayores que cero; (ii) incluir casos en los que los datos sean menores que cero; y (iii) garantizar que el hiperplano de separación respete las restricciones establecidas por el conjunto de ejemplos.

2.1.3.4.Redes Bayesianas

Según Alloghani et al (2019), afirman que las redes bayesianas están basadas en el teorema de bayes, proporcionando una posibilidad con condiciones que suceda un acontecimiento en un estudio, explicando cómo influye el resultado en el aprendizaje del programa, y calculando con ecuaciones.

También, es definido como un método de clasificar con datos generales, donde los rasgos son independientes y sin relación, es así como una particularidad específicamente de una clase no es afectado por el estado de otro. (Sisodia, 2018)

$$P\left(\frac{X}{Y}\right) = \frac{P\left(\frac{Y}{X}\right) x\ P(X)}{P(Y)}$$

Donde: P (X/Y) representa la probabilidad condicional posterior, P(X) corresponde a la probabilidad previa de la clase, P (Y) indica la probabilidad a priori de la clase, mientras que P (Y/X) hace referencia a la probabilidad condicional del predictor.

2.1.3.5.Vecinos más Cercanos (KNN)

Es un método de clasificación supervisada, que se basa en aportar un híper parámetro K, y de esta manera pronosticar la clase, como un grupo puede ser perteneciente a la información solicitada en cuestión y cual está más cerca a la mayoría, enfocándose en búsqueda de coeficiente de función y optimización. (De la Hoz y Fontalvo, 2019), que hace uso de la fórmula:

$$d\ (x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Por tanto, se establece que la distancia entre los puntos 'x' e 'y' equivale a la suma de las diferencias en los valores de 'y' al ser restados, lo que permite obtener el valor de 'x' dentro de la dimensión correspondiente.

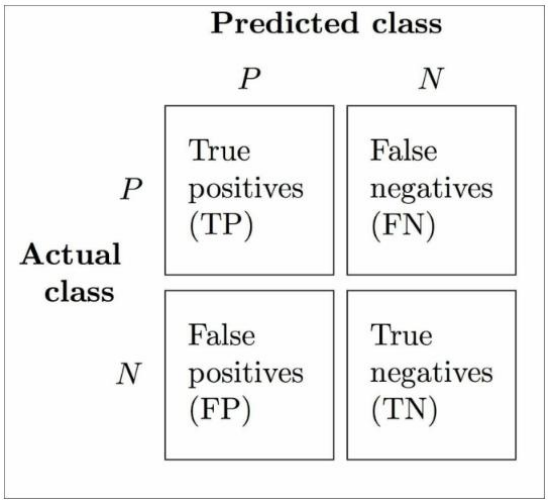
2.1.4. Métricas de Evaluación de técnicas de Machine Learning

Sirven para evaluar el desempeño de las técnicas de machine Learning entre ellas tenemos:

2.1.4.1.Matriz de confusión

Se destaca como una de las métricas más accesibles y comprensibles empleadas para evaluar la precisión y exactitud de un modelo, cuya aplicación se centra en problemas de clasificación que involucran dos o más categorías de salida, cabe resaltar que esta matriz abarca 2 aspectos clave: “actual” y “predicción” , así como conjuntos de clases en ambas dimensiones, por lo que en términos sencillos, las filas corresponden a las clases reales u observadas, mientras que las columnas indican las clases predichas por el modelo (Raschka & Mirjalili, 2017), la cual es representado como se muestra a continuación:

Figura 4 *Matriz de confusión*



Fuente: (Geron, 2019)

En la Figura 4 Se presenta en la matriz de confusión, donde True Positives, son las referentes a las predicciones positivas que realmente son positivos para la clase; True Negatives, son las predicciones negativas que concuerdan correctamente con la realidad de la clase False Positives, son las predicciones positivas que resultan ser incorrectas respecto a la clase; y False Negatives, son predicciones negativas que resultan ser incorrecta con respecto a la clase.

2.1.4.2.Exactitud (Accuracy)

(Esposito & Esposito, 2020) Indican cuán frecuentemente el modelo predictivo lo realiza de manera adecuada, incluyendo las positivas y las negativas. Se calcula mediante la siguiente fórmula:

$$\text{Exactitud} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{FP} + \text{FN} + \text{TN})}$$

2.1.4.3.Precisión (precisión)

Según Nieto et al (2018), la precisión se refiere a la relación de instancias positivas que han sido predichos adecuadamente al total de casos positivos predichos. Se define como:

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

Una baja precisión indica la presencia de un gran número de falsos positivos.

2.1.4.4.Sensibilidad (Recall or sensitivity)

Según Nieto et al., (2018), La sensibilidad es calculado como la relación del número de instancias que fueron predichas de manera correcta sobre el número total de positivos. Se calcula de la siguiente manera:

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

2.1.4.5.Especificidad (Specificity)

Para Musso, Rodríguez y Cascallar (2020), La especificidad es definida como la relación de no objetivos adecuadamente establecidos, de cada uno de no objetivo auténticos en agrupaciones. Se calcula mediante la siguiente fórmula:

$$\text{Especificidad} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

2.1.4.6.Puntuación F1 (F1-Score)

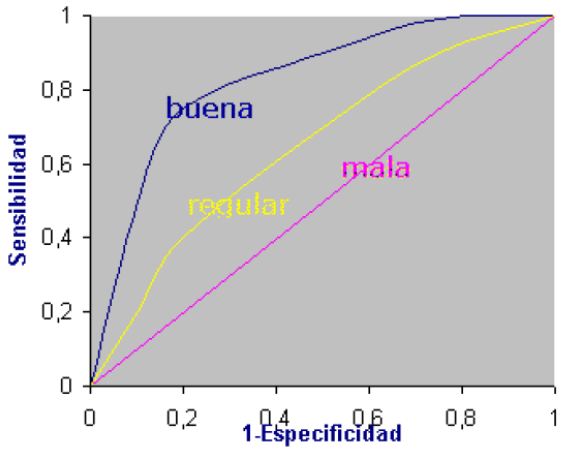
Geron (2019), menciona que la Puntuación F1 se mide a partir de los valores de sensibilidad y precisión (Valor doble de la precisión multiplicado por la sensibilidad dividido por el valor de la suma de sensibilidad y precisión); unifica la sensibilidad y precisión en una sola métrica y se calcula de la siguiente manera:

$$\text{Puntuación F1} = \frac{2 * (\text{Precision} * \text{Sensibilidad})}{\text{Precision} + \text{Sensibilidad}}$$

2.1.4.7. Curva ROC

Se trata de una representación de forma gráfica que ilustra la relación entre la tasa de verdaderos positivos (TPR) como la tasa de falsos positivos (FPR). Está determinada por los valores de especificidad y sensibilidad en cada posible punto de corte utilizado para evaluar los resultados del estudio. En una curva ROC, el eje x normalmente indica la tasa de falsos positivos (mostrado como FPR) y el eje y muestra la tasa de verdaderos positivos (TPR). Así, un valor más alto en el eje x (hacia la derecha) indica una tasa de falsos positivos más alta, mientras que un valor más alto en el eje y (hacia arriba) indica una tasa de verdaderos positivos más alta. También encontraremos el AUC (área bajo la curva) cuyos valores oscilan entre 1 (prueba perfecta) y 0,5 (prueba inútil). (Del Valle, 2017)

Figura 5 *Curva ROC*



Fuente: (Geron, 2019)

Como se presenta en la Figura 5 con respecto a la Curva ROC, donde se calcula el TPR y el FPR para distintos valores de retención, donde el FPR corresponde a la proporción de instancias negativas que han sido clasificadas erróneamente como positivas. Esta métrica equivale a uno menos la tasa de verdaderos negativos, la cual señala la proporción de casos negativos identificados de forma correcta como tales, así como la TNR también denominada especificidad. Por lo que, la curva ROC representa la sensibilidad frente a 1 - especificidad. Un modelo ideal presentaría una curva que atraviesa el punto (0, 1), lo cual indica un TPR de 1 (100% de verdaderos positivos) y un FPR de 0 (0% de falsos positivos).

2.1.5. Metodologías de Machine Learning

KDD, SEMMA y CRISP son metodologías enfocadas en procesos de minería de datos, cada una con sus propias fortalezas y limitaciones en relación con su aplicación en el entorno actual, siendo cada una desarrolladas para abordar distintos tanto tipos de procesos como de soluciones de una estrategia o un problema (Nitola, 2023), tal como se muestra en la Tabla 2.

Tabla 2. *Comparación de las Metodologías KDD, CRISP y SEMMA*

	KDD	CRISP	SEMMA
Enfoque	Enfocado en la detección de patrones óptimos para una tarea específica	Dirigido al cumplimiento de metas y necesidades del negocio	Centrado en el desarrollo estructurado del proceso de data mining
Uso	Productos orientados a la identificación de patrones en los datos	Metodología de acceso libre y sin restricciones de uso	Asociado a herramientas y soluciones de SAS

Metodología	Metodología de patrones arquitectónicos orientado a datos	Metodología de gestión de proyectos	Metodología aún no definida
Complejidad	Más complejo de implementar que los otros dos, ya que implica el desarrollo de un número significativo de fases.	Es el más sencillo de comprender e implementar, cuenta con una curva con alta capacidad de adaptación, lo que permite que sea accesible para desarrolladores con distintos niveles de experiencia y entornos de desarrollo	Es sencilla y bastante dinámica, con fases más alineadas al desarrollo ágil
# de fases de desarrollo	9	6	6
Siglas	Knowledge Discovery in Databases (Descubrimiento en bases de datos)	Cross-industry Standard Process (Proceso estándar en minería de datos)	Sample, Explore, Modify, Model and Access (Muestreo, análisis y modelado)
Relevancia actual	Baja	Alta	Media

Fuente: Información extraída del trabajo de Nitola (2023)

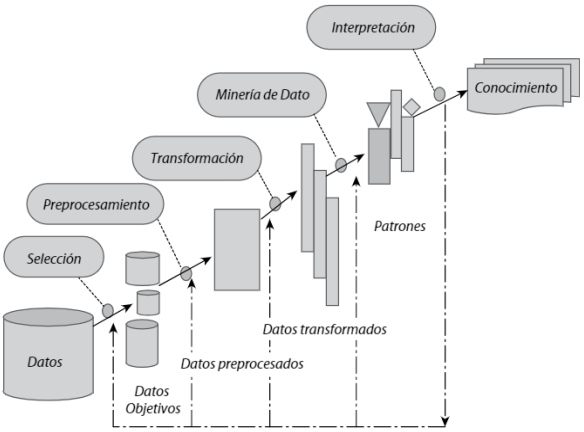
2.1.5.1.Metodología KDD

El proceso KDD o Knowledge Discovery in Databases, es aquella que usa base de datos junto con cada elección, con muestreos y evoluciones, aplicando minería de datos para detallar esquemas y examinar sus resultados, (Flores et al,2019)

Según Moreira y Ruete (2020), es definido como un procedimiento de saberes de data, en donde tiene que ver con minería de datos y la implementación de algoritmos.

Para la presente investigación se utilizará la metodología KDD (Ver Figura 6).

Figura 6 *Etapas del proceso KDD*



Fuente: (Geron, 2019)

A continuación, se procede a presentar una breve descripción de cada una de las fases que comprende la metodología KDD: (Moreira y Ruete (2020)

Fase I- Datos: Se trabaja en comprender el dominio de aplicación, tomando en cuenta las reglas, limitantes y todas las consideraciones necesarias para comprender el contexto y lograr una definición precisa de los objetivos del proyecto. (Moreira y Ruete (2020)

Fase II- Selección: En esta etapa, se lleva a cabo la elección de los conjuntos de datos que se analizarán, enfocándose en aquellos que son relevantes según la definición de los objetivos del proyecto. (Moreira y Ruete (2020)

Fase III- Pre-procesamiento: Se lleva a cabo la limpieza y pre-procesamiento de los datos, el cual se excluyen datos atípicos que puedan distorsionar los resultados, y se establecen estrategias para tratar datos faltantes. (Moreira y Ruete (2020)

Fase IV- Transformación: Se realiza la transformación de los datos, que puede incluir procesos como la discretización, donde se convierten valores numéricos en categóricos para lograr mejorar la precisión de las técnicas de aprendizaje automático, asimismo se realiza la reducción de datos con la meta de mejorar la calidad y especificidad de la información. (Moreira y Ruete (2020)

Fase V- Minería de datos: Acá, se hace la aplicación de data mining para identificar patrones o relaciones en el proceso, basándose en la hipótesis que se pretende probar. (Moreira y Ruete (2020)

Fase VI- Interpretación: Dentro de esta etapa, se interpreta la información extraída y se evalúan los patrones identificados, verificando el rendimiento obtenido en relación con los objetivos establecidos. (Moreira y Ruete (2020)

Fase VII: Conocimiento: En esta última fase, se llevan a cabo acciones que implican la aplicación del conocimiento adquirido en las etapas anteriores. (Moreira y Ruete (2020)

2.1.6. Rendimiento Académico

Según García, Pino y Muñoz (2019), es considerado como la evaluación del estudiante y es el principal criterio para hacer la determinación del éxito o el fracaso del alumno, es decir es una de las partes imprescindibles al momento de contrastar los logros alcanzados durante el desarrollo del proceso educativo.

No obstante, un factor significativo para lograr un buen rendimiento es el comportamiento de la data de la función del internet para pronosticar y lograr un buen rendimiento. (Xing et al,2019).

2.1.7. Predicción

Es una proposición de lo que sucederá en el futuro, es decir resulta de un procedimiento fundado, relacionándolo con el valor, una respuesta a una interrogante de sucesos en años. (Guillán, 2016)

Según Villareal (2016), es una valoración que puede ser cualitativa como cuantitativa de uno o más factores establecidos en el futuro, tomando en cuenta datos de la actualidad.

2.1.8. Predicción del rendimiento académico

Para predecir la data de los estudiantes, estos se deben convertir de texto a números para implementación de perspectivas del machine learning, utilizando técnicas para procesar datos de grado de importancia y calidad, de acuerdo con normas y funciones. (Negi y Jaiswal, 2016), por otra parte, se toma en cuenta factores demográficos, (Fernandes et al., 2019; Nithya et al., 2016), así como el promedio acumulado de materias y exámenes internas (cuestionarios y trabajos en clase) y socioeconómico. (Burgos et al., 2019).

De la misma forma, para lograr predecir el rendimiento académico, es importante saber cuáles son las variables indispensables para ayudar a predecir eficazmente. (Adelantado et al, 2018; Cao et al., 2018; Feng et al, 2019; Zhang et al., 2018)

2.1.9. Python

Python es definido como un leguaje de codificación de nivel alto, de ordenación fácil, con características que lo convierten en concreto en procedimientos de enseñar y aprender para programar (Shukla y Parmar, 2016).

Es un lenguaje moderno de programación de interpretación, de nivel superior, para muchas plataformas y de tipo dinámico con varios paradigmas, con escritura y programación de código. (Bahit, 2018; Sánchez et al. 2018)

Además, uno de los beneficios de este lenguaje de programación radica en que funciona en todo dispositivo, de forma fácil y distinguida que otros tipos de softwares (Prokopyev et al., 2020).

2.1.10. Anaconda

Es una distribución libre de Python creada para el manejo de grandes volúmenes de data, análisis predictivo y computación científica (Müller & Guido, 2017).

2.1.11. Jupyter Notebook

Se trata de una plataforma web de código abierto que permite la creación y portabilidad de documentos que tienen integrados código, fórmulas matemáticas e imágenes. Ofrece funcionalidades como la transformación y visualización de datos, modelado de aprendizaje, simulaciones numéricas, análisis estadístico, entre otras capacidades (Ionos, 2019).

2.1.12. Spyder

Se trata de un entorno integrado para el desarrollo de software (IDE) gratuito que viene con anaconda que incluye edición, pruebas y depuración en una única GUI, utilizado como un editor de Python. Spyder también soporta interesencia mediante introspección o reflexión. (Naik & Oza, 2019)

2.1.13. Escala de medición Razón

En este tipo de escala de medición, el valor 0 representa una ausencia completa de la variable medida, donde los valores se pueden ser representadas utilizando números enteros o decimales positivos, lo que facilita la clasificación y comparación de las mediciones, así como la realización de operaciones aritméticas. (Oyola-García, 2021)

2.2. Estado del Arte

En esta parte del documento, se presentará una revisión de investigaciones previas en distintos contextos geográficos: a nivel internacional, nacional y local, relacionadas con el uso de técnicas de aprendizaje automático en la predicción del rendimiento académico de los estudiantes. El enfoque principal es identificar investigaciones y hallazgos realizados por

diversos autores, los cuales se recopilarán de base de datos de renombre como Web of science, IEEE Xplore, Springer y Dialnet. También se incluirán tesis relacionadas con esta área de estudio, limitándolo a los últimos 7 años de publicaciones.

2.2.1. Internacionales

Buenaño, Gil y Luján (2019), analizaron un caso sobre la aplicación de machine learning en el pronóstico del desempeño de estudiantes universitarios de ingeniería informática, la finalidad del estudio fue predecir las puntuaciones de los estudiantes en base a ciertas características , la metodología utilizada fue elegir a un conjunto de 1,600 alumnos con características similares, usar el algoritmo supervisado para después realizar el procedimiento experimental, finalmente se tuvo como resultado que solo 4 alumnos son los que aprobaron las materias hasta 6to ciclo(6%) lograron avanzar el 62% de los cursos del plan, usando la técnica estructuras de datos, la tasa de alumnos desaprobados disminuyó en el semestre 2016-1 del 35% al 17% para 2018-1, mostrando lo efectivo que es el aprendizaje automático, concluyendo que se alcanzó optimizar las calificaciones y el desempeño académico de los alumnos, asimismo bajar la tasa de absentismo al final del periodo, demostrando la efectividad del método.

Harvey y Kumar (2019), ejecutaron un modelo práctico para que los docentes pronostiquen el rendimiento de los alumnos en machine learning, el objetivo fue la creación de un modelo de categorización para identificar con exactitud lo que afecta el rendimiento de los alumnos, entre la metodología se usó datos públicos de Massachusetts, escuelas del departamento de la web se descartaron datos innecesarios, enfocándose más en calificaciones de pruebas, para saber la correlación de variables, mostrándose diagramas, a la vez se utilizó 3 modelos: la regresión lineal, árbol de decisión y el Clasificador bayesiano ingenuo, teniendo como resultado que este último, mostró una exactitud de 71%, siendo la más alta y efectiva respecto a los puntajes de matemáticas, como conclusión se llegó a que el Clasificador bayesiano ingenuo es muy importante para predecir otras formas de desempeño de los estudiantes y mejorar su aprendizaje, como también evaluar las correlaciones de los datos .

Lau, Sun y Yang (2019), realizaron un artículo, en donde presentaron un enfoque que combina el análisis estadístico convencional con la modelización y predicción del rendimiento de los estudiantes mediante redes neuronales, cuya muestra fue de 1,000 estudiantes, así como el ANOVA, y el T-Test, la técnica utilizada fue la red neuronal artificial, el cual fue modelado con 11 variables de entrada, dos capas de neuronas ocultas y una capa de salida, donde se evaluaron las métricas de: error, exactitud, sensibilidad, especificidad, precisión y AUC. Cuyos resultados muestran que el algoritmo logró los siguientes valores: Exactitud=84.8%,

Error=15.2%, Sensibilidad= 94.8%, Especificidad= 54.6%, Precisión= 86.3% y AUC=86.00%, por lo que se concluye que la configuración de los modelos educativos a través de redes neuronales sirve para realizar una buena evaluación del rendimiento de los estudiantes.

Yousafzai, Hayat y Afzal (2020), aplicaron machine learning y minería de datos en el pronóstico del rendimiento de escolares, donde tuvieron como objetivo hacer el análisis sobre la calidad de la educación y predecir las notas a partir de datos académicos, para ello se trabajó con un grupo de escolares de arte se utilizó la técnica del machine learning supervisado, a la vez los datos fueron recopilados de la Junta Federal de Educación Intermedia y Secundaria Islamabad Pakistán, y se pasó a procesar, utilizando el árbol de decisión y modelo de regresión, donde el sistema pronosticó el grado y marcas (SSC-1, SSC-2, HSSC-1 y HSSC-2.), los resultados muestran que el algoritmo genético basado en el clasificador árbol de decisión alcanzó el mejor valor de exactitud del 96.64%, siendo superior a los otros algoritmos, por lo que se evidencia lo importante que es usar el machine learning para la predicción del desempeño de los estudiantes, ya que los estudiantes que tomaron un examen SSC1 para el próximo año será SSC2,lo que muestra un avance, concluyendo que es una buen instrumento para analizar y predecir notas de las asignatura, para comparar con lo del año anterior, siempre y cuando sea alumnos del mismo grupo.

Singh y Pal (2020), desarrollaron un modelo híbrido que combina los algoritmos de Bagging y Boosting con el objetivo de predecir el rendimiento estudiantil, para ello utilizaron diversas técnicas de machine learning como: KNN, el Clasificador bayesiano ingenuo, Árbol de decisión y árboles extremadamente aleatorio, para lo cual se hizo uso de un conjunto de datos de 1000 instancias y 22 atributos. donde los resultados reportaron una exactitud de 86,83%, Sensibilidad de 81,55% y Puntuación F1 de 72,78% utilizando el Clasificador bayesiano ingenuo, sin embargo, Agregación de arranque alcanzó el mejor valor de exactitud =91.76%. En consecuencia, se determina que la aplicación de técnicas de machine learning contribuye a mejorar la calidad de la educación superior, permitiendo además identificar a los estudiantes con bajo desempeño y prestarles mayor atención para mejorar su rendimiento.

Katarya et al (2021), llevaron a cabo un análisis sobre el uso de machine learning enfocado en la predicción del desempeño académico del educando, la finalidad fue conocer cual técnica es la más conveniente y que características se toman en cuenta para predecir el rendimiento académico, entre la metodología usada fue hacer una encuesta sobre las técnicas usadas en bastantes trabajos, para predecir las calificaciones se usó Análisis de Gartner con el Modelo de ascendencia, con datos puntuados por estudiantes en un curso STEM, y 8 algoritmos, entre ellos la regresión lineal y el Clasificador bayesiano ingenuo, soporte de vectores,

clasificador híbrido. Teniendo como resultado que el que pronosticó mejor fue el de regresión lineal, se pudo concluir que ese método es usado bastante en áreas educativas y datos exactos y de manera general, siendo favorable para las investigaciones y tiene efectos directos para estudiantes en dificultades que tuvieran con la finalidad de mejorar y lograr su crecimiento, de manera más eficiente.

Khan et al (2021), efectuaron una investigación centrada en la predicción del rendimiento estudiantil mediante el uso de técnicas de machine learning destacando sus posibles beneficios, el propósito del estudio fue examinar el comportamiento de los estudiantes e identificar las características de preferencias de las personas que lo utilizan con el fin de predecir el rendimiento estudiantil en diferentes ambientes pedagógicos. Entre la metodología presentada en el estudio fue proponer una variedad de conceptos para catalogarlos como latentes o dinámicos, para esto se aplicó una red neuronal artificial con un alumno que fue supervisado con una serie de datos y comparando el rendimiento de los estudiantes para un curso basado en el Selector de funciones de correlación (CFS), siendo este el que mide la correlación de Pearson con cada características con valores bajos, teniendo como resultados que con el modelo se consigue una exactitud del 52%, eliminando las características latentes y mayor rendimiento, obteniendo un valor bajo, concluyendo que los algoritmos del machine learning son importantes para asegurar un mayor aprendizaje a través de técnicas, siendo la dinámica buena y necesaria la aplicación de la red neuronal artificial, usando atributos con mayor correlación.

Yağcı (2022), planteó en su investigación un modelo apoyado en algoritmos de aprendizaje automático, con el propósito de hacer predicciones sobre las calificaciones de los exámenes finales de los universitarios, utilizando como fuente de datos 1854 estudiantes matriculados en la asignatura de Lengua Turca-I de una universidad estatal de Turquía en el semestre 2019-2020 y el uso de 6 técnicas de machine learning: Bosque aleatorio, Redes neuronales, Maquina de Vectores, Regresión Logística, el Clasificador bayesiano ingenuo y KNN, obteniendo como resultados que este último obtuvo las mejores puntuaciones de las métricas: Exactitud y Sensibilidad alcanzados del 69,90%, precisión del 69,10% y F1-Score del 69,4%, lo que significa que las técnicas de machine learning Bosque aleatorio, Redes neuronales y Maquina de vectores tuvieron mejores resultados, siendo estos eficientes para predecir el rendimiento académico.

2.2.2. Nacionales

Menacho (2017), su objetivo fue aplicar las técnicas de minería de datos (TMD) para la predicción de las notas finales de los alumnos inscritos en una materia de estadística, cuya

información es correspondiente a los historiales académicos localizados en la Oficina de Estudios de la UNALM, para lo cual se tomó en cuenta una muestra de 914 alumnos inscritos en el periodo 2013 II y 2014 I en la materia denominada Estadística. En esta investigación, las técnicas de minería de datos fueron las redes neuronales, árboles de decisión, redes bayesianas y regresión logística, cabe resaltar que para evaluar las TMD, en el estudio se propone el uso de una matriz de confusión, coeficiente Kappa y curva ROC. Los resultados lograron determinar que la red del Clasificador bayesiano ingenuo arrojó una mayor precisión, obteniendo el 71,0% de correcta clasificación y una Curva ROC de 62,0%; el cual muestra un grado de satisfacción aceptable en las técnicas, esto al ser mayor a 0.5. Concluyendo que las TMD han demostrado ser métodos eficaces para generar modelos que sirven como predictivos del rendimiento de los estudiantes que se encuentran inscritos en Estadística General.

Yamao (2018), centró su investigación en la predicción del rendimiento académico de los estudiantes que se matricularon en Ingeniería de Computación y Sistemas en la Universidad San Martin de Porres durante el primer ciclo haciendo uso de técnicas de data mining, el cual se recopilaron información de 1,304 nuevos estudiantes, los cuales se clasificaron en tres categorías: factores académicos, sociales y económicos. Asimismo, se llevaron a cabo predicciones mediante tres métodos: Máquina de vector de soporte, regresión lineal y árbol de decisión. El resultado más destacado fue árbol de decisión, con una precisión=82.87%. Entre los diversos factores en cuanto al rendimiento académico fueron: Edad, puntuación en el examen de admisión, la distancia entre su lugar de residencia y la institución educativa y género. A través de las técnicas de minería de datos, se logró pronosticar el rendimiento académico de nuevos ingresantes y los que pudieran tener dificultades entre su estancia universitaria.

Orihuela (2019), realizó un trabajo de investigación que tuvo como objetivo la predicción del rendimiento académico de los alumnos de la Universidad Nacional del Centro del Perú, dentro de la escuela de Ingeniería de Sistemas, mediante la aplicación de Ciencia de datos; el tipo y nivel de investigación utilizada es tecnológica - explicativa causal, de diseño pre-experimental, la población se conformó de estudiantes en curso de Ingeniería de Sistemas del periodo académico 2016-I al 2019-I, para la recolección de información se hizo uso del cuestionario el mismo que se aplicó a los alumnos de 6to semestre de dicha escuela profesional, además que para modelar los datos se utilizó dos técnicas de machine learning: Bosque aleatorio y Regresión logística; los resultados de la investigación arrojan que el modelo Bosque aleatorio alcanzó los mejores valores (Test) de los indicadores, siendo estos: Curva ROC=82.00%, Precisión= 76.00%, Sensibilidad=76.00% y Puntuación F1=76%. Por lo que se concluye que se logró el pronóstico del nivel académico de los alumnos de Ingeniería de Sistemas.

Candia (2019), orientó su investigación a la predicción del rendimiento académico de los alumnos de la Universidad Nacional de San Antonio Abad del Cusco (UNSAAC), se enmarcó como no experimental, cuantitativa y correlacional, cuya población en estudio estuvo conformada por los alumnos que ingresaron a la UNSAAC, durante el periodo 2014-I y 2018-I con una muestra de 12698 alumnos ingresantes, utilizó la metodología CRISP-DM, también hizo uso de los cinco técnicas que fueron: J48, Bosque aleatorio, Perceptrón multicapa, KNN, y Función de Regresión Logística, a través de su análisis se permitieron determinar que Random Forest tuvo el mejor valor de precisión= 69.35%, así como Regresión Logística=68.33%; Concluyendo que es posible la predicción del rendimiento académico mediante la información de admisión a la UNSAAC, logrando una efectividad del 69% haciendo uso de técnicas de machine learning.

Espinoza & León (2020), los autores en su investigación tuvieron como finalidad el mejoramiento del procedimiento de clasificación de los alumnos del centro de idiomas haciendo uso machine learning; su investigación corresponde al tipo aplicada y se desarrolló bajo un diseño pre-experimental; la población se constituyó por los estudiantes de idiomas matriculados en el año 2018; cuyos instrumentos utilizados en la investigación fueron un cuestionario y entrevista, asimismo se hizo uso de la metodología CRISP. Utilizó la técnica de regresión logística. Los resultados determinaron que se logró reducir el Tiempo Promedio de clasificación de los alumnos en base a su rendimiento académico en un 74.60%; como también se vio reflejada el aumento de la Exactitud =82.08%, finalmente se visualizó incrementos con referente al grado de satisfacción del personal= 71.35%, asimismo del alumno= 30%, haciendo uso de la técnica predictiva. El estudio concluye que el modelo de Machine Learning ayudó a cumplir los objetivos del estudio.

Garcia (2021), en su estudio tuvo como objetivo fue la determinación de que porcentaje las técnicas de machine learning ayudan en la predicción de rendimiento académico. Dentro de la metodología usada, se compararon 3 técnicas que fueron los siguientes: KNN, Árbol de decisión y Maquina de Vectores, asimismo se tomó en cuenta una muestra de 87 estudiantes, a la vez que la investigación fue experimental, aplicada y pre-experimental. Los resultados muestran que la técnica Maquina de Vectores (SVM) alcanzó los mejores valores de las métricas estudiadas (precisión, especificidad y sensibilidad), siendo esta de 100.00%. El autor llega a concluir que las técnicas de Machine Learning son eficientes para la predicción del desempeño académico del nivel superior, por ende, la técnica que logró los valores óptimos fue SVM.

Aronés (2021), en su trabajo de investigación, cuyo objetivo fue diseñar un enfoque de machine learning para realizar la predicción del desempeño académico de alumnos de ingeniería, asimismo el estudio fue descriptivo, transversal, retrospectivo y observacional, en la cual la población y la muestra estuvieron integradas por el historial de los estudiantes de ingeniería dentro de los años 2016 y 2019. Por otro lado, se hizo uso de los siguientes instrumentos: ficha de registro y ficha bibliográfica y 5 técnicas de machine learning (Regresión logística, SVM, Random Forest, KNN y Árbol de decisión). Como resultados se pudo observar que SVM obtuvo los mejores valores de las métricas estudiadas, siendo estas: Sensibilidad= 99.80%, Especificidad=99.80%, Curva ROC=74.30% y Validación cruzada=62.00%. Por último, se concluye que las técnicas de machine learning es útil, ya que aporta datos valiosos acerca del número de estudiantes que aprobaron que el algoritmo predice para la toma de decisiones más certeras de manera temprana, y de esta manera obtener estrategias eficaces.

2.2.3. Locales

A nivel local, no se encontraron estudios de investigación.

III. MATERIALES Y MÉTODOS

3.1. Lugar de Ejecución

El desarrollo del estudio tuvo lugar en la Universidad Nacional Agraria de la Selva, localizada en la ciudad de Tingo María, departamento de Huánuco cuyas coordenadas UTM son 9º 17'08 de latitud Sur, 8969880 390552 18L, 75º 59'52 de longitud Oeste y 660 msnm y 24 °C.

3.2. Materiales y métodos

3.2.1. Metodología

3.2.1.1.Tipo de la Investigación

Es de tipo aplicada, ya que el estudio se enfocó a los problemas de educación, en este caso a problemas de rendimiento académico en los alumnos universitarios de la UNAS, a través de ello se aplicó el machine learning para identificar a los alumnos que tienen bajo, medio o alto rendimiento académico (Lozada, 2014).

3.2.1.2.Enfoque de Investigación

El enfoque fue cuantitativo, porque siguió un proceso secuencial y asimismo se tomó datos numéricos y se usó el método estadístico para obtener evidencias que permitieron demostrar las hipótesis planteadas en relación con Machine learning para la predicción del rendimiento académico. Por ende, (Hernández & Mendoza, 2018) y Barrantes (2014), indican que este enfoque utiliza la recepción de información con el propósito de coincidir la hipótesis asumiendo el empleo de los números y la disciplina estadística que apruebe fijar aspectos de actitud.

3.2.1.3.Alcance de la Investigación

Es descriptivo, ya que estos tienen el objetivo de reconocer rasgos y cualidades de los individuos o colectivos sujeto al análisis. En otras palabras, buscan únicamente recabar datos acerca de las variables a las que hacen referencia, ya sea de manera independiente o en conjunto.

Es útil para describir con precisión la perspectiva o dirección de un fenómeno, evento, comunidad, entorno o situación. (Hernández & Mendoza, 2018).

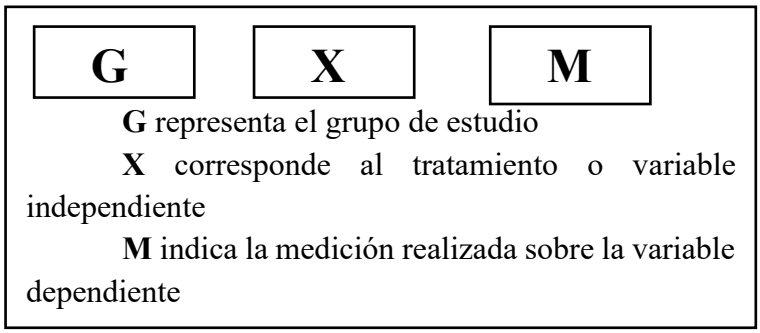
3.2.1.4.Diseño de la Investigación

El diseño empleado corresponde a un enfoque experimental de tipo pre-experimental (ver Figura 7), ya que se trata de un estudio con un solo grupo y un nivel mínimo de control

sobre las variables. Este tipo de diseño suele ser útil como aproximación inicial con la situación problemática en su contexto real (Hernández, Fernández y Baptista, 2014). Por su parte, (Hernández & Mendoza, 2018) señalan que el tipo pre-experimental, se basa en la administración de una incitación o tratamiento de un grupo, para posteriormente aplicarse una comprobación de una o más variables, con la finalidad de observar el nivel del grupo.

El esquema correspondiente se presenta a continuación:

Figura 7 *Diseño preexperimental con un único grupo*



Fuente: (Hernández & Mendoza, 2018)

Donde:

G: Datos de los alumnos de la UNAS

X: Técnicas de Machine Learning

M: Medición de las Métricas de precisión

3.2.1.5.Población y muestra

Vara (2012), define a la población como al universo, este puede estar constituido por individuos, cosas, especies, registros u otros, asimismo, comparten características similares en un mismo espacio, y son cambiantes en el tiempo.

En ese sentido, la investigación estuvo conformada por los promedios ponderados de 4584 estudiantes inscritos en el ciclo académico 2021-II de la Universidad Nacional Agraria de la Selva.

Por otra parte, la muestra consiste en una fracción de la población total, con la cual se desarrollará el estudio. En este apartado, la muestra fue igual que la población, debido a que es un valor mínimo para poder trabajar con las técnicas de machine learning.

Con respecto al muestreo empleado, este fue no probabilístico y por conveniencia, porque se eligió los elementos por las causas relaciones, y no dependió de la probabilidad. Como los determinados criterios de selección establecidos (Hernández & Mendoza, 2018). Asimismo, fue de tipo por conveniencia, ya que, al momento de la selección de las unidades,

estos tuvieron que responder a criterios subjetivos, que estén de acorde a los objetivos de la investigación (Cea, 1988).

3.2.1.6.Técnicas e instrumentos de recolección de datos

Técnicas:

Para recolectar los datos, se utilizó la técnica de Documentos y registros empleando así información histórica de los promedios ponderados de 4584 estudiantes matriculados en el periodo académico 2021-II que han sido extraídas de las bases de datos pertenecientes a la UNAS, referente al rendimiento académico. Para la obtención de estos datos, fue necesario solicitar información a la Oficina de Asuntos Académicos. En consecuencia, se presentó una solicitud formal (Ver **Anexo 7**).

Instrumentos de recolección de datos

Ficha de registro, este instrumento permitió recopilar los datos de los estudiantes. Esta información fue fundamental para la construcción del modelo predictivo. El instrumento utilizado es mostrado en el **Anexo 6.**

Herramientas:

Distribución Anaconda, IDE de desarrollo Jupyter, lenguaje de programación Python y Spyder, para trabajar estadísticamente la información en cual este permitió procesar y obtener los datos.

3.2.1.7.Variables de la investigación

Definición de Variables

- ✓ **Técnicas de Machine Learning**

  Definición conceptual:

Se basan en algoritmos que aprenden a partir de un conjunto de ejemplos especificando para una entrada dada cuál debe ser la salida, de modo que cuando se les da una entrada nueva entradas, producirán la salida correcta.

- ✓ **Predicción del Rendimiento académico**

  Definición conceptual:

Sirve para anticipar el rendimiento académico de los estudiantes utilizando métricas de evaluación de predicción, permitiendo así identificar a estudiantes que tienen un bajo rendimiento académico.

Dimensiones:

Como dimensiones se consideran las métricas de evaluación de las técnicas de machine Learning.

Indicadores:

Precisión, Exactitud, sensibilidad, especificidad, y puntuación F1 y Curva ROC

Escala de medición:

Razón, comprendido entre 0 y 1.

3.2.1.8.Operacionalización de Variables

La matriz correspondiente a la operacionalización de las variables se presenta en el Anexo 2.

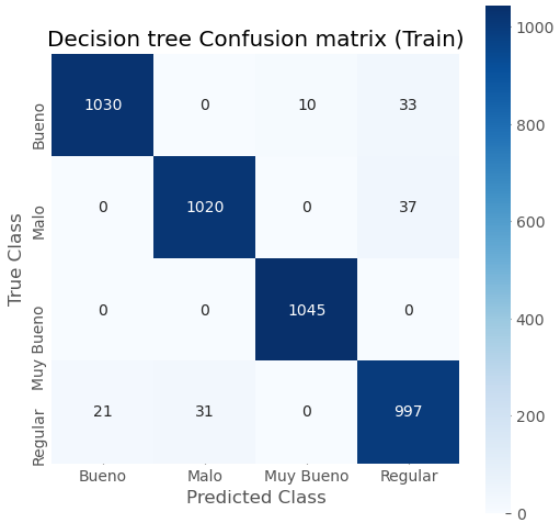
IV. RESULTADOS Y DISCUSIÓN

Se muestran los resultados alcanzados en el estudio basados en los indicadores de exactitud, precisión, sensibilidad, especificidad, F1-Score y Curva ROC. Estos resultados han sido comparados entre las 5 técnicas de machine learning seleccionadas para esta investigación: Arboles de decisión, Máquinas de Vector de Soporte, Redes Bayesianas, Redes Neuronales y Vecinos más Cercanos, siendo el objetivo identificar cuál de ellas tiene el valor óptimo, tomando como referencia la escala de calificación detallada en la Tabla 1. Finalmente, se realiza la demostración de las hipótesis que han sido planteadas. Cabe resaltar que todos los datos han sido procesados utilizando el lenguaje de programación Python; cuyo desarrollo se encuentra en el **Anexo 3**.

