UNIVERSIDAD CÉSAR VALLEJO

FACULTAD DE INGENIERÍA Y ARQUITECTURA

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

# Sistema inteligente con Machine Learning basado en selección de variables para predecir el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana”

**TESIS PARA OBTENER EL TÍTULO PROFESIONAL DE:**

**Ingeniero de Sistemas**


## AUTORES:

Huertas Fabian, Nicole Valery (orcid.org/0009-0001-0120-6292)

Salcedo Zapata, Juan Antonio (orcid.org/0000-0003-4979-5708)

## ASESOR:

Dr. Daza Vergaray, Alfredo (orcid.org/0000-0002-2259-1070)


## LÍNEA DE INVESTIGACIÓN:

Sistema de Información y Comunicaciones

## LÍNEA DE RESPONSABILIDAD UNIVERSITARIA:

Desarrollo económico, empleo y emprendimiento

**LIMA - PERÚ**

**2023**

## Dedicatoria

A mis abuelitos, Esperanza, Zenobia y Gabriel por ser una constante motivación de superación para cumplir con todas mis metas.

A mis padres, Gilmer y Lizzet, por brindarme su amor, sabiduría y apoyo incondicional desde siempre.

Ustedes son los más valioso que tengo y son mi inspiración para seguir adelante.

**Nicole**

A mis padres, Rosa y Simón,

Gracias por ser mi mayor apoyo y motivación, siempre han estado ahí para mí, en las buenas y en las malas.

Su amor, su comprensión y su aliento me han ayudado a ser la persona que soy hoy, son mi mayor orgullo y mi mejor ejemplo para seguir.

**Juan**

## Agradecimiento

A la I.E.P Nuestra Señora de Copacabana por permitirme utilizar sus datos para el desarrollo de este trabajo.
A mi asesor, el Dr. Daza Vergaray Alfredo, por su orientación y apoyo durante todo el proceso de investigación.
Finalmente, quiero agradecer a Diego por su ayuda para que la empresa aceptara participar en este proyecto.

**Nicole**

A mi asesor, Dr. Daza Vergaray Alfredo,
Gracias por su guía, apoyo y paciencia inquebrantable durante este proyecto. Su conocimiento y experiencia han sido invaluables para la elaboración de esta investigación.

**Juan**

# Índice de Contenidos

# Índice de Tablas

# Índice de Figuras

# Índice de Ecuaciones

# Resumen

Esta investigación tiene como objetivo general desarrollar un Sistema Inteligente con Machine Learning basado en selección de variables para predecir el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana”. La tesis es de tipo aplicada y como población tuvo un total de 256 estudiantes del nivel secundaria, donde se realizó un muestreo no probabilístico y para la recolección de datos se utilizó el cuestionario. Para el desarrollo del estudio se empleó las siguientes etapas de la metodología KDD: La primera etapa que es la selección de datos, se realizó la recopilación de la información de los estudiantes, en la segunda etapa, se llevó a cabo el preprocesamiento de los datos, realizando la limpieza y el análisis exploratorio, en la tercera etapa que es la transformación de los datos, se hizo la asignación de valor numérico y exportación en la base de datos, en la cuarta etapa se realizó la minería de datos en base al entrenamiento del modelo con diez algoritmos de aprendizaje automático (DT, RF, SVM, ANN, GBM, MP, K-NN, NB, LR y AB). Finalmente, en la quinta etapa, se hizo la interpretación de los datos y el diseñó del sistema inteligente para predecir el rendimiento académico en base a las métricas de evaluación (especificidad, precisión, sensibilidad, exactitud y F1-Score). Dados que los resultados de los algoritmos DT, SVM, GBM y ADA, obtuvieron con un 100% en todas las métricas, se concluye que el sistema realiza la predicción del rendimiento académico de forma muy precisa.

**Palabras Clave:** Sistema Inteligente, Maching Learning (ML), selección de variables, predicción, rendimiento académico.

# Abstract

The general objective of this research is to develop an Intelligent System with Machine Learning based on variable selection to predict the academic performance of the I.E.P. "Nuestra Señora de Copacabana". The thesis is of applied type and as population had a total of 256 students of the secondary level, where a non-probabilistic sampling was performed and for data collection the questionnaire was used. For the development of the study, the following stages of the KDD methodology were used: The first stage which is the data selection, the collection of the students' information was performed, in the second stage, the data preprocessing was carried out, performing the cleaning and exploratory analysis, in the third stage which is the data transformation, the assignment of numerical value and export in the database was done, in the fourth stage data mining was performed based on the training of the model with ten machine learning algorithms (DT, RF, SVM, ANN, GBM, MP, K-NN, NB, LR and AB). Finally, in the fifth stage, the interpretation of the data and the design of the intelligent system to predict academic performance based on the evaluation metrics (specificity, precision, sensitivity, accuracy and F1-Score) were performed. Given that the results of the DT, SVM, GBM and ADA algorithms, obtained with 100% in all metrics, it is concluded that the system performs the prediction of academic performance in an accurate way.



**Keywords:** Intelligent System, Maching Learning (ML), variable selection, prediction, academic performance.

# I. INTRODUCCIÓN

En este primer capítulo, se desarrolla la realidad problemática a nivel internacional, en América Latina, a nivel nacional y luego en la institución educativa. Posteriormente, se detalla un caso de éxito, la propuesta de solución, las preguntas de investigación, la justificación, los objetivos e hipótesis.

En este contexto, las instituciones educativas se encuentran lidiando en la transición de los estudiantes desde la antigua modalidad presencial, dada la disminución del control durante la etapa remota. Esta situación ha tenido un impacto adverso en el rendimiento académico de los estudiantes, generando un desafío actual que requiere medidas efectivas para mejorar y restablecer la disciplina en el proceso de aprendizaje.

Al respecto, Estrada (2018) menciona que, aunque la educación experimenta cambios a lo largo del tiempo, persisten las prácticas tradicionales en la forma de enseñar, utilizando técnicas y estrategias que siguen el paradigma tradicionalista. Además, subraya la importancia del interés tanto del docente por enseñar como del alumno por aprender como un factor importante para determinar el rendimiento académico (p. 2). A partir de esta reflexión, surge el conflicto relacionado con el bajo rendimiento académico, ya que la falta de innovación en los métodos de enseñanza conduce a un estancamiento en el proceso educativo. Al mantenerse en lo tradicional, aunque pueda proporcionar resultados aceptables, resulta insuficiente para generar un cambio significativo a largo plazo.

Con ello, **a nivel internacional**, en Estados Unidos, el Centro Nacional de Estadísticas Educativas (NCEA), entidad líder en recopilar y analizar datos educativos en el país, reveló que hasta el año 2022, los estudiantes de 9 años experimentaron una disminución en sus promedios de lectura y matemáticas, con una reducción de 5 y 7 puntos, respectivamente, en comparación con los resultados de 2020. Este declive, señalado como el peor rendimiento en dos décadas según la British Broadcasting Corporation News Mundo (2022, párr. 9). Por tal motivo, se deduce que el cambio en el rendimiento académico se vio afectado por el entorno de aprendizaje virtual en ese intervalo de años, siendo esta una causa principal de la caída en el rendimiento académico general de todos los países. Asimismo, cabe resaltar que Estados Unidos ya anticipaba resultados desfavorables debido a los cambios derivados de la pandemia. En respuesta, han implementado un plan de

estudios para mejorar el rendimiento en el presente año, reconociendo la necesidad de adaptarse y mitigar los efectos negativos del aprendizaje virtual en el ámbito académico.

**En América Latina**, López et al. (2022) manifiesta una semejanza leve en el rendimiento académico de los estudiantes colombianos. La evaluación abarcó a 343 alumnos de niveles primario y secundario, asignándoles calificaciones en una escala de 0 a 10, con resultados que oscilaron entre un límite superior de 6,80 y un límite inferior de 6,44. Aunque estos datos sugieren una relativa homogeneidad en el rendimiento, es crucial subrayar que la consideración de un rendimiento académico óptimo se sitúa en un valor igual o superior a ocho. En este sentido, se destaca que la diferencia observada en el rendimiento académico no supera un porcentaje significativo. No obstante, para alcanzar un rendimiento que se clasifique como bueno, los resultados deben situarse en niveles iguales o superiores a ocho. Este umbral resalta la importancia de no solo observar la uniformidad superficial, sino también considerar las metas establecidas para la excelencia académica. Estos hallazgos, respaldados por la evaluación de un número considerable de estudiantes, ofrecen una visión más completa del panorama educativo en Colombia. Aunque la uniformidad podría interpretarse como un indicador positivo, la atención a los estándares de rendimiento más elevados sigue siendo esencial para asegurar la calidad educativa y la preparación académica de los estudiantes en el país.

Por otro lado, **a nivel nacional**, en la evaluación muestral (EM) de 396 mil estudiantes de 2°, 4° y 6° de primaria, y 2° de secundaria, llevada a cabo entre noviembre y diciembre de 2022 durante el retorno a la presencialidad, se evaluaron en los siguientes cursos: Ciencia y Tecnología, Matemática y Lectura, revelando una disminución en los resultados de aprendizaje en comparación con el año 2019 en la mayoría de las materias impartidas en los centros de estudios. En detalle, en el curso de Matemáticas experimentó un impacto significativo; en 2° de primaria, solo el 11,8% de los estudiantes lograron alcanzar un nivel satisfactorio, representando una disminución del 5,12% en comparación con los resultados obtenidos en el año 2019. De manera similar, en 4° de primaria, el 23,3% alcanzó el nivel satisfactorio, mostrando una disminución del 10,7% en relación con los

resultados de 2019. En el caso de 2° de secundaria, el 12,7% alcanzó el nivel satisfactorio, registrando una disminución del 5% en comparación con los resultados del año anterior. Estas cifras reflejan una tendencia generalizada a la baja en el rendimiento académico en estas etapas específicas. En el área de Ciencia y Tecnología, evaluada solo en 2° de secundaria, se observó una disminución de 2 puntos en el rendimiento académico, obteniendo 499 en comparación con la media promedio del año anterior. Por último, en el área de Lectura, la situación varía: en 2° de primaria, el 37,6% alcanzó el nivel satisfactorio, mientras que en 4° de primaria, el rendimiento bajó al 30%. En 2° de secundaria, se observaron mejoras con un 19,1% en el nivel satisfactorio, un aumento del 4,6% en comparación con 2019. Respecto a 6° de primaria, el 25,2% y el 15% de los estudiantes alcanzaron el nivel satisfactorio en las áreas de Lectura y Matemáticas, respectivamente (Ministerio de Educación del Perú, 2023, pár. 1 – 19).

Es así como, los resultados obtenidos demuestran una proyección de cómo está el rendimiento académico en el Perú, ya sea, en instituciones públicas o privadas, lo que permite tener una mejor perspectiva sobre los puntos a mejorar en los años posteriores.

En referente a la I.E.P “Nuestra Señora de Copacabana”, tras realizar entrevistas con los docentes, escuchar las opiniones del director y realizar observaciones continuas al entorno académico, se pudo determinar que el rendimiento académico ha disminuido mucho en comparación a años anteriores, este declive se atribuye principalmente a la normalización de la falta de disciplina entre los alumnos, a los desafíos familiares que enfrentan y a la carencia de herramientas tecnológicas para proporcionar el apoyo necesario a los alumnos con dificultades. De igual importancia es la distribución aleatoria de los alumnos, que está resultando contraproducente al mezclar a aquellos estudiantes dedicados con aquellos sin motivación para estudiar. Además, el área de psicología, encargada del bienestar mental y emocional de los estudiantes, ha experimentado cambios en su personal, lo que significa que están en la fase inicial de conocer a los alumnos que presentan problemas. Estas causas han tenido un impacto negativo en el rendimiento académico de los estudiantes, lo que ha reducido sus oportunidades

de continuar sus estudios en las universidades y ha incrementado su probabilidad de involucrarse en otro tipo de actividades. Véase el ANEXO 7.

En relación con la evaluación del rendimiento académico de los estudiantes en la institución, se establece el siguiente: Las evaluaciones se llevan a cabo a través de exámenes semanales de avance académico (ESAA), exámenes de simulacro tipo admisión y exámenes bimestrales, que contribuyen al 20%, 10% y 20% respectivamente de la calificación final. Como resultado, el conjunto de actividades realizadas durante las horas de clase, que incluye el trabajo en el cuaderno y libro, así como la participación y prácticas, constituye el 50% restante de la evaluación final. Véase el ANEXO 9.

Rajendran, et al. (2022), en su estudio “Predecir el rendimiento académico de estudiantes de secundaria y preparatoria utilizando algoritmos de aprendizaje automático”, se basó en algoritmos de aprendizaje automático (LR, ANN, RF, GBM, Apilamiento) para predecir el rendimiento académico de estudiantes de secundaria y preparatoria. En base al modelo de rendimiento “Bajo”, obtuvieron los siguientes resultados: LR obtuvo una precisión de 34.96%, recall 38.14% y F1 – score 36.48%, ANN obtuvo una precisión de 54.72%, recall 55.67% y F1 – score 55.19%, RF obtuvo una precisión de 79.07%, recall 80.89% y F1 – score 79.97%, GBM obtuvo una precisión de 63.28%, recall 67.69% y F1 – score 65.41% y el algoritmo de apilamiento obtuvo una precisión de 78.03%, recall 77.68% y F1 – score 77.85%, en cuanto al modelo de rendimiento “Moderado”, LR obtuvo una precisión de 55.35%, recall 53.96% y F1 – score 54.65%, ANN obtuvo una precisión de 52.75%, recall 45.58% y F1 – score 58.90%, RF obtuvo una precisión de 76.11%, recall 72.26% y F1 – score 74.13%, GBM obtuvo una precisión de 72.19%, recall 57.26% y F1 – score 63.86% y el algoritmo de apilamiento obtuvo una precisión de 77.28%, recall 72.82% y F1 – score 74.98%, Finalmente en el modelo de rendimiento “Alto”, LR obtuvo una precisión de 57.91%, recall 55.52% y F1 – score 56.69%, ANN obtuvo una precisión de 49.76%, recall 60.50% y F1 – score 54.61%, RF obtuvo una precisión de 69.65%, recall 73.10% y F1 – score 71.33%, GBM obtuvo una precisión de 54.94%, recall 69.80% y F1 – score 61.49% y el algoritmo de apilamiento obtuvo una precisión de 71.37%, recall 79.12% y F1 – score 75.05%. La conclusión del estudio indica que el algoritmo de apilamiento demostró la mayor

precisión, recuperación y puntuación F1 en general, seguido por el aumento de gradiente, el bosque aleatorio, la red neuronal artificial y, por último, el algoritmo de regresión logística. Estos hallazgos sugieren que una variedad de factores, incluyendo tanto variables sociodemográficas como variables relacionadas con el estilo de vida, pueden influir en el rendimiento académico de los estudiantes.