Seguidamente, se exponen los resultados referentes a la matriz de confusión de cada técnica de machine learning mediante el uso de las herramientas de Python:

Árbol de decisión

Figura 8 *Matriz de confusión del árbol de decisión (Train)*



Fuente: Elaboración propia

Tabla 3. *Matriz de confusión generada por el árbol de decisión (Train)*

RENDIMIENTO ACÁDEMICO	Bueno	Malo	Muy bueno	Regular	Total
Bueno	1030	0	10	33	1073
Malo	0	1020	0	37	1057
Muy bueno	0	0	1045	0	1045
Regular	21	31	0	997	1049

SUBTOTAL	**1051**	**1051**	**1055**	**1067**	**4224**
TOTAL		**4224**			

Fuente: Elaboración propia

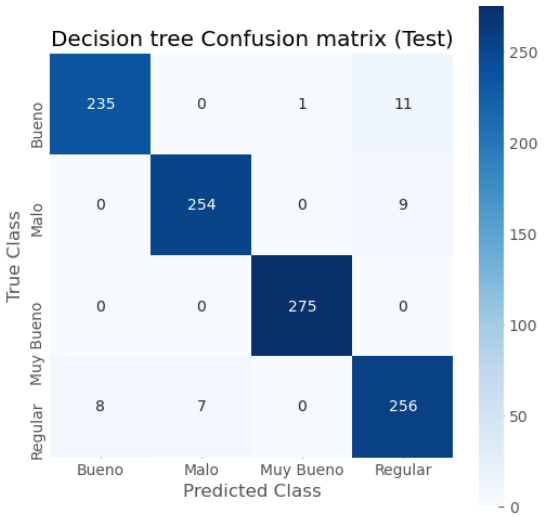
Según lo representado en la figura 8 y en la Tabla 3, de un total de 4,224 casos, se identificaron 1,030 casos correctamente y 43 incorrectamente en relación con el rendimiento bueno. De manera similar, en relación con el rendimiento malo, se predijeron correctamente 1,020 casos y 37 incorrectamente. Asimismo, en cuanto al rendimiento muy bueno, se predijeron de forma correcta 1,045 casos y 0 incorrectamente, mientras que, para el rendimiento regular, se identificaron 997 casos de forma correcta y 52 incorrectas.

Tabla 4. *Matriz de observación del árbol de decisión (Train)*

Clase	Medidas			
	TP	TN	FP	FN
Bueno	1030	3130	21	43
Malo	1020	3136	31	37
Muy bueno	1045	3169	10	0
Regular	997	3105	70	52
TOTALES	**4092**	**12540**	**132**	**132**

Fuente: Elaboración propia

Figura 9 *Matriz de confusión del árbol de decisión (Test)*



Fuente: Elaboración propia

Tabla 5. *Matriz de confusión del árbol de decisión (Test)*

RENDIMIENTO ACÁDEMICO	Bueno	Malo	Muy bueno	Regular	Total
Bueno	235	0	1	11	247
Malo	0	254	0	9	263
Muy bueno	0	0	275	0	275

Regular	8	7	0	256	271
SUBTOTAL	**243**	**261**	**276**	**276**	**1056**
TOTAL		**1056**			

Fuente: Elaboración propia

De acuerdo con lo representado en la Figura 9 y Tabla 5, de un total de 1,056 casos, se identificaron 235 casos correctamente y 12 incorrectamente en relación con el rendimiento bueno. De manera similar, en relación con el rendimiento malo, se predijeron correctamente 254 casos y 9 incorrectamente. Asimismo, en cuanto al rendimiento muy bueno, se predijeron de forma correcta 275 casos y 0 incorrectamente, mientras que, para el rendimiento regular, se identificaron 256 casos correctos y 15 incorrectos.

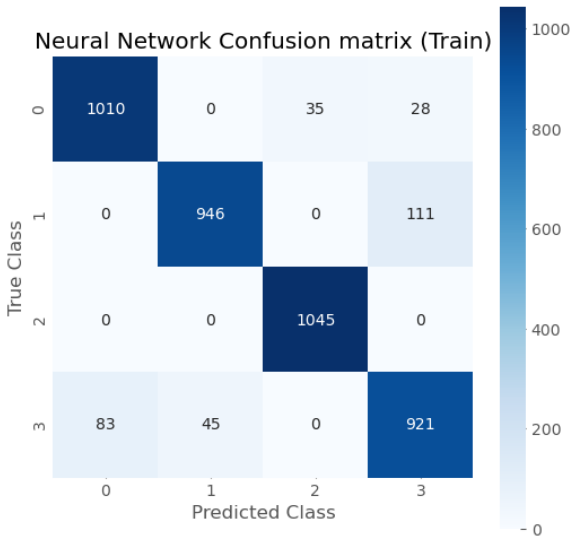
Tabla 6. *Matriz de observación del árbol de decisión (Test)*

Clase	Medidas			
	TP	TN	FP	FN
Bueno	235	801	8	12
Malo	254	786	7	9
Muy bueno	275	780	1	0
Regular	256	765	20	15
TOTALES	**1020**	**3132**	**36**	**36**

Fuente: Elaboración propia

Redes Neuronales

Figura 10 *Matriz de confusión de Redes Neuronales (Train)*



Fuente: Elaboración propia

Tabla 7. *Matriz de confusión de Redes Neuronales (Train)*

RENDIMIENTO ACÁDEMICO	Bueno	Malo	Muy bueno	Regular	Total
Bueno	1010	0	35	28	1073
Malo	0	946	0	111	1057
Muy bueno	0	0	1045	0	1045
Regular	83	45	0	921	1049
SUBTOTAL	**1093**	**991**	**1080**	**1060**	**4224**
TOTAL		**4224**			

Fuente: Elaboración propia

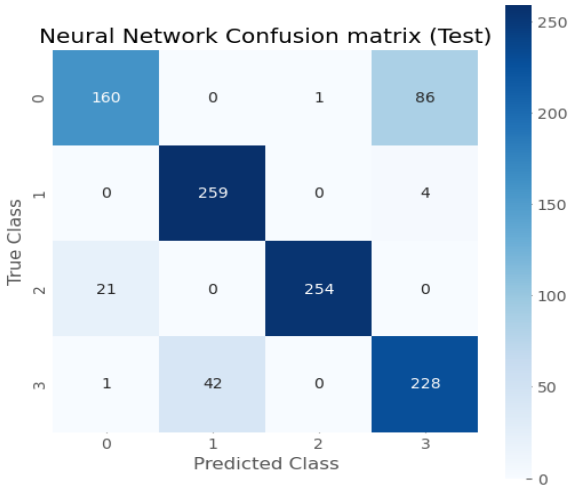
Según lo ilustrado en la Figura 10 y Tabla 7, de un total de 4,224 casos, se identificaron 1,010 casos correctamente y 63 incorrectamente en relación con el rendimiento bueno. De manera similar, en relación con el rendimiento malo, se predijeron correctamente 946 casos y 111 incorrectamente. Asimismo, en cuanto al rendimiento muy bueno, se predijeron de forma correcta 1,045 casos y 0 incorrectamente, mientras que, para el rendimiento regular, se identificaron 921 casos de valor correcto y 128 de valor incorrecto.

Tabla 8. *Matriz de observación de Redes Neuronales (Train)*

Clase	Medidas			
	TP	TN	FP	FN
Bueno	1010	3068	83	63
Malo	946	3122	45	111
Muy bueno	1045	3144	35	0
Regular	921	3036	139	128
TOTALES	**3922**	**12370**	**302**	**302**

Fuente: Elaboración propia

Figura 11 *Matriz de confusión de Redes Neuronales (Test)*



Fuente: Elaboración propia

Tabla 9. *Matriz de confusión de Redes Neuronales (Test)*

RENDIMIENTO ACÁDEMICO	Bueno	Malo	Muy bueno	Regular	Total
Bueno	160	0	1	86	247
Malo	0	259	0	4	263
Muy bueno	21	0	254	0	275
Regular	1	42	0	228	271
SUBTOTAL	**182**	**301**	**255**	**318**	**1056**
TOTAL		**1056**			

Fuente: Elaboración propia

Conforme se detalla en la Figura 11 y Tabla 9, de un total de 1,056 casos, se identificaron 160 casos correctamente y 87 incorrectamente en relación con el rendimiento bueno. De manera similar, en relación con el rendimiento malo, se predijeron correctamente 259 casos y 4 incorrectamente. Asimismo, en cuanto al rendimiento muy bueno, se predijeron de forma correcta 254 casos y 21 incorrectamente, mientras que, para el rendimiento regular, se identificaron 228 casos correctos e incorrectos en 43.

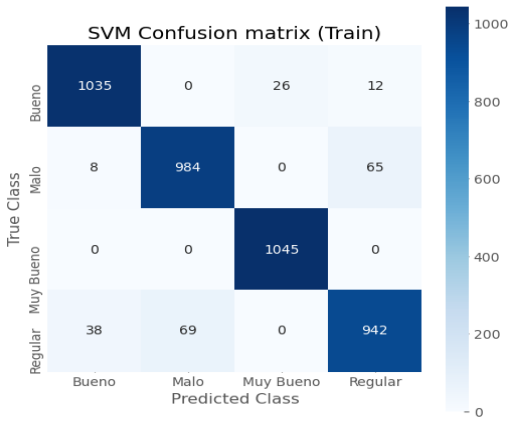
Tabla 10. *Matriz de observación – Redes Neuronales (Test)*

Clase	Medidas			
	TP	TN	FP	FN
Bueno	160	787	22	87
Malo	259	751	42	4
Muy bueno	254	780	1	21
Regular	228	695	90	43
TOTALES	**901**	**3013**	**155**	**155**

Fuente: Elaboración propia

Máquinas de Vector de Soporte (SVM)

Figura 12 *Matriz de confusión de SVM (Train)*



Fuente: Elaboración propia

Tabla 11. *Matriz de confusión de SVM (Train)*

RENDIMIENTO ACÁDEMICO	Bueno	Malo	Muy bueno	Regular	Total
Bueno	1035	0	26	12	1073
Malo	8	984	0	65	1057
Muy bueno	0	0	1045	0	1045
Regular	38	69	0	942	1049
SUBTOTAL	**1081**	**1053**	**1071**	**1019**	**4224**
TOTAL		**4224**			

Fuente: Elaboración propia

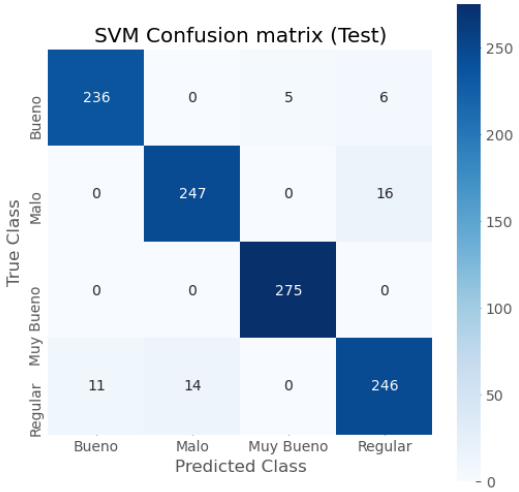
Según se expone en la Figura 12 y Tabla 11, de un total de 4,224 casos, se identificaron 1,035 casos correctamente y 38 incorrectamente en relación con el rendimiento bueno. De manera similar, en relación con el rendimiento malo, se predijeron correctamente 984 casos y 73 incorrectamente. Asimismo, en cuanto al rendimiento muy bueno, se predijeron de forma correcta 1,045 casos y 0 incorrectamente, mientras que, para el rendimiento regular, se identificaron 942 casos correctamente y como incorrectas 107.

Tabla 12. *Matriz de observación de SVM (Train)*

Clase	Medidas			
	TP	TN	FP	FN
Bueno	1035	3105	46	38
Malo	984	3098	69	73
Muy bueno	1045	3153	26	0
Regular	942	3098	77	107
TOTALES	**4006**	**12454**	**218**	**218**

Fuente: Elaboración propia

Figura 13 *Matriz de confusión de SVM (Test)*



Fuente: Elaboración propia

Tabla 13. *Matriz de confusión de SVM (Test)*

RENDIMIENTO ACÁDEMICO	Bueno	Malo	Muy bueno	Regular	Total
Bueno	236	0	5	6	247
Malo	0	247	0	16	263
Muy bueno	0	0	275	0	275
Regular	11	14	0	246	271
SUBTOTAL	**247**	**261**	**280**	**268**	**1056**
TOTAL		**1056**			

Fuente: Elaboración propia

Según lo evidenciado en la Figura 13 y Tabla 13, de un total de 1,056 casos, se identificaron 236 casos correctamente y 11 incorrectamente en relación con el rendimiento bueno. De manera similar, en relación con el rendimiento malo, se predijeron correctamente 247 casos y 16 incorrectamente. Asimismo, en cuanto al rendimiento muy bueno, se predijeron de forma correcta 275 casos y 0 incorrectamente, mientras que, para el rendimiento regular, se identificaron 246 casos que fueron correctas y 25 de forma incorrecta.

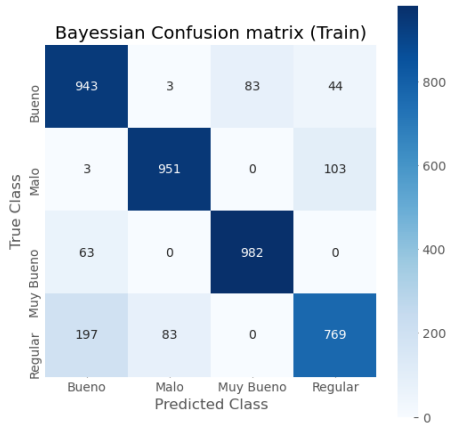
Tabla 14. *Matriz de observación de SVM (Test)*

Clase	Medidas			
	TP	TN	FP	FN
Bueno	236	798	11	11
Malo	247	779	14	16
Muy bueno	275	776	5	0
Regular	246	763	22	25
TOTALES	**1004**	**3116**	**52**	**52**

Fuente: Elaboración propia

Redes Bayesianas

Figura 14 *Matriz de confusión de Redes Bayesianas (Train)*



Fuente: Elaboración propia

Tabla 15. *Matriz de confusión de Redes Bayesianas (Train)*

RENDIMIENTO ACÁDEMICO	Bueno	Malo	Muy bueno	Regular	Total
Bueno	943	3	83	44	1073
Malo	3	951	0	103	1057
Muy bueno	63	0	982	0	1045
Regular	197	83	0	769	1049
SUBTOTAL	**1206**	**1037**	**1065**	**916**	**4224**
TOTAL		**4224**			

Fuente: Elaboración propia

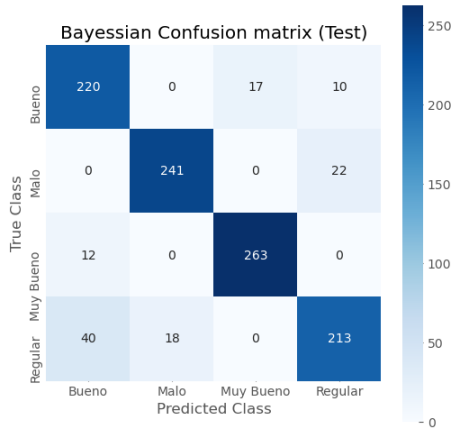
Así como es mostrado en la Figura 14 y Tabla 15, de un total de 4,224 casos, se identificaron 943 casos correctamente y 130 incorrectamente en relación con el rendimiento bueno. De manera similar, en relación con el rendimiento malo, se predijeron correctamente 951 casos y 106 incorrectamente. Asimismo, en cuanto al rendimiento muy bueno, se predijeron de forma correcta 982 casos y 63 incorrectamente, mientras que, para el rendimiento regular, se identificaron 769 casos correctamente frente a 280 que fueron incorrectos.

Tabla 16. *Matriz de observación de Redes Bayesianas (Train)*

Clase	Medidas			
	TP	TN	FP	FN
Bueno	943	2888	263	130
Malo	951	3081	86	106
Muy bueno	982	3096	83	63
Regular	769	3028	147	280
TOTALES	**3645**	**12093**	**579**	**579**

Fuente: Elaboración propia

Figura 15 *Matriz de confusión de Redes Bayesianas (Test)*



Fuente: Elaboración propia

Tabla 17. *Matriz de confusión de Redes Bayesianas (Test)*

RENDIMIENTO ACÁDEMICO	Bueno	Malo	Muy bueno	Regular	Total
Bueno	220	0	17	10	247
Malo	0	241	0	22	263
Muy bueno	12	0	263	0	275
Regular	40	18	0	213	271
SUBTOTAL	**272**	**259**	**280**	**245**	**1056**
TOTAL		**1056**			

Fuente: Elaboración propia

Conforme se detalla en la Figura 15 y Tabla 17, de un total de 1,056 casos, se identificaron 220 casos correctamente y 27 incorrectamente en relación con el rendimiento bueno. De manera similar, en relación con el rendimiento malo, se predijeron correctamente 241 casos y 22 incorrectamente. Asimismo, en cuanto al rendimiento muy bueno, se predijeron de forma correcta 263 casos y 12 incorrectamente, mientras que, para el rendimiento regular, se identificaron 213 casos correctamente y 58 incorrectamente.

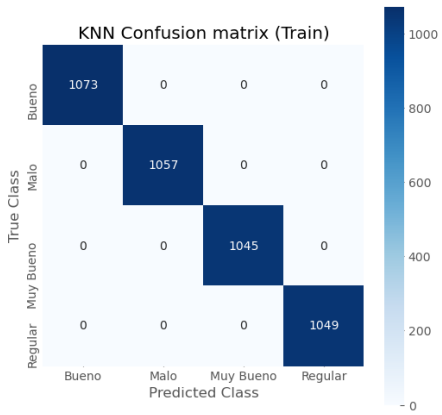
Tabla 18. *Matriz de observación de Redes Bayesianas (Test)*

Clase	Medidas			
	TP	TN	FP	FN
Bueno	220	757	52	27
Malo	241	775	18	22
Muy bueno	263	764	17	12
Regular	213	753	32	58
TOTALES	**937**	**3049**	**119**	**119**

Fuente: Elaboración propia

Vecinos más Cercanos (KNN)

Figura 16 *Matriz de confusión de KNN (Train)*



Fuente: Elaboración propia

Tabla 19. *Matriz de confusión de KNN (Train)*

RENDIMIENTO ACÁDEMICO	Bueno	Malo	Muy bueno	Regular	Total
Bueno	1073	0	0	0	1073
Malo	0	1057	0	0	1057
Muy bueno	0	0	1045	0	1045
Regular	0	0	0	1049	1049
SUBTOTAL	**1073**	**1057**	**1045**	**1049**	**4224**
TOTAL		**4224**			

Fuente: Elaboración propia

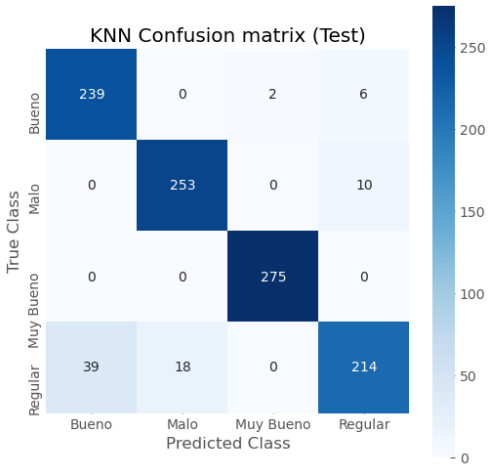
Tal como se llega a indicar en la Figura 16 y Tabla 19, de un total de 4,224 casos, se identificaron 1,073 casos correctamente y 0 incorrectamente en relación con el rendimiento bueno. De manera similar, en relación con el rendimiento malo, se predijeron correctamente 1,057 casos y 0 incorrectamente. Asimismo, en cuanto al rendimiento muy bueno, se predijeron de forma correcta 1,045 casos y 0 incorrectamente, mientras que, para el rendimiento regular, se identificaron 1,049 casos de forma correcta y 0 de forma incorrecta.

Tabla 20. *Matriz de observación de KNN (Train)*

Clase	Medidas			
	TP	TN	FP	FN
Bueno	1073	3151	0	0
Malo	1057	3167	0	0
Muy bueno	1045	3179	0	0
Regular	1049	3175	0	0
TOTALES	**4224**	**12672**	**0**	**0**

Fuente: Elaboración propia

Figura 17 *Matriz de confusión de KNN (Test)*



Fuente: Elaboración propia

Tabla 21. *Matriz de confusión de KNN (Test)*

RENDIMIENTO ACÁDEMICO	Bueno	Malo	Muy bueno	Regular	Total
Bueno	239	0	2	6	247
Malo	0	253	0	10	263
Muy bueno	0	0	275	0	275
Regular	39	18	0	214	271
SUBTOTAL	**278**	**271**	**277**	**230**	**1056**
TOTAL		**1056**			

Fuente: Elaboración propia

De acuerdo con lo representado en la Figura 17 y Tabla 21, de un total de 1,056 casos, se identificaron 239 casos correctamente y 8 incorrectamente en relación con el rendimiento bueno. De manera similar, en relación con el rendimiento malo, se predijeron correctamente 253 casos y 10 incorrectamente. Asimismo, en cuanto al rendimiento muy bueno, se predijeron de forma correcta 275 casos y 0 incorrectamente, mientras que, para el rendimiento regular, se identificaron 214 casos de forma correcta y 57 de forma incorrecta.

Tabla 22. *Matriz de observación de KNN (Test)*

Clase	Medidas			
	TP	TN	FP	FN
Bueno	239	770	39	8
Malo	253	775	18	10
Muy bueno	275	779	2	0
Regular	214	769	16	57
TOTALES	**981**	**3093**	**75**	**75**

Fuente: Elaboración propia

Seguidamente, se presenta la demostración de las hipótesis especificas formuladas en el marco de la investigación. Para realizar la contrastación de hipótesis, primero se realizó el experimento con las 5 técnicas propuestas las cuales se encuentran los árboles de decisión, las redes neuronales, los modelos SVM, las redes bayesianas y el método de los k vecinos más cercanos con 4224 datos del conjunto de entrenamiento que consta de: 1073 casos de la clase Malo, 1057 casos para la clase regular, 1045 casos para la clase Bueno y 1049 casos para la clase Muy Bueno, y para validar si el modelo es robusto y confiable se procedió hacer la validación de los datos con 1056 datos del conjunto de test donde: 247 son casos de la clase Malo, 263 casos de la clase Regular, 275 casos de la clase Bueno y 271 casos de la clase Muy Bueno, para luego obtener los valores de las métricas en 10 repeticiones que nos brinda el software python en donde se presenta en la Tabla 23 los resultados de Exactitud, en la Tabla 26 los resultados de Precisión, en la tabla 29 los resultados de sensibilidad, en la Tabla 32 los

resultados de especificidad, en la Tabla 35 los resultados de F1-Score, y en la Tabla 38 los resultados de la curva ROC.

Los resultados obtenidos son confiables e indican que redes neuronales mostró un desempeño superior en comparación con las demás técnicas, con una exactitud que supera el 95 por ciento, siendo esta mayor con respecto a las técnicas antes mencionadas. Finalmente, los datos recopilados en las tablas mencionadas anteriormente se ingresaron en el software SPSS stadistic y luego se aplicó la prueba de normalidad en donde analizamos el resultado en base a shapiro-wilk debido a que son 10 valores, que nos permitió determinar que la técnica estadística adecuada a usar es la Prueba de Kruskal-Wallis (no paramétrica).

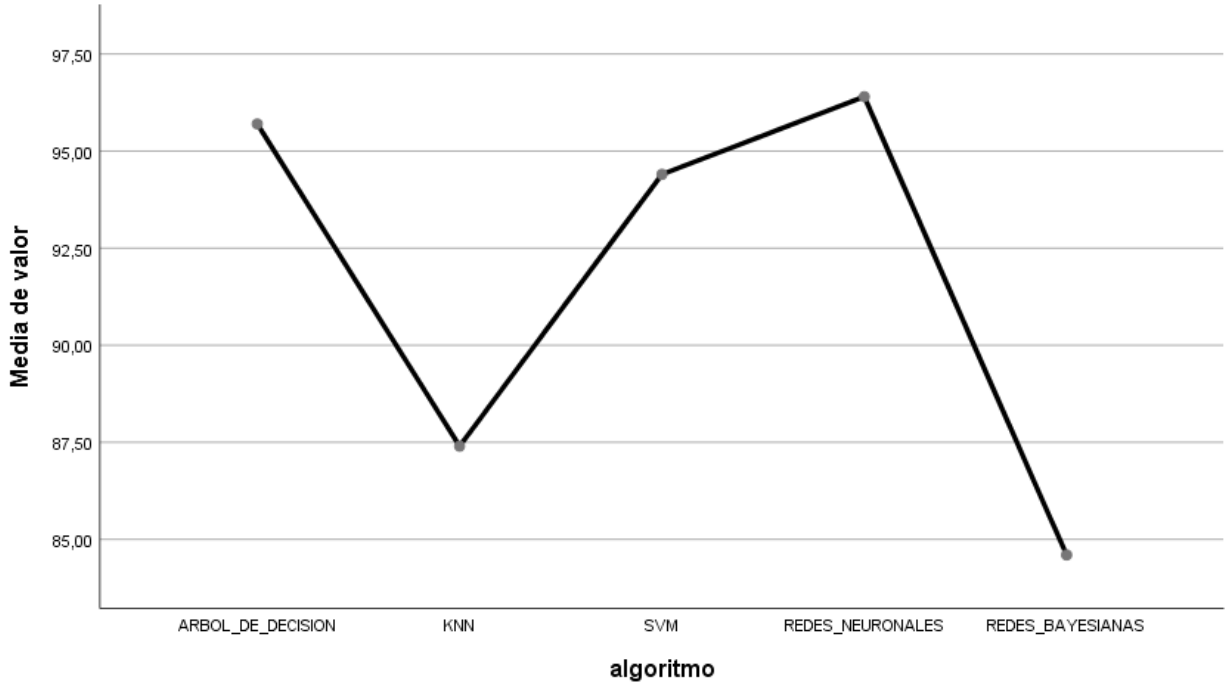
HE1: Existe diferencia estadística significativa en la exactitud de las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS.

La Tabla 23 presenta los resultados obtenidos para la métrica de 'Exactitud', aplicando 10 particiones de los datos, donde los valores de cv varían desde el cv = 2 hasta el cv = 11.

Tabla 23. *Resultados de la métrica Exactitud*

cv	Árbol de decisión	KNN	SVM	Redes Neuronales	Redes Bayesianas
2	97.06%	88.07%	94.70%	97.73%	85.70%
3	95.64%	88.26%	94.32%	96.59%	84.94%
4	95.17%	86.55%	93.75%	96.69%	83.43%
5	95.27%	86.08%	95.36%	95.83%	85.61%
6	95.74%	87.59%	95.08%	95.45%	84.56%
7	95.27%	87.97%	94.03%	95.74%	84.56%
8	95.93%	87.31%	93.37%	95.83%	85.23%
9	95.64%	87.88%	93.66%	96.31%	83.71%
10	95.45%	88.64%	94.89%	95.55%	84.09%
11	95.83%	84.94%	94.70%	96.69%	82.95%
total	957.00%	873.29%	943.86%	962.41%	844.78%
promedio	**95.70%**	**87.33%**	**94.39%**	**96.24%**	**84.47%**

Figura 18 *Resultados de la métrica Exactitud*



Fuente: SPSS Statistics v.25

Interpretación: Tal como se observa en la tabla 23 y Figura 18, la técnica con el % óptimo referente a la métrica exactitud que permite la predicción del rendimiento académico de los estudiantes de la UNAS correctamente es “Redes neuronales” =96.24%, a la vez de Árbol de decisión” = 95.70% y “SVM” =94.39%, luego “KNN” =87.33%, y finalmente “Redes bayesianas” = 84.47%.

Esto es refutado con el estudio de Harvey y Kumar (2019), se pudo observar que la técnica de Naive Bayes, mostró una exactitud de 71.00%, siendo la más alta y efectiva respecto a los puntajes de matemáticas, demostrando de esta manera que la técnica mencionada con anterioridad es muy importante para predecir otras formas de desempeño de los estudiantes y mejorar su aprendizaje, como también evaluar las correlaciones de los datos. De forma similar, en el estudio de Khan et al (2021), donde concluyeron que la técnica predictiva utilizada (Red neuronal artificial) usado para pronosticar el rendimiento de los estudiantes alcanzó una exactitud del 52.00%, lo cual dice que las técnicas de aprendizaje automático son indispensables para garantizar un aprendizaje más eficiente a través de técnicas, siendo la dinámica buena y necesaria la aplicación de la red neuronal artificial, utilizando atributos con mayor correlación.

Para realizar la contrastación de la hipótesis de la métrica de exactitud se realizó la prueba de normalidad de todos los grupos independientes obteniendo los resultados de acuerdo con lo presentado en la Tabla 24.

Tabla 24. *Prueba de normalidad de exactitud.*

Pruebas de normalidad						
	Kolmogorov-Smirnov			Shapiro-Wilk		
Técnicas	Estadístico	gl	Sig	Estadístico	gl	Sig.
Arbol_decision	,272	10	0,035	0,802	10	0,015
KNN	,295	10	0,014	0,873	10	0,108
SVM	,305	10	0,009	0,781	10	0,008
Redes_neuronales	,282	10	0,023	0,890	10	0,172
Redes_bayesianas	,245	10	0,090	0,892	10	0,177

El en la Tabla 24, se presenta el valor de gl es menor que 30 por lo cual analizaremos la prueba de normalidad en base a shapiro-wilk, en donde se puede apreciar que los grupo de KNN, redes neuronales y redes bayesianas tiene un nivel de significancia mayor a 0.05, así que se muestran que son paramétricas y siguen una distribución normal, así mismo los grupos de Árbol de decisión y SVM tiene un nivel de significancia inferior que 0.05 lo que indica que no son paramétricas y no siguen una distribución normal, por lo tanto para la contratación de hipótesis se usó el test estadístico Kruskal-Wallis (ver Tabla 25)

Tabla 25. *Prueba de Kruskal-Wallis de exactitud.*

Estadísticos de prueba	
	Valor
H de Kruskal-Wallis	44,119
gl	4
Sig. asintótica	,000

Interpretación de los resultados:

1. **Hipótesis de la prueba:**
   - Hipótesis nula (H0): No hay diferencias significativas entre las medianas de los grupos (técnicas).
   - Hipótesis alternativa (H1): Por lo menos un grupo (técnica) tiene una mediana significativamente diferente.
2. **Estadístico de Kruskal-Wallis (H)**
   - El valor calculado de la estadística H=44.119 señala la magnitud de la diferencia entre los grupos.
   - Este valor se compara con una distribución chi-cuadrado con gl=4 (grados de libertad).

3. **Grados de libertad (gl):**
    - La prueba se realizó con 4 grados de libertad, lo que significa que se están comparando 5 grupos en total.
4. **Significación asintótica (p-valor):**
    - El valor de significancia reportado es p=.000, lo que significa que es menor que el nivel típico de significancia (α=0.05).
    - Dado que p <0 .05, se descarta la hipótesis nula.

Conclusión:

Según lo mostrado en la Tabla 25, el valor p es inferior a 0.05, lo que indica diferencias estadísticamente significativas entre las medianas de las distintas técnicas analizadas con relación a la exactitud. Esto sugiere que al menos una de las técnicas presenta un rendimiento diferente en comparación con los demás.

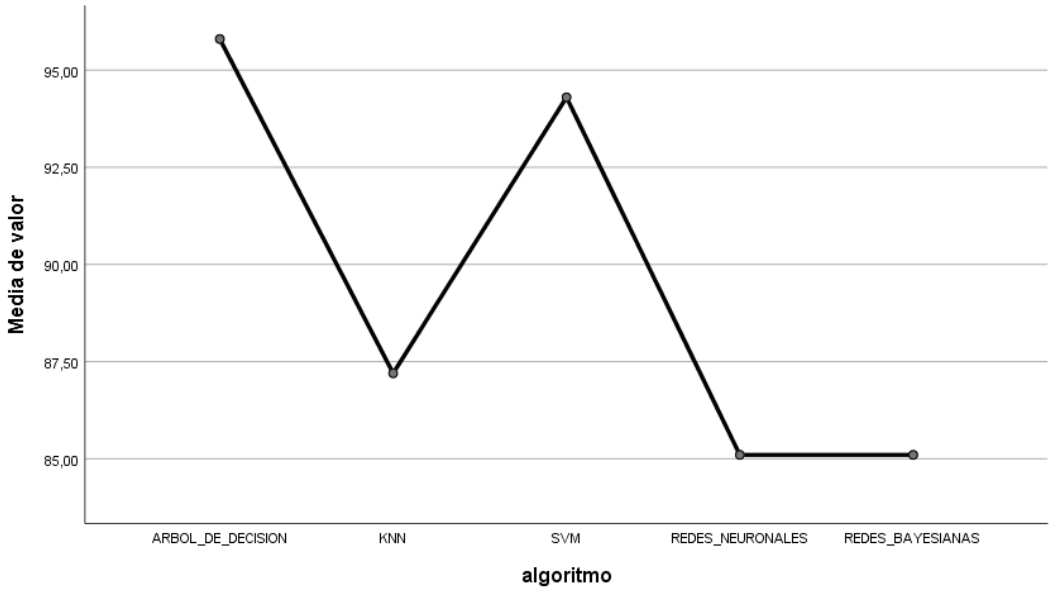
HE2: Existe diferencia estadística significativa en la precisión de las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS.

En la Tabla 26 se detallan los valores de precisión calculados mediante validación cruzada, variando el número de particiones desde cv = 2 hasta cv = 11

Tabla 26. *Resultados de la métrica Precisión*

cv	Árbol de decisión	KNN	SVM	Redes Neuronales	Redes Bayesianas
2	97.07%	87.90%	94.74%	86.39%	86.39%
3	95.75%	88.15%	94.29%	85.89%	85.89%
4	95.26%	86.61%	93.86%	84.64%	84.64%
5	95.20%	85.75%	95.26%	86.30%	86.30%
6	95.65%	87.18%	94.97%	85.22%	85.22%
7	95.24%	87.73%	93.94%	85.09%	85.09%
8	95.87%	86.75%	93.14%	85.43%	85.43%
9	95.54%	87.40%	93.47%	84.14%	84.14%
10	95.64%	88.55%	94.90%	85.34%	85.34%
11	95.93%	84.95%	94.68%	83.85%	83.85%
total	957.15%	870.97%	943.25%	852.29%	852.29%
promedio	**95.72%**	**87.10%**	**94.33%**	**85.23%**	**85.22%**

Figura 19 *Resultados de la métrica Precisión*



Fuente: SPSS Statistics v.25

Interpretación: Tal como se aprecia en la Tabla 26 y Figura 19, la técnica que alcanza el mayor porcentaje referente a la métrica precisión que permite predicir el rendimiento académico de los alumnos de la UNAS correctamente es “Árbol de decisión” = 95.72%, a la vez de “SVM” = 94.33%, luego “K-NN” con 87.10%, asimismo sigue “Redes neuronales” con un valor del 85.23% y por último “Redes bayesianas” con 85.22%.

Estos datos son contrastados con el estudio de Candia (2019), donde se pudo observar que Random Forest tuvo el mejor valor de precisión, logrando predecir hasta un 69.35% el rendimiento de los estudiantes, concluyendo la utilización de técnicas de aprendizaje automático es efectivo para la predicción del desempeño académico, partiendo de información de admisión a la UNSAAC. A la vez, Yamao (2018), concluyó que árbol de decisión obtuvieron el mejor resultado de precisión, siendo este de 82.87%, lo que quiere decir que a utilizar técnicas de minería de datos permite predecir el rendimiento académico de los alumnos ingresantes que pueden enfrentarse a problemas de estudio.

Para realizar la contrastación de la hipótesis de la métrica de precisión se realizó la prueba de normalidad de todos los grupos independientes obteniendo los resultados como se expone en la Tabla 27.

Tabla 27. *Prueba de normalidad de precisión.*

Pruebas de normalidad						
	Kolmogorov-Smirnov			Shapiro-Wilk		
Técnicas	Estadístico	gl	Sig	Estadístico	gl	Sig.
Arbol_decision	,324	10	0,004	0,794	10	0,012
KNN	,230	10	0,143	0,933	10	0,479
SVM	,302	10	0,010	0,781	10	0,008
Redes_neuronales	,254	10	0,067	0,833	10	0,036
Redes_bayesianas	,254	10	0,067	0,833	10	0,036

Se ilustra en la Tabla 27, que el valor de gl es menor que 30 por lo cual analizaremos la prueba de normalidad en base a shapiro-wilk, en donde se puede apreciar que el grupo de KNN, tiene un nivel de significancia mayor a 0.05 lo que exhibe que son paramétricas y siguen una distribución de nivel normal, así mismo los grupos de Árbol de decisión, SVM, redes neuronales y redes bayesianas tiene un nivel de significancia menor que 0.05 lo que indica que no son paramétricas y no siguen una distribución normal, por lo tanto para la contratación de hipótesis se usó la prueba estadística no paramétrica de Kruskal-Wallis (ver Tabla 28.)

Tabla 28. *Prueba de Kruskal-Wallis de precision*

Estadísticos de prueba	
	Valor
H de Kruskal-Wallis	43,092
gl	4
Sig_ asintótica	,000

Interpretación de los resultados:

1. **Hipótesis de la prueba:**
   - Hipótesis nula (H0): No hay diferencias significativas entre las medianas de los grupos (técnicas).
   - Hipótesis alternativa (H1): Al menos un grupo (técnica) tiene una mediana significativamente diferente.
2. **Estadístico de Kruskal-Wallis (H)**
   - El valor calculado de la estadística H=43.092 evidencia la magnitud de la diferencia entre los grupos.
   - Este valor se compara con una distribución chi-cuadrado con gl=4 (grados de libertad).
3. **Grados de libertad (gl):**
   - La prueba se realizó con 4 grados de libertad, lo que significa que se están comparando 5 grupos en total.

4. **Significación asintótica (p-valor):**
    - El valor de significancia reportado es p=.000, lo que significa que es menor que el nivel típico de significancia, la cual es (α=0.05).
    - Dado que p <0 .05, permite que se rechace la hipótesis nula.

Conclusión:

La Tabla 28 muestra un p-valor que es menos que 0.05, lo cual permite concluir que hay diferencias entre las medianas de las técnicas comparadas con relación a la precisión. Esto sugiere que al menos una de las técnicas presenta un rendimiento diferente en comparación con las demás.

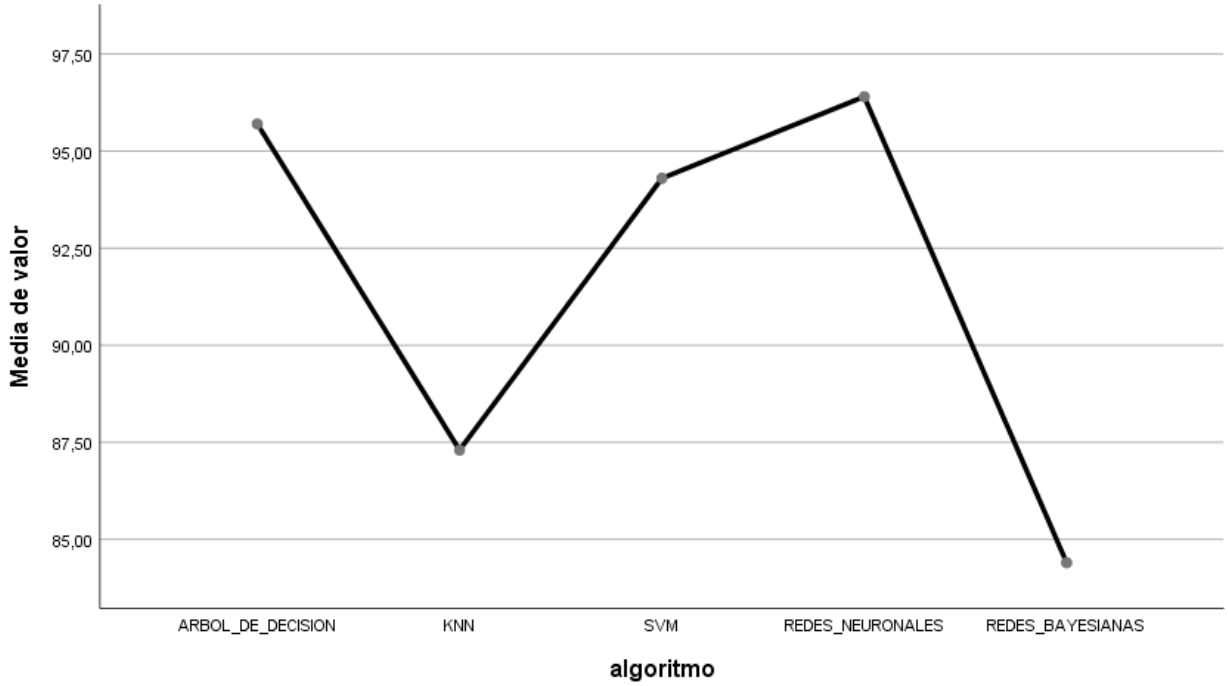
HE3: Existe diferencia estadística significativa en la sensibilidad las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS.

En la Tabla 29 se presentan los valores correspondientes a la métrica 'Sensibilidad', calculados mediante validación cruzada con particiones que van desde cv = 2 hasta cv = 11.

Tabla 27. *Resultados de la métrica Sensibilidad*

cv	Árbol de decisión	KNN	SVM	Redes Neuronales	Redes Bayesianas
2	97.10%	88.16%	94.77%	97.74%	85.79%
3	95.59%	88.36%	94.36%	96.56%	84.85%
4	95.39%	87.10%	94.01%	96.81%	83.98%
5	95.18%	85.78%	95.27%	95.83%	85.22%
6	95.57%	87.25%	94.99%	95.44%	83.99%
7	95.23%	87.91%	93.98%	95.71%	84.52%
8	95.74%	86.84%	93.25%	95.67%	84.58%
9	95.51%	87.48%	93.44%	96.14%	83.29%
10	95.47%	88.77%	94.93%	95.58%	84.08%
11	96.01%	85.80%	94.94%	96.84%	83.38%
total	956.79%	873.45%	943.94%	962.32%	843.68%
promedio	**95.68%**	**87.35%**	**94.39%**	**96.23%**	**84.37%**

Figura 20 *Resultados de la métrica Sensibilidad*



Fuente: SPSS Statistics v.25

Interpretación: Como se evidencia en la Tabla 29 y la Figura 20, la técnica que presenta el mayor porcentaje referente a la métrica sensibilidad que ayuda en la predicción del rendimiento académico de los alumnos de la UNAS correctamente es “Redes neuronales” = 96.23%, a la vez de “Árbol de decisión” = 95.68%, luego “SVM” con 94.39%, asimismo sigue “KNN” con un valor del 87.35% y por último “Redes bayesianas” con 84.37%.

Estos resultados son refutados con el estudio de Orihuela (2019), donde predijeron el rendimiento académico en una universidad del Perú, cuyos resultados muestran que Random Forest alcanzó el valor de sensibilidad (Test) más alta de 76.00%, concluyendo que se logra predecir el rendimiento. Sin embargo, es confirmado con el estudio de Garcia (2021), donde elaboró un enfoque de machine learning para la predicción del rendimiento académico e identificar quienes tienen la probabilidad de éxito o fracaso en sus cursos. Por lo que, SVM fue el que logró un mejor valor de sensibilidad de 100.00%, lo que significa que este es el más eficiente al momento de predecir el rendimiento.

Para realizar la contrastación de la hipótesis de la métrica de sensibilidad se realizó la prueba de normalidad de todos los grupos independientes obteniendo los resultados como se puede observar en la Tabla 27.

Tabla 30. *Prueba de normalidad de sensibilidad.*

Pruebas de normalidad						
	Kolmogorov-Smirnov			Shapiro-Wilk		
Técnicas	Estadístico	gl	Sig	Estadístico	gl	Sig.
Arbol_decision	,272	10	0,035	0,802	10	0,015
KNN	,224	10	0,168	0,911	10	0,287
SVM	,302	10	0,010	0,781	10	0,008
Redes_neuronales	,282	10	0,023	0,890	10	0,172
Redes_bayesianas	,233	10	0,133	0,904	10	0,245

En la Tabla 30, se presenta que el valor de gl es menor que 30 por lo cual analizaremos la prueba de normalidad en base a shapiro-wilk, en donde se puede apreciar que el grupo de KNN, redes neuronales y redes bayesianas tiene un nivel de significancia mayor a 0.05 lo que indica que son paramétricas y siguen una distribución normal, así mismo los grupos de Árbol de decisión y SVM, tiene un valor nivel de significancia menos que 0.05 lo que indica que no son paramétricas y no siguen una distribución normal, por lo tanto para la contratación de hipótesis se usó Kruskal-Wallis. (Ver tabla 31)

Tabla 31. *Prueba de Kruskal-Wallis de sensibilidad.*

Estadísticos de prueba	
	Valor
H de Kruskal-Wallis	44,438
gl	4
Sig. asintótica	,000

Interpretación de los resultados:

1. **Hipótesis de la prueba:**
   - Hipótesis nula (H0): No hay diferencias significativas entre las medianas de los grupos (técnicas).
   - Hipótesis alternativa (H1): Al menos un grupo (técnica) tiene una mediana significativamente diferente.
2. **Estadístico de Kruskal-Wallis (H)**
   - El valor calculado de la estadística H=44,438 expone el nivel de la magnitud de la diferencia entre los grupos.
   - Este valor se compara con una distribución chi-cuadrado con gl=4 (grados de libertad).
3. **Grados de libertad (gl):**
   - La prueba se realizó con 4 grados de libertad, lo que significa que se están comparando 5 grupos en total.
4. **Significación asintótica (p-valor):**

- El valor de significancia reportado es p=.000, lo que significa que es menor que el nivel típico de significancia, que se indica como (α=0.05).
- Al encontrarse el p-valor por debajo de 0.05, se descarta la hipótesis nula.

Conclusión:

En la Tabla 31 se puede observar que el p-valor es menor a 0.05, se concluye que existen diferencias significativas entre las medianas de las distintas técnicas analizadas. Esto sugiere que al menos una de las técnicas presenta un rendimiento diferente en comparación con los demás.

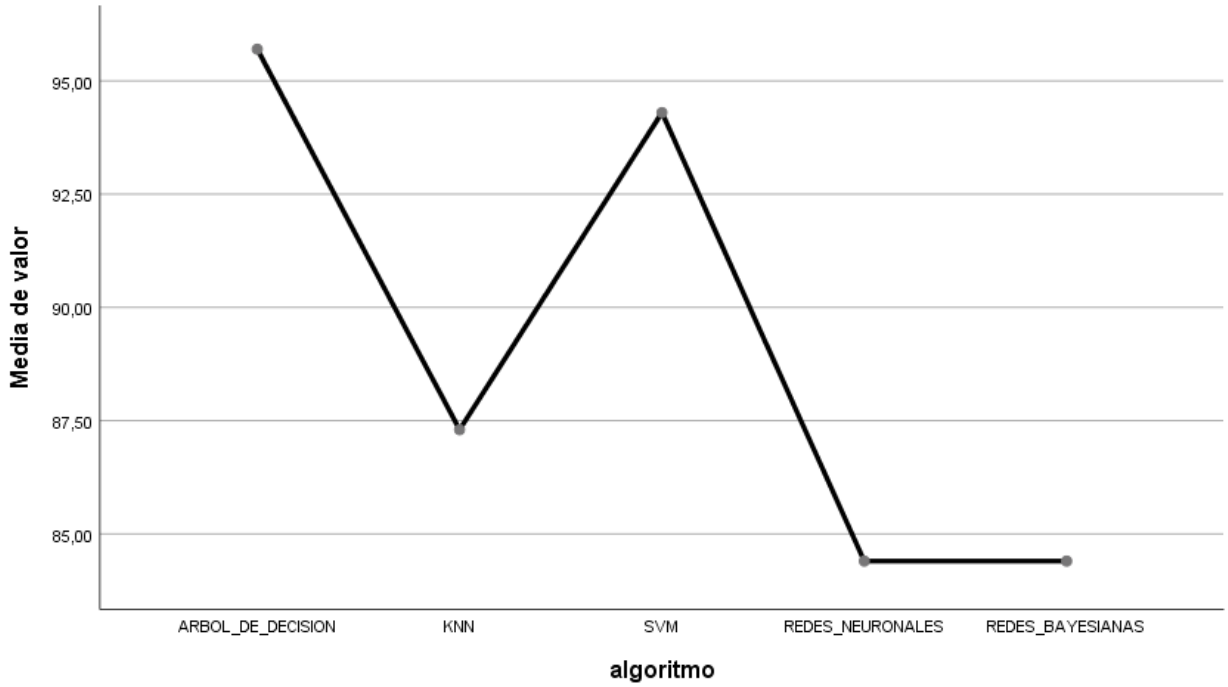
HE4: Existe diferencia estadística significativa en la especificidad las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS.

La Tabla 32 presenta los resultados que se han obtenido para la métrica de especificidad, utilizando validación cruzada con 10 particiones, donde los valores de cv oscilan entre 2 y 11.

Tabla 28. *Resultados de la métrica Especificidad*

cv	Árbol de decisión	KNN	SVM	Redes Neuronales	Redes Bayesianas
2	97.10%	88.16%	94.77%	85.79%	85.79%
3	95.59%	88.36%	94.36%	84.85%	84.85%
4	95.39%	87.10%	94.01%	83.98%	83.98%
5	95.18%	85.78%	95.27%	85.22%	85.22%
6	95.57%	87.25%	94.99%	83.99%	83.99%
7	95.23%	87.91%	93.98%	84.52%	84.52%
8	95.74%	86.84%	93.25%	84.58%	84.58%
9	95.51%	87.48%	93.44%	83.29%	83.29%
10	95.47%	88.77%	94.93%	84.08%	84.08%
11	96.01%	85.80%	94.94%	83.38%	83.38%
total	956.79%	873.45%	943.94%	843.68%	843.68%
promedio	**95.68%**	**87.35%**	**94.39%**	**84.37%**	**84.36%**

Figura 21 *Resultados de la métrica Especificidad*



Fuente: SPSS Statistics v.25

Interpretación: La tabla 32 y Figura 21 indican que la técnica con el porcentaje óptimo referente a la métrica especificidad que ayuda en la predicción del rendimiento académico de los alumnos de la UNAS correctamente es “Árbol de decisión” = 95.68%, a la vez que “SVM” = 94.39%, luego “K-NN” con 87.35%, asimismo sigue “Redes neuronales” con un valor del 84.37% y por último “Redes bayesianas” con 84.36%.

Esto es contrastado con el estudio de Lau, Sun y Yang (2019) en base al resultado de la técnica utilizada (red neuronal artificial) se obtuvo una especificidad de 54.6%. Por lo que se concluye que el uso de técnicas de machine learning con configuración de modelos educativos ayuda a realizar predicciones más exactas del rendimiento de los alumnos. No obstante, es confirmado por el estudio de Aronés (2021), quien en su estudio empleó un enfoque de machine learning para la predicción del rendimiento, donde se evidenció que SVM obtuvo el mejor valor de especificidad de 99.80%, lo que quiere decir que sirve de ayuda para obtener información útil y valiosa acerca de los alumnos aprobados de forma temprana.

Para realizar el contraste de la hipótesis de especificidad se realizó la prueba de normalidad de todos los grupos independientes obteniendo los resultados como se exhiben en la Tabla 33.

Tabla 33. *Prueba de normalidad de Especificidad.*

Pruebas de normalidad						
	Kolmogorov-Smirnov			Shapiro-Wilk		
Técnicas	Estadístico	gl	Sig	Estadístico	gl	Sig.
Arbol_decision	,272	10	0,035	0,802	10	0,015
KNN	,224	10	0,168	0,911	10	0,287
SVM	,302	10	0,010	0,781	10	0,008
Redes_neuronales	,233	10	0,133	0,904	10	0,245
Redes_bayesianas	,233	10	0,133	0,904	10	0,245

En la Tabla 30, se manifiesta que el valor de gl es menor que 30 por lo cual analizaremos la prueba shapiro-wilk, en donde se puede apreciar que el grupo de KNN, redes neuronales y redes bayesianas tiene un nivel de significancia mayor a 0.05 lo que indica que son paramétricas y siguen una distribución normal, así mismo los grupos de Árbol de decisión y SVM, tiene un nivel de significancia menor que 0.05 lo que indica que no son paramétricas y no siguen una distribución normal, por lo tanto para la contratación de hipótesis se usó el test Kruskal-Wallis (Ver Tabla 34.)

Tabla 34. *Prueba de Kruskal-Wallis de Especificidad.*

Estadísticos de prueba	
	Valor
H de Kruskal-Wallis	44,212
gl	4
Sig. asintótica	,000

Interpretación de los resultados:

1. **Hipótesis de la prueba:**
   - Hipótesis nula (H0): No hay diferencias significativas entre las medianas de los grupos (técnicas).
   - Hipótesis alternativa (H1): Al menos un grupo (técnica) tiene una mediana significativamente diferente.
2. **Estadístico de Kruskal-Wallis (H)**
   - El valor calculado de la estadística H=44,212 manifiesta la magnitud de la diferencia entre los grupos.
   - Este valor permite la comparación con un chi-cuadrado con gl=4 (grados de libertad).

3. **Grados de libertad (gl):**
    - La prueba se realizó con 4 grados de libertad, lo que significa que se están comparando 5 grupos en total.
4. **Significación asintótica (p-valor):**
    - El valor de significancia reportado es p=.000, lo que significa que es inferior al nivel típico de significancia que es (α=0.05).
    - El resultado estadístico ($p < 0.05$) permite rechazar la hipótesis nula.

Conclusión:

En la Tabla 34 se puede observar que el p-valor es menor a 0.05, se concluye que se presentan diferencias significativas entre las medianas de las distintas técnicas analizadas con relación a la especificidad. Esto sugiere que al menos una de las técnicas presenta un rendimiento diferente en comparación con los demás.

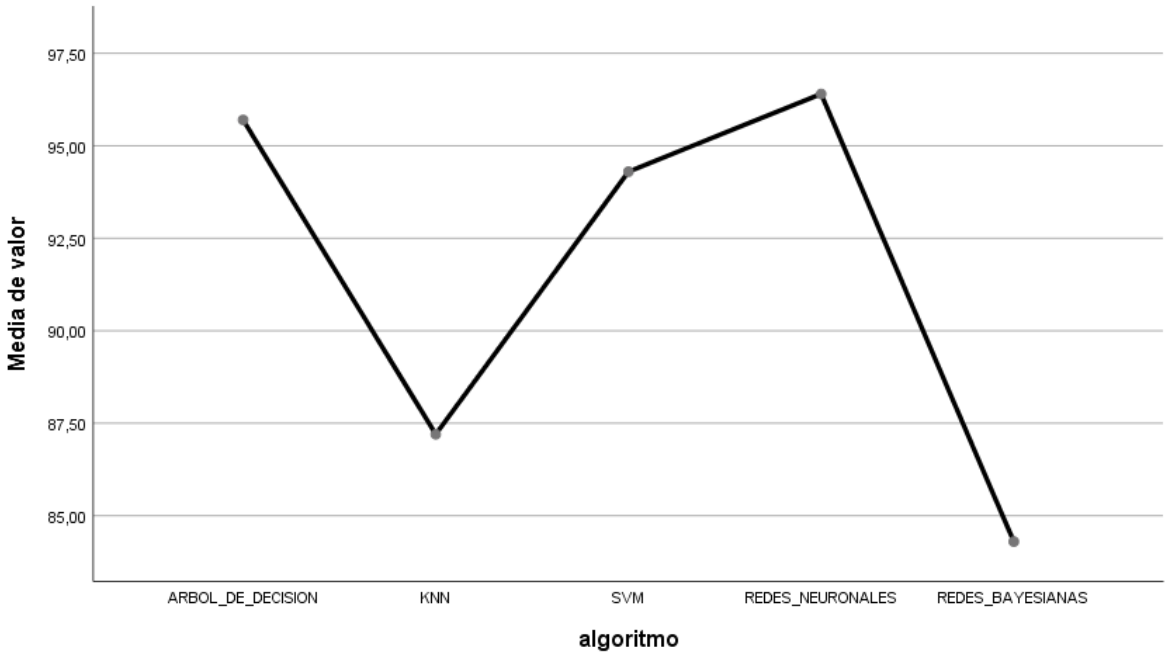
HE5: Existe diferencia estadística significativa en la puntuación F1 las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS.

La Tabla 35 presenta los resultados obtenidos para la métrica 'Puntuación F1', utilizando validación cruzada con 10 particiones, donde los valores de cv varían entre 2 y 11.

Tabla 29. *Resultados de la métrica Puntuación F1*

cv	Árbol de decisión	KNN	SVM	Redes Neuronales	Redes Bayesianas
2	97.07%	87.96%	94.74%	97.75%	85.69%
3	95.61%	88.20%	94.31%	96.58%	84.95%
4	95.27%	86.78%	93.92%	96.77%	83.59%
5	95.17%	85.72%	95.26%	95.79%	85.26%
6	95.60%	87.18%	94.96%	95.35%	83.96%
7	95.17%	87.75%	93.96%	95.68%	84.40%
8	95.79%	86.78%	93.17%	95.67%	84.73%
9	95.52%	87.30%	93.44%	96.14%	83.46%
10	95.44%	88.62%	94.90%	95.56%	84.19%
11	95.95%	85.26%	94.79%	96.78%	83.28%
total	956.59%	871.55%	943.45%	962.07%	843.51%
promedio	**95.66%**	**87.16%**	**94.35%**	**96.21%**	**84.35%**

Figura 22 *Resultados de la métrica Puntuación F1*



Fuente: SPSS Statistics v.25

Interpretación: Así como es mostrado en la tabla 35 y Figura 22, la técnica con el porcentaje de nivel óptimo referente a la métrica puntuación F1 que ayuda a predecir el rendimiento académico de los alumnos de la UNAS correctamente es “Redes neuronales” = 96.21%, a la vez que “Árbol de decisión” = 95.66%, luego “SVM” con 94.35%, asimismo sigue “KNN” con un valor del 87.16% y por último “Redes bayesianas” con 84.35%.

Estos resultados se contrastan con el estudio de Yağcı (2022), propusieron un modelo en base a técnicas de machine learning para la predicción de las calificaciones de los exámenes finales de los alumnos de educación superior, donde se evidencia que la técnica KNN alcanzó una puntuación F1 del 69.4%, lo que significa que el empleo de técnicas de machine learning es eficaz para la predicción. Asimismo, en el estudio de Singh y Pal (2020) desarrollaron un modelo combinado en base las técnicas de machine learning Bagging y Boosting y compararon 4 técnicas, donde se puede observar que Naive Bayes alcanzó el mejor valor de F1-Score=72.78%, lo cual quiere decir que las técnicas tienen eficiencia para mejorar la calidad de la enseñanza e identificar que estudiantes requieren más atención.

Para realizar la contrastación de la hipótesis de la puntuación F1 se realizó la prueba de normalidad de todos los grupos independientes obteniendo los resultados como se muestra en la Tabla 36.

Tabla 36. *Prueba de normalidad de Puntuación F1*

Pruebas de normalidad						
	Kolmogorov-Smirnov			Shapiro-Wilk		
Técnicas	Estadístico	gl	Sig	Estadístico	gl	Sig.
Arbol_decision	,272	10	0,035	0,802	10	0,015
KNN	,230	10	0,143	0,933	10	0,479
SVM	,302	10	0,010	0,781	10	0,008
Redes_neuronales	,282	10	0,023	0,890	10	0,172
Redes_bayesianas	,224	10	0,168	0,911	10	0,287

En la Tabla 36, se expone que el valor de gl es menor que 30 por lo cual analizaremos la prueba de normalidad en base a shapiro-wilk, en la cual se puede apreciar que el grupo de KNN, redes neuronales y redes bayesianas tiene un nivel de significancia mayor a 0.05 lo que indica que son paramétricas y siguen una distribución normal, así mismo los grupos de Árbol de decisión y SVM, tiene un nivel de significancia inferior que 0.05 lo que indica que no son paramétricas y no siguen una distribución normal, por lo tanto para la contratación de hipótesis se usó la prueba Kruskal-Wallis (Ver Tabla 37.)

Tabla 37. *Prueba de Kruskal-Wallis de Especificidad.*

Estadísticos de prueba	
	Valor
H de Kruskal-Wallis	44,261
gl	4
Sig asintótica	,000

Interpretación de los resultados:

1. **Hipótesis de la prueba:**
   - Hipótesis nula (H0): No hay diferencias significativas entre las medianas de los grupos (técnicas).
   - Hipótesis alternativa (H1): Al menos un grupo (técnica) tiene una mediana significativamente diferente.
2. **Estadístico de Kruskal-Wallis (H)**
   - El valor H = 44,261 indica el grado de discrepancia entre los distintos grupos
   - Este valor se compara con una distribución chi-cuadrado con gl=4 (grados de libertad).
3. **Grados de libertad (gl):**
   - La prueba se realizó con 4 grados de libertad, lo que significa que se están comparando 5 grupos en total.
4. **Significación asintótica (p-valor):**
   - El valor de significancia reportado es p=.000, lo que significa que es menor que el nivel típico de significancia (α=0.05).
   - Como el valor p es menor a 0.05, se procede a rechazar la hipótesis nula.

Conclusión:

La Tabla 37 muestra un p-valor inferior a 0.05, lo que permite concluir que hay diferencias estadísticamente significativas entre las medianas de las distintas técnicas analizadas con relación a la especificidad. Esto sugiere que al menos una de las técnicas presenta un rendimiento diferente en comparación con los demás.

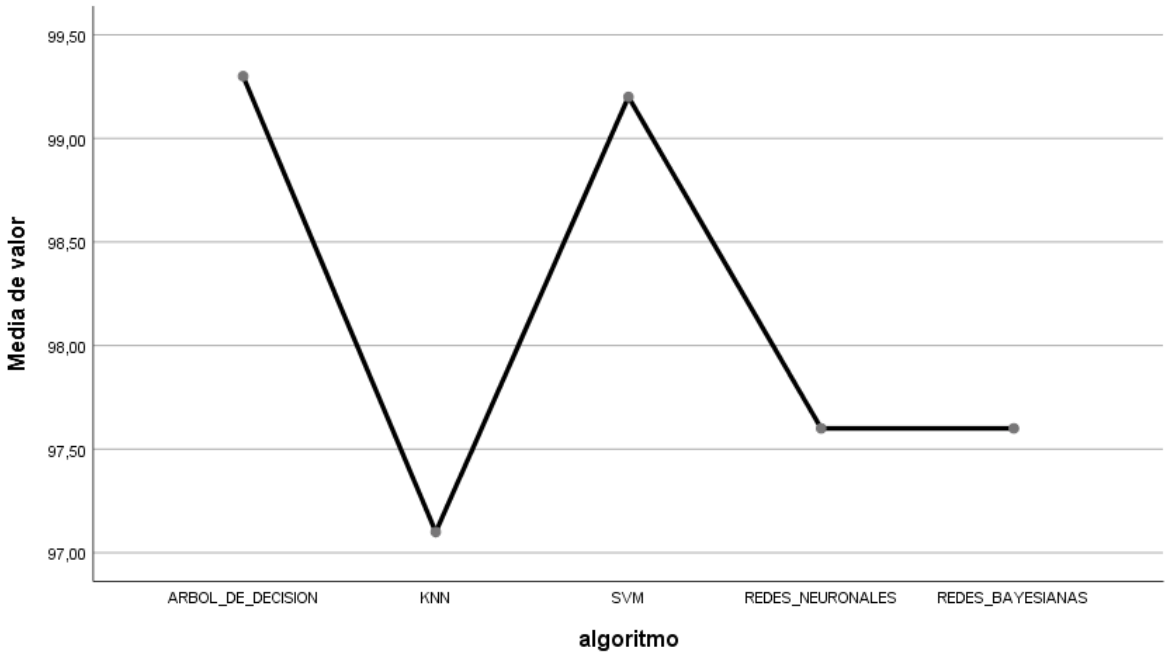
HE6: Existe diferencia estadística significativa en la curva ROC las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS.

En la Tabla 38 se exhiben los resultados obtenidos para la métrica 'Curva ROC', utilizando validación cruzada con 10 particiones de los datos, donde los valores de cv varían desde 2 hasta 11

Tabla 30. Resultados de la métrica Curva ROC

cv	Árbol de decisión	KNN	SVM	Redes Neuronales	Redes Bayesianas
2	99.55%	97.60%	99.61%	97.58%	97.58%
3	99.52%	97.61%	99.58%	97.66%	97.66%
4	98.89%	96.91%	99.28%	97.46%	97.46%
5	99.03%	96.74%	99.45%	97.78%	97.78%
6	99.24%	97.47%	99.46%	97.49%	97.49%
7	99.34%	97.01%	99.41%	97.67%	97.67%
8	99.29%	96.58%	99.26%	97.50%	97.50%
9	99.46%	97.36%	99.44%	97.33%	97.33%
10	99.39%	97.44%	99.41%	97.71%	97.71%
11	99.53%	96.29%	99.34%	97.22%	97.22%
total	993.24%	971.01%	994.24%	975.40%	975.40%
promedio	**99.32%**	**97.10%**	**99.42%**	**97.54%**	**97.54**

Figura 23 *Resultados de la métrica Curva ROC*



Fuente: SPSS Statistics v.25

Interpretación: Tal como se indica en la tabla 38 y Figura 23, la técnica con % óptimo referente a la Curva ROC que ayuda a predecir correctamente es tanto "SVM" =99.42%, seguido de "Árbol de decisión" = 99.32%, a la vez que tanto "Redes neuronales" como "Redes bayesianas" = 97.54% y finalmente "K-NN" =97.10%.

Esto es contrastado con la investigación de Menacho (2017), donde aplicaron técnicas de data mining para predecir la clasificación final de los estudiantes pertenecientes a la Universidad Nacional Agraria La Molina, cuyos resultados muestran que el Clasificador bayesiano ingenuo logró un alto valor de Curva ROC=62.00%, lo que quiere decir que las técnicas de data mining son herramientas eficaces para la obtención de modelos que ayuden a predecir los resultados de los estudiantes. A la vez, se reafirma con la investigación de Orihuela (2019), donde se evidencia que Bosque aleatorio obtuvo un valor de Curva ROC de 82.00%, esto demuestra que esta técnica logra predecir el rendimiento académico.

Para realizar la contrastación de la hipótesis de la curva ROC se realizó la prueba de normalidad de todos los grupos independientes obteniendo los resultados como se expone en la Tabla 39.

Tabla 39. *Prueba de normalidad de Curva ROC*

Pruebas de normalidad						
	Kolmogorov-Smirnov			Shapiro-Wilk		
Técnicas	Estadístico	gl	Sig	Estadístico	gl	Sig.
Arbol_decision	,433	10	0,000	0,594	10	0,000
KNN	,370	10	0,000	0,752	10	0,004
SVM	,482	10	0,000	0,509	10	0,000
Redes_neuronales	,381	10	0,000	0,640	10	0,000
Redes_bayesianas	,381	10	0,000	0,640	10	0,000

En la Tabla 39, se puede observar que el valor de gl es menor que 30 por lo cual analizaremos la prueba de normalidad en base a shapiro-wilk, en la cual se puede apreciar que el grupo KNN, redes neuronales, redes bayesianas, Árbol de decisión y SVM, tiene un nivel de significancia inferior al 0.05 lo que indica que no son paramétricas y no se sigue una distribución normal, por lo tanto, para la contratación de hipótesis se usó el test estadístico no paramétrico de Kruskal-Wallis (Ver Tabla 40.)

Tabla 40. *Prueba de Kruskal-Wallis de Curva ROC.*

Estadísticos de prueba	
	Valor
H de Kruskal-Wallis	39,727
gl.	4
Sig asintótica	,000

Interpretación de los resultados:

1. **Hipótesis de la prueba:**
   - Hipótesis nula (H0): No existen diferencias significativas entre las medianas de los grupos (técnicas).
   - Hipótesis alternativa (H1): Por lo menos un grupo (técnica) tiene una mediana significativamente diferente.
2. **Estadístico de Kruskal-Wallis (H)**
   - El valor calculado de la estadística H=39,727 revela la magnitud de la diferencia entre los grupos.
   - Dicho valor se contrasta con una distribución chi-cuadrado con gl=4 (grados de libertad).
3. **Grados de libertad (gl):**
   - La prueba se realizó con 4 grados de libertad, lo que significa que se están comparando 5 grupos en total.

4. **Significación asintótica (p-valor):**
    - El valor de significancia reportado es p=.000, lo que significa que es menor que el nivel típico de significancia que es (α=0.05).
    - Debido a que el p-valor es menor a 0.05, se descarta la hipótesis nula

Conclusión:

En la Tabla 40 se indica que el p-valor es menor a 0.05, se concluye que se identifican diferencias significativas entre las medianas de las distintas técnicas analizadas con relación a la métrica de la curva ROC. Esto sugiere que al menos una de las técnicas presenta un rendimiento diferente en comparación con los demás.

V. CONCLUSIONES

1. Se determinó que la técnica que proporciona la mejor exactitud para predecir el rendimiento académico de los estudiantes de la UNAS después de realizar la comparación de los algoritmos en el test es “Redes Neuronales” con un 96.24%. Luego se aplicó la prueba estadística de Kruskal-Wallis con un nivel de significancia de 0.00. Esto significa que el 96.24% de las predicciones del modelo son correctas. Este hallazgo se corrobora con investigaciones que han encontrado que las redes neuronales supera a otras técnicas en tareas de predicción complejas, como la identificación de estudiantes que se encuentran en riesgo de bajo rendimiento académico.

2. Se determinó que la técnica que brinda la mejor precisión para la predicción el rendimiento académico de los alumnos de la UNAS después de realizar la comparación de los algoritmos en el test es “Árbol de decisión” con un 95.72%. Luego se aplicó la prueba estadística de Kruskal-Wallis, indicando como nivel de significancia 0.00, lo que indica que el 95.72% de las predicciones de “excelente” del modelo son correctas, siendo esta métrica importante cuando el costo de los falsos positivos es elevado, ya que mide la calidad de las predicciones positivas del modelo. Este hallazgo tiene una relevancia práctica significativa, ya que aborda el problema real de predecir el rendimiento académico en la UNAS, permitiendo la implementación de estrategias de intervención temprana para los estudiantes en riesgo, lo que la alta precisión del árbol de decisión facilita una toma de decisiones más efectiva contribuyendo a mejorar el apoyo y las oportunidades de éxito académico de los estudiantes.
3. Se determinó que la técnica que muestra la mejor sensibilidad para predecir el rendimiento académico de los alumnos de la UNAS después de realizar la comparación de los algoritmos en el test es “Redes Neuronales” con un 96.23%. Luego se aplicó la prueba estadística de Kruskal-Wallis con un nivel de significancia de 0.00. Esto indica que el modelo tiene la capacidad de reconocer de forma correcta el 96.23% de los estudiantes tienen un rendimiento “excelente”, siendo esta métrica de gran importancia cuando la meta es obtener gran cantidad posible de instancias positivas, sobre todo cuando los errores por omisión (falsos negativos) resultan críticos. Este hallazgo tiene importancia práctica, ya que estudios de este tipo contribuyen directamente a mejorar la calidad educativa y el futuro académico de los estudiantes, donde la alta sensibilidad del modelo permite identificar de manera precisa a los estudiantes con un buen

rendimiento, facilitando la implementación de medidas de apoyo personalizado y oportuno.

4. Se determinó que la técnica que indica la mejor especificidad para predecir el rendimiento académico de los alumnos de la UNAS después de realizar la comparación de los algoritmos en el test es “Árbol de decisión” con un 95.68%. Luego se utilizó la prueba Kruskal-Wallis, teniendo un nivel de significancia de 0.00. Lo que indica que el modelo tiene la capacidad de identificar correctamente el 95.68% de los estudiantes que no tienen un rendimiento “excelente”, siendo está métrica fundamental cuando es crucial identificar correctamente las instancias negativas, principalmente cuando el costo de los falsos positivos es elevado. Este hallazgo tiene importancia práctica, ya que no solo optimiza los procesos académicos, sino que también tiene un impacto positivo en el entorno educativo, promoviendo una atención más equilibrada y equitativa para todos los estudiantes, reforzando la capacidad del Árbol de decisión como una herramienta beneficiosa para la mejora de la calidad de vida académica.
5. Se determinó que la técnica que produce la mejor puntuación F1 para predecir el rendimiento académico de los alumnos de la UNAS después de realizar la comparación de los algoritmos en el test es “Redes Neuronales” con un 96.21%. Luego se aplicó la prueba estadística de Kruskal-Wallis, contando con un nivel de significancia de 0.00. Esto significa que la puntuación F1 del modelo para la categoría “excelente” es 96.21%, lo que indica un equilibrio moderado entre la precisión y la sensibilidad, siendo una métrica clave para evaluar los modelos de clasificación, especialmente cuando hay un desbalance entre las clases o cuando se desea encontrar un balance entre la precisión y la sensibilidad. Este hallazgo se corrobora teóricamente, al contribuir al conocimiento del comportamiento de diversas variables que presentan influencia sobre rendimiento académico, donde el uso de técnicas como Redes neuronales no solo fortalece la comprensión del proceso predictivo, sino que también ofrece una base sólida para futuras investigaciones.
6. Se determinó que la técnica que brinda la mejor Curva ROC para predecir después de realizar la comparativa de los algoritmos en el test es “SVM” con un 99.42%. Luego se aplicó el test Kruskal-Wallis, presentando un nivel de significancia de 0.00. Lo que significa que esta métrica proporciona una forma poderosa de evaluar la capacidad de un modelo de clasificación binaria (uno contra todos) para hacer la distinción entre las clases positivas y negativas a través de diferentes umbrales acerca de la decisión, como es el caso, para “excelente” versus “no excelente” y luego repetir el proceso para las

otras categorías (“bueno” versus “no bueno”, etc.). Este hallazgo proporciona una evaluación detallada de como el alto desempeño del modelo SVM en términos de Curva ROC refuerza su utilidad en la predicción académica y destaca su capacidad para ofrecer una diferenciación precisa entre categorías de rendimiento.

7. En conclusión, en cuanto al objetivo general, después de haber realizado la comparación de los experimentos con todas las técnicas seleccionadas, se puede determinar que Redes neuronales presentó el mejor desempeño relacionado a la exactitud, precisión, sensibilidad, especificidad, puntuación F1 y Curva ROC, superando a Árbol de decisión, siendo las más eficaces para predecir el rendimiento académico de los alumnos de la UNAS. Sin embargo, KNN y Redes bayesianas presentaron valores más bajos, lo que sugiere que no son las más adecuadas para este conjunto de datos específicos. Estos hallazgos refuerzan la importancia de seleccionar el modelo adecuado para maximizar el impacto positivo en la calidad educativa, promoviendo una intervención temprana y personalizada que responda a las necesidades de los estudiantes.

VI. RECOMENDACIONES

Se sugiere desarrollar enfoques alternativos de machine learning, como son técnicas híbridas que permitan procesar datos de entrada con el objetivo de estimar valores de salida dentro de márgenes aceptables relacionados con la predicción del rendimiento académico, tales como métodos de ensamblaje, modelos en cascada, árboles extremadamente aleatorios y técnicas de agrupamiento.

Graduar la sensibilidad del modelo, donde sea capaz de detectar a estudiantes con rendimiento bueno, malo, muy bueno o regular, esto para formular un modelo más preciso y que pudiera discernir y dar luces del grado de rendimiento que presenta el alumno.

Se recomienda actualizar periódicamente la base de datos en el modelo predictivo para que cuente con información relevante frente a los problemas actuales, favoreciendo la toma de mejores decisiones.

En base a los resultados, se evidenció que las redes neuronales obtuvieron los mejores valores en las métricas, lo que indica que son las más eficientes. Por ello se sugiere la construcción de técnicas optimizadas que tengan en su estructura hiperparámetros que permitan mejorar los resultados como las redes neuronales artificiales con arquitecturas como perceptrón multicapa, (MLP) o redes neuronales profundas que son adecuadas para tareas de clasificación con datos discretos y continuos. Por ello, se recomienda continuar la línea de esta investigación desarrollando modelos con tales características.

Dar facilidades en brindar la información requerida para que la investigación sea relevante.

Finalmente, se sugiere desarrollar un mecanismo de recopilación de información, esto para actualizar los datos del modelo con nuevos datos, concretos acerca del rendimiento de los estudiantes tales como exámenes, y desarrollar metodologías más complejas que requieren mayor cantidad de información como son las técnicas de regresión.

VII. REFERENCIAS

Alban, M. (2019). Contribuciones a la Predicción de la Deserción Universitaria a través de Minería de Datos. In *Universidad Nacional Mayor de San Marcos*. https://cybertesis.unmsm.edu.pe/handle/20.500.12672/10776

Arrascue, M. (2015). El rendimiento académico de los estudiantes de la UNAS.

Bahit, E. (2018). Introducción al lenguaje Python (1° ed.). Creative Commons Atribución 4.0. https://www.researchgate.net/publication/333965199_Introduccion_al_Lenguaje_Python

Bernal, C. (2010). Metodología de la investigación (3ª. Ed.). Pearson Education.

Buenaño, D., Gil, D. y Luján, S. (2019). Application of Machine Learning in Predicting Performance for Computer Engineering Students: A Case Study. Sustainability, 11(10), 2833. https://doi.org/10.3390/su11102833

Burgos, C. et al (2018). Data mining for modeling students' performance: A tutoring action plan to prevent academic dropout. Computers and Electrical Engineering, 66(1), 541–556.https://doi.org/10.1016/j.compeleceng.2017.03.005

Candia Oviedo, D. I. (2019). Predicción del rendimiento académico de los estudiantes de la UNSAAC a partir de sus datos de ingreso utilizando algoritmos de aprendizaje automático. http://hdl.handle.net/20.500.12918/4120

Cao, X., Masood, A., Luqman, A., & Ali, A. (2018). Excessive use of mobile social networking sites and poor academic performance: Antecedents and consequences from stressor-strain-outcome perspective. Computers in Human Behavior, 85, 163–174. https://doi.org/10.1016/j.chb.2018.03.023.

Cea, M. (1998). Metodología cuantitativa: estrategias y técnicas de investigación social. Editorial Síntesis.

Chay, J. (2016). Principales Factores que influyen en el bajo rendimiento de los estudiantes en las áreas de Matemáticas y Comunicación y Lenguaje L1 del Instituto Nacional de Educación Básica INEB, Santo Tomás la Unión, Suchitepéquez. In *Universidad de San Carlos de Guatemala*.

Contreras, L., Fuentes, H. y Rodríguez, J. (2020). Predicción del rendimiento académico como indicador de éxito/fracaso de los estudiantes de ingeniería, mediante aprendizaje automático. Formación Universitaria,13(5), 233-246. https://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-

50062020000500233&lang=pt

Dabhade, P. et al (2021). Educational data mining for predicting students' academic performance using machine learning algorithms. 47(1), 5260-5267.https://www.sciencedirect.com/science/article/pii/S2214785321042735

Del Valle, A. (2017). Curvas ROC (Receiver-Operating-Characteristic) y sus aplicaciones. [Tesis de pregrado, Universidad de Sevilla]. Archivo digital. https://idus.us.es/bitstream/handle/11441/63201/Valle%20Benavides%20Ana%20Roc%C3%ADo%20del%20TFG.pdf

De La Hoz, E., y Fontalvo, T. (2019). Methodology of Machine Learning for the classification and Prediction of users in Virtual Education Environments. Información Tecnológica, 30, 247–254. https://doi.org/10.4067/S0718-07642019000100247

Đurđević, I. (2017). Machine learning methods in predicting the student academic motivation. Croatian Operational Research Review. 443–461. https://www.researchgate.net/publication/323170777_Machine_learning_methods_in_predicting_the_student_academic_motivation

Espinoza Airac, G. X., & León Muñoz, E. F. (2020). Modelo de Machine Learning para la clasificación de estudiantes de acuerdo a su rendimiento académico en el Centro de Idiomas de la Universidad Nacional del Santa. http://repositorio.uns.edu.pe/handle/UNS/3588

Esposito, D., & Esposito, F. (2020). *Introducing Machine Learning*.