En este sentido, se propone la implementación de un Sistema Inteligente con Machine Learning basado en la selección de variables, con el objetivo de predecir de manera individual el rendimiento de cada alumno. Esta aproximación posibilitaría la clasificación de los estudiantes en diferentes categorías, teniendo en cuenta sus capacidades, actitudes y aptitudes. Además, la aplicación de este sistema permitiría una visión más precisa de la realidad académica de los alumnos, facilitando así la formulación de un marco de trabajo y un plan académico adaptado a las necesidades.

Por tal motivo, se ha generado la siguiente **pregunta de investigación** ante la problemática general: ¿Cómo un Sistema Inteligente con Machine Learning basado en selección de variables permitirá predecir el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana”? y como preguntas que responden a los problemas específicos, en primer lugar: ¿Cómo un Sistema Inteligente con Machine Learning basado en selección de variables permitirá predecir con especificidad el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana”?, en segundo lugar: ¿Cómo un Sistema Inteligente con Machine Learning basado en selección de variables permitirá predecir con precisión el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana”?; en tercer lugar: ¿Cómo un Sistema Inteligente con Machine Learning basado en selección de variables permitirá predecir con sensibilidad el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana”?; en cuarto lugar: ¿Cómo un Sistema Inteligente con Machine Learning basado en selección de variables permitirá predecir con exactitud el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana”?; finalmente: ¿Cómo un Sistema Inteligente con Machine Learning basado en selección de variables permitirá predecir con F1-score el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana”?

Frente a lo mencionado, el presente trabajo cuenta con una justificación teórica, debido a que está respaldado por artículos científicos, revistas indexadas y fuentes de búsqueda de información confiables como lo es el MyLOFT. Además de basarse en las siguientes teorías: inteligencia artificial, donde sostiene que las máquinas pueden aprender y adaptarse a su entorno; teoría de la predicción, donde sostiene que es posible predecir el comportamiento futuro a partir de datos históricos; y la teoría de la selección de variables, donde sostiene que la selección de variables puede mejorar la precisión de las predicciones. Asimismo, presenta una justificación social, porque fomenta el apoyo a los jóvenes estudiantes al predecir el rendimiento académico, identificando a los que requerirán mayor apoyo por parte del docente. Adicionalmente, se justifica de forma tecnológica por el desarrollo de un sistema de predicción que abre camino a una educación modernizada, esto en busca de generar algún cambio que mejore el rendimiento académico del país. Por último, brinda facilidad a la hora de clasificar a los alumnos y permite generar un plan de trabajo adecuado para el desarrollo de los jóvenes.

En base a todo lo investigado, se planteó el siguiente **objetivo general**: Desarrollar un Sistema Inteligente con Machine Learning basado en selección de variables para predecir el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana”, y como objetivos específicos; en primer lugar: Desarrollar un Sistema Inteligente con Machine Learning basado en selección de variables para predecir con especificidad el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana”; en segundo lugar: Desarrollar un Sistema Inteligente con Machine Learning basado en selección de variables para predecir con precisión el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana”; en tercer lugar: Desarrollar un Sistema Inteligente con Machine Learning basado en selección de variables para predecir con sensibilidad el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana”; en cuarto lugar: Desarrollar un Sistema Inteligente con Machine Learning basado en selección de variables para predecir con exactitud el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana”; finalmente: Desarrollar un Sistema Inteligente con Machine Learning basado en selección de variables para predecir con F1-score el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana”.

Por ende, se establece como **hipótesis general**: El desarrollo de un Sistema Inteligente con Machine Learning basado en selección de variables predice el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana”, y como hipótesis específicas; en primer lugar: El desarrollo de un Sistema Inteligente con Machine Learning basado en selección de variables predice con especificidad el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana”; en segundo lugar: El desarrollo de un Sistema Inteligente con Machine Learning basado en selección de variables predice con precisión el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana”; en tercer lugar: El desarrollo de un Sistema Inteligente con Machine Learning basado en selección de variables predice con sensibilidad el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana”; en cuarto lugar: El desarrollo de un Sistema Inteligente con Machine Learning basado en selección de variables predice con exactitud el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana”; finalmente: El desarrollo de un Sistema Inteligente con Machine Learning basado en selección de variables predice con F1-score el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana”.

# II. MARCO TEÓRICO

En este capítulo, se ha llevado a cabo la búsqueda de fuentes de información, abarcando antecedentes a nivel nacional e internacional (véase el ANEXO 3), incorporando teorías y bases conceptuales relevantes (véase el ANEXO 4). A continuación, se presenta una lista detallada de los términos y conceptos claves que sustentarán el desarrollo de la investigación, proporcionando una visión clara de que serán empleados en este capítulo y en el contenido de la investigación.

**Tabla N° 1: Diccionario de Términos**

| | |
|---|---|
| **KDD** | Descubrimiento de conocimiento en bases de datos o Knowledge Discovery in Databases. |
| **DT** | Árbol de decisión o Decision Tree. |
| **RF** | Bosque aleatorio o Random Forest. |
| **RT** | Árbol aleatorio o Random Tree. |
| **NB** | Bayes ingenuos o Naïve Bayes. |
| **SVM** | Máquina de vectores de soporte o Support Vector Machines. |
| **KNN** | K-vecinos más cercanos o K-Nearest Neighbors. |
| **MP** | Perceptrón múltiple o Multi Perceptron. |
| **LB** | LogitBoost. |
| **GBM** | Máquina de aumento de gradiente o Gradient Boosting Machine. |
| **SGD** | Descenso del gradiente estocástico o Stochastic Gradient Descent. |
| **ROC CURVE** | Curva de características de operación del receptor o Receiver Operating Characteristic Curve. |
| **SLD** | Discapacidad de aprendizaje específica o Specific Learning Disability. |
| **ML** | Aprendizaje automático o Machine Learning. |
| **ADA** | Impulso Adaptativo o Adaptive Boosting. |
| **ANN** | Red Neuronal Artificial o Artificial Neural Network. |
| **LR** | Regresión Logística o Logistic Regression. |
| **CNN** | Red neuronal convolucional o Convolutional Neural Network. |
| **AUC** | Área bajo la curva o Area Under the Curve. |
| **MCC** | Coeficiente de correlación de Matthews o Matthews Correlation Coefficient. |
| **XGB** | Refuerzo de gradientes extremo o Extreme Gradient Boosting. |

| | |
|---|---|
| **RFE** | Eliminación Recursiva de Características o Recursive Feature Elimination. |
| **PCA** | Análisis de Componentes Principales o Principal Component Analysis. |
| **SPSS** | Paquete Estadístico para Ciencias Sociales o Statistical Package for Social Sciences. |
| **IDE** | Entorno de Desarrollo Integrado o Integrated Development Environment. |
| **SSMS** | Estudio de administración de SQL Server o SQL Server Management Studio. |
| **PART** | Búsqueda por partición o Partition Search. |
| **ODBC** | Conectividad abierta de base de datos o Open Database Connectivity. |

**Fuente: Elaboración Propia**

Según, Orsoni et al. (2023), indican que su investigación consistió en identificar la solución óptima de agrupamiento mediante la comparación de dos métodos basados en funciones cognitivas más distintivas entre estudiantes con SLD. Además, se propusieron desarrollar y validar un modelo de Machine Learning (ML) utilizando los algoritmos ADA y ANN. El objetivo era determinar si se podían predecir los perfiles cognitivos de 292 estudiantes mediante métricas de evaluación como exactitud y F1-score. Para lo cual, la investigación, llevada a cabo en la región de Emilia-Romaña en Italia, incluyó la selección de participantes de varias escuelas, quienes fueron sometidos a pruebas estandarizadas y a un juego digital en línea para evaluar sus funciones cognitivas. Asimismo, empleó una metodología basada en dos niveles: en el primer nivel, se utilizó un algoritmo de autoorganización de mapas de Kohonen y en el segundo nivel, se usó el algoritmo de agrupación K-medias. Los resultados revelaron que ADA logró una exactitud del 77.2% y un F1-score del 89.9%, mientras que ANN alcanzó una exactitud del 91.7% y un F1-score del 91.6%. Estos hallazgos respaldan la capacidad del modelo de ML desarrollado para predecir con precisión los perfiles cognitivos de los estudiantes. De este modo, se obtuvo que el modelo de ML desarrollado fue capaz de predecir con precisión los perfiles cognitivos de los estudiantes. Por lo que, estos hallazgos sugieren que el uso de modelos de aprendizaje automático puede ser una herramienta útil para

identificar y comprender las diferencias cognitivas entre estudiantes con SLD, lo que podría ayudar a desarrollar estrategias educativas más personalizadas y efectivas. (p. 1 - 2).

Igualmente, Kannan, Abarna y Vairachilai (2023), buscaron mejorar el antiquisimo sistema educativo de la India para evitar el abandono académico de los estudiantes de grado superior, siendo un total de 4424 estudiantes, proponiendo así el desarrollo de un modelo de ML que emplee los siguientes algoritmos: KNN, DT, RF, SVM, ANN, NB, GBM y ADA que fueron medidos a través de la métrica de exactitud. La finalidad es conseguir una predicción del rendimiento académico y así poder brindar apoyo a los alumnos. Para lo cual, se utilizaron cuatro enfoques: la recopilación de datos, la selección de características y procesamiento, el análisis de datos y la clasificación de los alumnos; aplicando técnicas de ingeniería, validación cruzada k-fold y algoritmos de predicción respectivamente, con una rigurosa evaluación para la fase final. Como resultado, KNN obtuvo una exactitud de 70.12%, DT una exactitud de 74.17%, RF una exactitud de 80.55%, SVM una exactitud de 71.42%, ANN una exactitud de 80.23%, NB una exactitud de 66.23%, GBM una exactitud de 79.05% y ADA una exactitud de 75.74%. Es así como, se pudo crear un modelo preciso que facilitaba la predicción del rendimiento académico, que contribuyó en la toma de decisiones tempranas de algunas instituciones educativas de enseñanza superior. (p. 1 - 4)

Por otro lado, Al-Alawi et al. (2022), buscaron identificar y analizar los factores que afectan el rendimiento académico de los estudiantes universitarios, especialmente aquellos que se encuentran en situación de bajo rendimiento académico. Para esto, usaron la metodología KDD, que es ampliamente utilizada por los investigadores científicos en el campo de la minería de datos, este proceso iterativo consta de ocho fases: especificación del problema, obtención de recursos, limpieza de datos, preprocesamiento, minería de datos, evaluación, interpretación y explotación. La muestra aplicada fue de 6514 estudiantes universitarios, haciendo uso de los algoritmos individuales: DT, RF, RT, NB, SVM, KNN y MP; y algoritmos en conjunto: LB, Vote y Bagging que fueron evaluados a través de la métrica de exactitud. Por consiguiente, se determinó que los principales factores que afectan el rendimiento académico de los estudiantes universitarios incluyen la duración del

estudio en la universidad y el rendimiento previo en la escuela secundaria. Además, se encontró que los algoritmos de aprendizaje automático empleados son muy efectivos para identificar patrones ocultos en los datos y predecir el rendimiento académico de los estudiantes con una alta precision, siendo los resultados los siguientes: DT obtuvo una exactitud de 82.4%, RF una exactitud de 78.3%, RT una exactitud de 76.1%, NB una exactitud de 71.4%, MP una exactitud de 80.9%, SVM una exactitud de 70.8%, KNN una exactitud de 76.1%, Vote una exactitud de 81.3%, LB una exactitud de 80.7% y Bagging una exactitud de 82.2%. En conclusión, el algoritmo de Decisión Tree  es el óptimo para su uso comparado a los demás. De igual manera, los resultados experimentales revelaron que la duración del estudio, el puntaje de la escuela secundaria y el género son los principales factores identificados que resultaron en que los estudiantes universitarios quedaran bajo período de prueba (p. 15).

En relación, Wang et al. (2022), manifiestan que su objetivo es analizar la oscilación del rendimiento académico de estudiantes ordinarios chinos y desarrollar un modelo de predicción certero. Por consiguiente, se emplearon cuestionarios para el proceso de recopilación de datos y se aplicó la prueba de Chi-Cuadrado para el análisis e identificación de los factores principales. Respecto a la metodología empleada, esta se basó en la recopilación de los datos, preprocesamiento, selección de características, construcción del modelo, evaluación de modelo y análisis de los resultados. Gracias a esto, se propusieron cuatro modelos de ML, haciendo uso de los algoritmos de LR, SVM, NB y RF, y métricas de evaluación como exactitud, precisión y sensibilidad, dónde se identificaron características únicas de los alumnos con un bajo rendimiento. Siendo los resultados los siguientes: LR con una exactitud de 85.04%, una precisión de 81.61% y una sensibilidad de 77.78%, NB con una exactitud de 80.49%, una precisión de 73.57% y una sensibilidad de 76.27%, RF con una exactitud de 83.71%, una precisión de 78.42% y una sensibilidad de 75.76%. Por lo tanto, se pudo determinar que el modelo de SVM es el mejor y más preciso para la predicción del rendimiento académico siendo la tasa promedio de exactitud: 86.18%, la tasa promedio de precisión: 80,96% y la tasa promedio de sensibilidad: 82.83%.