Fernandes, E. et al (2019). Educational data mining: Predictive analysis of academic performance of public-school students in the capital of Brazil. Journal of Business Research, 94, 335–343. https://doi.org/10.1016/j.jbusres.2018.02.012

Flores, G. et al (2019). Minería de datos como herramienta estratégica. Revista Científica Mundo de la Investigación y el Conocimiento, 3(1), 1–16. https://www.recimundo.com/index.php/es/article/view/400

Garcia, K., Pino, J. y Muñoz, J.(2019), Learning Analytics as an analysis factor of university academic performance. CEUR Workshop Proceedings, 2231, 42-50. http://ceur-ws.org/Vol-2231/LALA_2018_paper_14.pdf

Geron, A. (2019). *Hands-on Machine Learning with Scikit-Learn, Keras & Tensorflow* (Second Edi). O'Reilly Media, Inc.

Gonzalez, J. L. (2020, julio 13). *Tipos de aprendizaje automático*. Medium.

https://medium.com/soldai/tipos-de-aprendizaje-autom%C3%A1tico-6413e3c615e2

Guillán, A. (2016). Predicción científica y valores: Análisis de la dimensión estructural y de la componente dinámica. Formación universitaria, 13(1), 1-19. https://scielo.conicyt.cl/scielo.php?pid=S0718-50062020000100093&script=sci_arttext

Harvey, J. y Kumar, S. (2019). A Practical Model for Educators to Predict Student Performance in K-12 Education using Machine Learning. Conference 2019 IEEE Symposium Series on Computational Intelligence (SSCI), 3004-3011. https://doi.org/10.1109/SSCI44817.2019.9003147

Henriquez, C., Salcedo, D. y Sanchez. G. (2022). El aprendizaje automático en entornos educativos universitarios: Caso deserción académica. Prospectiva, 20(1),1-12. https://doi.org/10.15665/rp.v20i1.2736

Hernández, R., Fernández, C., y Baptista, P. (2014). *Metodología de la investigación* (6ª. Ed.). McGraw-Hill.

Hernández, S., & Mendoza, C. (2018). *Metodología de la Investigación*.

Hurtado, J. (2012). Feng, S. et al (2019). The internet and facebook usage on academic distraction of college students. Computers & Education, 134, 41–49. https://doi.org/10.1016/j.compedu.2019.02.005 ed.). Bogotá-Caracas: Ciea-Sypal y Quirón.

Ionos. (2019). *Jupyter Notebook: la herramienta de Python para procesar datos - IONOS*. https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/jupyter-notebook/

Katarya, R., Gaba, J., Garg, A. y Verma, V. (2021), A review on machine learning based student's academic performance prediction systems, 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), 254-259. https://doi.org/10.1109/ICAIS50930.2021.9395767

Khan, I., Rahim, A., Jabeur, N. y Mahdi, M. (2021). A Conceptual Framework to Aid Attribute Selection in Machine Learning Student Performance Prediction Models. International Journal of Interactive Mobile Technologies (iJIM), 15(15),4. https://doi.org/10.3991/ijim.v15i15.20019

Lau, E. T., Sun, L., & Yang, Q. (2019). Modelling, prediction and classification of student academic performance using artificial neural networks. SN Applied Sciences, 1(9), 1–10. https://doi.org/10.1007/s42452-019-0884- 7

Lozada, J. (2014). Investigación aplicada: Definición, propiedad intelectual e industria. Revista de divulgación científica de la Universidad Tecnológica Indoamérica, 3(1), 47-50.

Maimon, O. & Rokach, L., (2010). *Data mining and knowledge discovery handbook* (2nd ed.). Springer New York. https://doi.org/10.1007/978-0-387-09823-4

Menacho Chiok, C. H. (2017). Predicción del rendimiento académico aplicando técnicas de minería de datos. https://dialnet.unirioja.es/descarga/articulo/6171237.pdf

Ministerio de Economia y Finanzas. (2021). Poder Ejecutivo presenta proyecto de Ley de Presupuesto para el Año Fiscal 2022, por S/ 197 mil millones. https://www.mef.gob.pe/index.php/?option=com_content&view=article&id=7140&Itemid=101108&lang=es

Moreira, J. y Ruete,D.(2020). Propuesta metodológica basada en la aplicación de minería de datos mediante un modelo de gestión de proyectos para apoyar la toma de decisiones académicas en una institución de educación superior. [Tesis de pregrado, Universidad Andrés Bello]. Archivo digital. http://repositorio.unab.cl/xmlui/bitstream/handle/ria/14749/a130617_Moreira_J_Propuesta_metodologica_basada_en_la_aplicacion_2020_Tesis.pdf?sequence=1&isAllowed=y

Musso, M., Rodríguez, C. y Cascallar, E. (2020). Predicting key educational outcomes in academic trajectories: a machine-learning approach. Higher Education. 80, 875–894. https://doi.org/10.1007/s10734-020-00520-7

Müller, A. C., & Guido, S. (2017). Introduction to with Python Learning Machine. In *Proceedings of the Speciality Conference on Infrastructure Condition Assessmenr: Art, Science, Practice*.

Naik, P. & Oza, K. (2019). Python with Spyder: An Experiential Learning Perspective. Shashwat publication.

Negi, A. y Jaiswal, V(2016). A first attempt to develop a diabetes prediction method based on different global datasets. Parallel, Distributed and Grid Computing (PDGC), 2016 Fourth International Conference on IEEE. https://ieeexplore.ieee.org/document/7913152

Nieto, Y. et al (2018). Supporting academic decision making at higher educational institutions using machine learning-based algorithms. Soft Computing, 23, 4145–4153. https://doi.org/10.1007/s00500-018-3064-6

Nithya, P., Umamaheswari, B., y Umadevi, A. (2016). A Survey on Educational Data Mining in Field of Education. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) ,16(1), 145-153. https://www.semanticscholar.org/paper/A-Survey-on-Educational-Data-Mining-in-Field-of-Nithya-Umamaheswari/2df5ca68026fc4c93bc88ec713678c7d02e2c49c

Olortegui, L. (2024). Estrategia educativa para potenciar el rendimiento académico universitario desde la satisfacción estudiantil. [Tesis de doctorado, Universidad San Ignacio de Loyola]. Archivo digital.

https://repositorio.usil.edu.pe/server/api/core/bitstreams/96f218ae-3496-4c2d-84ad-f4a6caa67a12/content

Oyola-García, A. E. (2021). La variable. Revista del Cuerpo Médico Hospital Nacional Almanzor Aguinaga Asenjo, 14(1), 90-93. https://doi.org/10.35434/rcmhnaaa.2021.141.905

Orihuela Maita, G. Y. (2019). Aplicación de Data Science para la predicción del rendimiento académico de los estudiantes de la Facultad de Ingeniería de Sistemas de la Universidad Nacional del Centro del Perú. http://hdl.handle.net/20.500.12894/5837

Pavón, F. (2016). Generación de Conocimiento basado en Aprendizaje Automático y Aplicación en Diferentes Sectores. UNED. Universidad Nacional de Educación a Distancia (España). https://dialnet.unirioja.es/ servlet/tesis?codigo=66958

Pedamkar, P. (2022). *Técnicas de aprendizaje automático | Las 4 técnicas principales de aprendizaje automático*. Retrieved March 8, 2022, from https://www.educba.com/machine-learning-techniques/

Sánchez, C.S., Salas-Cernades, H. H., Maldonado, A. R., & Aguirre, E. J. (2022). Rendimiento académico de estudiantes, en una universidad pública peruana: un diagnóstico significativo para la toma de decisiones. *Paidagogo*, *4*(1), 4-20. https://doi.org/10.52936/p.v4i1.98

Prokopyev, M. et al (2020). Development of a Programming Course for Students of a Teacher Training Higher Education Institution Using the Programming Language Python. Propósitos y Representaciones, 8(3).http://www.scielo.org.pe/scielo.php?script=sci_arttext&pid=S2307-79992020000400023

Radhwan A., Abbas A, y Ali S. (2017). Popular Decision Tree Algorithms of Data Mining Techniques. International Journal of Computer Science and Mobile Computing, 6(6), 133–142 . https://www.researchgate.net/publication/317731072_Popular_Decision_Tree_Algorithms_of_Data_Mining_Techniques_A_Review

Raschka, S., & Mirjalili, V. (2017). Python machine learning (2nd ed.). Packt Publishing.

Russo, C. et al. (2016). Tratamiento masivo de datos utilizando técnicas de machine learning. XVIII Workshop de Investigadores en Ciencias de la Computación WICC 2016. https://bit.ly/31Vub1S

Sana, E. et al (2020). Predicting Students' Academic Performance Through Supervised Machine Learning. International Conference on Information Science and Communication

Technology IEEE.1-6. https://ieeexplore.ieee.org/document/9080033

Sánchez, P. y Garcia, J. (2017). A new methodology for neural network training ensures error reduction in time series forecasting. Journal of Computer Science, 13, 211–217. https://doi.org/10.3844/jcssp.2017.211.217

Sánchez, M. et al (2018). Sistema informático para la gestión y publicación de la producción científica de la Universidad Nacional de Loja. Iberian Conference on Information Systems and Technologies. 1 – 6. https://doi.org/10.23919/CISTI.2018.8398637

Shukla, X. y Parmar D. (2016) Python – A comprehensive yet free programming language for statisticians. Journal of Statistics and Management Systems, 19(2), 277 – 284.https://doi.org/10.1080/09720510.2015.1103446

Singh, R. and Pal, S. (2020). Application of Machine Learning Algorithms to Predict Students Performance. International Journal of Advanced Research in Computer Science, 29(5), 7249-7261.
https://www.researchgate.net/publication/342065464_Application_of_Machine_Learning_Algorithms_to_Predict_Students_Performance

Sisodia, D. (2018). Prediction of Diabetes using Classification Algorithms. Procedia Computer Science, 132, 1578–1585.https://doi.org/10.1016/j. procs.2018.05.122

Schuster, F. (2005). Explicación y Predicción
La validez del conocimiento en ciencias sociales. Editorial CLACSO. http://bibliotecavirtual.clacso.org.ar/ar/libros/secret/schuster/schuster.htm

Subasi A. (2020). *Practical machine learning for data analysis using python*. Academic Press.

Tamayo y Tamayo, M. (2008). El Proceso de la Investigación Científica. (4ª ed.). Editorial Limusa.

Timarán-Pereira, S. R., Hernández-Arteaga, I., Caicedo-Zambrano, S. J., Hidalgo-Troya, A. y AlvaradoPérez, J. C. (2016). El proceso de descubrimiento de conocimiento en bases de datos. En Descubrimiento de patrones de desempeño académico con árboles de decisión en las competencias genéricas de la formación profesional (pp. 63-86). Bogotá: Ediciones Universidad Cooperativa de Colombia. doi: http://dx.doi.org/10.16925/9789587600490

Veliz, C. (2020). *Aprendizaje automático* (Fondo Editorial de la Pontificia Universidad Catolica del Peru (ed.); Fondo Edit).

Villareal, F.(2016). Introducción a las técnicas de Pronósticos. Universidad Nacional del Sur, 1–121.
https://www.matematica.uns.edu.ar/uma2016/material/Introduccion_a_los_Modelos_de_Pronosticos.pdf

Viswanathan, S., & Vengatesh, S. (2021). Study Of Students' Performance Prediction Models Using Machine Learning. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, *12*(2), 3085–3091. https://doi.org/10.17762/turcomat.v12i2.2351

Watt, J., Borhani, R., & Katsaggelos, A. k. (2020). *Machine Learning Refined* (Segunda).

Xing, X. et al (2019). Prediction of academic performance associated with internet usage behaviors using machine learning algorithms. Computers in human behavior, 98(1), 166–173. https://www.sciencedirect.com/science/article/abs/pii/S0747563219301554

Yağcı, M. (2022). Educational data mining: prediction of students' academic performance using machine learning algorithms. Smart Learning Environments, 9(1), 11. https://doi.org/10.1186/s40561-022-00192-z

Yamao, E. (2018). Predicción del rendimiento académico mediante minería de datos en estudiantes del primer ciclo de las Escuela Profesional de Ingeniería de Computación y Sistemas, Universidad de San Martín de Porres, Lima-Perú. https://hdl.handle.net/20.500.12727/3555

Yousafzai, B., Hayat, M. y Afzal, S. (2020). Application of machine learning and data mining in predicting the performance of intermediate and secondary education level student. Education and information Technologies, 25(3), 4677–4697. https://doi.org/10.1007/s10639-020-10189-1

Zhang, Y., Qin, X., & Ren, P. (2018). Adolescents' academic engagement mediates the association between Internet addiction and academic achievement: The moderating effect of classroom achievement norm. Computers in Human Behavior, 89, 299–307. https://doi.org/10.1016/j.chb.2018.08.018.

ANEXOS

Anexo 1

Tabla 31. *Matriz de Consistencia*

Variable	Problema	Hipótesis	Objetivos	Dimensiones	Indicadores
	General		**General**		
Técnicas de Machine Learning	¿Existe diferencia en las métricas de evaluación de las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS?	Existe diferencia significativa en las métricas de evaluación de las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS	Comparar las técnicas de machine learning para predecir el rendimiento académico de los estudiantes de la UNAS		
	Específicos	**Específicos**	**Específicos**		
Predicción del Rendimiento Académico	¿Existe diferencia en la exactitud de las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS?	Existe diferencia estadística significativa en la exactitud de las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS	Evaluar la diferencia en la exactitud de las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS	Exactitud	Porcentaje de Exactitud
	¿Existe diferencia en la precisión de las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS?	Existe diferencia estadística significativa en la precisión de las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS	Evaluar la diferencia en la precisión de las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS	Precisión	Porcentaje de Precisión
	¿Existe diferencia en la sensibilidad de las técnicas de	Existe diferencia estadística significativa en la sensibilidad las	Evaluar la diferencia en la sensibilidad de las técnicas	Sensibilidad	Porcentaje de Sensibilidad

	Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS?	técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS	de Machine Learning en la predicción del rendimiento académico los estudiantes de la UNAS		
	¿Existe diferencia en la especificidad de las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS?	Existe diferencia estadística significativa en la especificidad las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS	Evaluar la diferencia en la especificidad de las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS	Especificidad	Porcentaje de Especificidad
	¿Existe diferencia en la puntuación F1 las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS?	Existe diferencia estadística significativa en la puntuación F1 las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS	Evaluar la diferencia en la puntuación F1 de las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS	Puntuación F1	Porcentaje de Puntuación F1
	¿Existe diferencia en la curva ROC de las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS?	Existe diferencia estadística significativa en la curva ROC las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS	Evaluar la diferencia en la curva ROC de las técnicas de Machine Learning en la predicción del rendimiento académico de los estudiantes de la UNAS	Curva ROC	Porcentaje de Curva ROC

Fuente: Elaboración propia

Anexo 2
Operacionalización de Variables

Tabla 32. *Matriz de Operacionalización de Variables*

Variables	Definición conceptual	Definición Operacional	Dimensiones	Indicadores	Escala de medición
Técnicas de Machine Learning	Se basan en algoritmos que aprenden a partir de un conjunto de ejemplos especificando para una entrada dada cuál debe ser la salida, de modo que cuando se les da una entrada nueva entradas, producirán la salida correcta.	Para desarrollar las técnicas de machine learning se utilizará la metodología KDD con sus respectivos pasos: Comprensión del dominio y Objetivos de KDD, Selección y adición, Preprocesamiento-Limpieza de datos, Transformación, Minería de datos, Evaluación e interpretación, Discovered Knowledge (Visualización e integración). (Maimon yRokach,2010)			
Predicción del Rendimiento Académico	Sirve para anticipar el rendimiento académico de los estudiantes haciendo uso de métricas de evaluación de predicción, permitiendo así identificar a estudiantes que tienen un bajo rendimiento académico.	Para medir la predicción del rendimiento académico de los estudiantes de la Universidad Nacional Agraria de la selva se hará uso de las métricas de evaluación de predicción con el fin de evaluar el desempeño de las técnicas de machine learning.	Exactitud	Porcentaje de Exactitud	Razón
			Precisión	Porcentaje de Precisión	Razón
			Sensibilidad	Porcentaje de Sensibilidad	Razón
			Especificidad	Porcentaje de Especificidad	Razón
			Puntuación F1	Porcentaje de Puntuación F1	Razón
			Curva ROC	Porcentaje de Curva ROC	Razón

Fuente: Elaboración propia

Anexo 3

Desarrollo de las técnicas de Machine Learning

a) SVM

Figura 24 *Conexión a los datos de Excel (SVM)*

Conexion a los datos Excel

```
In [1]: #librerias para el desarrollo del proyecto
import matplotlib.pyplot as plt #libreria para graficas
from matplotlib.colors import ListedColormap
import matplotlib.patches as mpatches

import seaborn as sb

#librerias pandas para el manejo de los datos
import pandas as pd
import pandas as pq
import pandas as pf
import pandas as filtro_filas

#enlazando los datos en el archivo excel
data = pd.read_excel("./data/Reporte_alumnos.xlsx",index_col=None)
data_over = pd.read_excel("./data/Reporte_alumnos_over.xlsx",index_col=None)
import numpy as np
#invocando a la libreria de clasificacion
from sklearn.tree import DecisionTreeClassifier
pd.options.mode.chained_assignment = None
```

Figura 25 *Datos de los atributos de rendimiento académico*

Mostrar Datos

```
In [2]: data
```

Out[2]:

	Id	appaterno	apmaterno	nombre	sexo	Departamento	Provincia	Distrito	Fecha_Nacimiento	Edad	...	anio_ingreso	ciclo_academico
0	1	ABAD	LINARES	CESAR ALEJANDRO	M	HUANUCO	LEONCIO PRADO	RUPA-RUPA	2001-11-13	20	...	2019	5
1	2	ABAD	RIVERA	ANGIE BRIGITTE	F	HUANUCO	LEONCIO PRADO	RUPA-RUPA	1998-10-29	23	...	2018	7
2	3	ABAD	RIVERA	JOIS SHIRLEY	F	HUANUCO	LEONCIO PRADO	RUPA-RUPA	2000-08-03	22	...	2017	8
3	4	ABAL	NIETO	MARIA CELENIA	F	HUANUCO	HUANUCO	AMARILIS	2000-05-27	22	...	2019	7
4	5	ABENDAÑO	MEZA	CESAR JHULINNIO	M	HUANUCO	LEONCIO PRADO	RUPA-RUPA	1997-11-13	24	...	2015	8
...	...	...	...	...	...	...	...	...	...	...	...	...	...
4579	4580	ZUÑIGA	TOLENTINO	JHORQUEENS YOHAN	M	HUANUCO	LEONCIO PRADO	RUPA-RUPA	2003-03-16	19	...	2013	7
4580	4581	ZURITA	SANTA CRUZ	MONICA AMANDA	M	SAN MARTIN	LAMAS	ZAPATERO	2000-11-07	21	...	2020	2
4581	4582	ZUTA	PAREDES	SEAN AKIRA	F	LORETO	UCAYALI	CONTAMANA	2004-06-05	18	...	2021	2
4582	4583	ZUTA	PAREDES	SOON YI IXANKA	F	LORETO	UCAYALI	CONTAMANA	1996-07-11	26	...	2014	10
4583	4584	ZUTA	PAREDES	ZYANKO KATIUSKA YVETTE	F	LORETO	UCAYALI	CONTAMANA	1994-01-10	28	...	2014	10

4584 rows × 26 columns

Figura 26 *Atributos de rendimiento académico con columnas eliminadas*

Eliminar columnas

```
In [5]: #se elimino 9 columnas quedadnod = 26 -9 = 17

pq = data.drop(['Id','appaterno','apmaterno','nombre','Fecha_Nacimiento','codigo_alumno',
                'beneficiario_pronabec', 'ciclo_academico','codcurricula'], axis=1)
```

```
In [6]: pq
```

Out[6]:

	sexo	Departamento	Provincia	Distrito	Edad	Estado_Civil	modalidad_ingreso	tipo_colegio_procedencia	escuela_profesinal	anio_ingreso	Total_C
0	M	HUANUCO	LEONCIO PRADO	RUPA-RUPA	20	SOLTERO	Centro Pre Universitario	Publico	INGENIERIA EN RECURSOS NATURALES RENOVABLES	2019	
1	F	HUANUCO	LEONCIO PRADO	RUPA-RUPA	23	SOLTERO	Centro Pre Universitario	Publico	CONTABILIDAD	2018	
2	F	HUANUCO	LEONCIO PRADO	RUPA-RUPA	22	SOLTERO	Exámen Ordinario	Publico	CONTABILIDAD	2017	
3	F	HUANUCO	HUANUCO	AMARILIS	22	SOLTERO	Exámen Ordinario	Publico	INGENIERIA EN INDUSTRIAS ALIMENTARIAS	2019	
4	M	HUANUCO	LEONCIO PRADO	RUPA-RUPA	24	SOLTERO	Centro Pre Universitario	Publico	AGRONOMIA	2015	
...	...	...	...	...	...	...	...	...	...	...	
4579	M	HUANUCO	LEONCIO PRADO	RUPA-RUPA	19	SOLTERO	Exonerados Primeros Puestos	Publico	INGENIERIA AMBIENTAL	2013	
4580	M	SAN MARTIN	LAMAS	ZAPATERO	21	SOLTERO	Convenios Especiales	Publico	ZOOTECNIA	2020	
4581	F	LORETO	UCAYALI	CONTAMANA	18	NaN	Exámen Ordinario	Publico	INGENIERIA EN RECURSOS NATURALES	2021	

Figura 27 *Visualización de datos incompletos*

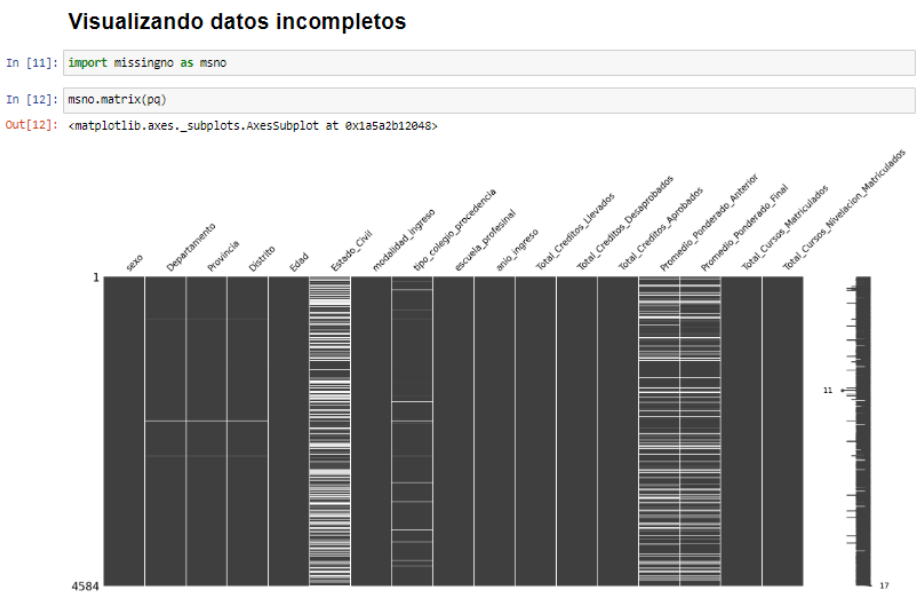


Figura 28 *Visualización de datos limpios*

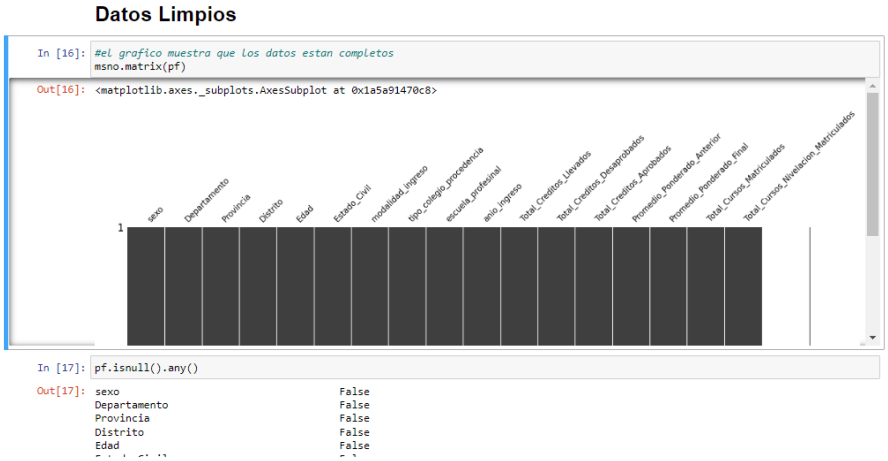


Figura 29 *Datos explorados (Cantidad de columnas)*

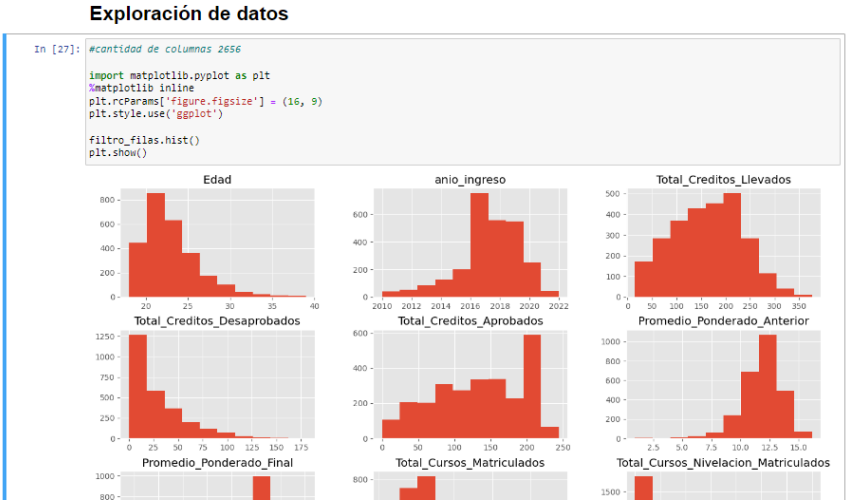


Figura 30 *Transformación de datos del promedio final*

Transformando datos de promedio final

```
In [30]: condicion_rango=[ ((pf['Promedio_Ponderado_Final'] <= 20) & (pf['Promedio_Ponderado_Final'] >= 18)),
                          ((pf['Promedio_Ponderado_Final'] < 18) & (pf['Promedio_Ponderado_Final'] >= 15)),
                          ((pf['Promedio_Ponderado_Final'] < 15) & (pf['Promedio_Ponderado_Final'] >= 13)),
                          ((pf['Promedio_Ponderado_Final'] < 13) & (pf['Promedio_Ponderado_Final'] >= 11)),
                          (pf['Promedio_Ponderado_Final'] < 11) ]
         escala_calificacion=['Excelente','Muy Bueno','Bueno','Regular','Malo']
         pf['CALIFICACION']=np.select(condicion_rango, escala_calificacion)

In [31]: condicion_rango=[ ((pf['Promedio_Ponderado_Final'] <= 20) & (pf['Promedio_Ponderado_Final'] >= 18)),
                          ((pf['Promedio_Ponderado_Final'] < 18) & (pf['Promedio_Ponderado_Final'] >= 15)),
                          ((pf['Promedio_Ponderado_Final'] < 15) & (pf['Promedio_Ponderado_Final'] >= 13)),
                          ((pf['Promedio_Ponderado_Final'] < 13) & (pf['Promedio_Ponderado_Final'] >= 11)),
                          (pf['Promedio_Ponderado_Final'] < 11) ]
         escala_calificacion=[5,4,3,2,1]
         pf['CALIFICACION_2']=np.select(condicion_rango, escala_calificacion)

In [32]: pf
```

Out[32]:

	sexo	Departamento	Provincia	Distrito	Edad	Estado_Civil	modalidad_ingreso	tipo_colegio_procedencia	escuela_profesinal	anio_ingreso	Total_C
0	M	HUANUCO	LEONCIO PRADO	RUPA-RUPA	20	SOLTERO	Centro Pre Universitario	Publico	INGENIERIA EN RECURSOS NATURALES RENOVABLES	2019	
1	F	HUANUCO	LEONCIO PRADO	RUPA-RUPA	23	SOLTERO	Centro Pre Universitario	Publico	CONTABILIDAD	2018	
2	F	HUANUCO	LEONCIO PRADO	RUPA-RUPA	22	SOLTERO	Exámen Ordinario	Publico	CONTABILIDAD	2017	

Figura 31 *Transformación de datos del estado civil*

Transformando datos de estado civil

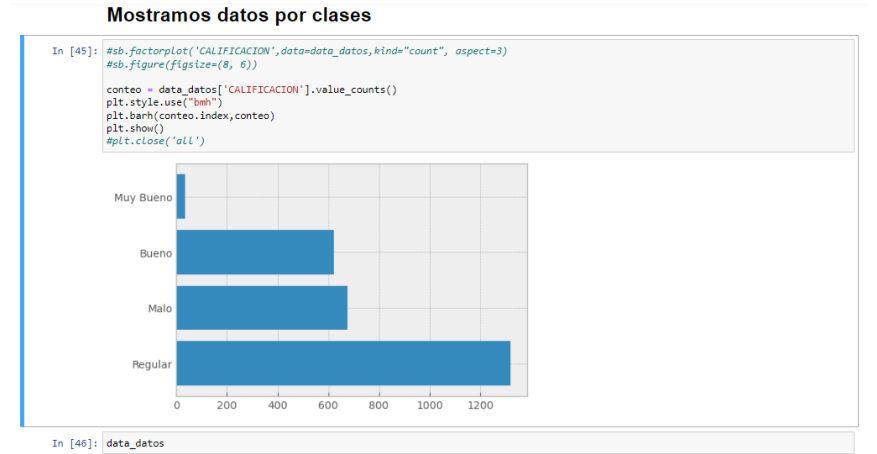
```
In [35]: # convertir estado civil
         condicion_estadocivil=[
             (pf['Estado_Civil'] == 'SOLTERO'),
             (pf['Estado_Civil'] == 'VIUDO'),
             (pf['Estado_Civil'] == 'CASADO'),
             (pf['Estado_Civil'] == 'DIVORCIADO')
         ]
         escala_estadocivil=['1','2','3','4']
         pf['ES_2']=np.select(condicion_estadocivil, escala_estadocivil)

In [36]: pf['ES_2']
```

Out[36]:

```
0       1
1       1
2       1
3       1
4       1
       ..
4578    1
4579    1
4580    1
4582    1
4583    1
Name: ES_2, Length: 2656, dtype: object
```

```
In [37]: condicion_TCP=[
             (pf['tipo_colegio_procedencia'] == 'Publico'),
             (pf['tipo_colegio_procedencia'] == 'Privado')
         ]
         escala_TCP=['0','1']
```

Figura 32 *Visualización de los datos por clases*



Mostramos datos por clases

```
In [45]: #sb.factorplot('CALIFICACION',data=data_datos,kind="count", aspect=3)
         #sb.figure(figsize=(8, 6))

         conteo = data_datos['CALIFICACION'].value_counts()
         plt.style.use("bmh")
         plt.barh(conteo.index,conteo)
         plt.show()
         #plt.close('all')
```



```
In [46]: data_datos
```

Figura 33 *Matriz de correlación de variables (SVM)*

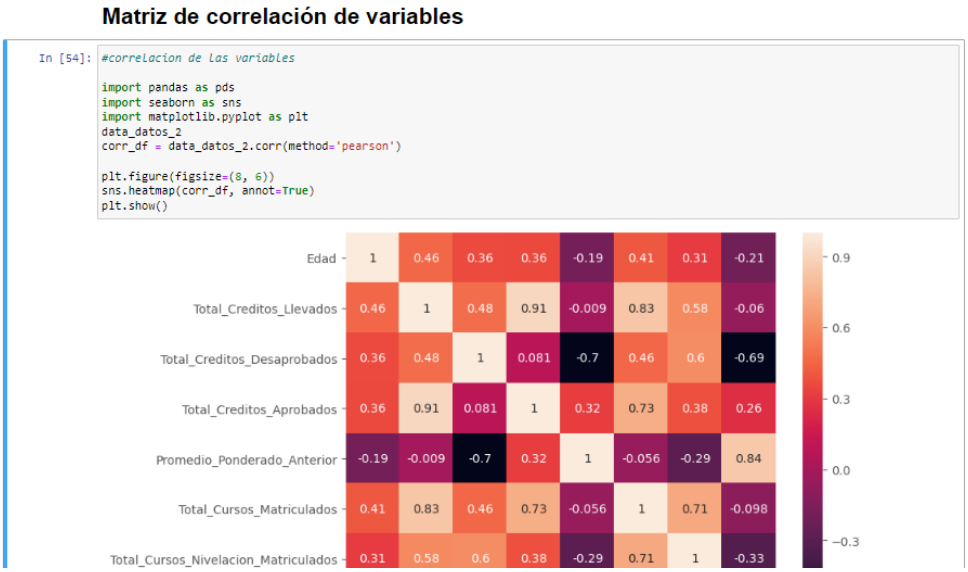


Figura 34 *Selección de variables predichas y predictoras*

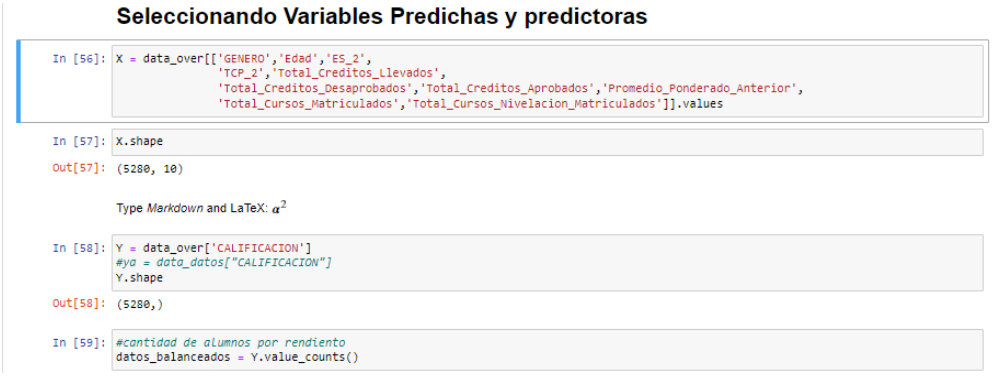


Figura 35 *Balanceo de datos*

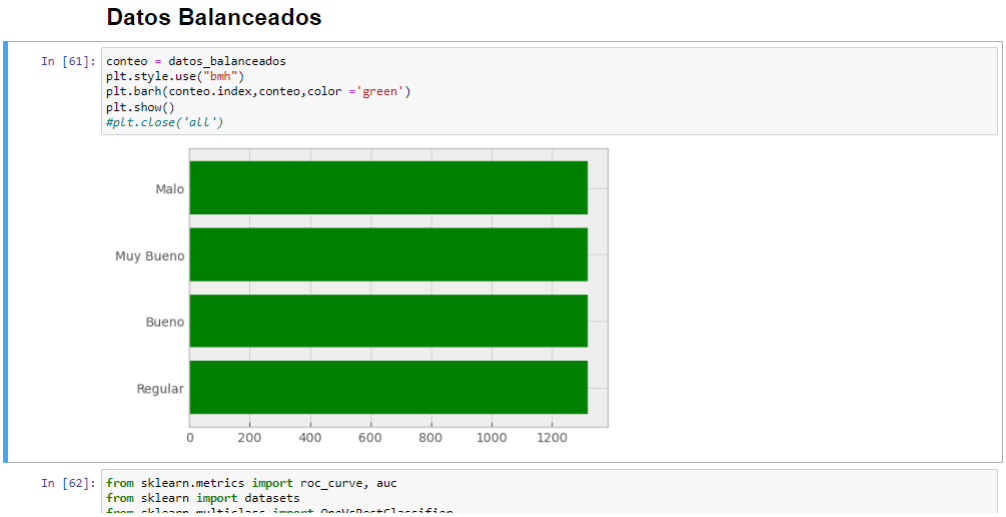


Figura 36 *Modelo SVM*

Modelo SVM

```
In [64]: from sklearn.model_selection import GridSearchCV
         from sklearn.svm import SVC

         # defining parameter range
         param_grid = {'C': [0.01, 0.1, 1],
                       'gamma': ['scale', 'auto'],
                       'kernel': ['rbf', 'linear']
                      }

         grid = GridSearchCV(SVC(probability=True),
                             param_grid,
                             refit = True,
                             verbose = 3,
                             cv = 3)

         # fitting the model for grid search
         grid.fit(X_trainset, y_trainset)
```

```
Fitting 3 folds for each of 12 candidates, totalling 36 fits
[CV 1/3] END ...C=0.01, gamma=scale, kernel=rbf;, score=0.478 total time=   8.1s
[CV 2/3] END ...C=0.01, gamma=scale, kernel=rbf;, score=0.494 total time=   7.4s
[CV 3/3] END ...C=0.01, gamma=scale, kernel=rbf;, score=0.525 total time=   7.4s
[CV 1/3] END C=0.01, gamma=scale, kernel=linear;, score=0.903 total time=   2.9s
[CV 2/3] END C=0.01, gamma=scale, kernel=linear;, score=0.902 total time=   2.8s
[CV 3/3] END C=0.01, gamma=scale, kernel=linear;, score=0.920 total time=   2.9s
[CV 1/3] END ....C=0.01, gamma=auto, kernel=rbf;, score=0.254 total time=  11.5s
[CV 2/3] END ....C=0.01, gamma=auto, kernel=rbf;, score=0.254 total time=  11.1s
[CV 3/3] END ....C=0.01, gamma=auto, kernel=rbf;, score=0.254 total time=  12.1s
[CV 1/3] END .C=0.01, gamma=auto, kernel=linear;, score=0.903 total time=   3.1s
[CV 2/3] END .C=0.01, gamma=auto, kernel=linear;, score=0.902 total time=   3.0s
```

Figura 37 *Resultados de las métricas del modelo SVM (Train)*

Métricas del Modelo (Train)

```
In [69]: pred = model.predict(X_trainset)
```

```
In [70]: from sklearn.metrics import confusion_matrix
         from sklearn.metrics import classification_report
```

```
In [71]: #accuracy solo es la diagonal / total
         from sklearn.metrics import accuracy_score
         print('acuracy =' , accuracy_score(y_trainset,pred))

         #Recall sensibilidad
         from sklearn.metrics import recall_score
         print('sensibilidad =',recall_score(y_trainset, pred, average='macro'))

         #precision
         from sklearn.metrics import precision_score
         print('precision =', precision_score(y_trainset, pred, average='macro'))

         #f1_score
         from sklearn.metrics import f1_score
         print('f1_score =',f1_score(y_trainset, pred, average='macro'))
```

```
acuracy = 0.9483901515151515
sensibilidad = 0.948379995352057
precision = 0.9480197717653517
f1_score = 0.9481105044475984
```

```
In [72]: print(classification_report(y_trainset, pred))
```

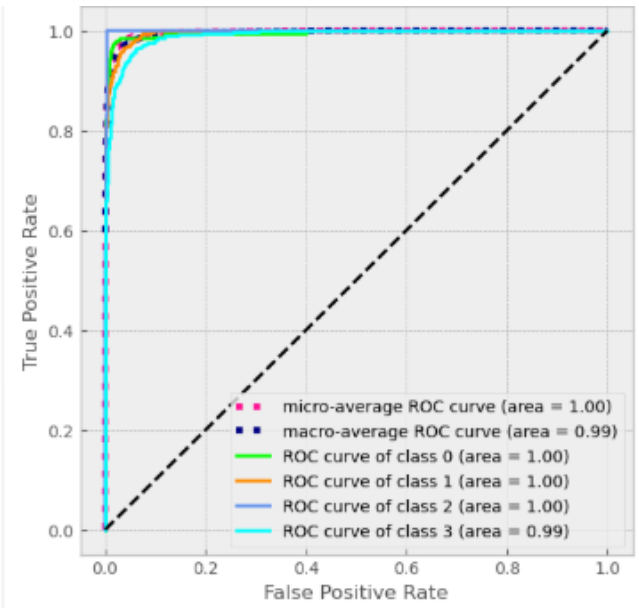


Figura 38 *Matriz de Confusión del modelo SVM (Train)*

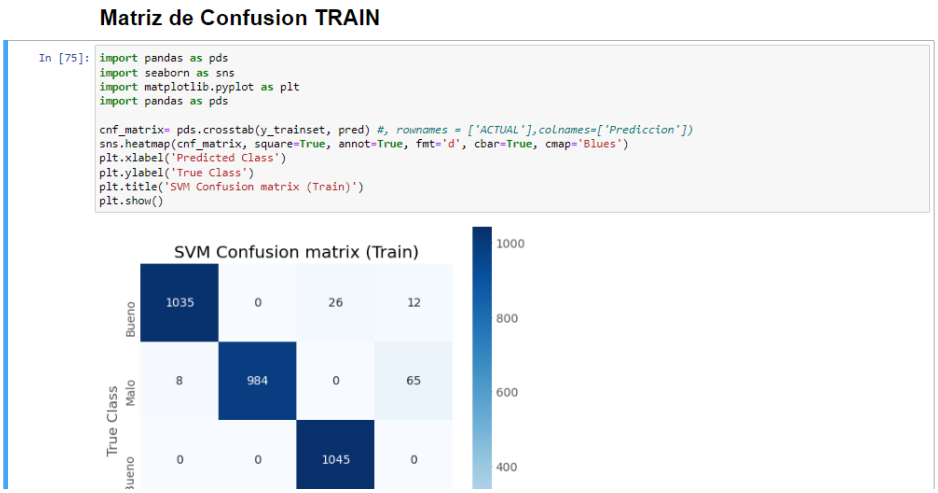
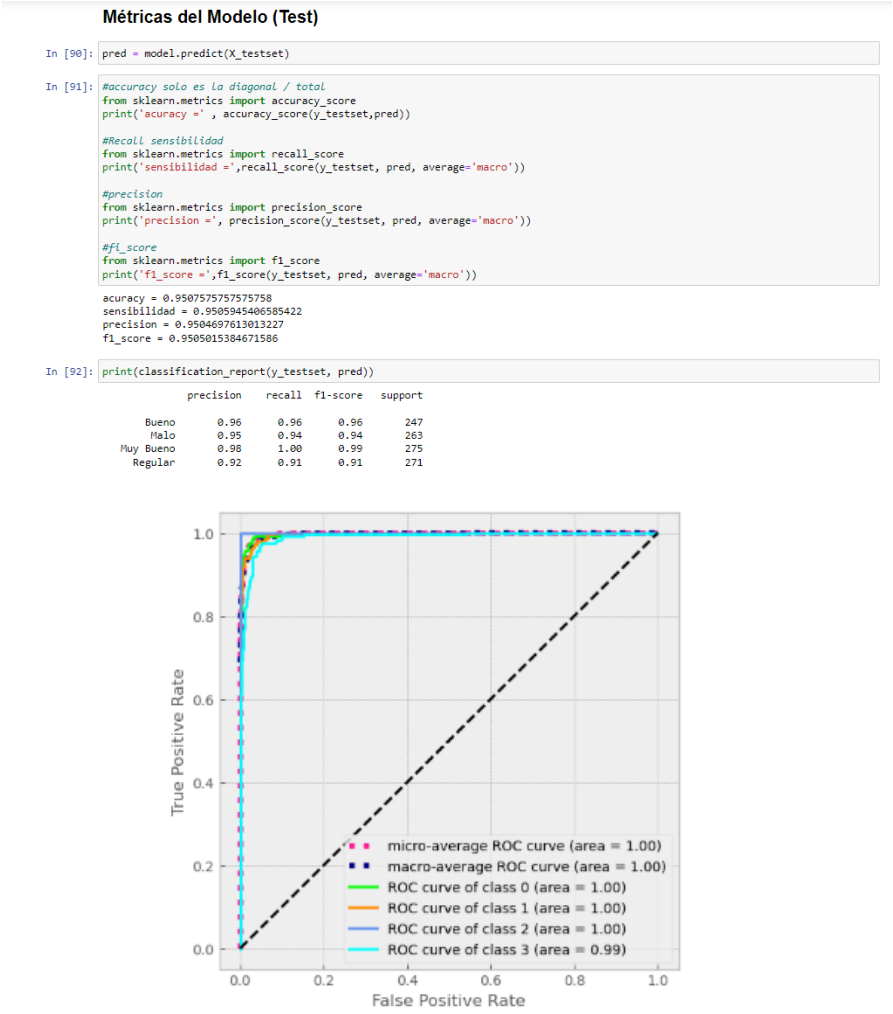


Figura 39 *Resultados de las métricas del modelo SVM (Test)*



1. Accuracy (95.07%): El modelo logró una precisión del 95.07%, lo que indica que más del 95% de las predicciones realizadas fueron correctas al clasificar el rendimiento académico de los estudiantes en las diferentes categorías. La universidad puede identificar a los estudiantes tendrán un bajo rendimiento académico. Esto permite implementar programas de apoyo como tutorías personalizadas, asesorías psicológicas o justes en los métodos de enseñanza. Así como también puede impactar en la optimización de recursos educativos, diseño de estrategias pedagógicas, evaluación y mejora de planes de estudio.