Además, los datos recopilados se podrían utilizar para el desarrollo de medidas coherentes para el apoyo a los estudiantes. (p. 1)

De forma similar, Owusu-Boadu et al. (2021) mencionaron que el objetivo del estudio fue examinar el grado en que las características cognitivas y no cognitivas influyen en el rendimiento académico de los estudiantes de escuelas de segundo ciclo utilizando algoritmos de ML como el DT, KNN, ANN, LR, RF, ADA y SVM que fueron evaluados con las siguientes métricas: AUC y exactitud. De igual manera, comparar y evaluar el rendimiento predictivo de diferentes métodos de clasificación para predecir el rendimiento. Asimismo, la metodología utilizada incluyó la obtención de 480 registros con dieciséis características, el preprocesamiento del conjunto de datos y la codificación numérica. Finalmente, los resultados obtenidos fueron: RF con una exactitud de 77.1% y un AUC de 90.3%, LR con una exactitud de 77.9% y un AUC de 90%, ANN con una exactitud de 76% y un AUC de 89.5% superando a los algoritmos de KNN con una exactitud de 63.8% y un AUC de 82.6%, SVM con una exactitud de 72.7% y un AUC de 80%, DT con una exactitud de 73.3% y un AUC de 87.6% y ADA con una exactitud de 74.8% y un AUC de 80.8%. Además de que las características cognitivas como las no cognitivas son importantes predictores del rendimiento académico, y que los modelos de aprendizaje automático pueden ser efectivos para dicha predicción, por lo que este enfoque puede ser útil para mejorar la toma de decisiones educativas y ayudar a identificar a los estudiantes que necesitan apoyo adicional. (p. 12).

De igual importancia, García (2021), mencionó en su tesis para obtener el título profesional de Ingeniero de Sistemas, que su objetivo era desarrollar un modelo de ML para predecir el rendimiento académico, aprovechando la metodología KDD y herramientas como SPSS statistic y modeler. Además, buscaba identificar que tan exacto, sensible y específico puede ser el ML para la predicción del rendimiento académico. Para lo cual, se realizó una recolección de datos de la población establecida por 82 alumnos de ingeniería de sistemas, para luego utilizar la metodología KDD y dejar los datos listos para aplicar técnicas de ML, considerando utilizar los algoritmos de SVM, DT y KNN y así buscar la validación final con una matriz de confusión y el coeficiente Kappa de Cohen. Los resultados fueron los siguientes: SVM con una exactitud, sensibilidad y especificidad de 100%,

DT con una exactitud de 89.51%, una sensibilidad de 85.05% y una especificidad de 91.92%, KNN con una exactitud de 79.75%, una sensibilidad de 72.41% y una especificidad de 84%. Por consiguiente, se determinó que SVM puede realizar predicciones con todos los criterios mencionados y se verificó la muy buena concordancia de los datos gracias al índice de Kappa de Cohen. (p. 26 – 41)

Además, los autores Ojajuni et al. (2021) tienen como finalidad desarrollar un modelo de ML que clasifique y prediga el éxito académico de los estudiantes basándose en datos históricos, así como también identificar los factores clave que afectan el desempeño académico. Para su desarrollo, se empleó el uso de algoritmos de ML supervisados como: DT, RF, SVM, LR, ADA, XGBoost y CNN que serán medidos según su exactitud. Los materiales y métodos usados se basan en: primero, definir las herramientas, Python junto con las bibliotecas Scikit-learn y TensorFlow; segundo, establecer el conjunto de datos, rendimiento académico de 1044 estudiantes; tercero, preprocesamiento de datos e ingeniería de funciones; cuarto, modelo de clasificación de ML; quinto, evaluación del desempeño del modelo de ML, el conjunto de entrenamiento está compuesto por un 70% y el conjunto de prueba en un 30%. Los resultados fueron los siguientes: DR con una exactitud de 47.95%, RF con una exactitud de 92.60%, SVM con una exactitud de 42.88%, LR con una exactitud de 40.96%, ADA con una exactitud de 35.75%, SGD con una exactitud de 33.69%, XGBoost con una exactitud de 97.12%, CNN con una exactitud de 72.74%. Estos resultados demuestran que XGBoost puede predecir el rendimiento académico con una mayor exactitud. Además, se demostró que las características sociales y demográficas afectan el éxito académico de los estudiantes. Es así como, concluyen que la aplicación de ML en el aula ayudará a los educadores a identificar a estudiantes con bajo rendimiento para tomar las acciones necesarias. (p. 482 – 489).

Por otra parte, Dinh et al. (2020), tienen como objetivo de su estudio emplear las técnicas de ML para predecir el promedio de calificaciones final de los estudiantes en función de las características personales (género y lugar de residencia), puntajes de ingreso a la universidad, año sabático y su desempeño del primer y segundo año. Los datos son recopilados combinando la información de una encuesta con los datos del sistema de información de gestión estudiantil de la

universidad para un total de 525 estudiantes. Para ello, emplea una metodología que consta en: recopilar e integrar los datos, preprocesar los datos, y construir y evaluar el modelo. Mismo que es medido por la métrica de exactitud para las técnicas de ML como: NB, SMO, MP, DT, RF, RT, PART y OneR, teniendo en cuenta que se considera el 70% como conjunto de entrenamiento y 30% como conjunto de prueba. Los resultados fueron los siguientes: NB y MP con una exactitud de 86.19%, SMO con una exactitud de 85.64%, DT con una exactitud de 73.48%, RF con una exactitud de 80.70%, RT con una exactitud de 77.90%, PART con una exactitud de 74.59% y OneR con una exactitud de 76.24%. En conclusión, se puede observar como el algoritmo de NB y MP obtuvieron una mayor exactitud y a su vez una mayor tasa de TP para la mayoría de las clases. (p. 23 – 26).

De igual manera, Urkude y Gupta (2019), indican que el objetivo de su investigación es desarrollar un modelo de ML para predecir la probabilidad de que un estudiante pase un curso y determinar si era necesario una intervención, con la intención de mejorar la calidad del sistema educativo y predecir el rendimiento académico de los estudiantes. Para ello, se empleó una metodología que consta en los siguientes pasos: recopilación de datos, preprocesamiento, construcción del modelo y evaluación. Es así como, se utilizaron tres algoritmos de aprendizaje supervisado: SVM, DT y NV para predecir el número de estudiantes que obtendrían éxito en un curso y evaluar la calidad del método de enseñanza-aprendizaje. Por ello, se utiliza la métrica de evaluación de modelo, F1-score, para evaluar la precisión de las predicciones. Donde, se pudo demostrar que, para una muestra de 300 estudiantes, el SVM obtuvo un F1-score de 78.38%, el DT un F1-score de 66.13%, el NV un F1-score de 76.34%. Además, se menciona que los resultados obtenidos pueden ser utilizados por las instituciones educativas para mejorar los resultados del curso y comprender mejor los patrones de inscripción de los estudiantes. (p. 1 - 4)

También, Mounika y Persis (2019), tienen como objetivo desarrollar un modelo de predicción del desempeño académico de los estudiantes, utilizando métodos de clasificación como: KNN, SVM, DT, RF y GBM que fueron evaluados por la métrica de exactitud. Para su desarrollo, utilizaron una metodología de ML supervisado basada en las siguientes fases: carga del conjunto, cálculo de la

media, separación de datos, resumen de los datos y hacer predicciones. En resumen, el algoritmo se entrena en un conjunto de datos etiquetado, mismo que está compuesto por información sobre las características de los estudiantes, como el sexo, la edad, la procedencia, el puntaje en el examen de ingreso a la universidad y las notas en los exámenes semestrales. Los resultados fueron: para el algoritmo KNN una exactitud de 78.86%, SVM con una exactitud de 98.60%, DT con una exactitud de 93.28%, RF con una exactitud de 91.61% y GBM con una exactitud de 99.66%, siendo este último el que generó la mayor exactitud. (p. 724)

Por otro lado, Carrión et al. (2021), comenta que el objetivo del estudio fue evaluar cómo la pandemia afectó el rendimiento académico y la salud mental de los estudiantes de farmacia que asisten a una universidad históricamente negra (HBCU). Para ello, se desarrolló una encuesta que recopiló información demográfica y respuestas de los estudiantes utilizando preguntas tipo Likert, de selección múltiple, con varias respuestas. Los resultados del estudio indicaron que la pandemia tuvo un impacto significativo en el rendimiento académico y la salud mental de los estudiantes de farmacia HBCU, lo que sugiere la necesidad de apoyo adicional para estos estudiantes durante y después de la pandemia. (p. 1). Considerando lo leído, la etapa virtual que se atravesó durante la pandemia es un precedente del porqué se presenta un bajo rendimiento académico, por lo cual, se requiere tomar las medidas necesarias para restablecer el marco de estudio y los métodos del aprendizaje.

Por otra parte, se hace uso de las siguientes **teorías y bases conceptuales**, a modo, de que sirvan como guía durante todo el desarrollo. Al respecto, la Universidad Internacional de La Rioja (UNIR) indica que **los sistemas inteligentes** aplican tecnologías como el Big Data, el internet de las cosas, el 5G, la inteligencia artificial, la visión artificial, la realidad aumentada, entre otros. Permitiendo así, la integración entre el mundo físico y el mundo virtual (párr. 2, 2022).

En cuanto al **ML**, Awad y Khanna (2015) lo definen como una rama de la inteligencia artificial que emplea una serie de algoritmos para analizar y simplificar la relación entre datos y así dotar a las computadoras de la capacidad de encontrar patrones y generar predicciones (p. 1).

Respecto a los **tipos de ML**, Oladipupo (2010) lo divide en 4 por orden jerárquica, tales como, el aprendizaje supervisado, aprendizaje no supervisado, aprendizaje semisupervisado y aprendizaje por refuerzo (p. 3). A continuación, el autor define un breve concepto por cada tipo de ML.

- **Aprendizaje Supervisado:** El algoritmo se encarga de generar una función que pueda asignar las entradas a las salidas deseadas.
- **Aprendizaje No Supervisado:** Modela un conjunto de entradas de datos.
- **Aprendizaje Semisupervisado:** Realiza la combinación de ejemplos etiquetados y no etiquetados parar generar un nuevo clasificador.
- **Aprendizaje por refuerzo:** El algoritmo establece como actuar en base a su percepción del mundo, dónde cada acción genera un impacto en el entorno, mismo que guía al algoritmo de aprendizaje.

Además, Sen (2021), añade también que en el **aprendizaje supervisado** el algoritmo desarrolla un modelo matemático con los datos de las entradas y los resultados esperados. Para el aprendizaje no supervisado se toma un conjunto de datos para detectar los patrones. En cuanto al aprendizaje semisupervisado es una forma híbrida de técnicas. Finalmente, el aprendizaje por refuerzo hace uso de software y máquinas para tomar una mejor decisión (p. 62 – 63).

Por otro lado, Albán y Calero (2017), menciona que el **rendimiento académico** comprende todo el proceso de aprendizaje realizado por el estudiante y el aprovechamiento de las herramientas e influencias brindadas a los alumnos (p. 214). Además, de que dependerá también de los procesos de aprendizaje que emplee el docente como las metodologías que aplique la institución educativa para la enseñanza de los cursos.

Respecto a los métodos que pueden aplicarse para facilitar el uso del ML, se encuentra **la selección de variables o características**, dónde la autora González (2015) infiere que este método se encarga de abordar la construcción o selección del modelo, teniendo en cuenta que, a mayor cantidad de variables, más parámetros por estimar y por ende produce una disminución en la precisión

individual. Por ese motivo, se busca seleccionar las variables con apoyo de filtros (p. 11).

Para ello, Hsuan et al. (2023), menciona que, en la selección de características mediante **RFE**, es utilizado para reducir el número de variables de entrada de los modelos de ML, para identificar las características importantes en este estudio (p. 3).

En cuanto al algoritmo **SelectkBest**, Meca (2011), indica que es una técnica de selección de características o variables mediante la relación entre la variable objetivo y la selección de las características más importantes según un puntaje especifico (p. 9).

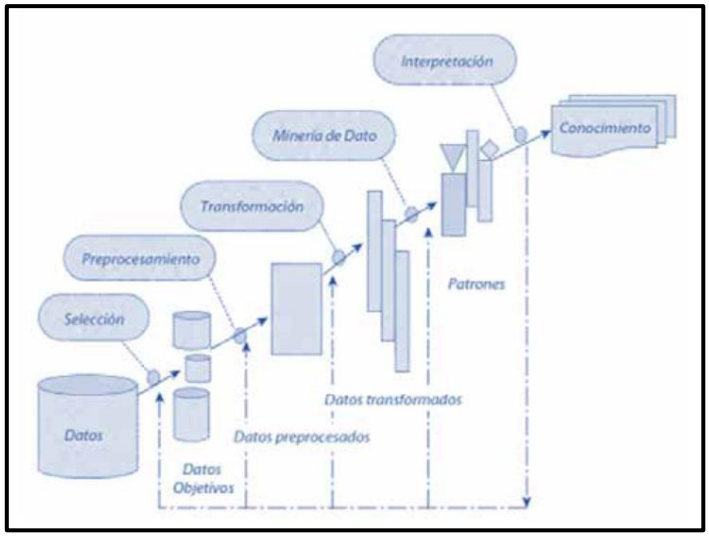
De igual manera, Canto (2021), indica que **SelectPercentile** es un método de selección que se utiliza para elegir o seleccionar las características más importantes de un conjunto de datos en función de un porcentaje especifico de las mejores características, se basa en una evaluación univariante de las variables (p. 67).

Asimismo, Chaudhari y Thakkar (2023), menciona que **PCA**, se utiliza para identificar patrones en los datos y reducir la dimensionalidad, y **VarianceThreshold**, realiza la selección de características al eliminar las características con baja varianza, ambas son técnicas utilizadas ML para el preprocesamiento y la reducción de la dimensionalidad de conjuntos de datos (p. 2).

Respecto al algoritmo de optimización de hiperparámetros, Canto (2021), menciona que el **GridSearchCV** se utiliza para realizar una búsqueda exhaustiva de los mejores hiperparámetros, entrena el modelo y evalúa su rendimiento utilizando validación cruzada para un modelo de aprendizaje automático (p. 72).

Con relación a la metodología que se va a emplear, Corona, Jiménez y Cortés (2020) menciona que **KDD**, es un proceso iterativo que se basa en los requerimientos del usuario final, dónde se combinan los descubrimientos y análisis con el fin de identificar los patrones para que el usuario los analice. (p. 19).

**Figura N° 1: Etapas de la metodología KDD**




**Fuente: Corona, Jiménez y Cortés (2020)**

Los mismos autores definen que la primera etapa de selección consta en identificar el conocimiento importante desde la perspectiva del usuario final, creándose un conjunto de datos objetivo o seleccionando la totalidad de los datos (p. 20).

Respecto a la segunda etapa, preprocesamiento, los mismos autores añaden que se analiza la calidad de la información haciendo uso de las operaciones básicas, tales como, la remoción de datos ruidosos, la selección de estrategias para el manejo de todo tipo de datos y la conceptualización de técnicas estadísticas para su reemplazo en caso sea requerido (p. 20).

En cuanto a la tercera etapa, transformación de datos, indican que se busca características para representar la información dependiendo de las metas del proceso. En este punto, se hace uso de métodos de reducción de dimensiones para hallar representaciones invariantes (p. 20).

En la cuarta etapa, minería de datos, los autores infieren que se busca descubrir los patrones de interés. Esto se logra aplicándose tareas de descubrimiento como lo son la clasificación, el agrupamiento, los patrones secuenciales, las asociaciones, etc. (p. 20).

Por último, en la etapa de interpretación y evaluación de datos, mencionan que se evalúa los datos hallados en la etapa anterior para determinar si estos se retornan por ser patrones redundantes o irrelevantes. Luego, los patrones útiles se traducen para que sean entendidos por los usuarios finales (p. 20).

Asimismo, dentro de las Métricas de Evaluación de Modelo, se definen las **métricas de especificidad, precisión, sensibilidad, exactitud y F1-score**, que serán evaluadas a través de una matriz de confusión, dónde los autores Abdulkadir y Derbew (2023) lo define como una herramienta simple de análisis de rendimiento que hace uso del aprendizaje automático supervisado, dónde cada columna representa la instancia en una clase predicha y cada fila la instancia en una clase real.

**Tabla N° 2: Matriz de confusión**

| | | **Estimación por el modelo** | |
|---|---|---|---|
| | | **Negativo (N)** | **Positivo (P)** |
| **Real** | **Negativo** | TN | FP |
| | **Positivo** | FN | TP |

**Fuente: Abdulkadir y Derbew (2023)**

Dónde:

- **True Negatives (TN):** Predicciones negativas que son realmente negativas.
- **False Positives (FP):** Predicciones positivas que son realmente negativas.
- **False Negatives (FN):** Predicciones negativas que son realmente positivas.
- **True Positives (TP):** Predicciones positivas que son realmente positivas.

Como una de las métricas a tratar, se busca que el modelo presente brinde resultados muy precisos, por lo que, según Burman y Som (2019), la especificidad proporciona una medición para conocer la tasa de verdaderos negativos o falsos, mostrando cuan bien el modelo puede detectar estos casos (p. 758). Gracias a

este, el modelo podrá clasificar correctamente los casos con resultados negativos y prevenir errores dentro de lo posible.

**Ecuación N° 1: Cálculo de la especificidad**

$$\boldsymbol{Especificidad} = \frac{\boldsymbol{TN}}{\boldsymbol{TN + FP}} * \mathbf{100}$$

**Fuente: Burman y Som (2019)**

Adicionalmente, Zapeta et al. (2022) indican que la precisión mide la cantidad de verdaderos positivos y falsos positivos para buscar una predicción precisa por medio del porcentaje de predicciones acertadas (p. 4632).

**Ecuación N° 2: Cálculo de la precisión**

$$\boldsymbol{Precisión} = \frac{\boldsymbol{TP}}{\boldsymbol{TP + FP}} * \mathbf{100}$$

**Fuente: Zapeta et al. (2019)**

Además, se busca que el modelo sea sensible, para lo cual, Burman y Som (2019), mencionan que esta métrica indica el porcentaje de casos positivos que fueron correctamente identificados según los parámetros del estudio (p. 758). Gracias a esto, se disminuirá la probabilidad de tener errores durante el estudio.

**Ecuación N° 3: Cálculo de la sensibilidad**

$$\boldsymbol{Sensibilidad} = \frac{\boldsymbol{TP}}{\boldsymbol{TP + FN}} * \mathbf{100}$$

**Fuente: Burman y Som (2019)**

También, se opta por obtener una mejor solución individual y seleccionar una mejora solución producida ya por un algoritmo en particular. De esta manera, se podrá crear un modelo óptimo. Para ello, se aplica la métrica de exactitud, que según Hossin y Sulaiman (2015) es el concepto que más se acopla a ello (p. 2).

**Ecuación N° 4: Cálculo de la exactitud**

$$Exactitud = \frac{TP + TN}{TP + TN + FP + FN} * 100$$

**Fuente: Hossin y Sulaiman (2015)**

Finalmente, el mismo autor añade que “la métrica F1-score es la media armónica de precisión y sensibilidad”, está métrica se usa para obtener un valor más objetivo (p. 47).

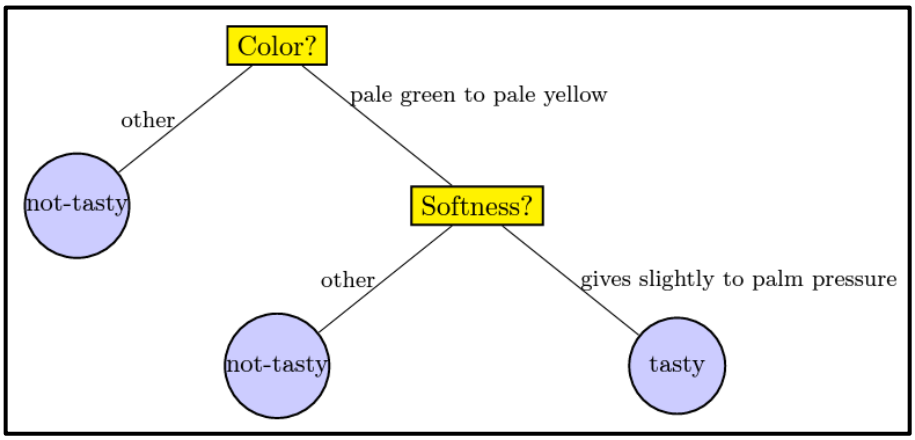
**Ecuación N° 5: Cálculo de F1 Score**

$$F1 - score = 2 * \frac{Precisión * Sensibilidad}{Precisión + Sensibilidad} * 100$$

**Fuente: Dalianis (2018)**

En cuanto a los algoritmos que serán evaluados por la matriz de confusión, son diez en su totalidad. Estos son los siguientes: **DT, RF, SVM, ANN, GBM, MP, KNN, NB, LR y ADA.** Respecto al primer algoritmo, Shalev y Ben (2014) definen al **DT** como un predictor que predice la etiqueta asociada con una instancia X para viajar desde un nodo raíz a una hoja (p. 250).

**Figura N° 2: Ejemplo del algoritmo Decision Tree**




**Fuente: Shalev y Ben (2014)**

Por ejemplo, tal como se muestra en la figura, para corroborar si una papaya es sabrosa o no, el árbol primero examina su color. Si este color no está en el rango

de verde pálido a amarillo pálido, entonces la papaya no es sabrosa. En caso si tenga el color planteado, mide la suavidad con la presión de la palma, si no es blando no se considera sabroso, caso contrario, se define que la papaya si es sabrosa (p. 250).

Correspondiente al segundo algoritmo, Bonaccorso (2017) indica que **RF** es un conjunto de árboles de decisión creados mediante unas muestras aleatorias, cuya política para dividir el nodo es usar un subconjunto aleatorio de características por cada árbol para encontrar el mejor umbral que separe los datos. Para la interpretación de los resultados existen dos formas: la primera se basa en un voto mayoritario y la segunda en la implementación de scikit-learn para promediar los resultados. Ambos métodos conducen a resultados comparables (p. 167).

Por parte del tercer algoritmo, **SVM**, el mismo autor manifiesta que se pueden trabajar con escenarios lineales y no lineales. Además de que cuando se aplica con las redes neuronales representan la mejor opción para diversas tareas de las cuales no son fáciles hallar un buen hiperplano de separación (p. 133).

En cuanto al cuarto algoritmo, Contreras, Fuentes y Rodríguez (2020), indica que **ANN** es un modelo de procesamiento de información basado en un sistema de interconexión de nodos que se utilizan para realizar tareas de aprendizaje automático, estos están compuestos por capas de nodos que tienen asociado un peso que modifica la señal que se transmite entre ellos (p. 238).

Para el quinto algoritmo, Rivera (2020), menciona que **GBM** construye una serie de modelos predictivos simples luego realiza predicciones para mejorar la precisión general, utiliza tareas de regresión, clasificación y da prioridad a la corrección de los errores de los modelos anteriores, lo que permite que el modelo final sea una combinación ponderada de todos los modelos individuales. (p. 29).

Continuando con el sexto algoritmo, Contreras, Fuentes y Rodríguez (2020) manifiesta que **MP** es una variante del algoritmo de redes neuronales, se utiliza para resolver problemas de clasificación, se basa en una red de múltiples capas de perceptrones, que son los elementos básicos de una red neuronal y lo procesan a través de una función de activación para producir una salida (p.238).

Respecto al séptimo algoritmo, Shalev y Ben (2014) lo definen como uno de los algoritmos más simples del ML y su función radica en memorizar el conjunto de entrenamientos para luego predecir la etiqueta de una nueva instancia sobre la base de las etiquetas de sus vecinos más cercanos (p. 258).

Correspondiente al octavo algoritmo, Bonaccorso (2017) lo define como clasificadores potentes y fáciles de entrenar. Su función es que determinan la probabilidad de un resultado dado un conjunto de condiciones utilizando el teorema de Bayes, dónde busca que las probabilidades condicionales se inviertan para que la consulta se exprese como una función de cantidades medibles (p. 120).

**Ecuación N° 6: Teorema de Bayes**

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

**Fuente: Bonaccorso (2017)**

En la presente ecuación, se consideran dos eventos probabilísticos A y B, dónde P(A) y P(B) son las probabilidades marginales y P(A|B) y P(B|A) son las probabilidades condicionales.

Por parte del noveno algoritmo, el mismo autor añade que el algoritmo LR "es un método de clasificación que se basa en la probabilidad de que una muestra pertenezca a una clase" (p. 97).

Finalmente, para el décimo algoritmo **ADA**, Meza y Chue (2020), lo definen como un algoritmo de ML utilizado principalmente para la clasificación de datos y la resolución de problemas de predicción, este asigna pesos a cada instancia de datos de entrenamiento y los ajusta combinando los clasificadores débiles ponderados de manera adecuada para obtener un clasificador final más robusto (p.29).

Por la parte tecnológica, a continuación, se define lo que se usará. Según Halvorsen (2020), **Python** es un lenguaje de programación abierto que tiene multipropósito, por lo mismo que cuenta con una extensión múltiple, tales como: computación científica, cálculo y desarrollo web (p. 17).

Según, Cabrera y Díaz (2018), **Jupyter Notebooks** es un proyecto de código abierto que permite el uso de diferentes lenguajes de programación en una plataforma computacional. Su popularidad en el rubro académico ha crecido enormemente debido a su gran capacidad y fácil acceso a través de un navegador (p. 1).

Adicionalmente, Mérette et al. (2020) define a **Anaconda** como un software libre que proporciona un conjunto de herramientas para la investigación científica, también, permite la codificación en Python (p. 4).

De igual manera, Weissman y Van de Laar (2020), menciona que **SQL server** es un sistema de gestión de bases de datos relacional que se utiliza comúnmente en entornos empresariales para gestionar grandes volúmenes de datos y realizar tareas como transacciones de bases de datos, análisis de datos, generación de informes para almacenar y recuperar datos (p. 2).

Asimismo, el mismo autor añade que, **SSMS** es un entorno integrado el cual proporciona herramientas para configurar, monitorear y administrar instancias de SQL server y sus bases de datos, que permite a los usuarios realizar tareas de forma visual o mediante la escritura de comandos SQL (p. 35).

Por otra parte, Villamagua (2021), menciona que **Spyder** es un IDE para Python, diseñado específicamente para la ciencia de datos y el análisis numérico, este se destaca por proporcionar una interfaz gráfica de usuario potente para el desarrollo, la depuración, la programación interactiva y el análisis de datos (p. 17).

# III. METODOLOGÍA

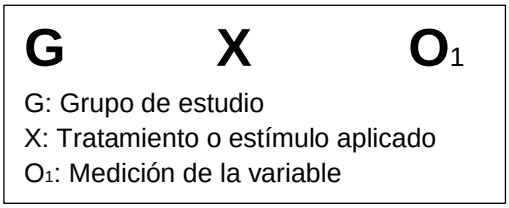
En este capítulo, se detalla el tipo y diseño de investigación, abordando aspectos fundamentales como la definición y operacionalización de las variables. Además, se delimita la población, se describe el proceso de selección de la muestra y se ofrece un análisis detallado de las técnicas e instrumentos utilizados en la recolección de datos del mismo modo en el método de análisis de datos y los aspectos éticos del trabajo.

### 3.1 Tipo y diseño de investigación

El presente trabajo de investigación es de tipo aplicada, ya que, empleó un Sistema Inteligente con ML para obtener un resultado de la predicción del rendimiento académico de cada alumno. Al respecto, Arias (2021) menciona que la investigación aplicada se caracteriza por utilizar la teoría para resolver problemas prácticos, apoyándose en hallazgos y descubrimientos relacionados con los objetivos del estudio, haciendo uso de conocimientos para resolver problemas y en beneficios de los grupos participantes, buscando un rápido uso para el conocimiento existente. (p. 68).

Adicionalmente, Hernández, Fernández y Baptista (2014), infieren que la investigación pre – experimental consiste en suministrar un estímulo o procedimiento a un conjunto en específico, para luego poder medir una o más variables según sea el caso y así observar el nivel en el que se encuentra dicho grupo o conjunto. (p. 141). Por ende, el diseño de investigación que se empleó es experimental de tipo pre – experimental con una sola medición, debido a que se aplicó el sistema inteligente para únicamente predecir el rendimiento académico dónde una vez obtenido los resultados de predicción, se brindó la información a la institución educativa para las mejoras respectivas. Se puede diagramar de la siguiente manera.