2. Sensibilidad (Recall) (95.05%): Con un valor de 95.05%, esta métrica refleja la capacidad del modelo para identificar correctamente a los estudiantes dentro de cada categoría de rendimiento académico, garantizando que los casos relevantes sean detectados con alta eficacia. Este nivel alto de sensibilidad impacta en la detección temprana y focalizada de los estudiantes asegurando que la decisiones y estrategias implementadas sean inclusivas y efectivas.

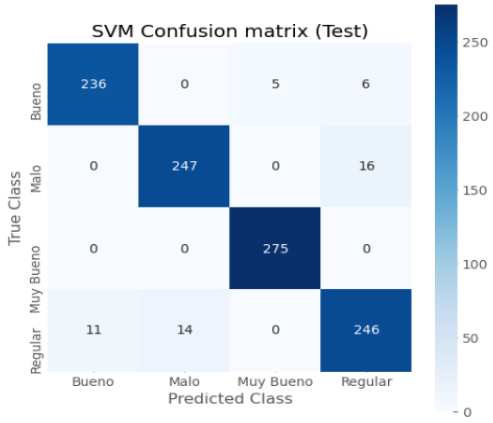
3. Precisión (Precisión) (95.04%): La precisión obtenida fue del 95.04%, lo que evidencia que el modelo realiza predicciones positivas con un alto grado de confianza, minimizando los falsos positivos en la clasificación del rendimiento estudiantil.

4. F1-Score (95.05%): El F1-Score fue de 95.05%, lo que demuestra que el modelo equilibra de manera efectiva la sensibilidad y la precisión, optimizando su desempeño incluso en situaciones donde las clases pueden estar desbalanceadas

5. Curva ROC y AUC (Área bajo la curva) (1.00): El modelo alcanzó un AUC de 1.00, lo que evidencia su capacidad sobresaliente para distinguir entre las diferentes categorías de rendimiento académico de los estudiantes. Este resultado sugiere un alto nivel de discriminación entre las clases.

Figura 40 *Matriz de Confusión del modelo SVM (Test)*

```
import pandas as pds
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pds

cnf_matrix= pds.crosstab(y_testset, pred) #, rownames = ['ACTUAL'],colnames=['Prediccion'])
sns.heatmap(cnf_matrix, square=True, annot=True, fmt='d', cbar=True, cmap='Blues')
plt.xlabel('Predicted Class')
plt.ylabel('True Class')
plt.title('SVM Confusion matrix (Test)')
plt.show()
```



La matriz muestra cómo el modelo clasifica las instancias entre las clases:

- **Diagonal principal (valores correctos):** Los números altos en la diagonal indican que el modelo clasifica correctamente la mayoría de las instancias de todas las clases:
  - 236 casos de la clase "Bueno" fueron clasificados correctamente.
  - 247 casos de la clase "Malo" fueron clasificados correctamente.
  - 275 casos de la clase "Muy Bueno" fueron clasificados correctamente.
  - 246 casos de la clase "Regular" fueron clasificados correctamente.
- **Errores de clasificación:**
  - Por ejemplo, 6 casos de la clase "Bueno" fueron clasificados como "Regular", y 14 de la clase "Regular" fueron clasificados como "Malo".

b) Redes neuronales

Figura 41 *Datos de los atributos de rendimiento académico (Redes neuronales)*

Mostrar Datos

```
In [2]: data
```

Out[2]:

	Id	appaterno	apmaterno	nombre	sexo	Departamento	Provincia	Distrito	Fecha_Nacimiento	Edad	...	anio_ingreso	ciclo_academico
0	1	ABAD	LINARES	CESAR ALEJANDRO	M	HUANUCO	LEONCIO PRADO	RUPA-RUPA	2001-11-13	20	...	2019	
1	2	ABAD	RIVERA	ANGIE BRIGITTE	F	HUANUCO	LEONCIO PRADO	RUPA-RUPA	1998-10-29	23	...	2018	
2	3	ABAD	RIVERA	JOIS SHIRLEY	F	HUANUCO	LEONCIO PRADO	RUPA-RUPA	2000-08-03	22	...	2017	
3	4	ABAL	NIETO	MARIA CELENIA	F	HUANUCO	HUANUCO	AMARILIS	2000-05-27	22	...	2019	
4	5	ABENDAÑO	MEZA	CESAR JHULINNIO	M	HUANUCO	LEONCIO PRADO	RUPA-RUPA	1997-11-13	24	...	2015	
...	...	...	...	...	...	...	...	...	...	...	...	...	
4579	4580	ZUÑIGA	TOLENTINO	JHORQUEENS YOHAN	M	HUANUCO	LEONCIO PRADO	RUPA-RUPA	2003-03-16	19	...	2013	
4580	4581	ZURITA	SANTA CRUZ	MONICA AMANDA	M	SAN MARTIN	LAMAS	ZAPATERO	2000-11-07	21	...	2020	
4581	4582	ZUTA	PAREDES	SEAN AKIRA	F	LORETO	UCAYALI	CONTAMANA	2004-06-05	18	...	2021	
4582	4583	ZUTA	PAREDES	SOON YI IXANKA	F	LORETO	UCAYALI	CONTAMANA	1996-07-11	26	...	2014	
4583	4584	ZUTA	PAREDES	ZYANKO KATIUSKA YVETTE	F	LORETO	UCAYALI	CONTAMANA	1994-01-10	28	...	2014	

4584 rows × 26 columns

Figura 42 *Atributos de rendimiento académico con columnas eliminadas*

Eliminar Columnas

```
In [5]: #se elimino 9 columnas quedadnod = 26 -9 = 17

        pq = data.drop(['Id','appaterno','apmaterno','nombre','Fecha_Nacimiento','codigo_alumno',
                        'beneficiario_pronabec', 'ciclo_academico','codcurricula'], axis=1)
```

```
In [6]: pq
```

Out[6]:

	sexo	Departamento	Provincia	Distrito	Edad	Estado_Civil	modalidad_ingreso	tipo_colegio_procedencia	escuela_profesinal	anio_ingreso	Total_C
0	M	HUANUCO	LEONCIO PRADO	RUPA-RUPA	20	SOLTERO	Centro Pre Universitario	Publico	INGENIERIA EN RECURSOS NATURALES RENOVABLES	2019	
1	F	HUANUCO	LEONCIO PRADO	RUPA-RUPA	23	SOLTERO	Centro Pre Universitario	Publico	CONTABILIDAD	2018	
2	F	HUANUCO	LEONCIO PRADO	RUPA-RUPA	22	SOLTERO	Exámen Ordinario	Publico	CONTABILIDAD	2017	
3	F	HUANUCO	HUANUCO	AMARILIS	22	SOLTERO	Exámen Ordinario	Publico	INGENIERIA EN INDUSTRIAS ALIMENTARIAS	2019	
4	M	HUANUCO	LEONCIO PRADO	RUPA-RUPA	24	SOLTERO	Centro Pre Universitario	Publico	AGRONOMIA	2015	
...	...	...	...	...	...	...	...	...	...	...	
4579	M	HUANUCO	LEONCIO PRADO	RUPA-RUPA	19	SOLTERO	Exonerados Primeros Puestos	Publico	INGENIERIA AMBIENTAL	2013	
4580	M	SAN MARTIN	LAMAS	ZAPATERO	21	SOLTERO	Convenios Especiales	Publico	ZOOTECNIA	2020	
4581	F	LORETO	UCAYALI	CONTAMANA	18	NaN	Exámen Ordinario	Publico	INGENIERIA EN RECURSOS	2021	

Figura 43 *Visualización de datos incompletos*

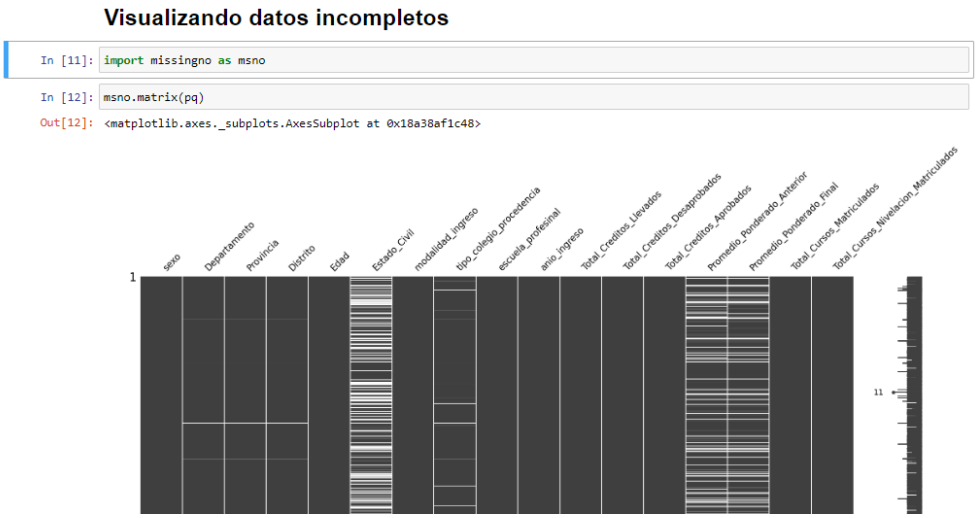


Figura 44 *Visualización de datos limpios*

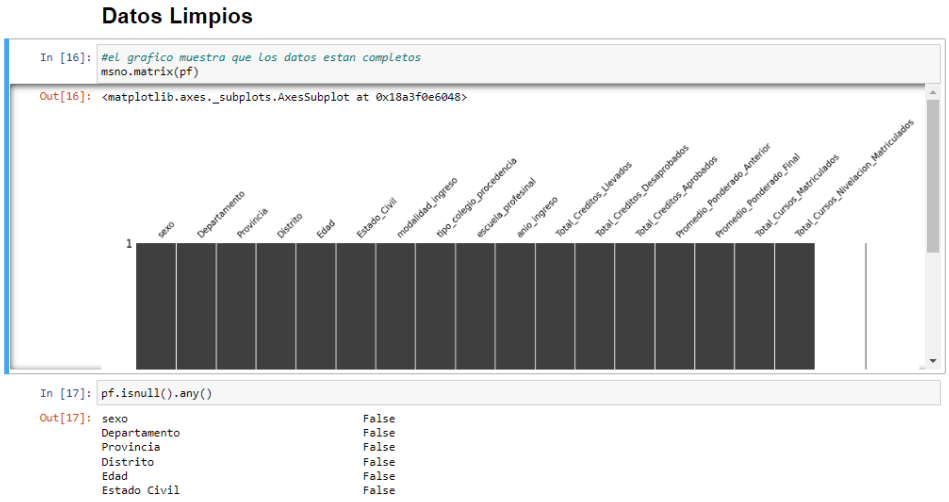


Figura 45 *Datos explorados (Cantidad de columnas)*

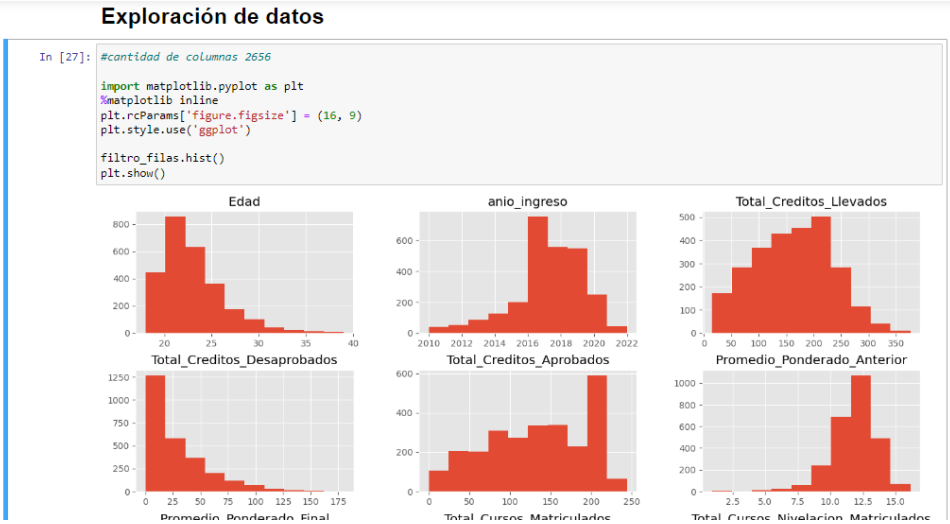


Figura 46 *Transformación de datos del promedio final*

Transformando datos de promedio final

```
In [30]: condicion_rango=[ ((pf['Promedio_Ponderado_Final'] <= 20) & (pf['Promedio_Ponderado_Final'] >= 18)),
                           ((pf['Promedio_Ponderado_Final'] < 18) & (pf['Promedio_Ponderado_Final'] >= 15)),
                           ((pf['Promedio_Ponderado_Final'] < 15) & (pf['Promedio_Ponderado_Final'] >= 13)),
                           ((pf['Promedio_Ponderado_Final'] < 13) & (pf['Promedio_Ponderado_Final'] >= 11)),
                           (pf['Promedio_Ponderado_Final'] < 11) ]
         escala_calificacion=['Excelente','Muy Bueno','Bueno','Regular','Malo']
         pf['CALIFICACION']=np.select(condicion_rango, escala_calificacion)

In [31]: condicion_rango=[ ((pf['Promedio_Ponderado_Final'] <= 20) & (pf['Promedio_Ponderado_Final'] >= 18)),
                           ((pf['Promedio_Ponderado_Final'] < 18) & (pf['Promedio_Ponderado_Final'] >= 15)),
                           ((pf['Promedio_Ponderado_Final'] < 15) & (pf['Promedio_Ponderado_Final'] >= 13)),
                           ((pf['Promedio_Ponderado_Final'] < 13) & (pf['Promedio_Ponderado_Final'] >= 11)),
                           (pf['Promedio_Ponderado_Final'] < 11) ]
         escala_calificacion=[4,3,2,1,0]
         pf['CALIFICACION_2']=np.select(condicion_rango, escala_calificacion)

In [32]: pf
```

Out[32]:

	sexo	Departamento	Provincia	Distrito	Edad	Estado_Civil	modalidad_ingreso	tipo_colegio_procedencia	escuela_profesinal	anio_ingreso	Total_C
0	M	HUANUCO	LEONCIO PRADO	RUPA-RUPA	20	SOLTERO	Centro Pre Universitario	Publico	INGENIERIA EN RECURSOS NATURALES RENOVABLES	2019	
1	F	HUANUCO	LEONCIO PRADO	RUPA-RUPA	23	SOLTERO	Centro Pre Universitario	Publico	CONTABILIDAD	2018	
2	F	HUANUCO	LEONCIO PRADO	RUPA-RUPA	22	SOLTERO	Exámen Ordinario	Publico	CONTABILIDAD	2017	
3	F	HUANUCO	HUANUCO	AMARILIS	22	SOLTERO	Exámen Ordinario	Publico	INGENIERIA EN INDUSTRIAS ALIMENTARIAS	2019	

Figura 47 *Transformación de datos del estado civil*

Transformando datos de estado civil

```
In [35]: # convertir estado civil
         condicion_estadocivil=[
             (pf['Estado_Civil'] == 'SOLTERO'),
             (pf['Estado_Civil'] == 'VIUDO'),
             (pf['Estado_Civil'] == 'CASADO'),
             (pf['Estado_Civil'] == 'DIVORCIADO')
         ]
         escala_estadocivil=['1','2','3','4']
         pf['ES_2']=np.select(condicion_estadocivil, escala_estadocivil)

In [36]: pf['ES_2']
```

Out[36]:

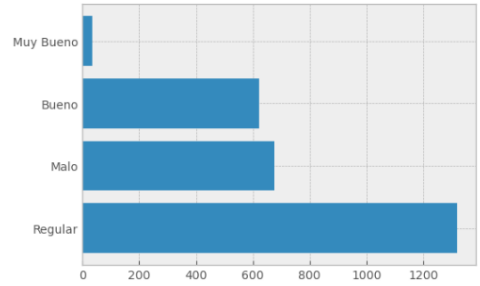
```
0       1
1       1
2       1
3       1
4       1
       ..
4578    1
4579    1
4580    1
4582    1
4583    1
Name: ES_2, Length: 2656, dtype: object
```

```
In [37]: condicion_TCP=[
             (pf['tipo_colegio_procedencia'] == 'Publico'),
             (pf['tipo_colegio_procedencia'] == 'Privado')
```

Figura 48 *Visualización de los datos por clases*

Mostramos datos por clases

```
In [45]: #sb.factorplot('CALIFICACION',data=data_datos,kind="count", aspect=3)
         #sb.figure(figsize=(8, 6))

         conteo = data_datos['CALIFICACION'].value_counts()
         plt.style.use("bmh")
         plt.barh(conteo.index,conteo)
         plt.show()
         #plt.close('all')
```




```
In [46]: data_datos
```

Figura 49 *Matriz de correlación de variables (Redes neuronales)*

Matriz de correlación de variables

```
In [54]: #correlacion de las variables

         import pandas as pds
         import seaborn as sns
         import matplotlib.pyplot as plt
         data_datos_2
         corr_df = data_datos_2.corr(method='pearson')

         plt.figure(figsize=(8, 6))
         sns.heatmap(corr_df, annot=True)
         plt.show()
```

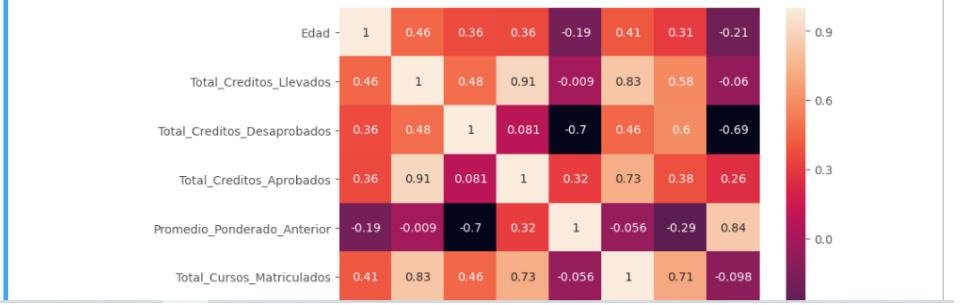


Figura 50 *Selección de variables predichas y predictoras*

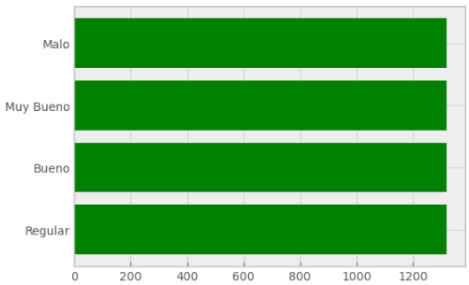
Seleccionando Variables Predichas y predictoras

```
In [56]: X = data_over[['GENERO','Edad','ES_2',
                       'TCP_2','Total_Creditos_Llevados',
                       'Total_Creditos_Desaprobados','Total_Creditos_Aprobados','Promedio_Ponderado_Anterior',
                       'Total_Cursos_Matriculados','Total_Cursos_Nivelacion_Matriculados']].values

In [57]: X.shape
Out[57]: (5280, 10)

         Type Markdown and LaTeX: α²

In [58]: Y = data_over['CALIFICACION_2'].values
         #ya = data_datos["CALIFICACION"]
         Y.shape
Out[58]: (5280,)

In [59]: #cantidad de alumnos por rendiento
         datos_balanceados = data_over['CALIFICACION'].value_counts()

In [60]: plt.rcParams['figure.figsize'] = (6, 4)
```

Figura 51 *Balanceo de datos*

Datos Balanceados

```
In [61]: conteo = datos_balanceados
         plt.style.use("bmh")
         plt.barh(conteo.index,conteo,color ='green')
         plt.show()
         #plt.close('all')
```




```
In [62]: from sklearn.metrics import roc_curve, auc
         from sklearn import datasets
         from sklearn.multiclass import OneVsRestClassifier
         from sklearn.svm import LinearSVC
```

Figura 52 *Modelo Redes neuronales*

Modelo Redes Neuronales

```
In [65]: import tensorflow as tf
         from tensorflow.keras.models import Sequential
         from tensorflow.keras.layers import Dense, Dropout

In [66]: y_trainset = tf.keras.utils.to_categorical(y_trainset, 4)
         y_testset = tf.keras.utils.to_categorical(y_testset, 4)

In [67]: model = Sequential()
         model.add(Dense(units=32, activation='relu', input_dim=10))
         model.add(Dense(units=16, activation='relu'))
         model.add(Dense(units=8, activation='relu'))
         model.add(Dropout(0.2))
         model.add(Dense(units=4, activation='softmax'))
         model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 32)                352
_________________________________________________________________
dense_1 (Dense)              (None, 16)                528
_________________________________________________________________
dense_2 (Dense)              (None, 8)                 136
_________________________________________________________________
dropout (Dropout)            (None, 8)                 0
_________________________________________________________________
dense_3 (Dense)              (None, 4)                 36
=================================================================
Total params: 1,052
Trainable params: 1,052
```

Figura 53 *Resultados de las métricas del modelo Redes neuronales (Train)*

Métricas del Modelo (Train)

```
In [70]: predictions = model.predict(X_trainset)

In [71]: y_pred = np.argmax(predictions, axis=1)
         y_real = np.argmax(y_trainset, axis=1)

In [72]: from sklearn.metrics import confusion_matrix
         from sklearn.metrics import classification_report

In [73]: #accuracy solo es la diagonal / total
         from sklearn.metrics import accuracy_score
         print('acuracy =' , accuracy_score(y_real, y_pred))

         #Recall sensibilidad
         from sklearn.metrics import recall_score
         print('sensibilidad =',recall_score(y_real, y_pred, average='macro'))

         #precision
         from sklearn.metrics import precision_score
         print('precision =', precision_score(y_real, y_pred, average='macro'))

         #f1_score
         from sklearn.metrics import f1_score
         print('f1_score =',f1_score(y_real, y_pred, average='macro'))
```

```
acuracy = 0.9285037878787878
sensibilidad = 0.9285627375595942
precision = 0.9287785132769073
f1_score = 0.9283379741939216
```

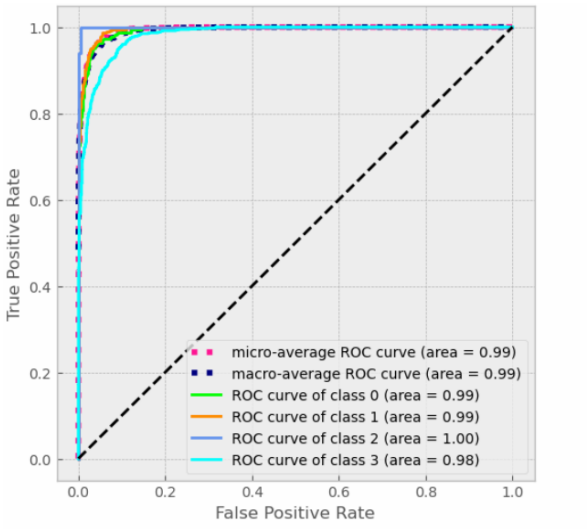


Figura 54 *Matriz de Confusión del modelo Redes neuronales (Train)*

Matriz de Confusion TRAIN

```
In [77]: import pandas as pds
         import seaborn as sns
         import matplotlib.pyplot as plt
         import pandas as pds

         cnf_matrix= pds.crosstab(y_real, y_pred) #, rownames = ['ACTUAL'],colnames=['Prediccion'])
         sns.heatmap(cnf_matrix, square=True, annot=True, fmt='d', cbar=True, cmap='Blues')
         plt.xlabel('Predicted Class')
         plt.ylabel('True Class')
         plt.title('Neural Network Confusion matrix (Train)')
         plt.show()
```

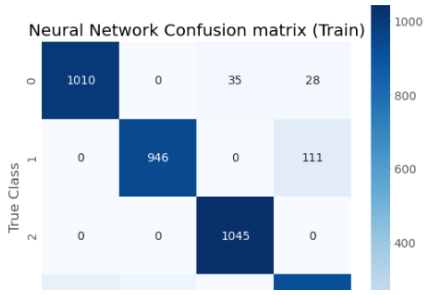
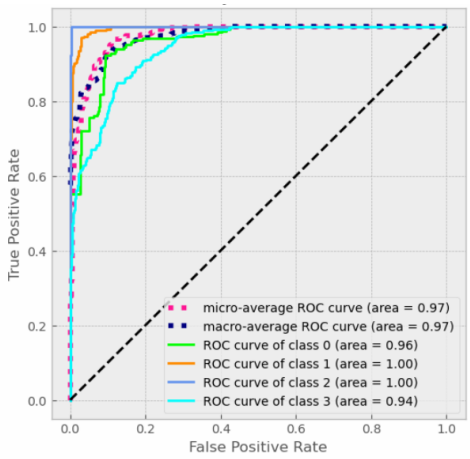


Figura 55 *Resultados de las métricas del modelo Redes neuronales (Test)*

Métricas del Modelo (Test)

```
In [87]: predictions = model.predict(X_testset)

In [88]: y_pred = np.argmax(predictions, axis=1)
         y_real = np.argmax(y_testset, axis=1)

In [89]: #accuracy solo es la diagonal / total
         from sklearn.metrics import accuracy_score
         print('acuracy =' , accuracy_score(y_real, y_pred))

         #Recall sensibilidad
         from sklearn.metrics import recall_score
         print('sensibilidad =',recall_score(y_real, y_pred, average='macro'))

         #precision
         from sklearn.metrics import precision_score
         print('precision =', precision_score(y_real, y_pred, average='macro'))

         #f1_score
         from sklearn.metrics import f1_score
         print('f1_score =',f1_score(y_real, y_pred, average='macro'))

         acuracy = 0.853219696969697
         sensibilidad = 0.8493822326993595
         precision = 0.8631613897119924
         f1_score = 0.8492611441644089

In [90]: print(classification_report(y_real, y_pred))
```



1. Accuracy (85.53%): El modelo logró una precisión del 85.30%, lo que indica que más del 85% de las predicciones realizadas fueron correctas al clasificar el rendimiento académico de los estudiantes en las diferentes categorías. La universidad puede identificar a los estudiantes tendrán un bajo rendimiento académico. Esto permite implementar programas de apoyo como tutorías personalizadas, asesorías psicológicas o justes en los métodos de enseñanza. Así como

también puede impactar en la optimización de recursos educativos, diseño de estrategias pedagógicas, evaluación y mejora de planes de estudio.

2. Sensibilidad (Recall) (84.93%): Con un valor de 84.93%, esta métrica refleja la capacidad del modelo para identificar correctamente a los estudiantes dentro de cada categoría de rendimiento académico, garantizando que los casos relevantes sean detectados con alta eficacia. Este nivel alto de sensibilidad impacta en la detección temprana y focalizada de los estudiantes asegurando que las decisiones y estrategias implementadas sean inclusivas y efectivas.

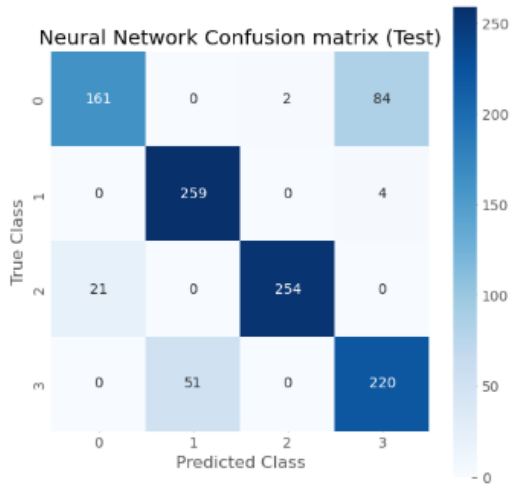
3. Precisión (Precisión) (86.31%): La precisión obtenida fue del **86.31%**, lo que evidencia que el modelo realiza predicciones positivas con un alto grado de confianza, minimizando los falsos positivos en la clasificación del rendimiento estudiantil.

4. F1-Score (84.92%): El F1-Score fue de 84.92%, lo que demuestra que el modelo equilibra de manera efectiva la sensibilidad y la precisión, optimizando su desempeño incluso en situaciones donde las clases pueden estar desbalanceadas

5. Curva ROC y AUC (Área bajo la curva) (0.97): El modelo alcanzó un AUC de 0.97, lo que evidencia su capacidad sobresaliente para distinguir entre las diferentes categorías de rendimiento académico de los estudiantes. Este resultado sugiere un alto nivel de discriminación entre las clases.

Figura 56 *Matriz de Confusión del modelo Redes neuronales (Test)*

```
import pandas as pds
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pds

cnf_matrix= pds.crosstab(y_real, y_pred) #, rownames = ['ACTUAL'],colnames=['Prediccion'])
sns.heatmap(cnf_matrix, square=True, annot=True, fmt='d', cbar=True, cmap='Blues')
plt.xlabel('Predicted Class')
plt.ylabel('True Class')
plt.title('Neural Network Confusion matrix (Test)')
plt.show()
```



Leyenda: Muy bueno =3, Bueno = 2, Regular = 1, Malo=0

La matriz muestra cómo el modelo clasifica las instancias entre las clases:

- **Diagonal principal (valores correctos):** Los números altos en la diagonal indican que el modelo clasifica correctamente la mayoría de las instancias de todas las clases:
  - 254 casos de la clase "Bueno" fueron clasificados correctamente.
  - 161 casos de la clase "Malo" fueron clasificados correctamente.
  - 220 casos de la clase "Muy Bueno" fueron clasificados correctamente.
  - 259 casos de la clase "Regular" fueron clasificados correctamente.
- **Errores de clasificación:**
  - Por ejemplo, 21 casos de la clase "Bueno" fueron clasificados como "Malo", y 4 de la clase "Regular" fueron clasificados como "Muy Bueno".

c) Árboles de decisión

Figura 57 *Datos de los atributos de rendimiento académico (Árbol de decisión)*

Mostrar Datos

```
In [2]: data
```

Out[2]:

	Id	appaterno	apmaterno	nombre	sexo	Departamento	Provincia	Distrito	Fecha_Nacimiento	Edad	...	anio_ingreso	ciclo_academico
0	1	ABAD	LINARES	CESAR ALEJANDRO	M	HUANUCO	LEONCIO PRADO	RUPA-RUPA	2001-11-13	20	...	2019	
1	2	ABAD	RIVERA	ANGIE BRIGITTE	F	HUANUCO	LEONCIO PRADO	RUPA-RUPA	1998-10-29	23	...	2018	
2	3	ABAD	RIVERA	JOIS SHIRLEY	F	HUANUCO	LEONCIO PRADO	RUPA-RUPA	2000-08-03	22	...	2017	
3	4	ABAL	NIETO	MARIA CELENIA	F	HUANUCO	HUANUCO	AMARILIS	2000-05-27	22	...	2019	
4	5	ABENDAÑO	MEZA	CESAR JHULINNIO	M	HUANUCO	LEONCIO PRADO	RUPA-RUPA	1997-11-13	24	...	2015	
...	...	...	...	...	...	...	...	...	...	...	...	...	...
4579	4580	ZUÑIGA	TOLENTINO	JHORQUEENS YOHAN	M	HUANUCO	LEONCIO PRADO	RUPA-RUPA	2003-03-16	19	...	2013	
4580	4581	ZURITA	SANTA CRUZ	MONICA AMANDA	M	SAN MARTIN	LAMAS	ZAPATERO	2000-11-07	21	...	2020	
4581	4582	ZUTA	PAREDES	SEAN AKIRA	F	LORETO	UCAYALI	CONTAMANA	2004-06-05	18	...	2021	
4582	4583	ZUTA	PAREDES	SOON YI IXANKA	F	LORETO	UCAYALI	CONTAMANA	1996-07-11	26	...	2014	
4583	4584	ZUTA	PAREDES	ZYANKO KATIUSKA YVETTE	F	LORETO	UCAYALI	CONTAMANA	1994-01-10	28	...	2014	

4584 rows × 26 columns

Figura 58 *Atributos de rendimiento académico con columnas eliminadas*

Eliminar Columnas

```
In [5]: #se elimino 9 columnas quedadnod = 26 -9 = 17

pq = data.drop(['Id','appaterno','apmaterno','nombre','Fecha_Nacimiento','codigo_alumno',
                'beneficiario_pronabec', 'ciclo_academico','codcurricula'], axis=1)
```

```
In [6]: pq
```

Out[6]:

	sexo	Departamento	Provincia	Distrito	Edad	Estado_Civil	modalidad_ingreso	tipo_colegio_procedencia	escuela_profesinal	anio_ingreso	Total_C
0	M	HUANUCO	LEONCIO PRADO	RUPA-RUPA	20	SOLTERO	Centro Pre Universitario	Publico	INGENIERIA EN RECURSOS NATURALES RENOVABLES	2019	
1	F	HUANUCO	LEONCIO PRADO	RUPA-RUPA	23	SOLTERO	Centro Pre Universitario	Publico	CONTABILIDAD	2018	
2	F	HUANUCO	LEONCIO PRADO	RUPA-RUPA	22	SOLTERO	Exámen Ordinario	Publico	CONTABILIDAD	2017	
3	F	HUANUCO	HUANUCO	AMARILIS	22	SOLTERO	Exámen Ordinario	Publico	INGENIERIA EN INDUSTRIAS ALIMENTARIAS	2019	
4	M	HUANUCO	LEONCIO PRADO	RUPA-RUPA	24	SOLTERO	Centro Pre Universitario	Publico	AGRONOMIA	2015	
...	...	...	...	...	...	...	...	...	...	...	...
4579	M	HUANUCO	LEONCIO PRADO	RUPA-RUPA	19	SOLTERO	Exonerados Primeros Puestos	Publico	INGENIERIA AMBIENTAL	2013	
4580	M	SAN MARTIN	LAMAS	ZAPATERO	21	SOLTERO	Convenios Especiales	Publico	ZOOTECNIA	2020	
4581	F	LORETO	UCAYALI	CONTAMANA	18	NaN	Exámen Ordinario	Publico	INGENIERIA EN RECURSOS	2021	