**Figura N° 3: Diagrama del diseño de investigación**



**Fuente: Hernández, Fernández y Baptista (2014)**

G: Alumnos del grado de secundaria de la I.E.P “Nuestra Señora de Copacabana”.

X: Sistema inteligente con ML basado en selección de variables para predecir el rendimiento académico.

O: Métricas de Evaluación de Modelo.

## 3.2 Variables y operacionalización

Según Espinoza (2019), las variables son piezas fundamentales para el desarrollo de una investigación. Estas pueden identificarse desde cuando se establece la problemática a investigar (p. 172). En el presente trabajo se determinó como variables: “Sistema Inteligente con ML” (variable independiente) y “predecir el rendimiento académico” (variable dependiente).

Respecto a la operacionalización, Arias (2021) lo define como un proceso que consiste en la descomposición ordenada de las variables para obtener sus dimensiones, y de estas obtener los indicadores necesarios para la investigación. (p. 54). En este trabajo, la variable dependiente, contó con una dimensión “Métricas de Evaluación de Modelo” que tuvo 5 indicadores: especificidad, precisión, sensibilidad, exactitud y F1-score, cada uno con su respectiva fórmula de cálculo. Se muestra el detalle en el ANEXO 1.

## 3.3 Población, muestra y muestreo

### 3.3.1 Población

Según, Lohr (2022), se conoce como población a todo un conjunto de elementos que se busca estudiar, los cuales deben compartir las mismas condiciones. Definir este rango es de suma importancia puesto que afectará profundamente en los resultados finales. (p. 4). Para el presente trabajo de investigación, se tuvo como población a todos los alumnos de grado secundaria, siendo un total de 256 estudiantes. Su selección estuvo basada en los siguientes criterios.

- **Criterios de inclusión:**

Se consideró a todos los estudiantes de sexo Masculino o Femenino, mayores a 11 años que se encuentren matriculados en el nivel de secundaria, desde primero a quinto, cursando las asignaturas correspondientes a su grado.

- **Criterios de exclusión:**

Se excluirá a los estudiantes pertenecientes a los grados de inicial y primaria, así como a los estudiantes del grado secundaria que se encuentren asistiendo a una academia particular o que estén asignados al aula "La Academia", puesto que ellos están realizando actividades distintas a los demás estudiantes.

### 3.3.2 Muestra

En cuanto a la muestra, Arias (2012) manifiesta que es una porción de la población que por su tamaño permite mayor facilidad para manejar los datos y así generalizar los resultados a la población restante con un margen de error aceptable. (p. 83). Sin embargo, en este estudio se consideró a la totalidad de la población, debido a que es más conveniente tener una mayor cantidad de datos para el desarrollo de este experimento.

### 3.3.2 Muestreo

Respecto al muestreo, el autor añade que es el conjunto de procedimientos llevados a cabo para analizar cómo ciertas características se distribuyen en toda una población. (p. 93). El presente trabajo aplicó un tipo de muestreo no probabilístico, por lo mismo que se consideraron todos los datos.

### 3.3.3 Unidad de Análisis

Un alumno de grado secundaria que se encuentre estudiando en la I.E.P. “Nuestra Señora de Copacabana”.

### 3.4 Técnicas e instrumento de recolección de datos

Al respecto Hernández, Fernández y Baptista (2014), los cuestionarios son usados en encuestas de cualquier índole, ya sea, para conocer el desempeño, identificar las necesidades o evaluar la percepción de un tema en general. Además, también define al cuestionario como un conjunto de preguntas relacionadas con una o más variables dentro de la investigación. (p. 217). Por tal motivo, para recopilar la información que servirá como data en la predicción del rendimiento académico, se empleó la técnica de la encuesta con su instrumento del cuestionario. Dicho instrumento, estuvo elaborado mediante Microsoft Word y se llevó de manera impresa a todos los estudiantes para su llenado, cabe resaltar que estuvo compuesto por 5 apartados, los datos personales del estudiante (2 ítems), la autopercepción (5 ítems), la educación familiar (3 ítems), sociabilidad (3 ítems) y la economía familiar (3 ítems), dichas preguntas fueron elaboradas con el apoyo del área de psicología de la propia institución educativa. Para mayor detalle del cuestionario empleado, véase el ANEXO 11.

Asimismo, se utilizó la técnica de observación con el instrumento de Fichas de Registro para recopilar la precisión de cada algoritmo a usar. Para esto, los autores definen a la observación como un método de recolección de datos basado en el registro confiable de comportamientos mediante un conjunto de categorías y subcategorías (p. 252). Véase el ANEXO 12 – 21.

### 3.5 Procedimientos

El presente trabajo, inició con la búsqueda de información e investigaciones similares al tema propuesto, tanto a nivel nacional como internacional, ya sea, de tesis o artículos científicos para plasmar la situación actual del rendimiento académico. En cuanto se obtuvo la información recopilada, se estableció los antecedentes y bases teóricas que apoyaron al desarrollo del trabajo, para posteriormente establecer la dimensión e indicadores de la variable dependiente de forma sustentada.

Para la recolección de datos se solicitó al director general de la I.E.P. “Nuestra Señora de Copacabana” una carta de aceptación para el uso y

manipulación de la información, tal como se visualiza en el ANEXO 8. Como segundo punto, se tomó los datos de registro de notas de los alumnos que se encuentra en una base de datos cargada en OneDrive, véase el ANEXO 10, y también se empleó un cuestionario. Dicho documento, se entregó de manera impresa a los docentes encargados de cada aula para que los alumnos puedan llenar la información correspondiente. Con ambas fuentes de información recopiladas, se importó la data a SSMS para que luego se ejecuten las consultas mediante la herramienta Jupyter Notebook y Anaconda. Luego se entrenó el modelo con los diez algoritmos: **DT, RF, SVM, ANN, GBM, MP, KNN, NB, LR y ADA**, así se pudo determinar los mejores modelos de predicción. Posteriormente, se construyó la interfaz del Sistema inteligente. Dicha interfaz, estuvo desarrollada en Spyder con el lenguaje de programación Python y SQL Server, véase el ANEXO 6.

Por último, para validar las hipótesis, se utilizaron dos herramientas: la matriz de confusión y el coeficiente de Kappa de Cohen, para ver cómo se desempeñaron los algoritmos de ML y, validar la concordancia y grado de significancia entre dos clasificadores respectivamente.

### 3.6 Método de análisis de datos

Este trabajo de investigación recopiló la información mediante la aplicación de un cuestionario de manera presencial en la institución, para posteriormente digitalizar la información y obtener una base de datos de los resultados. Luego, se empleó las fichas de registro según las métricas de evaluación de modelo y los algoritmos seleccionados. Finalmente, se usó Pyhton, Jupyter Notebook y Anaconda junto con el método de análisis descriptivo y predictivo, mismo que se encarga de utilizar los datos recopilados para encontrar los patrones y predecir lo que sucederá más adelante. De esta manera se estableció el mejor modelo de predicción para el desarrollo del Sistema Inteligente.

### 3.7 Aspectos éticos

El presente trabajo se comprometió con el cumplimiento de las normativas a nivel mundial para asegurar la ética del investigador. Asimismo, buscó respetar la

autoría de las fuentes citadas, referenciando así a los autores de los libros, tesis, artículos científicos, entre otros. Dichas referencias, se citaron según el manual de la norma ISO 690 y 690-2 brindado por el Fondo Editorial de la Universidad César Vallejo. Además, el trabajo se desarrolló aplicando la RESOLUCIÓN DE CONSEJO UNIVERSITARIO N° 0531-2021/UCV mostrado en el ANEXO 25, dando un enfoque a los artículos 7, 8, 24 para así garantizar un trabajo de calidad con ética moral y derechos de autor, logrando que pueda ser utilizado para extraer información en futuras investigaciones.

# IV. RESULTADOS

En esta sección, se presentan y demuestran los resultados que se obtuvieron en la investigación, basados en las métricas de especificidad, precisión, sensibilidad, exactitud y F1-score, que se obtuvieron realizando la comparación entre diez algoritmos, tales como: **DT, RF, SVM, ANN, GBM, MP, KNN, NB, LR y ADA**. A continuación, se profundiza en los resultados específicos obtenidos con respecto en la medición del rendimiento académico, la I.E.P. “Nuestra Señora de Copacabana”, lo clasificó de la siguiente manera.

**Tabla N° 3: Medición del rendimiento académico**

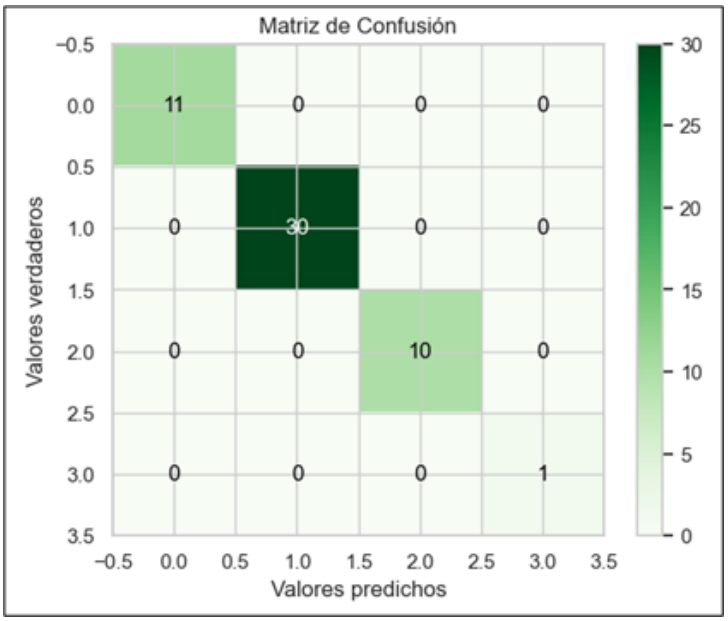
| PROMEDIO_CALIF_NOTAS | REND_ACAD | ESTADO_REND_ACAD |
|---|---|---|
| 0 al 10 | 1 | Desaprobado |
| 11 al 13 | 2 | Regular |
| 14 al 16 | 3 | Bueno |
| 17 al 18 | 4 | Distinguido |
| 19 al 20 | 5 | Excelente |

**Fuente: I.E.P “Nuestra Señora de Copacabana”**

A continuación, se demuestra la validación de cada hipótesis mediante la matriz de confusión y la Kappa de Cohen.

- **Decision Tree**

**Figura N° 4: Matriz de confusión – DT**




**Fuente: Elaboración Propia**

**Tabla N° 4: Matriz de observación – DT**

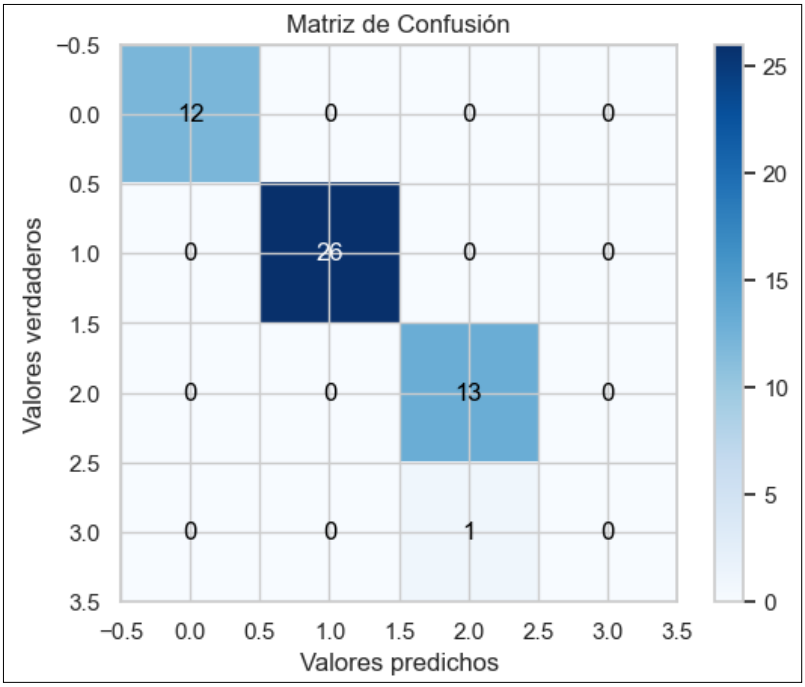
| Clases | Medidas | | | |
|---|---|---|---|---|
| | TP | TN | FP | FN |
| 2 | 11 | 41 | 0 | 0 |
| 3 | 30 | 22 | 0 | 0 |
| 4 | 10 | 42 | 0 | 0 |
| 5 | 1 | 51 | 0 | 0 |

**Fuente: Elaboración Propia**

Para un total de 52 registros en el conjunto de validación, se identificó a 11 alumnos con notas regulares de manera correcta, 30 alumnos con notas buenas de manera correcta, 10 alumnos con notas distinguidas de manera correcta y 1 alumno con nota excelente de manera correcta.

- **Random Forest**

**Figura N° 5 Matriz de confusión – RF**




**Fuente: Elaboración Propia**

**Tabla N° 5: Matriz de observación – RF**

| Clases | Medidas | | | |
|---|---|---|---|---|
| | TP | TN | FP | FN |
| 2 | 12 | 40 | 0 | 0 |

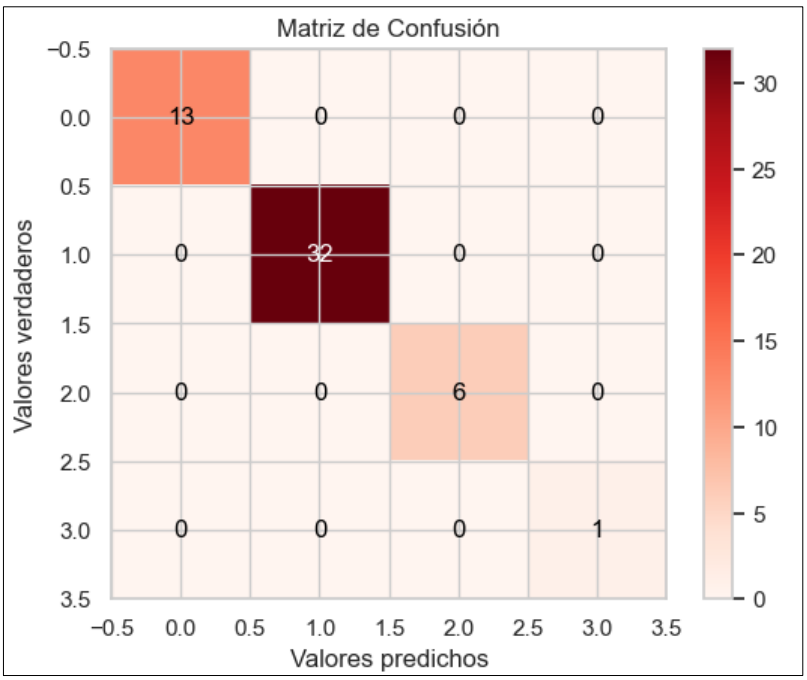
| | | | | |
|---|---|---|---|---|
| 3 | 26 | 26 | 0 | 0 |
| 4 | 13 | 38 | 1 | 0 |
| 5 | 0 | 51 | 0 | 1 |

**Fuente: Elaboración Propia**

Para un total de 52 registros, se identificó a 12 alumnos con notas regulares de manera correcta, 26 alumnos con notas buenas de manera correcta, 13 alumnos con notas distinguidas de manera correcta y 1 predicción incorrecta para un alumno con nota excelente.