Figura 59 *Visualización de datos incompletos*

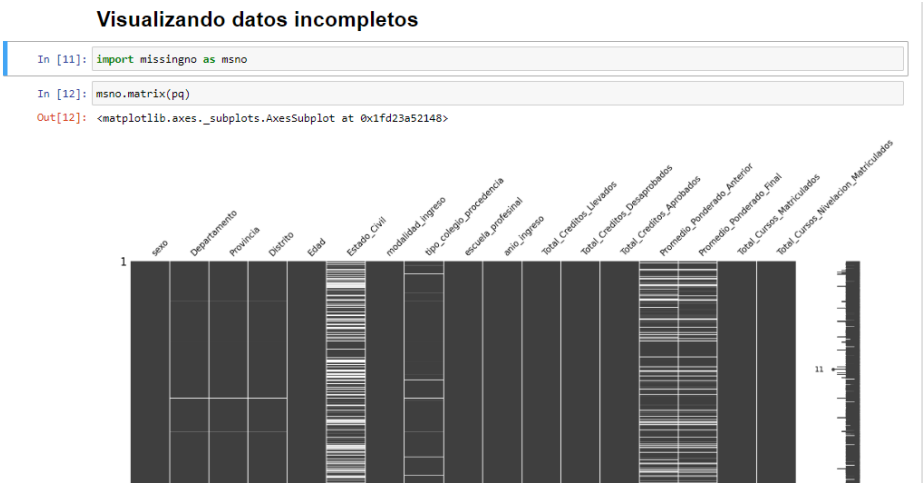


Figura 60 *Visualización de datos limpios*

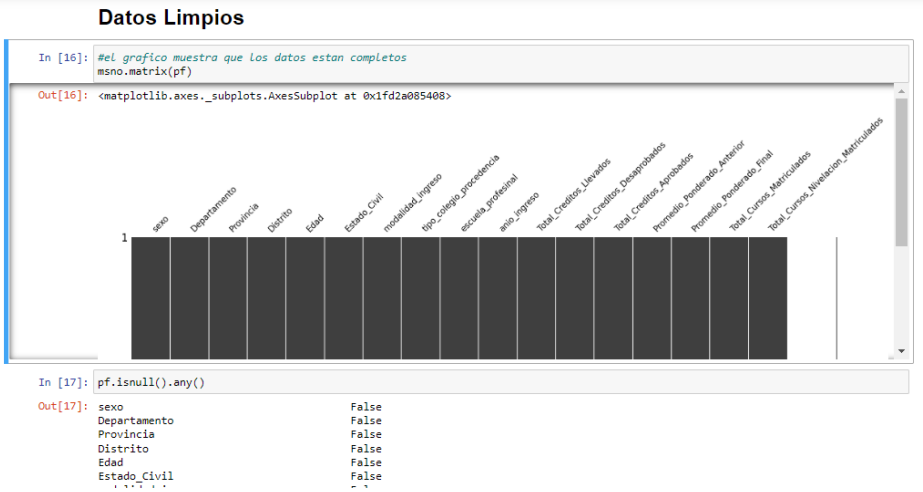


Figura 61 *Datos explorados (Cantidad de columnas)*

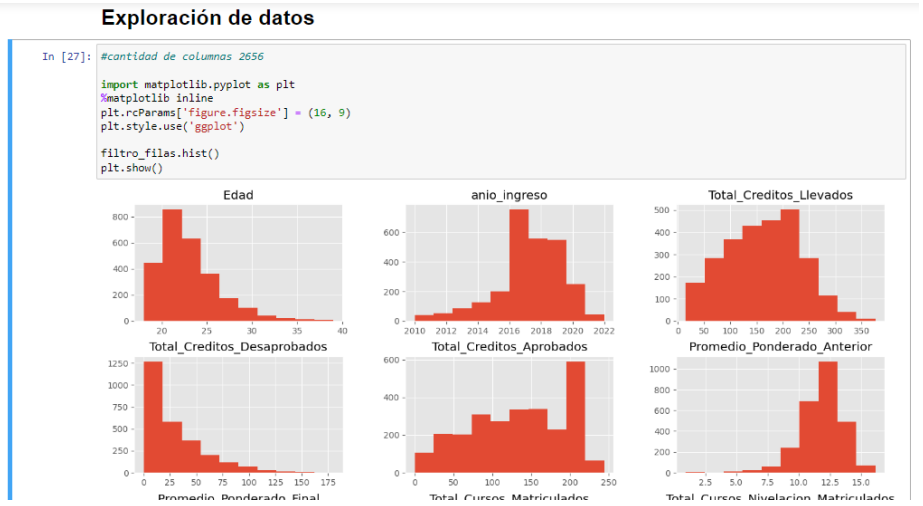


Figura 62 *Transformación de datos del promedio final*

Transformando datos de promedio final

```
In [30]: condicion_rango=[ ((pf['Promedio_Ponderado_Final'] <= 20) & (pf['Promedio_Ponderado_Final'] >= 18)),
                           ((pf['Promedio_Ponderado_Final'] < 18) & (pf['Promedio_Ponderado_Final'] >= 15)),
                           ((pf['Promedio_Ponderado_Final'] < 15) & (pf['Promedio_Ponderado_Final'] >= 13)),
                           ((pf['Promedio_Ponderado_Final'] < 13) & (pf['Promedio_Ponderado_Final'] >= 11)),
                           (pf['Promedio_Ponderado_Final'] < 11) ]
         escala_calificacion=['Excelente','Muy Bueno','Bueno','Regular','Malo']
         pf['CALIFICACION']=np.select(condicion_rango, escala_calificacion)

In [31]: condicion_rango=[ ((pf['Promedio_Ponderado_Final'] <= 20) & (pf['Promedio_Ponderado_Final'] >= 18)),
                           ((pf['Promedio_Ponderado_Final'] < 18) & (pf['Promedio_Ponderado_Final'] >= 15)),
                           ((pf['Promedio_Ponderado_Final'] < 15) & (pf['Promedio_Ponderado_Final'] >= 13)),
                           ((pf['Promedio_Ponderado_Final'] < 13) & (pf['Promedio_Ponderado_Final'] >= 11)),
                           (pf['Promedio_Ponderado_Final'] < 11) ]
         escala_calificacion=[5,4,3,2,1]
         pf['CALIFICACION_2']=np.select(condicion_rango, escala_calificacion)

In [32]: pf
```

Out[32]:

	sexo	Departamento	Provincia	Distrito	Edad	Estado_Civil	modalidad_ingreso	tipo_colegio_procedencia	escuela_profesinal	anio_ingreso	Total_C
0	M	HUANUCO	LEONCIO PRADO	RUPA-RUPA	20	SOLTERO	Centro Pre Universitario	Publico	INGENIERIA EN RECURSOS NATURALES RENOVABLES	2019	
1	F	HUANUCO	LEONCIO PRADO	RUPA-RUPA	23	SOLTERO	Centro Pre Universitario	Publico	CONTABILIDAD	2018	
2	F	HUANUCO	LEONCIO PRADO	RUPA-RUPA	22	SOLTERO	Exámen Ordinario	Publico	CONTABILIDAD	2017	
3	F	HUANUCO	HUANUCO	AMARILIS	22	SOLTERO	Exámen Ordinario	Publico	INGENIERIA EN INDUSTRIAS ALIMENTARIAS	2019	

Figura 63 *Transformación de datos del estado civil*

Transformando datos de estado civil

```
In [35]: # convertir estado civil
         condicion_estadocivil=[
             (pf['Estado_Civil'] == 'SOLTERO'),
             (pf['Estado_Civil'] == 'VIUDO'),
             (pf['Estado_Civil'] == 'CASADO'),
             (pf['Estado_Civil'] == 'DIVORCIADO')
         ]
         escala_estadocivil=['1','2','3','4']
         pf['ES_2']=np.select(condicion_estadocivil, escala_estadocivil)

In [36]: pf['ES_2']
```

Out[36]:

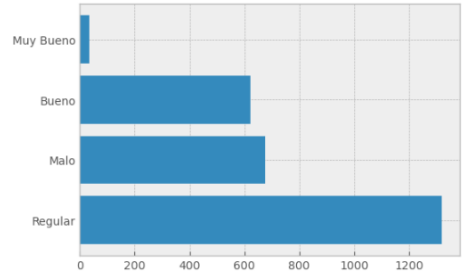
```
0       1
1       1
2       1
3       1
4       1
       ..
4578    1
4579    1
4580    1
4582    1
4583    1
Name: ES_2, Length: 2656, dtype: object
```

```
In [37]: condicion_TCP=[
             (pf['tipo_colegio_procedencia'] == 'Publico'),
             (pf['tipo_colegio_procedencia'] == 'Privado')
         ]
```

Figura 64 *Visualización de los datos por clases*

Mostramos datos por clases

```
In [45]: #sb.factorplot('CALIFICACION',data=data_datos,kind="count", aspect=3)
         #sb.figure(figsize=(8, 6))

         conteo = data_datos['CALIFICACION'].value_counts()
         plt.style.use("bmh")
         plt.barh(conteo.index,conteo)
         plt.show()
         #plt.close('all')
```




```
In [46]: data_datos
```

Figura 65 *Matriz de correlación de variables (Árbol de decisión)*

Matriz de correlación de variables

```
In [54]: #correlacion de las variables

         import pandas as pds
         import seaborn as sns
         import matplotlib.pyplot as plt
         data_datos_2
         corr_df = data_datos_2.corr(method='pearson')

         plt.figure(figsize=(8, 6))
         sns.heatmap(corr_df, annot=True)
         plt.show()
```

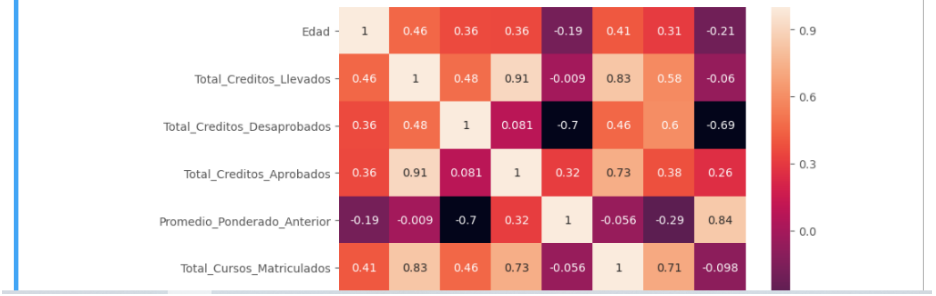


Figura 66 *Selección de variables predichas y predictoras*

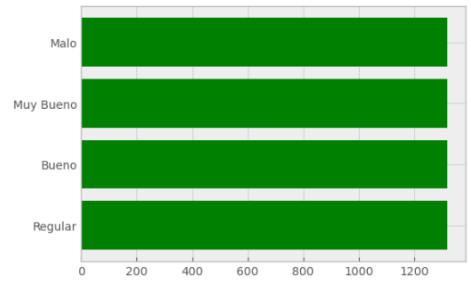
Seleccionando Variables Predichas y predictoras

```
In [56]: X = data_over[['GENERO','Edad','ES_2',
                       'TCP_2','Total_Creditos_Llevados',
                       'Total_Creditos_Desaprobados','Total_Creditos_Aprobados','Promedio_Ponderado_Anterior',
                       'Total_Cursos_Matriculados','Total_Cursos_Nivelacion_Matriculados']].values

In [57]: X.shape

Out[57]: (5280, 10)

         Type Markdown and LaTeX: α²

In [58]: Y = data_over['CALIFICACION']
         #ya = data_datos["CALIFICACION"]
         Y.shape

Out[58]: (5280,)

In [59]: #cantidad de alumnos por rendiento
         datos_balanceados = Y.value_counts()

In [60]: plt.rcParams['figure.figsize'] = (6, 4)
```

Figura 67 *Balanceo de datos*

Datos Balanceados

```
In [61]: conteo = datos_balanceados
         plt.style.use("bmh")
         plt.barh(conteo.index,conteo,color ='green')
         plt.show()
         #plt.close('all')
```




```
In [62]: from sklearn.metrics import roc_curve, auc
         from sklearn import datasets
         from sklearn.multiclass import OneVsRestClassifier
         from sklearn.svm import LinearSVC
```

Figura 68 *Modelo Árbol de decisión*

Modelo Árbol de Decisión

```
In [64]: from sklearn.model_selection import GridSearchCV

         # defining parameter range
         param_grid = {
             'max_depth': [2, 4, 5, None],
             'max_features': ['auto','sqrt', 'log2', None],
             'min_samples_leaf': [1, 5, 10],
             'criterion': ['gini', 'entropy'],
             'splitter': ['best', 'random']
         }

         grid = GridSearchCV(DecisionTreeClassifier(),
                             param_grid,
                             refit = True,
                             verbose = 3,
                             cv = 3)

         # fitting the model for grid search
         grid.fit(X_trainset, y_trainset)
```

```
Fitting 3 folds for each of 192 candidates, totalling 576 fits
[CV 1/3] END criterion=gini, max_depth=2, max_features=auto, min_samples_leaf=1, splitter=best;, score=0.729 total time=   0.
0s
[CV 2/3] END criterion=gini, max_depth=2, max_features=auto, min_samples_leaf=1, splitter=best;, score=0.732 total time=   0.
0s
[CV 3/3] END criterion=gini, max_depth=2, max_features=auto, min_samples_leaf=1, splitter=best;, score=0.719 total time=   0.
0s
[CV 1/3] END criterion=gini, max_depth=2, max_features=auto, min_samples_leaf=1, splitter=random;, score=0.305 total time=
0.0s
[CV 2/3] END criterion=gini, max_depth=2, max_features=auto, min_samples_leaf=1, splitter=random;, score=0.384 total time=
0.0s
[CV 3/3] END criterion=gini, max_depth=2, max_features=auto, min_samples_leaf=1, splitter=random;, score=0.419 total time=
```

Figura 69 *Resultados de las métricas del modelo Árbol de decisión (Train)*

Métricas del Modelo (Train)

```
In [69]: pred = model.predict(X_trainset)

In [70]: from sklearn.metrics import confusion_matrix
         from sklearn.metrics import classification_report

In [71]: #accuracy solo es la diagonal / total
         from sklearn.metrics import accuracy_score
         print('acuracy =' , accuracy_score(y_trainset,pred))

         #Recall sensibilidad
         from sklearn.metrics import recall_score
         print('sensibilidad =',recall_score(y_trainset, pred, average='macro'))

         #precision
         from sklearn.metrics import precision_score
         print('precision =', precision_score(y_trainset, pred, average='macro'))

         #f1_score
         from sklearn.metrics import f1_score
         print('f1_score =',f1_score(y_trainset, pred, average='macro'))
```

```
acuracy = 0.96875
sensibilidad = 0.9688374230740072
precision = 0.9688600348880709
f1_score = 0.9687980623371475
```

```
In [72]: print(classification_report(y_trainset, pred))
```

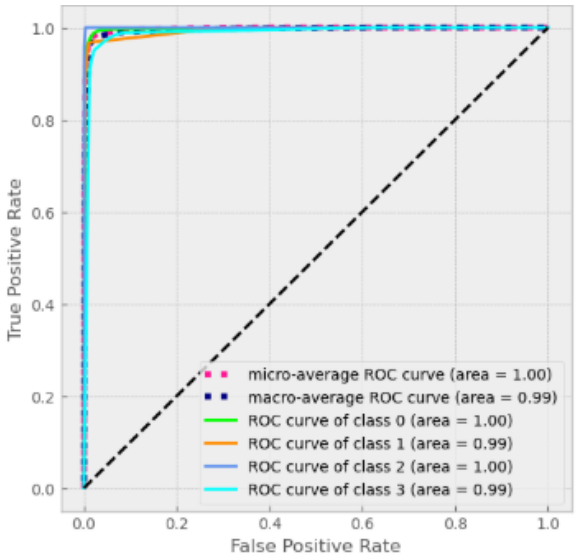


Figura 70 *Matriz de Confusión del modelo Árbol de decisión (Train)*

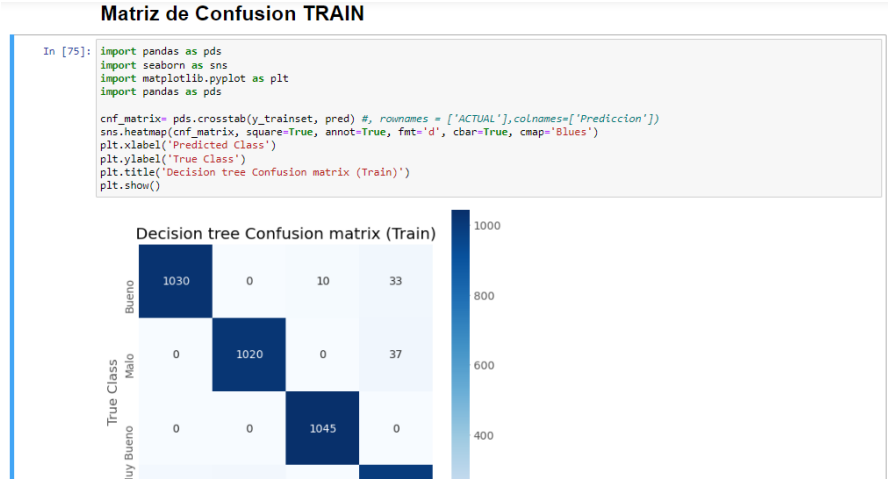
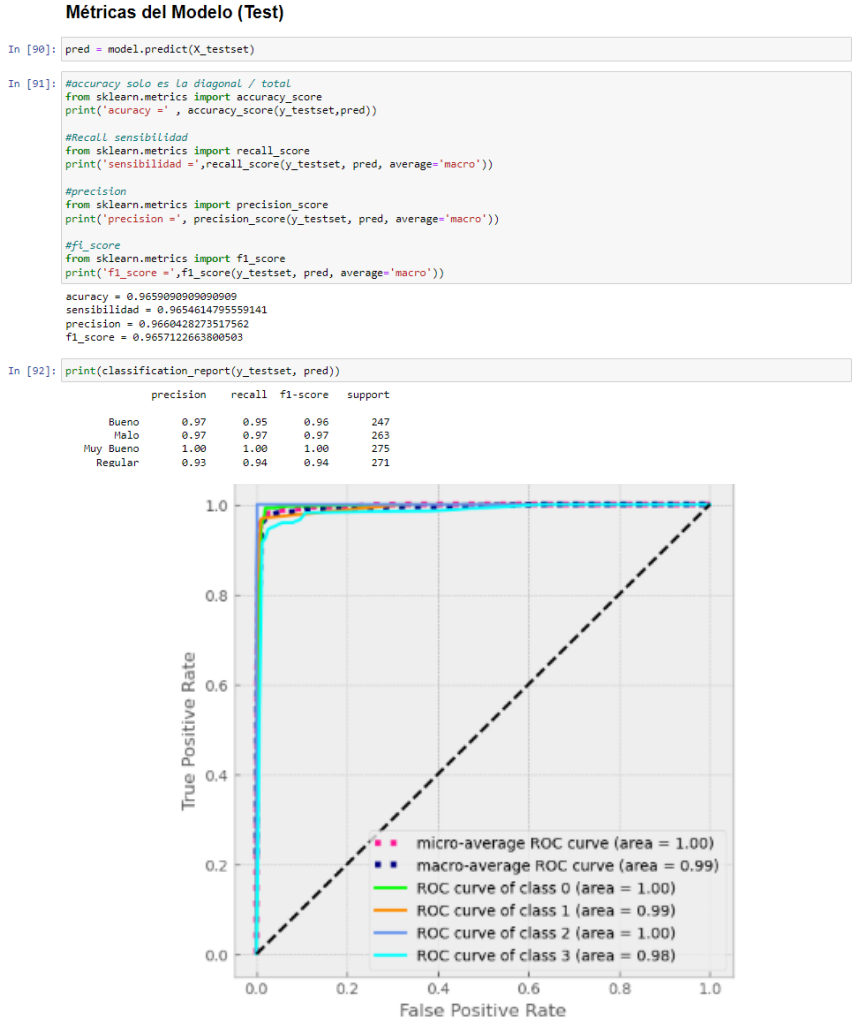


Figura 71 *Resultados de las métricas del modelo Árbol de decisión (Test)*



1. Accuracy (96.59%): El modelo logró una precisión del 96.59%, lo que indica que más del 96% de las predicciones realizadas fueron correctas al clasificar el rendimiento académico de los estudiantes en las diferentes categorías. La universidad puede identificar a los estudiantes tendrán un bajo rendimiento académico. Esto permite implementar programas de apoyo como

tutorías personalizadas, asesorías psicológicas o justes en los métodos de enseñanza. Así como también puede impactar en la optimización de recursos educativos, diseño de estrategias pedagógicas, evaluación y mejora de planes de estudio.

2. Sensibilidad (Recall) (96.54%): Con un valor de 96.54%, esta métrica refleja la capacidad del modelo para identificar correctamente a los estudiantes dentro de cada categoría de rendimiento académico, garantizando que los casos relevantes sean detectados con alta eficacia. Este nivel alto de sensibilidad impacta en la detección temprana y focalizada de los estudiantes asegurando que las decisiones y estrategias implementadas sean inclusivas y efectivas.

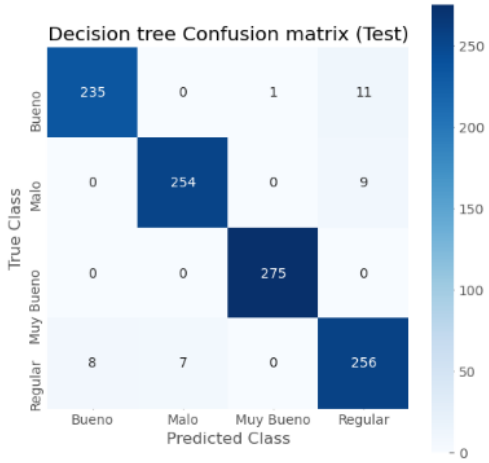
3. Precisión (Precisión) (96.60%): La precisión obtenida fue del **96.60%**, lo que evidencia que el modelo realiza predicciones positivas con un alto grado de confianza, minimizando los falsos positivos en la clasificación del rendimiento estudiantil.

4. F1-Score (96.57%): El F1-Score fue de 96.57%, lo que demuestra que el modelo equilibra de manera efectiva la sensibilidad y la precisión, optimizando su desempeño incluso en situaciones donde las clases pueden estar desbalanceadas

5. Curva ROC y AUC (Área bajo la curva) (0.99): El modelo alcanzó un AUC de 0.99, lo que evidencia su capacidad sobresaliente para distinguir entre las diferentes categorías de rendimiento académico de los estudiantes. Este resultado sugiere un alto nivel de discriminación entre las clases.

Figura 72 *Matriz de Confusión del modelo Árbol de decisión (Test)*

```
import pandas as pds
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pds

cnf_matrix= pds.crosstab(y_testset, pred) #, rownames = ['ACTUAL'],colnames=['Prediccion'])
sns.heatmap(cnf_matrix, square=True, annot=True, fmt='d', cbar=True, cmap='Blues')
plt.xlabel('Predicted Class')
plt.ylabel('True Class')
plt.title('Decision tree Confusion matrix (Test)')
plt.show()
```



La matriz muestra cómo el modelo clasifica las instancias entre las clases:

- **Diagonal principal (valores correctos):** Los números altos en la diagonal indican que el modelo clasifica correctamente la mayoría de las instancias de todas las clases:
  - 235 casos de la clase "Bueno" fueron clasificados correctamente.
  - 254 casos de la clase "Malo" fueron clasificados correctamente.
  - 275 casos de la clase "Muy Bueno" fueron clasificados correctamente.
  - 256 casos de la clase "Regular" fueron clasificados correctamente.
- **Errores de clasificación:**
  - Por ejemplo, 11 casos de la clase "Bueno" fueron clasificados como "Regular", y 7 de la clase "Regular" fueron clasificados como "Malo".

d) Redes Bayesianas

Figura 73 *Datos de los atributos de rendimiento académico (Redes bayesianas)*

Mostrar Datos

```
In [2]: data
```

Out[2]:

	Id	appaterno	apmaterno	nombre	sexo	Departamento	Provincia	Distrito	Fecha_Nacimiento	Edad	...	anio_ingreso	ciclo_academic
0	1	ABAD	LINARES	CESAR ALEJANDRO	M	HUANUCO	LEONCIO PRADO	RUPA-RUPA	2001-11-13	20	...	2019	
1	2	ABAD	RIVERA	ANGIE BRIGITTE	F	HUANUCO	LEONCIO PRADO	RUPA-RUPA	1998-10-29	23	...	2018	
2	3	ABAD	RIVERA	JOIS SHIRLEY	F	HUANUCO	LEONCIO PRADO	RUPA-RUPA	2000-08-03	22	...	2017	
3	4	ABAL	NIETO	MARIA CELENIA	F	HUANUCO	HUANUCO	AMARILIS	2000-05-27	22	...	2019	
4	5	ABENDAÑO	MEZA	CESAR JHULINNIO	M	HUANUCO	LEONCIO PRADO	RUPA-RUPA	1997-11-13	24	...	2015	
...	...	...	...	...	...	...	...	...	...	...	...	...	
4579	4580	ZUÑIGA	TOLENTINO	JHORQUEENS YOHAN	M	HUANUCO	LEONCIO PRADO	RUPA-RUPA	2003-03-16	19	...	2013	
4580	4581	ZURITA	SANTA CRUZ	MONICA AMANDA	M	SAN MARTIN	LAMAS	ZAPATERO	2000-11-07	21	...	2020	
4581	4582	ZUTA	PAREDES	SEAN AKIRA	F	LORETO	UCAYALI	CONTAMANA	2004-06-05	18	...	2021	
4582	4583	ZUTA	PAREDES	SOON YI IXANKA	F	LORETO	UCAYALI	CONTAMANA	1996-07-11	26	...	2014	
4583	4584	ZUTA	PAREDES	ZYANKO KATIUSKA YVETTE	F	LORETO	UCAYALI	CONTAMANA	1994-01-10	28	...	2014	

4584 rows × 26 columns

Figura 74 *Atributos de rendimiento académico con columnas eliminadas*

Eliminar Columnas

```
In [5]: #se elimino 9 columnas quedadnod = 26 -9 = 17

        pq = data.drop(['Id','appaterno','apmaterno','nombre','Fecha_Nacimiento','codigo_alumno',
                        'beneficiario_pronabec', 'ciclo_academico','codcurricula'], axis=1)
```

```
In [6]: pq
```

Out[6]:

	sexo	Departamento	Provincia	Distrito	Edad	Estado_Civil	modalidad_ingreso	tipo_colegio_procedencia	escuela_profesinal	anio_ingreso	Total_C
0	M	HUANUCO	LEONCIO PRADO	RUPA-RUPA	20	SOLTERO	Centro Pre Universitario	Publico	INGENIERIA EN RECURSOS NATURALES RENOVABLES	2019	
1	F	HUANUCO	LEONCIO PRADO	RUPA-RUPA	23	SOLTERO	Centro Pre Universitario	Publico	CONTABILIDAD	2018	
2	F	HUANUCO	LEONCIO PRADO	RUPA-RUPA	22	SOLTERO	Exámen Ordinario	Publico	CONTABILIDAD	2017	
3	F	HUANUCO	HUANUCO	AMARILIS	22	SOLTERO	Exámen Ordinario	Publico	INGENIERIA EN INDUSTRIAS ALIMENTARIAS	2019	
4	M	HUANUCO	LEONCIO PRADO	RUPA-RUPA	24	SOLTERO	Centro Pre Universitario	Publico	AGRONOMIA	2015	
...	...	...	...	...	...	...	...	...	...	...	
4579	M	HUANUCO	LEONCIO PRADO	RUPA-RUPA	19	SOLTERO	Exonerados Primeros Puestos	Publico	INGENIERIA AMBIENTAL	2013	
4580	M	SAN MARTIN	LAMAS	ZAPATERO	21	SOLTERO	Convenios Especiales	Publico	ZOOTECNIA	2020	
4581	F	LORETO	UCAYALI	CONTAMANA	18	NaN	Exámen Ordinario	Publico	INGENIERIA EN RECURSOS NATURALES	2021	

Figura 75 *Visualización de datos incompletos*

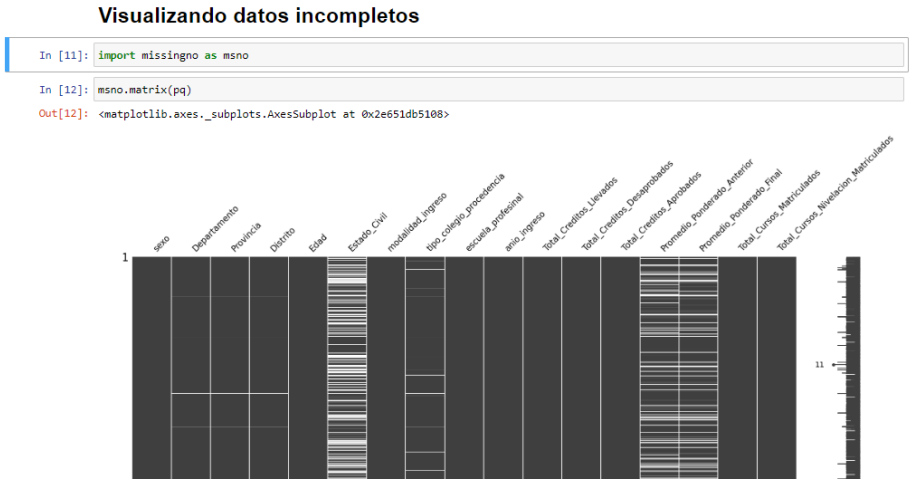


Figura 76 *Visualización de datos limpios*

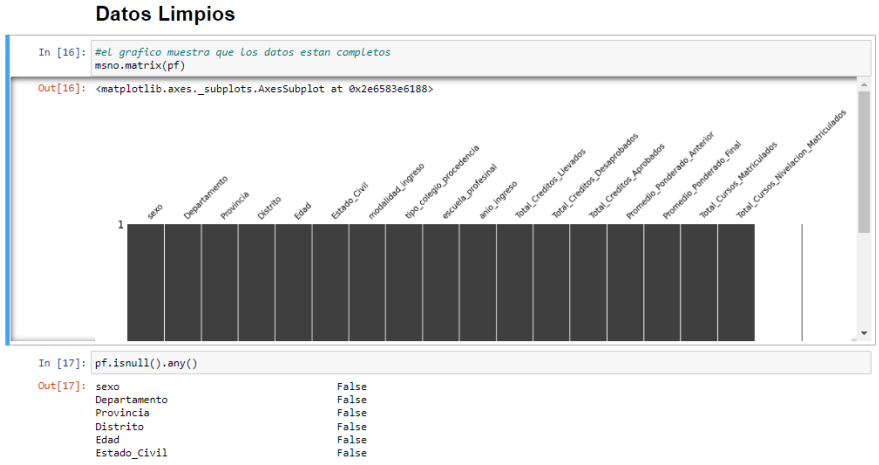


Figura 77 *Datos explorados (Cantidad de columnas)*

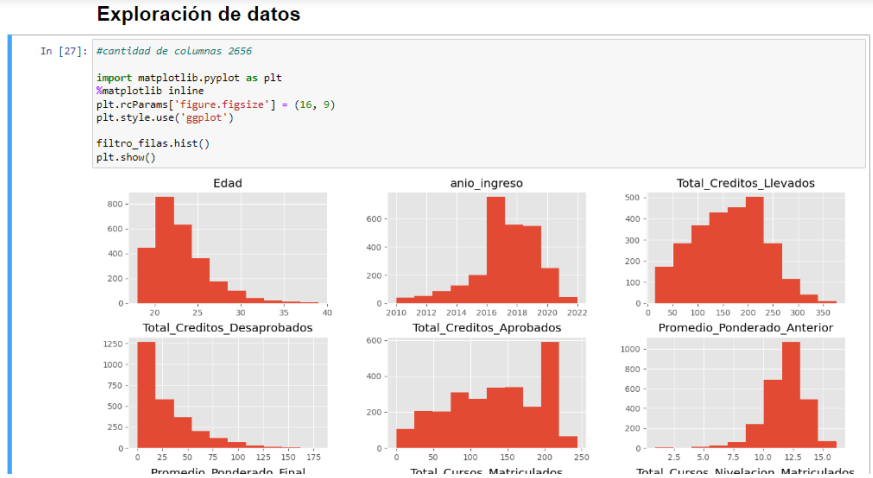


Figura 78 *Transformación de datos del promedio final*

Transformando datos de promedio final

```
In [30]: condicion_rango=[ ((pf['Promedio_Ponderado_Final'] <= 20) & (pf['Promedio_Ponderado_Final'] >= 18)),
                           ((pf['Promedio_Ponderado_Final'] < 18) & (pf['Promedio_Ponderado_Final'] >= 15)),
                           ((pf['Promedio_Ponderado_Final'] < 15) & (pf['Promedio_Ponderado_Final'] >= 13)),
                           ((pf['Promedio_Ponderado_Final'] < 13) & (pf['Promedio_Ponderado_Final'] >= 11)),
                           (pf['Promedio_Ponderado_Final'] < 11) ]
         escala_calificacion=['Excelente','Muy Bueno','Bueno','Regular','Malo']
         pf['CALIFICACION']=np.select(condicion_rango, escala_calificacion)

In [31]: condicion_rango=[ ((pf['Promedio_Ponderado_Final'] <= 20) & (pf['Promedio_Ponderado_Final'] >= 18)),
                           ((pf['Promedio_Ponderado_Final'] < 18) & (pf['Promedio_Ponderado_Final'] >= 15)),
                           ((pf['Promedio_Ponderado_Final'] < 15) & (pf['Promedio_Ponderado_Final'] >= 13)),
                           ((pf['Promedio_Ponderado_Final'] < 13) & (pf['Promedio_Ponderado_Final'] >= 11)),
                           (pf['Promedio_Ponderado_Final'] < 11) ]
         escala_calificacion=[5,4,3,2,1]
         pf['CALIFICACION_2']=np.select(condicion_rango, escala_calificacion)

In [32]: pf
```

Out[32]:

	sexo	Departamento	Provincia	Distrito	Edad	Estado_Civil	modalidad_ingreso	tipo_colegio_procedencia	escuela_profesinal	anio_ingreso	Total_C
0	M	HUANUCO	LEONCIO PRADO	RUPA-RUPA	20	SOLTERO	Centro Pre Universitario	Publico	INGENIERIA EN RECURSOS NATURALES RENOVABLES	2019	
1	F	HUANUCO	LEONCIO PRADO	RUPA-RUPA	23	SOLTERO	Centro Pre Universitario	Publico	CONTABILIDAD	2018	
2	F	HUANUCO	LEONCIO PRADO	RUPA-RUPA	22	SOLTERO	Exámen Ordinario	Publico	CONTABILIDAD	2017	
3	F	HUANUCO	HUANUCO	AMARILIS	22	SOLTERO	Exámen Ordinario	Publico	INGENIERIA EN INDUSTRIAS ALIMENTARIAS	2019	

Figura 79 *Transformación de datos del estado civil*

Transformando datos de estado civil

```
In [35]: # convertir estado civil
         condicion_estadocivil=[
             (pf['Estado_Civil'] == 'SOLTERO'),
             (pf['Estado_Civil'] == 'VIUDO'),
             (pf['Estado_Civil'] == 'CASADO'),
             (pf['Estado_Civil'] == 'DIVORCIADO')
         ]
         escala_estadocivil=['1','2','3','4']
         pf['ES_2']=np.select(condicion_estadocivil, escala_estadocivil)

In [36]: pf['ES_2']
```

Out[36]:

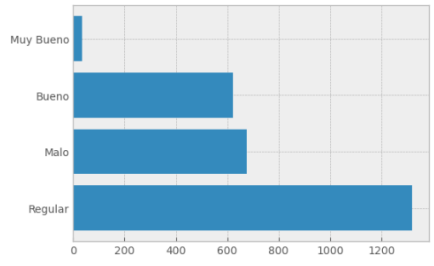
```
0       1
1       1
2       1
3       1
4       1
       ..
4578    1
4579    1
4580    1
4582    1
4583    1
Name: ES_2, Length: 2656, dtype: object
```

```
In [37]: condicion_TCP=[
             (pf['tipo_colegio_procedencia'] == 'Publico'),
             (pf['tipo_colegio_procedencia'] == 'Privado')
         ]
```

Figura 80 *Visualización de los datos por clases*

Mostramos datos por clases

```
In [45]: #sb.factorplot('CALIFICACION',data=data_datos,kind="count", aspect=3)
         #sb.figure(figsize=(8, 6))

         conteo = data_datos['CALIFICACION'].value_counts()
         plt.style.use("bmh")
         plt.barh(conteo.index,conteo)
         plt.show()
         #plt.close('all')
```




```
In [46]: data_datos
```

Figura 81 *Matriz de correlación de variables (Árbol de decisión)*

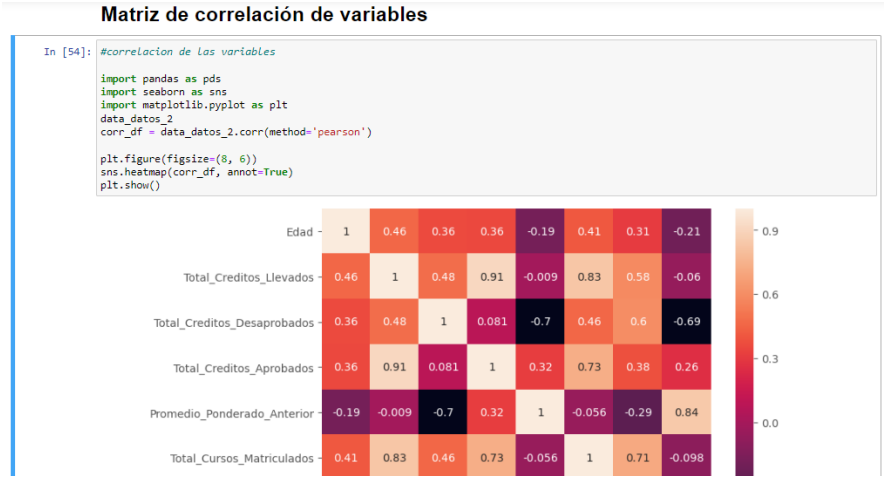


Figura 82 *Selección de variables predichas y predictoras*

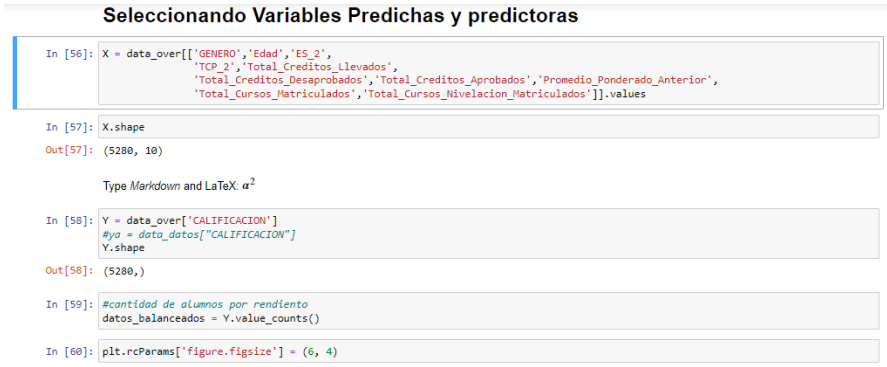


Figura 83 *Balanceo de datos*

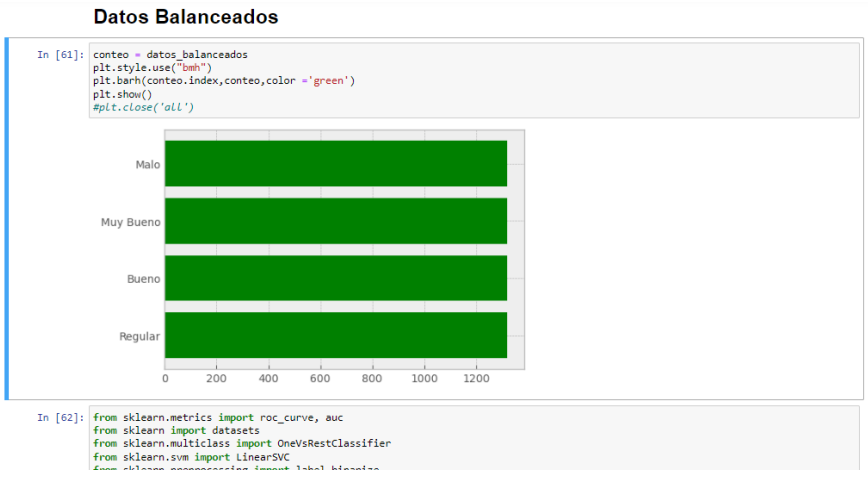


Figura 84 *Modelo Redes bayesianas*

Modelo Redes Bayesianas

```
In [64]: from sklearn.model_selection import GridSearchCV
         from sklearn.naive_bayes import GaussianNB

         # defining parameter range
         param_grid = {
             'var_smoothing': np.logspace(0,-9, num=100)
         }

         grid = GridSearchCV(GaussianNB(),
                             param_grid,
                             refit = True,
                             verbose = 3,
                             cv = 5)

         # fitting the model for grid search
         grid.fit(X_trainset, y_trainset)
```

```
Fitting 5 folds for each of 100 candidates, totalling 500 fits
[CV 1/5] END .................var_smoothing=1.0;, score=0.421 total time=   0.0s
[CV 2/5] END .................var_smoothing=1.0;, score=0.404 total time=   0.0s
[CV 3/5] END .................var_smoothing=1.0;, score=0.417 total time=   0.0s
[CV 4/5] END .................var_smoothing=1.0;, score=0.415 total time=   0.0s
[CV 5/5] END .................var_smoothing=1.0;, score=0.423 total time=   0.0s
[CV 1/5] END ..var_smoothing=0.8111308307896871;, score=0.428 total time=   0.0s
[CV 2/5] END ..var_smoothing=0.8111308307896871;, score=0.417 total time=   0.0s
[CV 3/5] END ..var_smoothing=0.8111308307896871;, score=0.421 total time=   0.0s
[CV 4/5] END ..var_smoothing=0.8111308307896871;, score=0.430 total time=   0.0s
[CV 5/5] END ..var_smoothing=0.8111308307896871;, score=0.431 total time=   0.0s
[CV 1/5] END ...var_smoothing=0.657933224657568;, score=0.445 total time=   0.0s
[CV 2/5] END ...var_smoothing=0.657933224657568;, score=0.437 total time=   0.0s
[CV 3/5] END ...var_smoothing=0.657933224657568;, score=0.428 total time=   0.0s
[CV 4/5] END ...var_smoothing=0.657933224657568;, score=0.440 total time=   0.0s
```

Figura 85 *Resultados de las métricas del modelo Redes bayesianas (Train)*

Métricas del Modelo (Train)

```
In [69]: pred = model.predict(X_trainset)
```

```
In [70]: from sklearn.metrics import confusion_matrix
         from sklearn.metrics import classification_report
```

```
In [71]: #accuracy solo es la diagonal / total
         from sklearn.metrics import accuracy_score
         print('acuracy =' , accuracy_score(y_trainset,pred))

         #Recall sensibilidad
         from sklearn.metrics import recall_score
         print('sensibilidad =',recall_score(y_trainset, pred, average='macro'))

         #precision
         from sklearn.metrics import precision_score
         print('precision =', precision_score(y_trainset, pred, average='macro'))

         #f1_score
         from sklearn.metrics import f1_score
         print('f1_score =',f1_score(y_trainset, pred, average='macro'))
```

```
acuracy = 0.8629261363636364
sensibilidad = 0.862838145274851
precision = 0.8651443899612606
f1_score = 0.8623420723497918
```

```
In [72]: print(classification_report(y_trainset, pred))
```

```
              precision    recall  f1-score   support

Bueno             0.78      0.88      0.83      1073
```

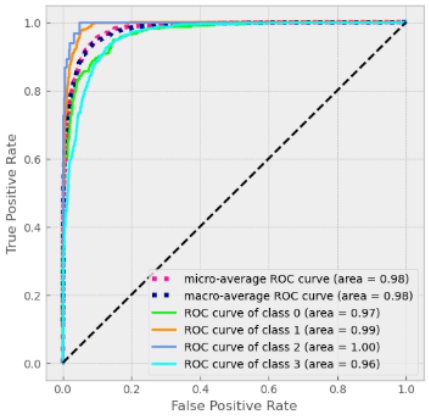


Figura 86 *Matriz de Confusión del modelo Redes bayesianas (Train)*

Matriz de Confusion TRAIN

```
In [75]: import pandas as pds
         import seaborn as sns
         import matplotlib.pyplot as plt
         import pandas as pds

         cnf_matrix= pds.crosstab(y_trainset, pred) #, rownames = ['ACTUAL'],colnames=['Prediccion'])
         sns.heatmap(cnf_matrix, square=True, annot=True, fmt='d', cbar=True, cmap='Blues')
         plt.xlabel('Predicted Class')
         plt.ylabel('True Class')
         plt.title('Bayessian Confusion matrix (Train)')
         plt.show()
```

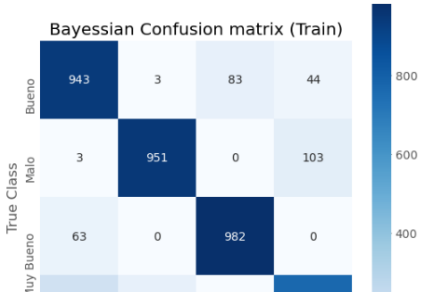
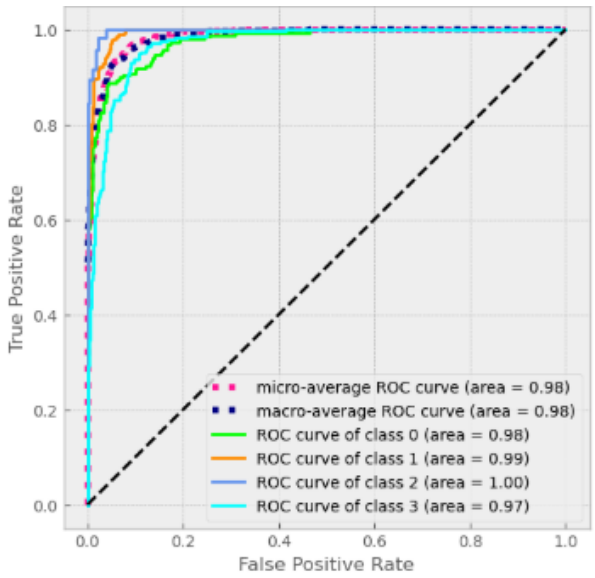


Figura 87 *Resultados de las métricas del modelo Redes bayesianas (Test)*

Métricas del Modelo (Test)

```
In [90]: pred = model.predict(X_testset)

In [91]: #accuracy solo es la diagonal / total
         from sklearn.metrics import accuracy_score
         print('acuracy =' , accuracy_score(y_testset,pred))

         #Recall sensibilidad
         from sklearn.metrics import recall_score
         print('sensibilidad =',recall_score(y_testset, pred, average='macro'))

         #precision
         from sklearn.metrics import precision_score
         print('precision =', precision_score(y_testset, pred, average='macro'))

         #f1_score
         from sklearn.metrics import f1_score
         print('f1_score =',f1_score(y_testset, pred, average='macro'))

acuracy = 0.8873106060606061
sensibilidad = 0.8873448912843693
precision = 0.8869997323253627
f1_score = 0.88612124774788

In [92]: print(classification_report(y_testset, pred))

              precision    recall  f1-score   support

       Bueno       0.81      0.89      0.85       247
       Malo        0.93      0.92      0.92       263
  Muy Bueno        0.94      0.96      0.95       275
    Regular        0.87      0.79      0.83       271
```



1. Accuracy (88.73%): El modelo logró una precisión del 88.73%, lo que indica que más del 88% de las predicciones realizadas fueron correctas al clasificar el rendimiento académico de los estudiantes en las diferentes categorías. La universidad puede identificar a los estudiantes tendrán un bajo rendimiento académico. Esto permite implementar programas de apoyo como tutorías personalizadas, asesorías psicológicas o justes en los métodos de enseñanza. Así como también puede impactar en la optimización de recursos educativos, diseño de estrategias pedagógicas, evaluación y mejora de planes de estudio.

2. Sensibilidad (Recall) 88.73%): Con un valor de 88.73%, esta métrica refleja la capacidad del modelo para identificar correctamente a los estudiantes dentro de cada categoría de rendimiento académico, garantizando que los casos relevantes sean detectados con alta eficacia. Este nivel alto de sensibilidad impacta en la detección temprana y focalizada de los estudiantes asegurando que las decisiones y estrategias implementadas sean inclusivas y efectivas.

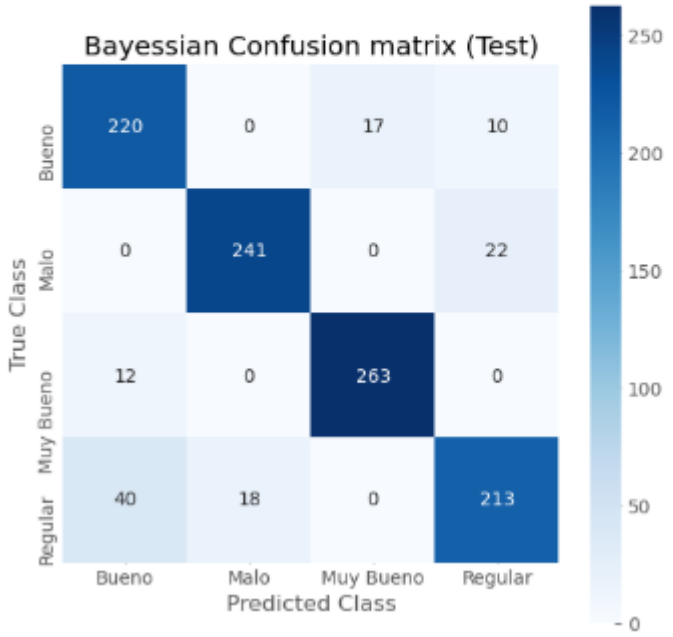
3. Precisión (Precisión) (88.69%): La precisión obtenida fue del **88.69%**, lo que evidencia que el modelo realiza predicciones positivas con un alto grado de confianza, minimizando los falsos positivos en la clasificación del rendimiento estudiantil.

4. F1-Score (88.61%): El F1-Score fue de 88.61%, lo que demuestra que el modelo equilibra de manera efectiva la sensibilidad y la precisión, optimizando su desempeño incluso en situaciones donde las clases pueden estar desbalanceadas

5. Curva ROC y AUC (Área bajo la curva) (0.98): El modelo alcanzó un AUC de 0.98, lo que evidencia su capacidad sobresaliente para distinguir entre las diferentes categorías de rendimiento académico de los estudiantes. Este resultado sugiere un alto nivel de discriminación entre las clases.

Figura 88 *Matriz de Confusión del modelo Redes bayesianas (Test)*

```
import pandas as pds
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pds

cnf_matrix= pds.crosstab(y_testset, pred) #, rownames = ['ACTUAL'],colnames=['Prediccion'])
sns.heatmap(cnf_matrix, square=True, annot=True, fmt='d', cbar=True, cmap='Blues')
plt.xlabel('Predicted Class')
plt.ylabel('True Class')
plt.title('Bayessian Confusion matrix (Test)')
plt.show()
```



La matriz muestra cómo el modelo clasifica las instancias entre las clases:

- **Diagonal principal (valores correctos):** Los números altos en la diagonal indican que el modelo clasifica correctamente la mayoría de las instancias de todas las clases:
    - 220 casos de la clase "Bueno" fueron clasificados correctamente.
    - 241 casos de la clase "Malo" fueron clasificados correctamente.
    - 263 casos de la clase "Muy Bueno" fueron clasificados correctamente.
    - 213 casos de la clase "Regular" fueron clasificados correctamente.
- **Errores de clasificación:**

Por ejemplo, 17 casos de la clase "Bueno" fueron clasificados como “Muy bueno” asi como 18 casos de la clase “regular” fueron clasificados como “Malo”.

e) Vecinos Cercanos (KNN)

Figura 89 *Datos de los atributos de rendimiento académico (KNN)*

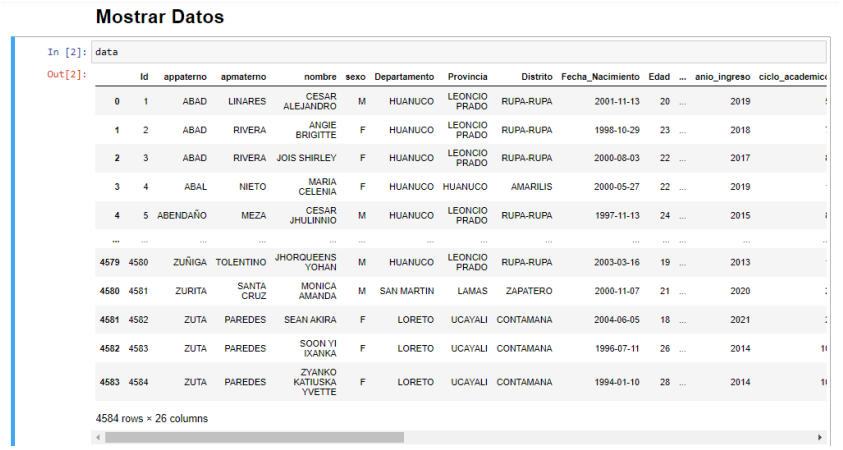


Figura 90 *Atributos de rendimiento académico con columnas eliminadas*

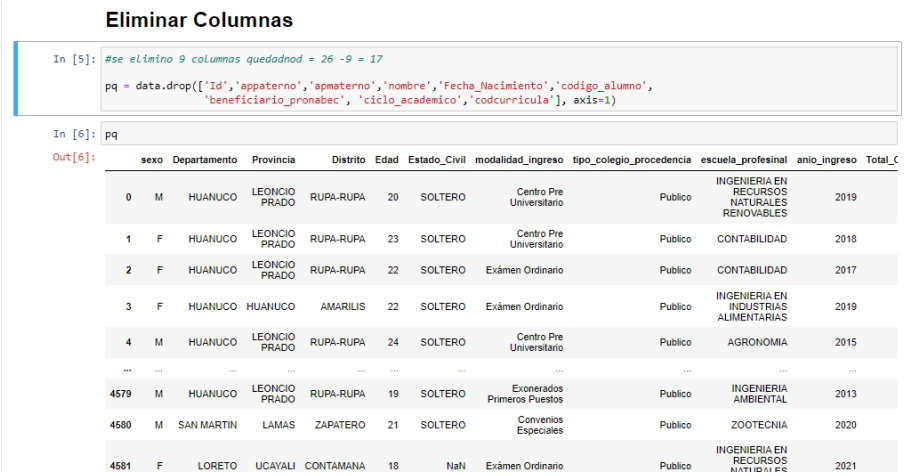


Figura 91 *Visualización de datos incompletos*

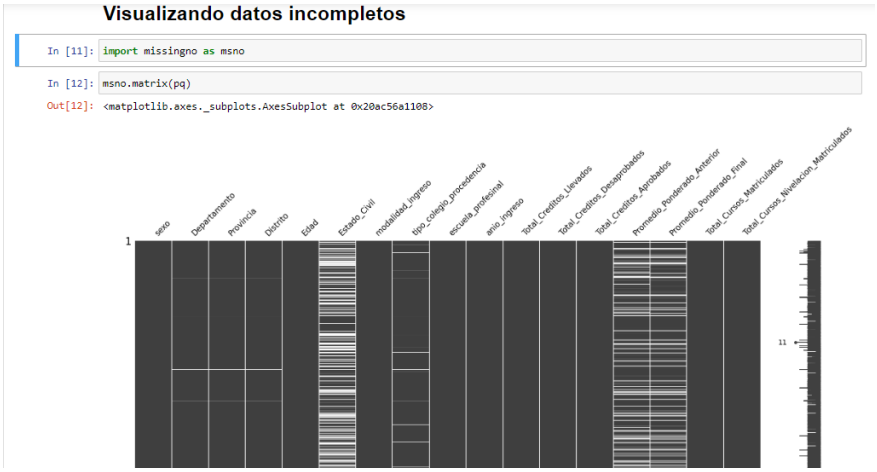


Figura 92 *Visualización de datos limpios*

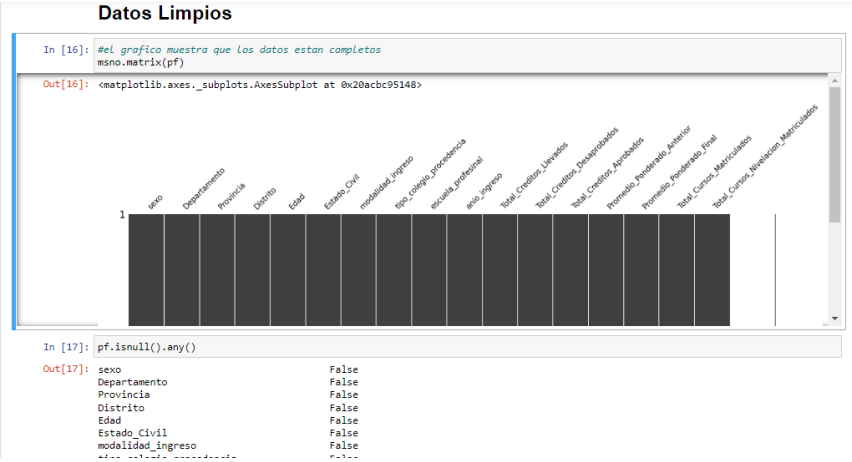


Figura 93 *Datos explorados (Cantidad de columnas)*

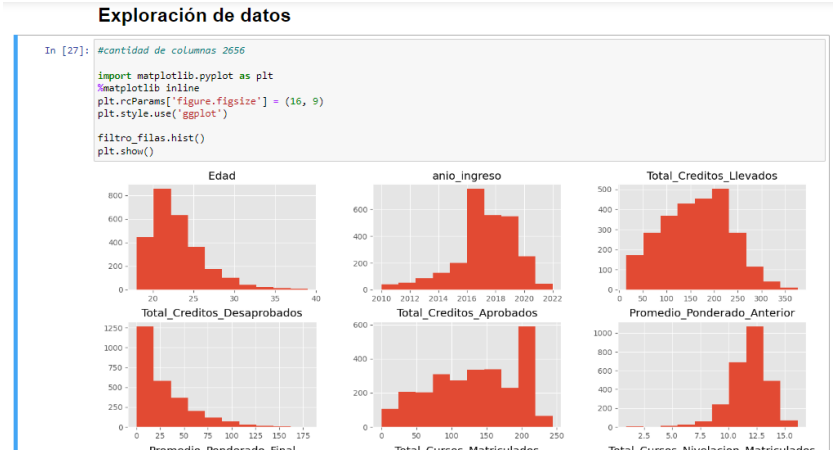


Figura 94 *Transformación de datos del promedio final*

Figura 95 *Transformación de datos del estado civil*

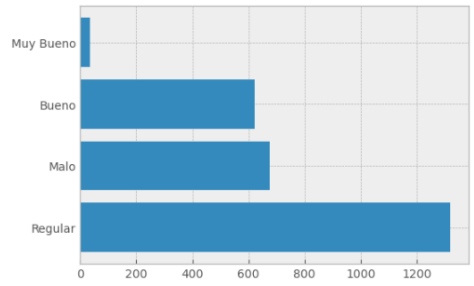
Transformando datos de estado civil

```
In [35]: # convertir estado civil
condicion_estadocivil=[
    (pf['Estado_Civil'] == 'SOLTERO'),
    (pf['Estado_Civil'] == 'VIUDO'),
    (pf['Estado_Civil'] == 'CASADO'),
    (pf['Estado_Civil'] == 'DIVORCIADO')
]
escala_estadocivil=['1','2','3','4']
pf['ES_2']=np.select(condicion_estadocivil, escala_estadocivil)

In [36]: pf['ES_2']

Out[36]: 0       1
         1       1
         2       1
         3       1
         4       1
                ..
         4578    1
         4579    1
         4580    1
         4582    1
         4583    1
         Name: ES_2, Length: 2656, dtype: object

In [37]: condicion_TCP=[
             (pf['tipo_colegio_procedencia'] == 'Publico'),
             (pf['tipo_colegio_procedencia'] == 'Privado')
         ]
```

Figura 96 *Visualización de los datos por clases*

Mostramos datos por clases

```
In [45]: #sb.factorplot('CALIFICACION',data=data_datos,kind="count", aspect=3)
         #sb.figure(figsize=(8, 6))

         conteo = data_datos['CALIFICACION'].value_counts()
         plt.style.use("bmh")
         plt.barh(conteo.index,conteo)
         plt.show()
         #plt.close('all')
```




```
In [46]: data_datos
```

Figura 97 *Matriz de correlación de variables (KNN)*

Matriz de correlación de variables

```
In [54]: #correlacion de las variables

         import pandas as pds
         import seaborn as sns
         import matplotlib.pyplot as plt
         data_datos_2
         corr_df = data_datos_2.corr(method='pearson')

         plt.figure(figsize=(8, 6))
         sns.heatmap(corr_df, annot=True)
         plt.show()
```

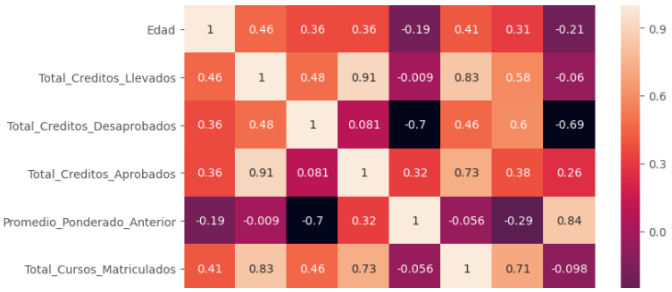


Figura 98 *Selección de variables predichas y predictoras*

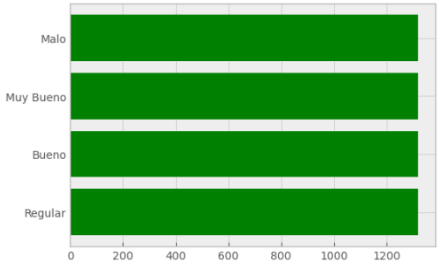
Seleccionando Variables Predichas y predictoras

```
In [56]: X = data_over[['GENERO','Edad','ES_2',
                        'TCP_2','Total_Creditos_Llevados',
                        'Total_Creditos_Desaprobados','Total_Creditos_Aprobados','Promedio_Ponderado_Anterior',
                        'Total_Cursos_Matriculados','Total_Cursos_Nivelacion_Matriculados']].values

In [57]: X.shape
Out[57]: (5280, 10)

         Type Markdown and LaTeX: α²

In [58]: Y = data_over['CALIFICACION']
         #ya = data_datos["CALIFICACION"]
         Y.shape
Out[58]: (5280,)

In [59]: #cantidad de alumnos por rendiento
         datos_balanceados = Y.value_counts()

In [60]: plt.rcParams['figure.figsize'] = (6, 4)
```

Figura 99 *Balanceo de datos*

Datos Balanceados

```
In [61]: conteo = datos_balanceados
         plt.style.use("bmh")
         plt.barh(conteo.index,conteo,color ='green')
         plt.show()
         #plt.close('all')
```




```
In [62]: from sklearn.metrics import roc_curve, auc
         from sklearn import datasets
         from sklearn.multiclass import OneVsRestClassifier
         from sklearn.svm import LinearSVC
```

Figura 100 *Modelo KNN*

Modelo KNN

```
In [64]: from sklearn.model_selection import GridSearchCV
         from sklearn.neighbors import KNeighborsClassifier

         # defining parameter range
         param_grid = {'n_neighbors' : [5,7,9,11,13,15],
                       'weights' : ['uniform','distance'],
                       'metric' : ['minkowski','euclidean','manhattan']
                        }

         grid = GridSearchCV(KNeighborsClassifier(),
                             param_grid,
                             refit = True,
                             verbose = 3,
                             cv = 5)

         # fitting the model for grid search
         grid.fit(X_trainset, y_trainset)

         Fitting 5 folds for each of 36 candidates, totalling 180 fits
         [CV 1/5] END metric=minkowski, n_neighbors=5, weights=uniform;, score=0.859 total time=   0.0s
         [CV 2/5] END metric=minkowski, n_neighbors=5, weights=uniform;, score=0.847 total time=   0.0s
         [CV 3/5] END metric=minkowski, n_neighbors=5, weights=uniform;, score=0.852 total time=   0.0s
         [CV 4/5] END metric=minkowski, n_neighbors=5, weights=uniform;, score=0.876 total time=   0.0s
         [CV 5/5] END metric=minkowski, n_neighbors=5, weights=uniform;, score=0.874 total time=   0.0s
         [CV 1/5] END metric=minkowski, n_neighbors=5, weights=distance;, score=0.909 total time=   0.0s
         [CV 2/5] END metric=minkowski, n_neighbors=5, weights=distance;, score=0.885 total time=   0.0s
         [CV 3/5] END metric=minkowski, n_neighbors=5, weights=distance;, score=0.889 total time=   0.0s
         [CV 4/5] END metric=minkowski, n_neighbors=5, weights=distance;, score=0.914 total time=   0.0s
         [CV 5/5] END metric=minkowski, n_neighbors=5, weights=distance;, score=0.914 total time=   0.0s
         [CV 1/5] END metric=minkowski, n_neighbors=7, weights=uniform;, score=0.850 total time=   0.0s
         [CV 2/5] END metric=minkowski, n_neighbors=7, weights=uniform;, score=0.839 total time=   0.0s
         [CV 3/5] END metric=minkowski, n_neighbors=7, weights=uniform;, score=0.860 total time=   0.0s
```

Figura 101 *Resultados de las métricas del modelo KNN (Train)*

Métricas del Modelo (Train)

```
In [69]: pred = model.predict(X_trainset)

In [70]: from sklearn.metrics import confusion_matrix
         from sklearn.metrics import classification_report

In [71]: #accuracy solo es la diagonal / total
         from sklearn.metrics import accuracy_score
         print('acuracy =' , accuracy_score(y_trainset,pred))

         #Recall sensibilidad
         from sklearn.metrics import recall_score
         print('sensibilidad =',recall_score(y_trainset, pred, average='macro'))

         #precision
         from sklearn.metrics import precision_score
         print('precision =', precision_score(y_trainset, pred, average='macro'))

         #f1_score
         from sklearn.metrics import f1_score
         print('f1_score =',f1_score(y_trainset, pred, average='macro'))

         acuracy = 1.0
         sensibilidad = 1.0
         precision = 1.0
         f1_score = 1.0

In [72]: print(classification_report(y_trainset, pred))
                      precision    recall  f1-score   support
```

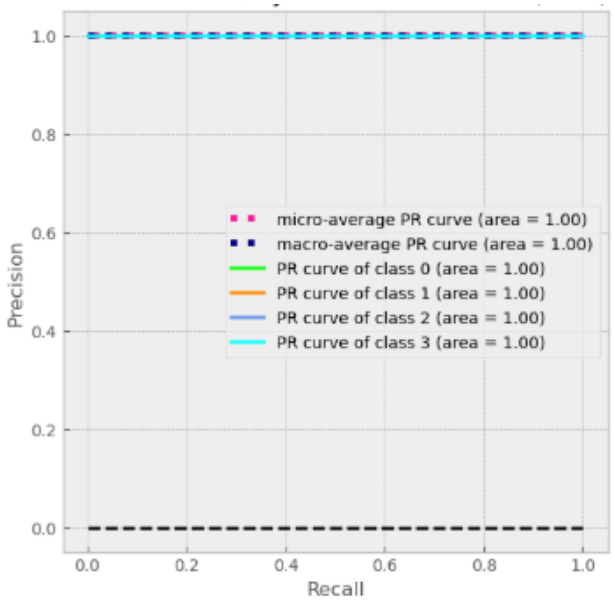


Figura 102 *Matriz de Confusión del modelo KNN (Train)*

Matriz de Confusion TRAIN

```
In [75]: import pandas as pds
         import seaborn as sns
         import matplotlib.pyplot as plt
         import pandas as pds

         cnf_matrix= pds.crosstab(y_trainset, pred) #, rownames = ['ACTUAL'],colnames=['Prediccion'])
         sns.heatmap(cnf_matrix, square=True, annot=True, fmt='d', cbar=True, cmap='Blues')
         plt.xlabel('Predicted Class')
         plt.ylabel('True Class')
         plt.title('KNN Confusion matrix (Train)')
         plt.show()
```

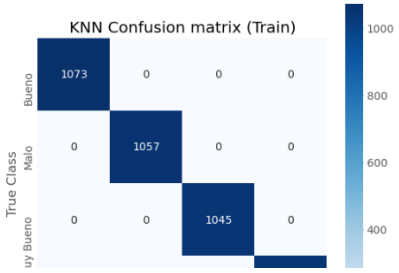
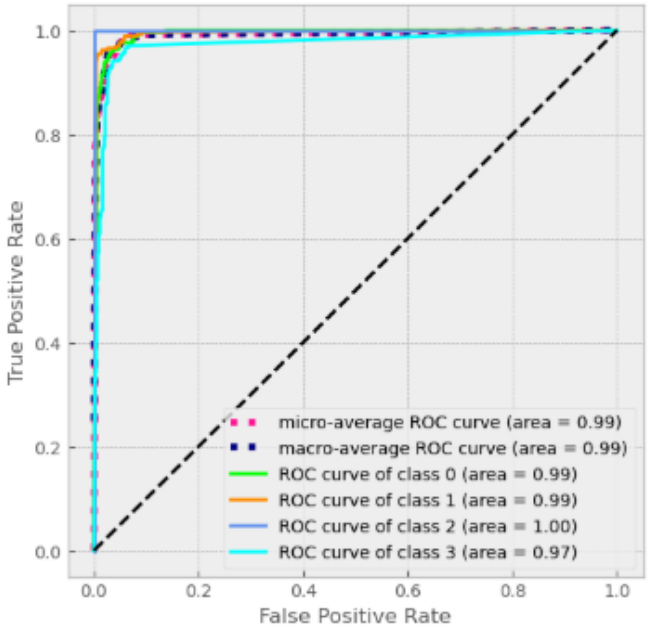


Figura 103 *Resultados de las métricas del modelo KNN (Test)*

Métricas del Modelo (Test)

```
In [90]: pred = model.predict(X_testset)

In [91]: #accuracy solo es la diagonal / total
from sklearn.metrics import accuracy_score
print('acuracy =' , accuracy_score(y_testset,pred))

#Recall sensibilidad
from sklearn.metrics import recall_score
print('sensibilidad =',recall_score(y_testset, pred, average='macro'))

#precision
from sklearn.metrics import precision_score
print('precision =', precision_score(y_testset, pred, average='macro'))

#f1_score
from sklearn.metrics import f1_score
print('f1_score =',f1_score(y_testset, pred, average='macro'))

acuracy = 0.9289772727272727
sensibilidad = 0.9298141047557856
precision = 0.9291265330028456
f1_score = 0.9271774905768058

In [92]: print(classification_report(y_testset, pred))

              precision    recall  f1-score   support

       Bueno       0.86      0.97      0.91       247
        Malo       0.93      0.96      0.95       263
   Muy Bueno       0.99      1.00      1.00       275
     Regular       0.93      0.79      0.85       271
```



1. Accuracy (92.89%): El modelo logró una precisión del 92.89%, lo que indica que más del 92% de las predicciones realizadas fueron correctas al clasificar el rendimiento académico de los estudiantes en las diferentes categorías. La universidad puede identificar a los estudiantes tendrán un bajo rendimiento académico. Esto permite implementar programas de apoyo como tutorías personalizadas, asesorías psicológicas o justes en los métodos de enseñanza. Así como también puede impactar en la optimización de recursos educativos, diseño de estrategias pedagógicas, evaluación y mejora de planes de estudio.

2. Sensibilidad (Recall) (92.98%): Con un valor de 92.98%, esta métrica refleja la capacidad del modelo para identificar correctamente a los estudiantes dentro de cada categoría de rendimiento académico, garantizando que los casos relevantes sean detectados con alta eficacia. Este nivel alto de sensibilidad impacta en la detección temprana y focalizada de los estudiantes asegurando que las decisiones y estrategias implementadas sean inclusivas y efectivas.

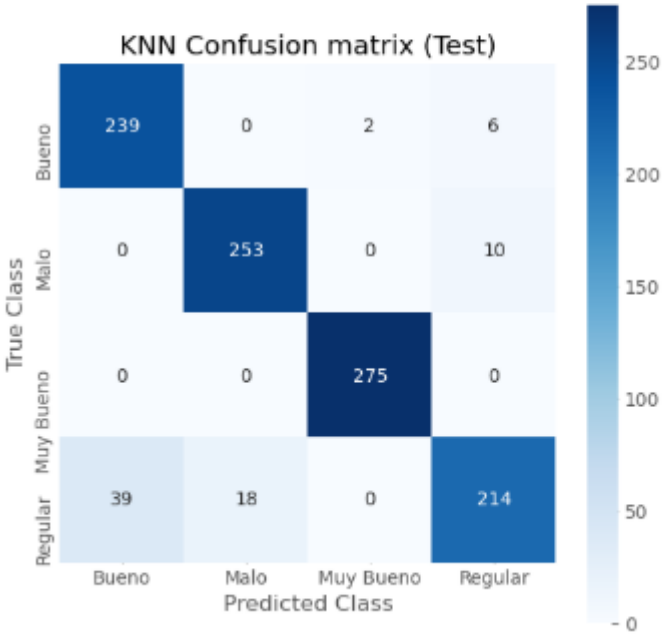
3. Precisión (Precisión) (92.91%): La precisión obtenida fue del **92.91%**, lo que evidencia que el modelo realiza predicciones positivas con un alto grado de confianza, minimizando los falsos positivos en la clasificación del rendimiento estudiantil.

4. F1-Score (92.71%): El F1-Score fue de 92.715%, lo que demuestra que el modelo equilibra de manera efectiva la sensibilidad y la precisión, optimizando su desempeño incluso en situaciones donde las clases pueden estar desbalanceadas

5. Curva ROC y AUC (Área bajo la curva) (0.99): El modelo alcanzó un AUC de 0.99, lo que evidencia su capacidad sobresaliente para distinguir entre las diferentes categorías de rendimiento académico de los estudiantes. Este resultado sugiere un alto nivel de discriminación entre las clases.

Figura 104 *Matriz de Confusión del modelo KNN (Test)*

```
import pandas as pds
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pds

cnf_matrix= pds.crosstab(y_testset, pred) #, rownames = ['ACTUAL'],colnames=['Prediccion'])
sns.heatmap(cnf_matrix, square=True, annot=True, fmt='d', cbar=True, cmap='Blues')
plt.xlabel('Predicted Class')
plt.ylabel('True Class')
plt.title('KNN Confusion matrix (Test)')
plt.show()
```



La matriz muestra cómo el modelo clasifica las instancias entre las clases:

- **Diagonal principal (valores correctos):** Los números altos en la diagonal indican que el modelo clasifica correctamente la mayoría de las instancias de todas las clases:
  - 239 casos de la clase "Bueno" fueron clasificados correctamente.

- 253 casos de la clase "Malo" fueron clasificados correctamente.
- 275 casos de la clase "Muy Bueno" fueron clasificados correctamente.
- 214 casos de la clase "Regular" fueron clasificados correctamente.

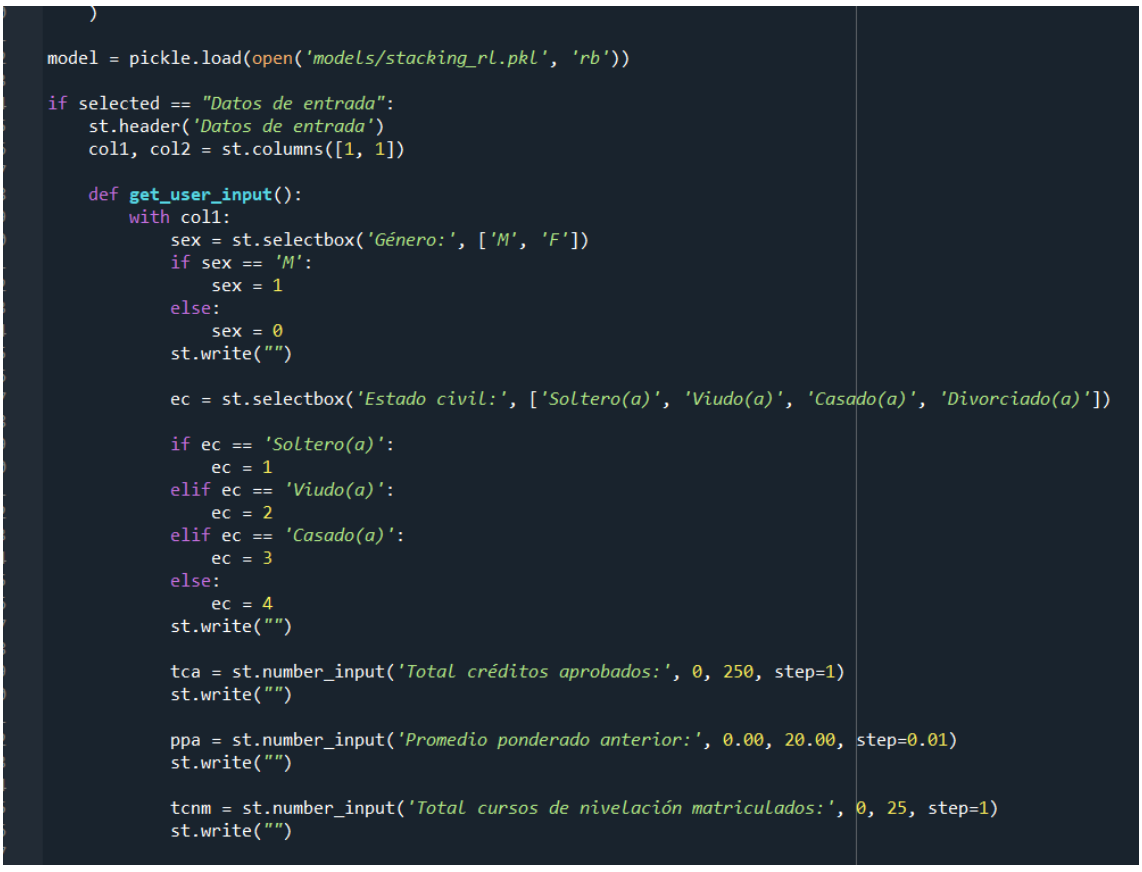
- **Errores de clasificación:**

Por ejemplo, 6 casos de la clase "Bueno" fueron clasificados como “Regular” asi como 18 casos de la clase “Regular” fueron clasificados Malo”.