- **Support Vector Machines**

**Figura N° 6 Matriz de confusión – SVM**




**Fuente: Elaboración Propia**

**Tabla N° 6: Matriz de observación – SVM**

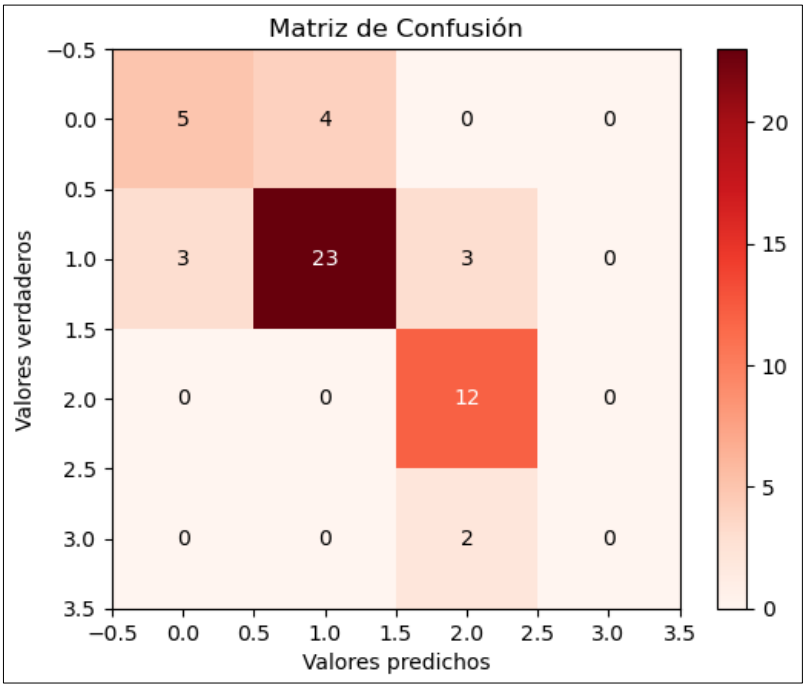
| Clases | Medidas | | | |
|---|---|---|---|---|
| | TP | TN | FP | FN |
| 2 | 13 | 39 | 0 | 0 |
| 3 | 32 | 20 | 0 | 0 |
| 4 | 6 | 46 | 0 | 0 |
| 5 | 1 | 51 | 0 | 0 |

**Fuente: Elaboración Propia**

Para un total de 52 registros, se identificó a 13 alumnos con notas regulares de manera correcta, 32 alumnos con notas buenas de manera correcta, 6 alumnos con notas distinguidas de manera correcta y 1 alumno con nota excelente de manera correcta.

- **Artificial Neural Network**

**Figura N° 7 Matriz de confusión – ANN**




**Fuente: Elaboración Propia**

**Tabla N° 7: Matriz de observación – ANN**

| Clases | Medidas | | | |
|---|---|---|---|---|
| | TP | TN | FP | FN |
| 2 | 5 | 40 | 3 | 4 |
| 3 | 23 | 19 | 4 | 6 |
| 4 | 12 | 35 | 5 | 0 |
| 5 | 0 | 50 | 0 | 2 |

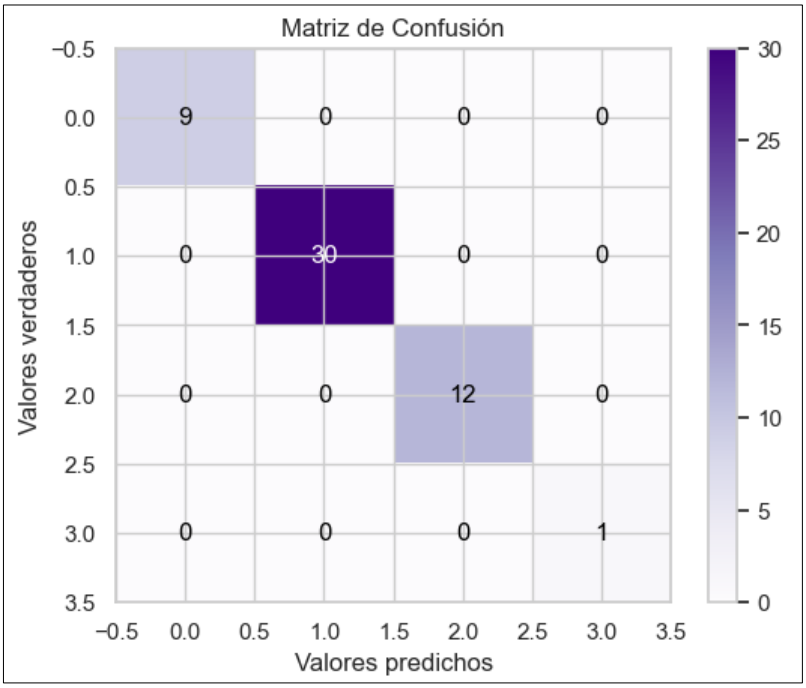
**Fuente: Elaboración Propia**

Para un total de 52 registros, se identificó a 5 alumnos con notas regulares de manera correcta y 4 de manera incorrecta, 23 alumnos con notas buenas de

manera correcta y 6 de manera incorrecta, 12 alumnos con notas distinguidas de manera correcta, y 2 predicciones incorrectas para alumnos con nota excelente.

- **Gradient Boosting Machines**

**Figura N° 8 Matriz de confusión – GBM**




**Fuente: Elaboración Propia**

**Tabla N° 8: Matriz de observación – GBM**

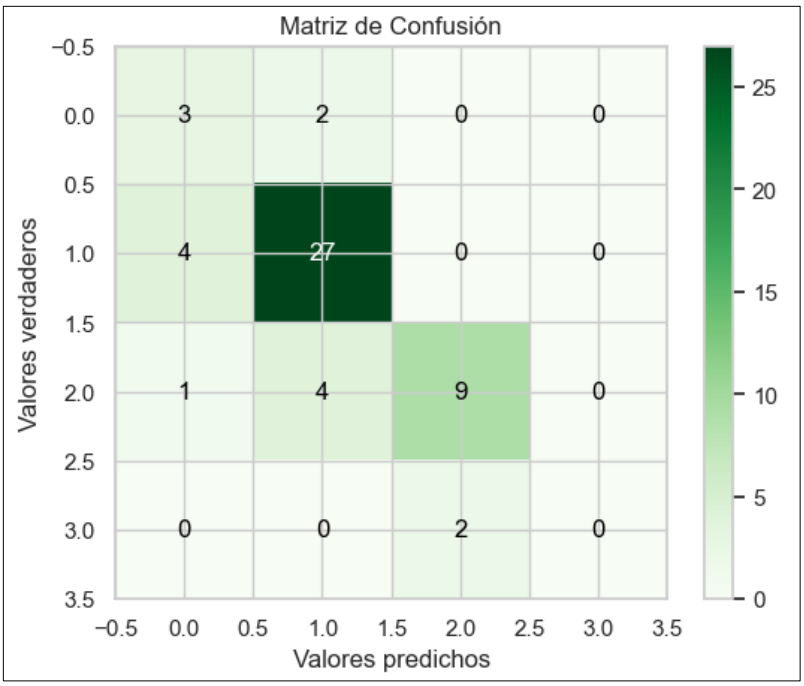
| Clases | Medidas | | | |
|---|---|---|---|---|
| | TP | TN | FP | FN |
| 2 | 9 | 43 | 0 | 0 |
| 3 | 30 | 22 | 0 | 0 |
| 4 | 12 | 40 | 0 | 0 |
| 5 | 1 | 51 | 0 | 0 |

**Fuente: Elaboración Propia**

Para un total de 52 registros, se identificó a 9 alumnos con notas regulares de manera correcta, 30 alumnos con notas buenas de manera correcta, 12 alumnos con notas distinguidas de manera correcta y 1 alumno con nota excelente de manera correcta.

- **Multi Perceptrón**

**Figura N° 9 Matriz de confusión – MP**




**Fuente: Elaboración Propia**

**Tabla N° 9: Matriz de observación – MP**

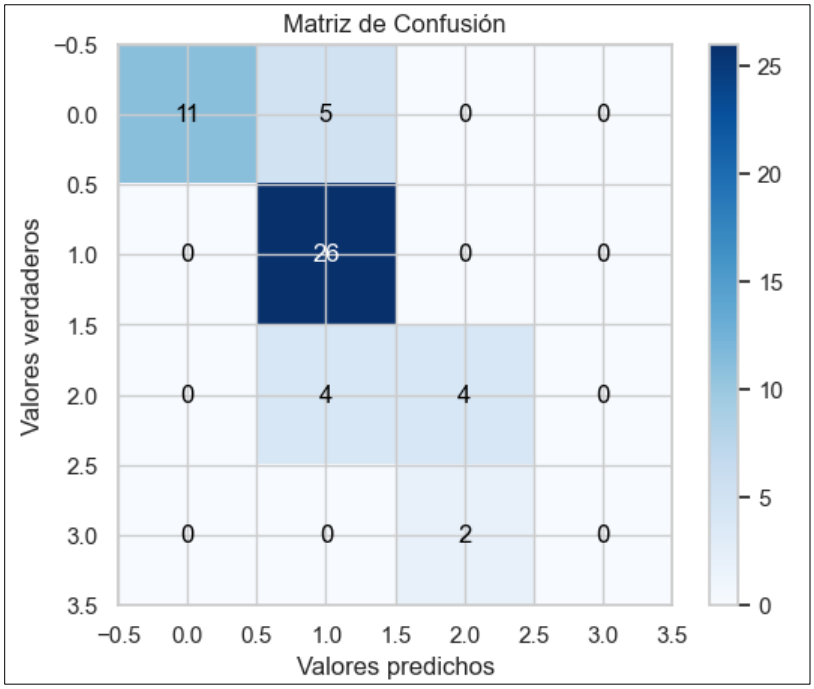
| Clases | Medidas | | | |
|---|---|---|---|---|
| | TP | TN | FP | FN |
| 2 | 3 | 42 | 5 | 2 |
| 3 | 27 | 15 | 6 | 4 |
| 4 | 9 | 36 | 2 | 5 |
| 5 | 0 | 50 | 0 | 2 |

**Fuente: Elaboración Propia**

Para un total de 52 registros, se identificó a 3 alumnos con notas regulares de manera correcta y 2 de manera incorrecta, 27 alumnos con notas buenas de manera correcta y 4 de manera incorrecta, 9 alumnos con notas distinguidas de manera correcta y 5 de manera incorrecta, y 2 predicciones de manera incorrecta para alumnos con nota excelente.

- **K-Nearest Neighboor**

**Figura N° 10 Matriz de confusión – KNN**




**Fuente: Elaboración Propia**

**Tabla N° 10: Matriz de observación – KNN**

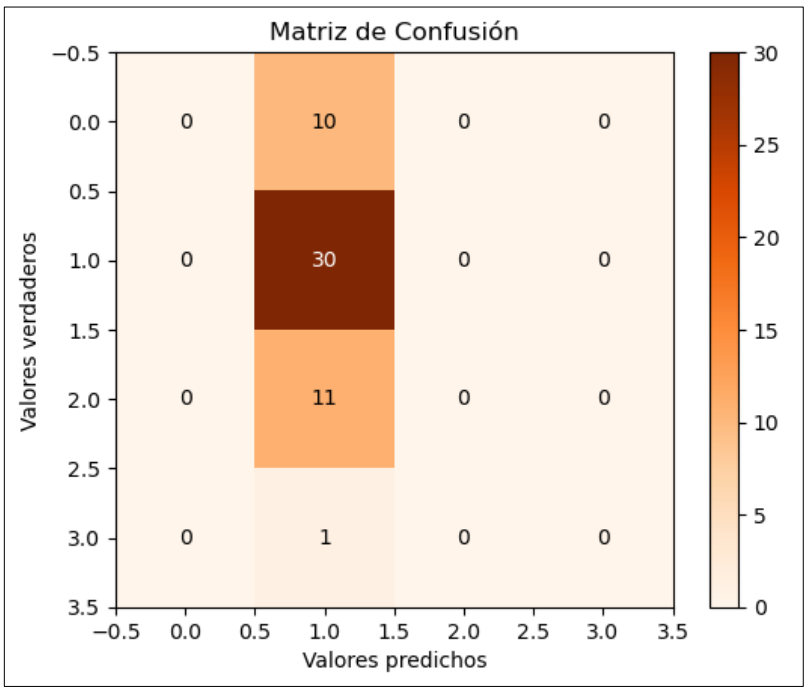
| Clases | Medidas | | | |
|---|---|---|---|---|
| | TP | TN | FP | FN |
| 2 | 11 | 36 | 0 | 5 |
| 3 | 26 | 17 | 9 | 0 |
| 4 | 4 | 42 | 2 | 4 |
| 5 | 0 | 50 | 0 | 2 |

**Fuente: Elaboración Propia**

Para un total de 52 registros, se identificó a 11 alumnos con notas regulares de manera correcta y 5 de manera incorrecta, 26 alumnos con notas buenas de manera correcta, 4 alumnos con notas distinguidas de manera correcta y 4 de manera incorrecta, y 2 predicciones de manera incorrecta para alumnos con nota excelente.