Figura 105 *Código del sistema inteligente-Parte 1*

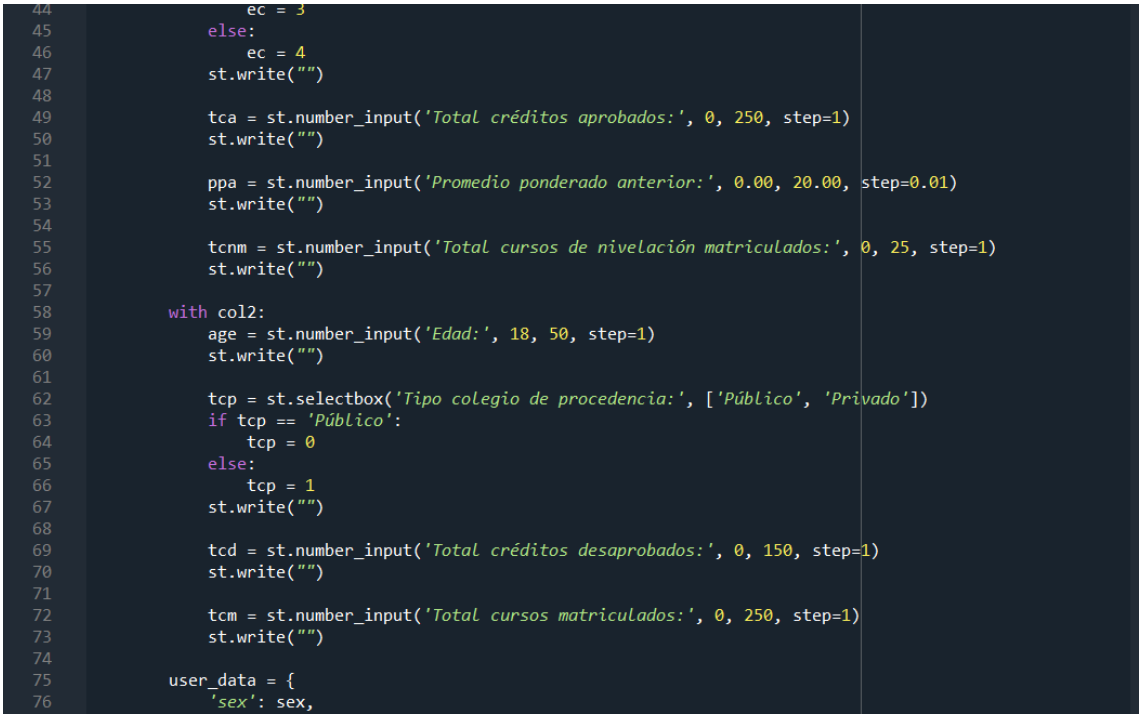
```python
from streamlit_option_menu import option_menu
import streamlit as st
import pandas as pd
import numpy as np
import pickle

st.set_page_config(
    page_title="Predicción de Rendimiento Académico",
    page_icon="https://images.emojiterra.com/google/android-jellybean/1f3eb.png",
    layout="wide",
    initial_sidebar_state="expanded"
)

with st.sidebar:
    selected = option_menu(
        menu_title=None,
        options=["Datos de entrada", "Subir datos"],
        icons=["mortarboard", "mortarboard-fill"],
        menu_icon="pencil-square"
    )

model = pickle.load(open('models/stacking_rl.pkl', 'rb'))
```

Figura 106 *Código del sistema inteligente-Parte 2*



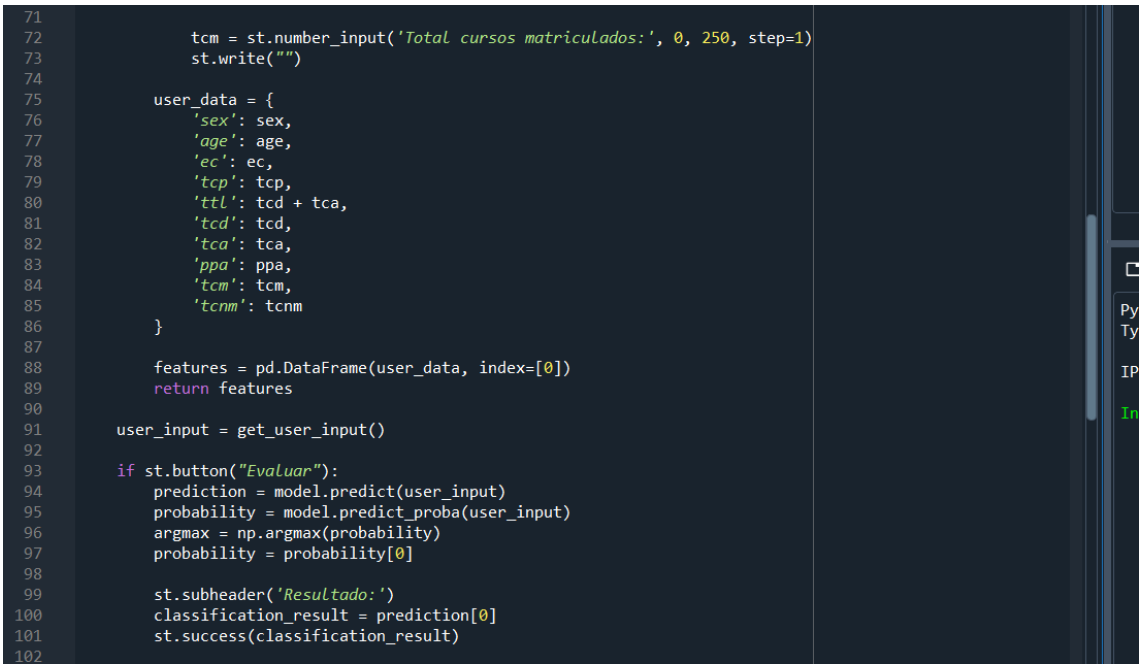
```python
)

model = pickle.load(open('models/stacking_rl.pkl', 'rb'))

if selected == "Datos de entrada":
    st.header('Datos de entrada')
    col1, col2 = st.columns([1, 1])

    def get_user_input():
        with col1:
            sex = st.selectbox('Género:', ['M', 'F'])
            if sex == 'M':
                sex = 1
            else:
                sex = 0
            st.write("")

            ec = st.selectbox('Estado civil:', ['Soltero(a)', 'Viudo(a)', 'Casado(a)', 'Divorciado(a)'])

            if ec == 'Soltero(a)':
                ec = 1
            elif ec == 'Viudo(a)':
                ec = 2
            elif ec == 'Casado(a)':
                ec = 3
            else:
                ec = 4
            st.write("")

            tca = st.number_input('Total créditos aprobados:', 0, 250, step=1)
            st.write("")

            ppa = st.number_input('Promedio ponderado anterior:', 0.00, 20.00, step=0.01)
            st.write("")

            tcnm = st.number_input('Total cursos de nivelación matriculados:', 0, 25, step=1)
            st.write("")
```

Figura 107 *Código del sistema inteligente-Parte 3*



```
                ec = 3
            else:
                ec = 4
            st.write("")

            tca = st.number_input('Total créditos aprobados:', 0, 250, step=1)
            st.write("")

            ppa = st.number_input('Promedio ponderado anterior:', 0.00, 20.00, step=0.01)
            st.write("")

            tcnm = st.number_input('Total cursos de nivelación matriculados:', 0, 25, step=1)
            st.write("")

        with col2:
            age = st.number_input('Edad:', 18, 50, step=1)
            st.write("")

            tcp = st.selectbox('Tipo colegio de procedencia:', ['Público', 'Privado'])
            if tcp == 'Público':
                tcp = 0
            else:
                tcp = 1
            st.write("")

            tcd = st.number_input('Total créditos desaprobados:', 0, 150, step=1)
            st.write("")

            tcm = st.number_input('Total cursos matriculados:', 0, 250, step=1)
            st.write("")

        user_data = {
            'sex': sex,
```

Figura 108 *Código del sistema inteligente-Parte 4*



```
            tcm = st.number_input('Total cursos matriculados:', 0, 250, step=1)
            st.write("")

        user_data = {
            'sex': sex,
            'age': age,
            'ec': ec,
            'tcp': tcp,
            'ttl': tcd + tca,
            'tcd': tcd,
            'tca': tca,
            'ppa': ppa,
            'tcm': tcm,
            'tcnm': tcnm
        }

        features = pd.DataFrame(user_data, index=[0])
        return features

    user_input = get_user_input()

    if st.button("Evaluar"):
        prediction = model.predict(user_input)
        probability = model.predict_proba(user_input)
        argmax = np.argmax(probability)
        probability = probability[0]

        st.subheader('Resultado:')
        classification_result = prediction[0]
        st.success(classification_result)
```

Figura 109 *Código del sistema inteligente-Parte 5*

```
    user_input = get_user_input()

    if st.button("Evaluar"):
        prediction = model.predict(user_input)
        probability = model.predict_proba(user_input)
        argmax = np.argmax(probability)
        probability = probability[0]

        st.subheader('Resultado:')
        classification_result = prediction[0]
        st.success(classification_result)

        st.subheader('Exactitud:')
        st.success(str((probability[argmax] * 100).round(2)) + "%")

if selected == "Subir datos":
    st.header('Evaluar datos subidos al sistema')
    uploaded_file = st.file_uploader(
        "Cargar archivos:", type=["xlsx"]
    )

    if uploaded_file:
        data = pd.read_excel(uploaded_file, index_col=None)

        X = data.values
        prediction = model.predict(X)
        probability = model.predict_proba(X)
        probs = []

        for i in probability:
            jhomby = i[np.argmax(i)]
            value = (jhomby * 100).round(2)
            probs.append(value)
```

Figura 110 *Código del sistema inteligente-Parte 6*

```
        for i in probability:
            jhomby = i[np.argmax(i)]
            value = (jhomby * 100).round(2)
            probs.append(value)

        #  = np.argmax(probability)
        probs = pd.DataFrame(probs, columns=['Exactitud']).astype(str)
        probs['Exactitud'] = probs['Exactitud'] + '%'

        df = data[['GENERO','Edad','ES_2',
               'TCP_2','Total_Creditos_Llevados',
               'Total_Creditos_Desaprobados','Total_Creditos_Aprobados','Promedio_Ponderado_Anterior',
               'Total_Cursos_Matriculados','Total_Cursos_Nivelacion_Matriculados']]

        df['Resultado'] = prediction
        df['Exactitud'] = probs['Exactitud']

        st.subheader('Resultados:')
        st.write(df)

# Hide markdown
hide_st_style = """
            <style>
            #MainMenu {visibility: hidden;}
            footer {visibility: hidden;}
            #header {visibility: hidden;}
            </style>
            """
st.markdown(hide_st_style, unsafe_allow_html=True)
```

Figura 111 *Sistema inteligente haciendo uso de la técnica Árbol de decisión-Parte 1*

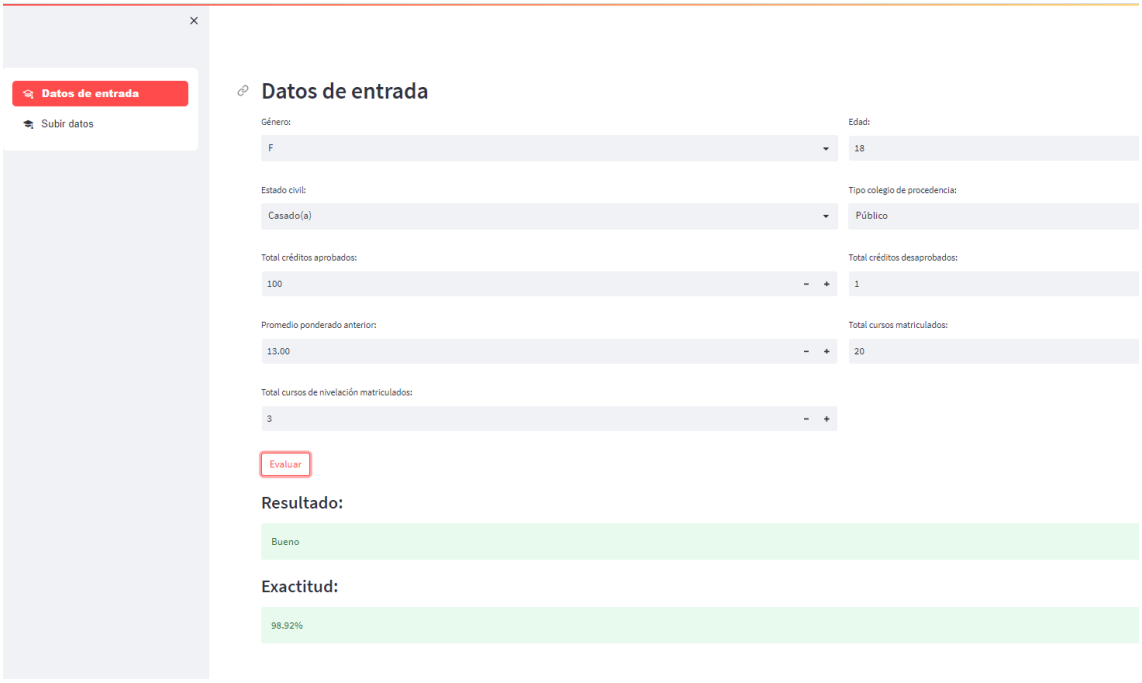
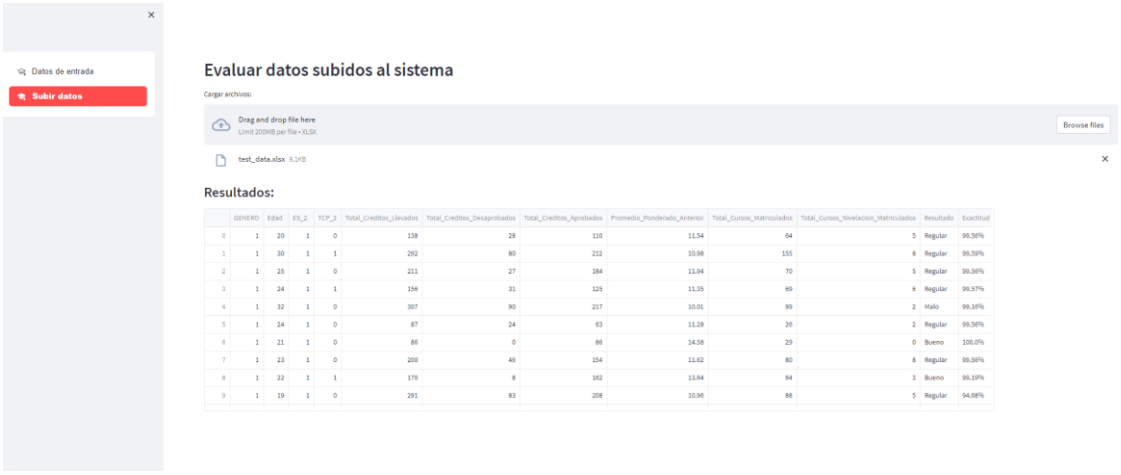


Figura 112 *Sistema inteligente haciendo uso de la técnica Árbol de decisión-Parte 2*



Anexo 4

Tabla 33. *Estudios complementarios utilizados para la investigación*

Autor	Año	Titulo	Técnicas usadas	Métricas
Buenaño, Gil y Luján	2019	Application of Machine Learning in Predicting Performance for Computer Engineering Students: A Case Study	Decision tree	Accuracy
Harvey y Kumar	2019	A Practical Model for Educators to Predict Student Performance in K-12 Education using Machine Learning	Regresión lineal, Árbol de decisión y Naive Bayes	Accuracy
Lau, Sun y Yang	2019	Modelling, prediction and classification of student academic performance using artificial neural networks	Red neuronal artificial	Accuracy, Error, Sensibilidad, Especificidad, Precisión y AUC
Yousafzai, Hayat y Afzal	2020	Application of machine learning and data mining in predicting the performance of intermediate and secondary education level student	Genetic algorithm	Accuracy
Singh y Pal	2020	Machine learning algorithms and ensemble technique to improve prediction of students performance	KNN, Naive Bayes, Decision Tree y ExtraTree	Accuracy, Recall y F1-Score
Katarya et al	2021	A review on machine learning based student's academic performance prediction systems	Gartner Analytics Ascendancy Model, Naive Bayes, SMO, ANN, Random Forest, KNN, Partial decision trees, REPTree, SVM, Decision tree y logistic regression	Accuracy, AUC, precision, R2 score
Khan et al	2021	A Conceptual Framework to Aid Attribute Selection in Machine Learning Student Performance Prediction Models	Red neuronal artificial	--

Yağcı	2022	Educational data mining: prediction of students' academic performance using machine learning algorithms	Random Forest, Redes neuronales, Maquina de Vectores, Regresión Logística, Naive Bayes y KNN	Accuracy, Precisión, F1-Score y Recall
Menacho	2017	Predicción del rendimiento académico aplicando técnicas de minería de datos	Regresión Logística, Árbol de Decisión J48, Red Neuronal y Naive de Bayes	Curva ROC, Coeficiente Kappa, precisión
Yamao	2018	Predicción del rendimiento académico mediante minería de datos en estudiantes del primer ciclo de las Escuela Profesional de Ingeniería de Computación y Sistemas	Regresión lineal, Árbol de decisiones y Máquina de vector de soporte	Precisión
Orihuela	2019	Aplicación de Data Science para la predicción del rendimiento académico de los estudiantes de la Facultad de Ingeniería de Sistemas de la Universidad Nacional del Centro del Perú	Regresión logística y Random Forest	Curva ROC, Precisión, Sensibilidad y F1-Score
Candia	2019	Predicción del rendimiento académico de los estudiantes de la UNSAAC a partir de sus datos de ingreso utilizando técnicas de Machine Learning	Árboles de decisión J48, Random Forest, Vecinos más cercanos (KNN), Función de Regresión Logística y Perceptrón multicapa	Precisión
Espinoza & León	2020	Modelo de Machine Learning para la clasificación de estudiantes de acuerdo con su rendimiento académico en el Centro de Idiomas de la Universidad Nacional del Santa	Regresión logística	Exactitud

Garcia	2021	Machine learning para predecir el rendimiento académico de los estudiantes universitarios	KNN, Árbol de decisión y Maquina de Vectores	Precisión, Especificidad y Sensibilidad
Aronés	2021	Predicción del rendimiento académico basado en Machine Learning, Escuela Profesional de Ingeniería de Sistemas, Ayacucho 2021	Regresión logística, SVM, Random Forest, KNN y Árbol de decisión	Sensibilidad, Especificidad, Curva ROC y Validación cruzada

Fuente: Elaboración propia

Anexo 5

Evaluación de métricas de las técnicas de Machine Learning

Árbol de decisión (Train)

Tabla 34. *Métrica de evaluación (Train) – Exactitud árbol de decisión*

Ítem	Métrica	Escala	Fórmula	Exactitud
1	Exactitud	Razón	$Exactitud = \left(\frac{TP + TN}{TP + TN + FP + FN}\right) * 100$	96.88%

Fuente: Elaboración propia

Interpretación: Al hacer uso de técnicas de machine learning, se logra predecir el rendimiento académico de los alumnos de la UNAS con una exactitud=96.88% empleando la técnica árbol de decisión.

Redes Neuronales (Train)

Tabla 35. *Métrica de evaluación (Train) – Exactitud Redes Neuronales*

Ítem	Métrica	Escala	Fórmula	Exactitud
1	Exactitud	Razón	$Exactitud = \left(\frac{TP + TN}{TP + TN + FP + FN}\right) * 100$	92.85%

Fuente: Elaboración propia

Interpretación: Al hacer uso de técnicas de machine learning, se logra predecir el rendimiento académico de los alumnos de la UNAS con una exactitud=92.85% empleando la técnica Redes neuronales.

SVM (Train)

Tabla 36. *Métrica de evaluación (Train) – Exactitud SVM*

Ítem	Métrica	Escala	Fórmula	Exactitud
1	Exactitud	Razón	$Exactitud = \left(\frac{TP + TN}{TP + TN + FP + FN}\right) * 100$	94.84%

Fuente: Elaboración propia

Interpretación: Al hacer uso de técnicas de machine learning, se logra predecir el rendimiento académico de los alumnos de la UNAS con una exactitud=94.84% empleando la técnica SVM.

Redes Bayesianas (Train)

Tabla 37. *Métrica de evaluación (Train) – Exactitud Redes Bayesianas*

Ítem	Métrica	Escala	Fórmula	Exactitud
1	Exactitud	Razón	$Exactitud = \left(\frac{TP + TN}{TP + TN + FP + FN}\right) * 100$	86.29%

Fuente: Elaboración propia

Interpretación: Al hacer uso de técnicas de machine learning, se logra predecir el rendimiento académico de los alumnos de la UNAS con una exactitud=86.29% empleando la técnica Redes Bayesianas.

KNN (Train)

Tabla 38. *Métrica de evaluación (Train) – Exactitud KNN*

Ítem	Métrica	Escala	Fórmula	Exactitud
1	Exactitud	Razón	$Exactitud = \left(\frac{TP + TN}{TP + TN + FP + FN}\right) * 100$	100.00%

Fuente: Elaboración propia

Interpretación: Al hacer uso de técnicas de machine learning, se logra predecir el rendimiento académico de los alumnos de la UNAS con una exactitud=100.00% empleando la técnica KNN.

Tabla 39. *Cuadro comparativo de resultados rendimiento académico de los estudiantes de la UNAS en base a la métrica Exactitud (Train)*

TÉCNICA	RESULTADO (%)
K-NN	100.00%
Árbol de decisión	96.88%
SVM	94.84%
Redes neuronales	92.85%
Redes bayesianas	86.29%

Fuente: Elaboración propia

Interpretación: Como se muestra en la tabla 67, la técnica con el resultado óptimo referente al indicador exactitud que predice el rendimiento académico de los estudiantes de la UNAS correctamente es K-NN=100.00%, a la vez de Árbol de decisión=96.88%, luego SVM=94.84%, de la misma manera Redes neuronales= 92.85% y finalmente Redes bayesianas=86.29%. ´

Árbol de decisión (Train)

Tabla 40. *Métrica de evaluación (Train) – Precisión árbol de decisión*

Ítem	Métrica	Escala	Fórmula	Precisión
2	Precisión	Razón	$\frac{TP}{TP + FP}$	96.89%

Fuente: Elaboración propia

Interpretación: Al hacer uso de técnicas de machine learning, se logra predecir el rendimiento académico de los alumnos de la UNAS con una precisión=96.89% empleando la técnica Árbol de decisión.

Redes Neuronales (Train)

Tabla 41. *Métrica de evaluación (Train) – Precisión Redes Neuronales*

Ítem	Métrica	Escala	Fórmula	Precisión
2	Precisión	Razón	$\frac{TP}{TP + FP}$	92.88%

Fuente: Elaboración propia

Interpretación: Al hacer uso de técnicas de machine learning, se logra predecir el rendimiento académico de los alumnos de la UNAS con una precisión=92.88% empleando la técnica Redes Neuronales.

SVM (Train)

Tabla 42. *Métrica de evaluación (Train) – Precisión SVM*

Ítem	Métrica	Escala	Fórmula	Precisión
2	Precisión	Razón	$\frac{TP}{TP + FP}$	94.80%

Fuente: Elaboración propia

Interpretación: Al hacer uso de técnicas de machine learning, se logra predecir el rendimiento académico de los alumnos de la UNAS con una precisión=94.80% empleando la técnica SVM.

Redes Bayesianas (Train)

Tabla 43. *Métrica de evaluación (Train) – Precisión Redes Bayesianas*

Ítem	Métrica	Escala	Fórmula	Precisión
2	Precisión	Razón	$\frac{TP}{TP + FP}$	86.51%

Fuente: Elaboración propia

Al hacer uso de técnicas de machine learning, se logra predecir el rendimiento académico de los alumnos de la UNAS con una precisión=86.51% empleando la técnica Redes Bayesianas.

KNN (Train)

Tabla 44. *Métrica de evaluación (Train) – Precisión KNN*

Ítem	Métrica	Escala	Fórmula	Precisión
2	Precisión	Razón	$\frac{TP}{TP + FP}$	100.00%

Fuente: Elaboración propia

Interpretación: Al hacer uso de técnicas de machine learning, se logra predecir el rendimiento académico de los alumnos de la UNAS con una precisión=100.00% empleando la técnica KNN.

Tabla 45. *Cuadro comparativo de resultados rendimiento académico de los estudiantes de la UNAS en base a la métrica Precisión (Train)*

TÉCNICA	RESULTADO (%)
K-NN	100.00%
Árbol de decisión	96.89%
SVM	94.80%
Redes neuronales	92.88%
Redes bayesianas	86.51%

Fuente: Elaboración propia

Interpretación: En la tabla 73 se evidencia que la técnica con mejor resultado en cuanto al indicador precisión que predice el rendimiento académico de los estudiantes de la UNAS correctamente es “K-NN” con un 100.00%, seguido de la técnica “Árbol de decisión” con un resultado igual a 96.89%, luego “SVM” con 94.80%, asimismo sigue “Redes neuronales” con 92.88% y por último “Redes bayesianas” con un valor del 86.51%

Árbol de decisión (Train)

Tabla 46. *Métrica de evaluación (Train)– Sensibilidad árbol de decisión*

Ítem	Métrica	Escala	Fórmula	Sensibilidad
3	Sensibilidad	Razón	$Sensibilidad = \left(\frac{TP}{TP + FN}\right) * 100$	96.88%

Fuente: Elaboración propia

Interpretación: Al hacer uso de técnicas de machine learning, se logra predecir el rendimiento académico de los alumnos de la UNAS con una sensibilidad =96.88% empleando la técnica Árbol de decisión.

Redes Neuronales (Train)

Tabla 47. *Métrica de evaluación (Train)– Sensibilidad Redes Neuronales*

Ítem	Métrica	Escala	Fórmula	Sensibilidad
3	Sensibilidad	Razón	$Sensibilidad = \left(\frac{TP}{TP + FN}\right) * 100$	92.86%

Fuente: Elaboración propia

Interpretación: Al hacer uso de técnicas de machine learning, se logra predecir el rendimiento académico de los alumnos de la UNAS con una sensibilidad=92.86% empleando la técnica Redes Neuronales.

SVM (Train)

Tabla 48. *Métrica de evaluación (Train)– Sensibilidad SVM*

Ítem	Métrica	Escala	Fórmula	Sensibilidad
3	Sensibilidad	Razón	$Sensibilidad = \left(\frac{TP}{TP + FN}\right) * 100$	94.84%

Fuente: Elaboración propia

Interpretación: Al hacer uso de técnicas de machine learning, se logra predecir el rendimiento académico de los alumnos de la UNAS con una sensibilidad=94.84% empleando la técnica SVM.

Redes Bayesianas (Train)

Tabla 49. *Métrica de evaluación (Train)– Sensibilidad Redes Bayesianas*

Ítem	Métrica	Escala	Fórmula	Sensibilidad
3	Sensibilidad	Razón	$Sensibilidad = \left(\frac{TP}{TP + FN}\right) * 100$	86.28%

Fuente: Elaboración propia

Interpretación: Al hacer uso de técnicas de machine learning, se logra predecir el rendimiento académico de los alumnos de la UNAS con una sensibilidad=86.28% empleando la técnica Redes Bayesianas.

KNN (Train)

Tabla 50. *Métrica de evaluación (Train)– Sensibilidad KNN*

Ítem	Métrica	Escala	Fórmula	Sensibilidad
3	Sensibilidad	Razón	$Sensibilidad = \left(\frac{TP}{TP + FN}\right) * 100$	100.00%

Fuente: Elaboración propia

Interpretación: Al hacer uso de técnicas de machine learning, se logra predecir el rendimiento académico de los alumnos de la UNAS con una sensibilidad=100.00% empleando la técnica KNN.

Tabla 51. *Cuadro comparativo de resultados rendimiento académico de los estudiantes de la UNAS en base a la métrica Sensibilidad (Train)*

TÉCNICA	RESULTADO (%)
K-NN	100.00%
Árbol de decisión	96.88%
SVM	94.84%
Redes neuronales	92.88%
Redes bayesianas	86.28%

Fuente: Elaboración propia

Interpretación: En la tabla 79 se evidencia que la técnica con mejor resultado en cuanto al indicador sensibilidad que predice el rendimiento académico de los estudiantes de la UNAS correctamente es “K-NN” con un 100.00%, seguido de “Árbol de decisión” con un resultado igual a 96.88%, luego “SVM” con 94.84%, asimismo sigue “Redes neuronales” con 92.86% y por último “Redes bayesianas” con un valor del 86.28%.

Árbol de decisión (Train)

Tabla 52. *Métrica de evaluación (Train) – Especificidad árbol de decisión*

Ítem	Métrica	Escala	Fórmula	Especificidad
4	Especificidad	Razón	$Especificidad = \left(\frac{TN}{TN + FP}\right) * 100$	98.96%

Fuente: Elaboración propia

Interpretación: Al hacer uso de técnicas de machine learning, se logra predecir el rendimiento académico de los alumnos de la UNAS con una especificidad=98.96% empleando la técnica Árbol de decisión.

Redes Neuronales (Train)

Tabla 53. *Métrica de evaluación (Train) – Especificidad Redes Neuronales*

Ítem	Métrica	Escala	Fórmula	Especificidad
4	Especificidad	Razón	$Especificidad = \left(\frac{TN}{TN + FP}\right) * 100$	97.62%

Fuente: Elaboración propia

Interpretación: Al hacer uso de técnicas de machine learning, se logra predecir el rendimiento académico de los alumnos de la UNAS con una especificidad=97.62% empleando la técnica Redes Neuronales.

SVM (Train)

Tabla 54. *Métrica de evaluación (Train) – Especificidad SVM*

Ítem	Métrica	Escala	Fórmula	Especificidad
4	Especificidad	Razón	$Especificidad = \left(\frac{TN}{TN + FP}\right) * 100$	98.28%

Fuente: Elaboración propia

Interpretación: Al hacer uso de técnicas de machine learning, se logra predecir el rendimiento académico de los alumnos de la UNAS con una especificidad=98.28% empleando la técnica SVM.

Redes Bayesianas (Train)

Tabla 55. *Métrica de evaluación (Train) – Especificidad Redes Bayesianas*

Ítem	Métrica	Escala	Fórmula	Especificidad
4	Especificidad	Razón	$Especificidad = \left(\frac{TN}{TN + FP}\right) * 100$	95.42%

Fuente: Elaboración propia

Interpretación: Al hacer uso de técnicas de machine learning, se logra predecir el rendimiento académico de los alumnos de la UNAS con una especificidad=95.42% empleando la técnica Redes Bayesianas.

KNN (Train)

Tabla 56. *Métrica de evaluación (Train) – Especificidad KNN*

Ítem	Métrica	Escala	Fórmula	Especificidad
4	Especificidad	Razón	$Especificidad = \left(\frac{TN}{TN + FP}\right) * 100$	100.00%

Fuente: Elaboración propia

Interpretación: Al hacer uso de técnicas de machine learning, se logra predecir el rendimiento académico de los alumnos de la UNAS con una especificidad=100.00% empleando la técnica KNN.

Tabla 57. *Cuadro comparativo de resultados rendimiento académico de los estudiantes de la UNAS en base a la métrica Especificidad (Train)*

TÉCNICA	RESULTADO (%)
K-NN	100.00%
Árbol de decisión	98.96%
SVM	98.28%
Redes neuronales	97.62%
Redes bayesianas	95.42%

Fuente: Elaboración propia

Interpretación: En la tabla 85 se evidencia que la técnica con mejor resultado en cuanto al indicador especificidad que predice el rendimiento académico de los estudiantes de la UNAS correctamente es “K-NN” con un 100.00%, seguido de la técnica “Árbol de decisión” con un

resultado igual a 98.96%, luego “SVM” con 98.28%, asimismo sigue “Redes neuronales” con 97.62% y por último “Redes bayesianas” con un valor del 95.42%.

Árbol de decisión (Train)

Tabla 58. *Métrica de evaluación (Train)– puntuación F1 árbol de decisión*

Ítem	Métrica	Escala	Fórmula	Puntuación F1
5	Puntuación F1	Razón	$\frac{2 * (Recall * Precisión)}{Recall + Precisión}$	96.88%

Fuente: Elaboración propia

Interpretación: Al hacer uso de técnicas de machine learning, se logra predecir el rendimiento académico de los alumnos de la UNAS con una puntuación F1=96.88% empleando la técnica Árbol de decisión.

Redes Neuronales (Train)

Tabla 59. *Métrica de evaluación (Train)– puntuación F1 Redes Neuronales*

Ítem	Métrica	Escala	Fórmula	Puntuación F1
5	Puntuación F1	Razón	$\frac{2 * (Recall * Precisión)}{Recall + Precisión}$	92.83%

Fuente: Elaboración propia

Interpretación: Al hacer uso de técnicas de machine learning, se logra predecir el rendimiento académico de los alumnos de la UNAS con una puntuación F1=92.83% empleando la técnica Redes Neuronales.

SVM (Train)

Tabla 60. *Métrica de evaluación (Train)– puntuación F1 SVM*

Ítem	Métrica	Escala	Fórmula	Puntuación F1
5	Puntuación F1	Razón	$\frac{2 * (Recall * Precisión)}{Recall + Precisión}$	94.81%

Fuente: Elaboración propia

Interpretación: Al hacer uso de técnicas de machine learning, se logra predecir el rendimiento académico de los alumnos de la UNAS con una puntuación F1=94.81% empleando la técnica SVM.

Redes Bayesianas (Train)

Tabla 61. *Métrica de evaluación (Train)– puntuación F1 Redes Bayesianas*

Ítem	Métrica	Escala	Fórmula	Puntuación F1
5	Puntuación F1	Razón	$\frac{2 * (Recall * Precisión)}{Recall + Precisión}$	86.23%

Fuente: Elaboración propia

Interpretación: Al hacer uso de técnicas de machine learning, se logra predecir el rendimiento académico de los alumnos de la UNAS con una puntuación F1=86.23% empleando la técnica Redes Bayesianas.

KNN (Train)

Tabla 62. *Métrica de evaluación (Train)– puntuación F1 KNN*

Ítem	Métrica	Escala	Fórmula	Puntuación F1
5	Puntuación F1	Razón	$\frac{2 * (Recall * Precisión)}{Recall + Precisión}$	100.00%

Fuente: Elaboración propia

Interpretación: Al hacer uso de técnicas de machine learning, se logra predecir el rendimiento académico de los alumnos de la UNAS con una puntuación F1=100.00% empleando la técnica KNN.

Tabla 63. *Cuadro comparativo de resultados rendimiento académico de los estudiantes de la UNAS en base a la métrica puntuación F1 (Train)*

TÉCNICA	RESULTADO (%)
K-NN	100.00%
Árbol de decisión	96.88%
SVM	94.81%
Redes neuronales	92.83%
Redes bayesianas	86.23%

Fuente: Elaboración propia

Interpretación: En la tabla 91 se evidencia que la técnica con mejor resultado en cuanto al indicador F1-Score que predice el rendimiento académico de los estudiantes de la UNAS correctamente es “K-NN” con un 100.00%, seguido de la técnica “Árbol de decisión” con un resultado igual a 96.88%, luego “SVM” con 94.81%, asimismo sigue “Redes neuronales” con 92.83% y por último “Redes bayesianas” con un valor del 86.23%

Árbol de decisión (Train)

Tabla 64. *Métrica de evaluación (Train)– Curva ROC árbol de decisión*

Ítem	Métrica	Escala	Fórmula	Curva ROC
6	Curva ROC	Razón	1-especificidad= FP/VN+FP	100.00%

Fuente: Elaboración propia

Interpretación: Al hacer uso de técnicas de machine learning, se logra predecir el rendimiento académico de los alumnos de la UNAS con una Curva ROC=100.00% empleando la técnica Árbol de decisión.

Redes Neuronales (Train)

Tabla 65. *Métrica de evaluación (Train)– Curva ROC Redes Neuronales*

Ítem	Métrica	Escala	Fórmula	Curva ROC
6	Curva ROC	Razón	1-especificidad= FP/VN+FP	99.00%

Fuente: Elaboración propia

Interpretación: Al hacer uso de técnicas de machine learning, se logra predecir el rendimiento académico de los alumnos de la UNAS con una Curva ROC=99.00% empleando la técnica Redes Neuronales.

SVM (Train)

Tabla 66. *Métrica de evaluación (Train)– Curva ROC SVM*

Ítem	Métrica	Escala	Fórmula	Curva ROC
6	Curva ROC	Razón	1-especificidad= FP/VN+FP	100.00%

Fuente: Elaboración propia

Interpretación: Al hacer uso de técnicas de machine learning, se logra predecir el rendimiento académico de los alumnos de la UNAS con una Curva ROC=100.00% empleando la técnica SVM.

Redes Bayesianas (Train)

Tabla 67. *Métrica de evaluación (Train)– Curva ROC Redes Bayesianas*

Ítem	Métrica	Escala	Fórmula	Curva ROC
6	Curva ROC	Razón	1-especificidad= FP/VN+FP	97.00%

Fuente: Elaboración propia

Interpretación: Al hacer uso de técnicas de machine learning, se logra predecir el rendimiento académico de los alumnos de la UNAS con una Curva ROC=97.00% empleando la técnica Redes Bayesianas.

KNN (Train)

Tabla 68. *Métrica de evaluación (Train)– Curva ROC KNN*

Ítem	Métrica	Escala	Fórmula	Curva ROC
6	Curva ROC	Razón	1-especificidad= FP/VN+FP	100.00%

Fuente: Elaboración propia

Interpretación: Al hacer uso de técnicas de machine learning, se logra predecir el rendimiento académico de los alumnos de la UNAS con una Curva ROC=100.00% empleando la técnica KNN.

Tabla 69. *Cuadro comparativo de resultados rendimiento académico de los estudiantes de la UNAS en base a la métrica Curva ROC (Train)*

TÉCNICA	RESULTADO (%)
K-NN	100.00%
Árbol de decisión	100.00%
SVM	100.00%
Redes neuronales	99.00%
Redes bayesianas	97.00%

Fuente: Elaboración propia

Interpretación: En la tabla 97 se evidencia que la técnica con mejor resultado en cuanto al indicador Curva ROC que predice el rendimiento académico de los estudiantes de la UNAS correctamente son las técnicas: “Árbol de decisión”, “SVM” y “K-NN” con un 100.00%, seguido de la técnica “Redes neuronales” con 99.00% y por último “Redes bayesianas” con un valor del 97.00%.

Anexo 6

Ficha de Registro

	Apellidos	Nombres	Apellidos y Nombres	Género	Departamento	Provincia	Distrito	Fecha de Nacimiento	Edad	Estado civil	Código del alumno	Beneficiario de Pronabec	Modalidad de ingreso	Tipo de Colegio de Procedencia	Escuela profesional	Año de ingreso	Ciclo académico actual	total de Créditos llevados	Créditos desaprobados	Créditos aprobados	Promedio ponderado anterior	Promedio ponderado final	Plan de estudio	Total de cursos matriculados	Total de cursos de nivelación llevados

Anexo 7

Solicitud de reporte académico de los estudiantes de la UNAS

SOLICITO REPORTE ACADÉMICO DE LOS ESTUDIANTES DE LA UNAS

Lima 14 de Julio del 2022

SRA.: OLIVIA PULGAR TAPIA
DIRECTORA DE LA OFICINA DE ASUNTOS ACADÉMICOS DE LA UNAS

Yo: **MARIA LUCI ZAMORA HERNÁNDEZ**, con DNI **43655298**, bachiller de la facultad de Informática y sistemas de la UNAS, me dirijo a usted con el debido respeto y expongo lo siguiente:

Que siendo necesaria contar con el reporte académico de los estudiantes de la Universidad Nacional Agraria de la Selva, correspondiente al periodo **2012-I** al **2021-II** para la elaboración demi tesis, con los siguientes datos:

ITEM	CAMPOS
1	Apellidos
2	Nombres
3	Apellidos y Nombres
4	Género
5	Departamento
6	Provincia
7	Distrito
8	Fecha de Nacimiento
9	Edad
10	Estado civil
11	Código del alumno
12	Beneficiario de Pronabec
13	Modalidad de ingreso

ITEM	CAMPOS
14	Tipo de Colegio de Procedencia
15	Escuela profesional
16	Año de ingreso
17	Ciclo académico actual
18	Total, de Créditos llevados
19	Créditos desaprobados
20	Créditos aprobados
21	Promedio ponderado anterior
22	Promedio ponderado final
23	Plan de estudio
24	Total de cursos matriculados
25	Total de cursos de nivelación llevados

Solicito a usted señora directora que tenga la bondad de asignar a quien corresponda para que me brinde lo solicitado.

Por lo expuesto
Le ruego a usted acceder a mi solicitud. Anticipo mis sinceros agradecimientos.

Maria Luci Zamora Hernandez
DNI: 43655298

Anexo 8

Respuesta a la Solicitud de reporte académico de los estudiantes de la UNAS

El mar, 6 sept 2022 a las 9:12, Jorge Luis Jara Linares - DICDA (<jorge.jara@unas.edu.pe>) escribió:

Buenos días, hago entrega de la información solicitada según el formato proporcionado

Saludos

Jorge Luis Jara Linares
Especialista de Sistemas
Resp. Área de Soporte Informático
DIRECCIÓN DE ASUNTOS ACADÉMICOS
UNIVERSIDAD NACIONAL AGRARIA DE LA SELVA

jorge.jara@unas.edu.pe
+51 961613274
@jorlujarlin

...

[Mensaje recortado] Ver todo el mensaje

1 archivo adjunto · Analizado por Gmail

Reporte_Luci.xlsx