- **Naïve Bayes**

**Figura N° 11 Matriz de confusión – NB**




**Fuente: Elaboración Propia**

**Tabla N° 11: Matriz de observación – NB**

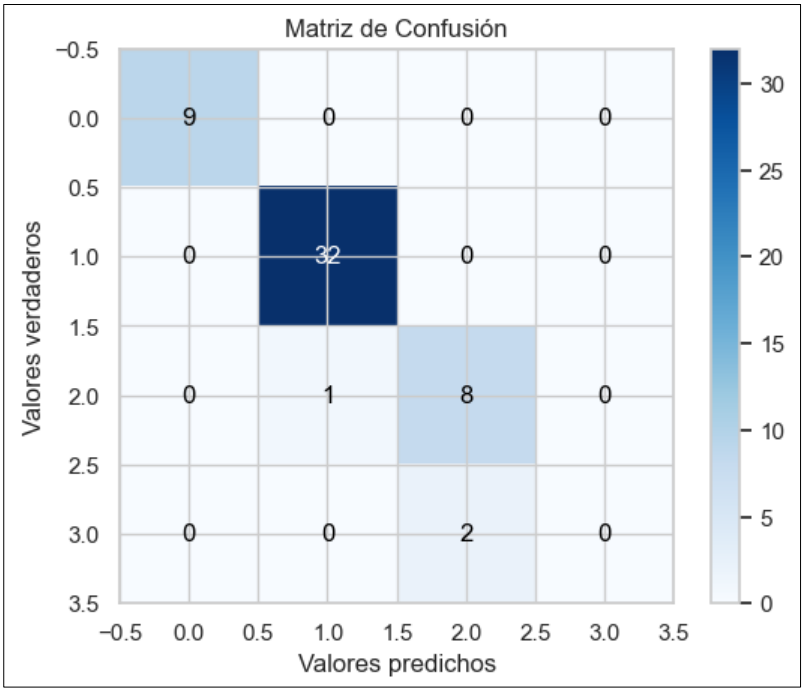
| Clases | Medidas | | | |
|---|---|---|---|---|
| | TP | TN | FP | FN |
| 2 | 0 | 42 | 0 | 10 |
| 3 | 30 | 0 | 22 | 0 |
| 4 | 0 | 41 | 0 | 11 |
| 5 | 0 | 51 | 0 | 1 |

**Fuente: Elaboración Propia**

Para un total de 52 registros, se identificó 10 predicciones incorrectas de alumnos con notas regulares y 0 de manera correcta, 30 alumnos con notas buenas de manera correcta y 0 de manera incorrecta, 11 predicciones incorrectas de alumnos con notas distinguidas y 0 de manera correcta, y 1 predicción de manera incorrecta para un alumno con nota excelente y 0 de manera correcta.

- **Logistic Regression**

**Figura N° 12 Matriz de confusión – LR**




**Fuente: Elaboración Propia**

**Tabla N° 12: Matriz de observación – LR**

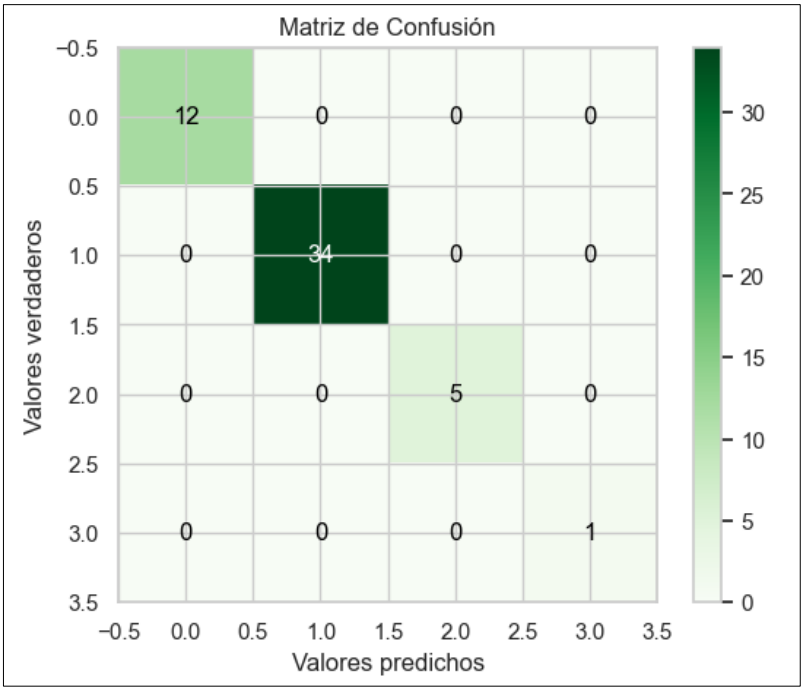
| Clases | Medidas | | | |
|---|---|---|---|---|
| | TP | TN | FP | FN |
| 2 | 9 | 43 | 0 | 0 |
| 3 | 32 | 19 | 1 | 0 |
| 4 | 8 | 41 | 2 | 1 |
| 5 | 0 | 50 | 0 | 2 |

**Fuente: Elaboración Propia**

Para un total de 52 registros, se identificó a 9 alumnos con notas regulares de manera correcta, 32 alumnos con notas buenas de manera correcta, 8 alumnos con notas distinguidas de manera correcta y 1 de manera incorrecta, y 2 predicciones incorrectas de alumnos con nota excelente.

- **Ada Boosting**

**Figura N° 13 Matriz de confusión – ADA**




**Fuente: Elaboración Propia**

**Tabla N° 13: Matriz de observación – ADA**

| Clases | Medidas | | | |
|---|---|---|---|---|
| | TP | TN | FP | FN |
| 2 | 12 | 40 | 0 | 0 |
| 3 | 34 | 18 | 0 | 0 |
| 4 | 5 | 47 | 0 | 0 |
| 5 | 1 | 51 | 0 | 0 |

**Fuente: Elaboración Propia**

Para un total de 52 registros, se identificó a 12 alumnos con notas regulares de manera correcta, 34 alumnos con notas buenas de manera correcta, 5 alumnos con notas distinguidas de manera correcta y 1 alumno con nota excelente de manera correcta.

**Hipótesis 1:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables predecirá con especificidad el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana”

- **Decision Tree**

**Tabla N° 14: Cálculo de la especificidad con el algoritmo DT**

| **Clases** | **Especificidad** (TN / (TN+FP)) * 100% | **Resultado** |
|---|---|---|
| 2 | (41/(41+0)) * 100% | 100% |
| 3 | (22/(22+0)) * 100% | 100% |
| 4 | (42/(42+0)) * 100 | 100% |
| 5 | (51/(51+0)) * 100 | 100% |
| Total | | 100% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una especificidad del 100% utilizando el algoritmo DT.

- **Random Forest**

**Tabla N° 15: Cálculo de la especificidad con el algoritmo RF**

| **Clases** | **Especificidad** (TN / (TN+FP)) * 100% | **Resultado** |
|---|---|---|
| 2 | (40/(40+0)) * 100% | 100% |
| 3 | (26/(26+0)) * 100% | 100% |
| 4 | (38/(38+1)) * 100 | 97.43% |
| 5 | (51/(51+0)) * 100 | 100% |
| Total | | 99.35% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una especificidad del 99.35% utilizando el algoritmo RF.

- **Support Vector Machines**

**Tabla N° 16: Cálculo de la especificidad con el algoritmo SVM**

| **Clases** | **Especificidad** (TN / (TN+FP)) * 100% | **Resultado** |
|---|---|---|
| 2 | (39/(39+0)) * 100% | 100% |
| 3 | (20/(20+0)) * 100% | 100% |
| 4 | (46/(46+0)) * 100 | 100% |
| 5 | (51/(51+0)) * 100 | 100% |
| Total | | 100% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una especificidad del 100% utilizando el algoritmo SVM.

- **Artificial Neural Network**

**Tabla N° 17: Cálculo de la especificidad con el algoritmo ANN**

| **Clases** | **Especificidad** (TN / (TN+FP)) * 100% | **Resultado** |
|---|---|---|
| 2 | (40/(40+3)) * 100% | 93.02% |
| 3 | (19/(19+4)) * 100% | 82.60% |
| 4 | (35/(35+5)) * 100 | 87.5% |
| 5 | (50/(50+0)) * 100 | 100% |
| Total | | 90.78% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una especificidad del 90.78% utilizando el algoritmo ANN.

- **Gradient Boosting Machines**

**Tabla N° 18: Cálculo de la especificidad con el algoritmo GBM**

| **Clases** | **Especificidad** (TN / (TN+FP)) * 100% | **Resultado** |
|---|---|---|
| 2 | (43/(43+0)) * 100% | 100% |
| 3 | (22/(22+0)) * 100% | 100% |
| 4 | (40/(40+0)) * 100 | 100% |
| 5 | (51/(51+0)) * 100 | 100% |
| Total | | 100% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una especificidad del 100% utilizando el algoritmo GBM.

- **Multi Perceptrón**

**Tabla N° 19: Cálculo de la especificidad con el algoritmo MP**

| **Clases** | **Especificidad** (TN / (TN+FP)) * 100% | **Resultado** |
|---|---|---|
| 2 | (42/(42+5)) * 100% | 89.36% |
| 3 | (15/(15+6)) * 100% | 71.42% |
| 4 | (36/(36+2)) * 100 | 94.73% |
| 5 | (50/(50+0)) * 100 | 100% |
| Total | | 88.88% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una especificidad del 88.88% utilizando el algoritmo MP.

- **K-Nearest Neighboor**

**Tabla N° 20: Cálculo de la especificidad con el algoritmo KNN**

| **Clases** | **Especificidad** (TN / (TN+FP)) * 100% | **Resultado** |
|---|---|---|
| 2 | (36/(36+0)) * 100% | 100% |
| 3 | 17/(17+9)) * 100% | 65.36% |
| 4 | (42/(42+2)) * 100 | 95.45% |
| 5 | (50/(50+0)) * 100 | 100% |
| Total | | 90.20% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una especificidad del 90.20% utilizando el algoritmo KNN.

- **Naïve Bayes**

**Tabla N° 21: Cálculo de la especificidad con el algoritmo NB**

| **Clases** | **Especificidad** (TN / (TN+FP)) * 100% | **Resultado** |
|---|---|---|
| 2 | (42/(42+0)) * 100% | 100% |
| 3 | (0/(0+22)) * 100% | 0% |
| 4 | (41/(41+0)) * 100 | 100% |
| 5 | (51/(51+0)) * 100 | 100% |
| Total | | 85.89% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una especificidad del 85.89% utilizando el algoritmo NB.

- **Logistic Regression**

**Tabla N° 22: Cálculo de la especificidad con el algoritmo LR**

| **Clases** | **Especificidad** (TN / (TN+FP)) * 100% | **Resultado** |
|---|---|---|
| 2 | (43/(43+0)) * 100% | 100% |
| 3 | (19/(19+1)) * 100% | 95% |
| 4 | (41/(41+2)) * 100 | 95.34% |
| 5 | (50/(50+0)) * 100 | 100% |
| Total | | 97.58% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una especificidad del 97.58% utilizando el algoritmo LR.

- **Ada Boosting**

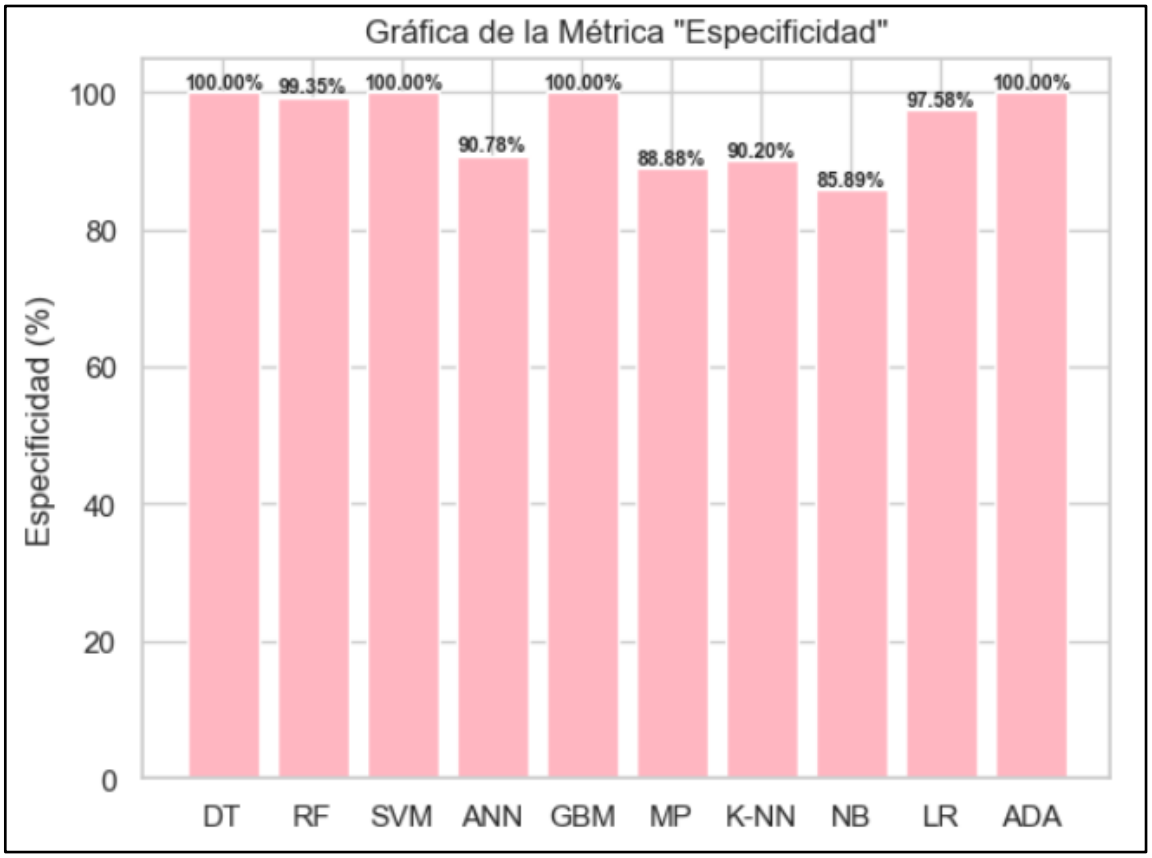
**Tabla N° 23: Cálculo de la especificidad con el algoritmo ADA**

| **Clases** | **Especificidad** (TN / (TN+FP)) * 100% | **Resultado** |
|---|---|---|
| 2 | (40/(40+0)) * 100% | 100% |
| 3 | (18/(18+0)) * 100% | 100% |
| 4 | (47/(47+0)) * 100 | 100% |
| 5 | (51/(51+0)) * 100 | 100% |
| Total | | 100% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una especificidad del 100% utilizando el algoritmo ADA.

**Figura N° 14: Resultados según la métrica de especificidad**




**Fuente: Elaboración Propia**

**Interpretación:** En este gráfico se puede observar que los algoritmos con mejores resultados respecto al indicador de especificidad fueron: DT, SVM, GBM y ADA con un 100% seguidos de RF con un 99.35% y LR con un 97.58%.

**Hipótesis 2:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables predecirá con precisión el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana”

- **Decision Tree**

**Tabla N° 24: Cálculo de la precisión con el algoritmo DT**

| **Clases** | **Precisión** (TP / (TP+FP)) * 100% | **Resultado** |
|---|---|---|
| 2 | (11/(11+0)) * 100% | 100% |
| 3 | (30/(30+0)) * 100% | 100% |
| 4 | (10/(10+0)) * 100 | 100% |
| 5 | (1/(1+0)) * 100 | 100% |
| Total | | 100% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una precisión de 100% utilizando el algoritmo DT.

- **Random Forest**

**Tabla N° 25: Cálculo de la precisión con el algoritmo RF**

| **Clases** | **Precisión** (TP / (TP+FP)) * 100% | **Resultado** |
|---|---|---|
| 2 | (12/(12+0)) * 100% | 100% |
| 3 | (26/(26+0)) * 100% | 100% |
| 4 | (13/(13+1)) * 100 | 92.85% |
| 5 | (0/(0+0)) * 100 | 0% |
| Total | | 73.21% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una precisión de 73.21% utilizando el algoritmo RF.

- **Support Vector Machines**

**Tabla N° 26: Cálculo de la precisión con el algoritmo SVM**

| Clases | Precisión (TP / (TP+FP)) * 100% | Resultado |
|---|---|---|
| 2 | (13/(13+0)) * 100% | 100% |
| 3 | (32/(32+0)) * 100% | 100% |
| 4 | (6/(6+0)) * 100 | 100% |
| 5 | (1/(1+0)) * 100 | 100% |
| Total | | 100% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una precisión del 100% utilizando el algoritmo SVM.

- **Artificial Neural Network**

**Tabla N° 27: Cálculo de la precisión con el algoritmo ANN**

| Clases | Precisión (TP / (TP+FP)) * 100% | Resultado |
|---|---|---|
| 2 | (5/(5+3)) * 100% | 62.5% |
| 3 | (23/(23+4)) * 100% | 85.18% |
| 4 | (12/(12+5)) * 100 | 70.58% |
| 5 | (0/(0+0)) * 100 | 0% |
| Total | | 54.56% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una precisión de 54.56% utilizando el algoritmo ANN.

- **Gradient Boosting Machines**

**Tabla N° 28: Cálculo de la precisión con el algoritmo GBM**

| **Clases** | **Precisión** (TP / (TP+FP)) * 100% | **Resultado** |
|---|---|---|
| 2 | (9/(9+0)) * 100% | 100% |
| 3 | (30/(30+0)) * 100% | 100% |
| 4 | (12/(12+0)) * 100 | 100% |
| 5 | (1/(1+0)) * 100 | 100% |
| Total | | 100% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una precisión de 100% utilizando el algoritmo GBM.

- **Multi Perceptrón**

**Tabla N° 29: Cálculo de la precisión con el algoritmo MP**

| **Clases** | **Precisión** (TP / (TP+FP)) * 100% | **Resultado** |
|---|---|---|
| 2 | (3/(3+5)) * 100% | 37.5% |
| 3 | (27/(27+6)) * 100% | 81.81% |
| 4 | (9/(9+2)) * 100 | 81.81% |
| 5 | (0/(0+0)) * 100 | 0% |
| Total | | 50.28% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una precisión del 50.28% utilizando el algoritmo MP.

- **K-Nearest Neighboor**

**Tabla N° 30: Cálculo de la precisión con el algoritmo KNN**

| **Clases** | **Precisión** (TP / (TP+FP)) * 100% | **Resultado** |
|---|---|---|
| 2 | (11/(11+0)) * 100% | 100% |
| 3 | (26/(26+9)) * 100% | 74.28% |
| 4 | (4/(4+2)) * 100 | 66.66% |
| 5 | (0/(0+0)) * 100 | 0% |
| Total | | 60.23% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una precisión de 60.23% utilizando el algoritmo KNN.

- **Naïve Bayes**

**Tabla N° 31: Cálculo de la precisión con el algoritmo NB**

| **Clases** | **Precisión** (TP / (TP+FP)) * 100% | **Resultado** |
|---|---|---|
| 2 | (0/(0+0)) * 100% | 0% |
| 3 | (30/(30+22)) * 100% | 57.69% |
| 4 | (0/(0+0)) * 100 | 0% |
| 5 | (0/(0+0)) * 100 | 0% |
| Total | | 14.42% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una precisión de 14.42% utilizando el algoritmo NB.

- **Logistic Regression**

**Tabla N° 32: Cálculo de la precisión con el algoritmo LR**

| **Clases** | **Precisión** (TP / (TP+FP)) * 100% | **Resultado** |
|---|---|---|
| 2 | (9/(9+0)) * 100% | 100% |
| 3 | (32/(32+1)) * 100% | 96.96% |
| 4 | (8/(8+2)) * 100 | 80% |
| 5 | (0/(0+0)) * 100 | 0% |
| Total | | 69.24% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una precisión de 69.24% utilizando el algoritmo LR.

- **Ada Boosting**

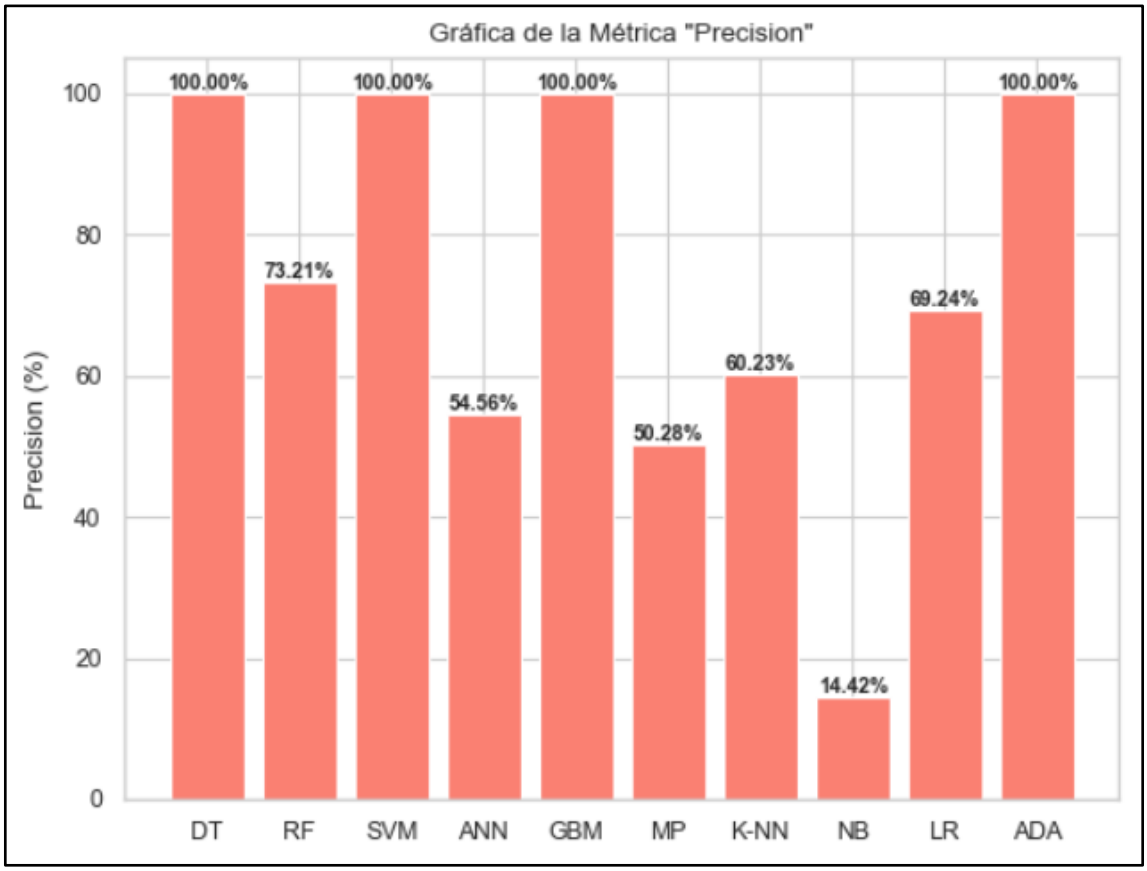
**Tabla N° 33: Cálculo de la precisión con el algoritmo ADA**

| **Clases** | **Precisión** (TP / (TP+FP)) * 100% | **Resultado** |
|---|---|---|
| 2 | (12/(12+0)) * 100% | 100% |
| 3 | (34/(34+0)) * 100% | 100% |
| 4 | (5/(5+0)) * 100 | 100% |
| 5 | (1/(1+0)) * 100 | 100% |
| Total | | 100% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una precisión de 100% utilizando el algoritmo ADA.

**Figura N° 15: Resultados según la métrica de precisión**




**Fuente: Elaboración Propia**

**Interpretación:** En este gráfico se puede observar que los algoritmos con mejores resultados respecto al indicador de precisión fueron: DT, SVM, GBM y ADA con un 100% seguidos de RF con un 73.21% y LR con un 69.24%.

**Hipótesis 3:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables predecirá con sensibilidad el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana”

- **Decision Tree**

**Tabla N° 34: Cálculo de la sensibilidad con el algoritmo DT**

| **Clases** | **Sensibilidad** (TP / (TP+FN)) * 100% | **Resultado** |
|---|---|---|
| 2 | (41/(41+0)) * 100% | 100% |
| 3 | (22/(22+0)) * 100% | 100% |
| 4 | (42/(42+0)) * 100 | 100% |
| 5 | (51/(51+0)) * 100 | 100% |
| Total | | 100% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una sensibilidad de 100% utilizando el algoritmo DT.

- **Random Forest**

**Tabla N° 35: Cálculo de la sensibilidad con el algoritmo RF**

| **Clases** | **Sensibilidad** (TP / (TP+FN)) * 100% | **Resultado** |
|---|---|---|
| 2 | (12/(12+0)) * 100% | 100% |
| 3 | (26/(26+0)) * 100% | 100% |
| 4 | (13/(13+0)) * 100 | 100% |
| 5 | (0/(0+1)) * 100 | 0% |
| Total | | 75% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una sensibilidad de 75% utilizando el algoritmo RF.

- **Support Vector Machines**

**Tabla N° 36: Cálculo de la sensibilidad con el algoritmo SVM**

| **Clases** | **Sensibilidad** (TP / (TP+FN)) * 100% | **Resultado** |
|---|---|---|
| 2 | (13/(13+0)) * 100% | 100% |
| 3 | (31/(31+0)) * 100% | 100% |
| 4 | (6/(6+0)) * 100 | 100% |
| 5 | (2/(2+0)) * 100 | 100% |
| Total | | 100% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una sensibilidad de 100% utilizando el algoritmo SVM.

- **Artificial Neural Network**

**Tabla N° 37: Cálculo de la sensibilidad con el algoritmo ANN**

| **Clases** | **Sensibilidad** (TP / (TP+FN)) * 100% | **Resultado** |
|---|---|---|
| 2 | (5/(5+4)) * 100% | 55.55% |
| 3 | (23/(23+6)) * 100% | 79.31% |
| 4 | (12/(12+0)) * 100 | 100% |
| 5 | (0/(0+2)) * 100 | 0% |
| Total | | 58.71% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una sensibilidad de 58.71% utilizando el algoritmo ANN.

- **Gradient Boosting Machines**

**Tabla N° 38: Cálculo de la sensibilidad con el algoritmo GBM**

| **Clases** | **Sensibilidad** (TP / (TP+FN)) * 100% | **Resultado** |
|---|---|---|
| 2 | (9/(9+0)) * 100% | 100% |
| 3 | (30/(30+0)) * 100% | 100% |
| 4 | (12/(12+0)) * 100 | 100% |
| 5 | (1/(1+0)) * 100 | 100% |
| Total | | 100% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una sensibilidad de 100% utilizando el algoritmo GBM.

- **Multi Perceptrón**

**Tabla N° 39: Cálculo de la sensibilidad con el algoritmo MP**

| **Clases** | **Sensibilidad** (TP / (TP+FN)) * 100% | **Resultado** |
|---|---|---|
| 2 | (3/(3+2)) * 100% | 60% |
| 3 | (27/(27+4)) * 100% | 87.09% |
| 4 | (9/(9+5)) * 100 | 64.28% |
| 5 | (0/(0+2)) * 100 | 0% |
| Total | | 52.84% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una sensibilidad de 52.84% utilizando el algoritmo MP.

- **K-Nearest Neighboor**

**Tabla N° 40: Cálculo de la sensibilidad con el algoritmo KNN**

| **Clases** | **Sensibilidad** (TP / (TP+FN)) * 100% | **Resultado** |
|---|---|---|
| 2 | (11/(11+5)) * 100% | 68.75% |
| 3 | (26/(26+0)) * 100% | 100% |
| 4 | (4/(4+4)) * 100 | 50% |
| 5 | (0/(0+2)) * 100 | 0% |
| Total | | 54.68% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una sensibilidad de 54.68% utilizando el algoritmo KNN.

- **Naïve Bayes**

**Tabla N° 41: Cálculo de la sensibilidad con el algoritmo NB**

| **Clases** | **Sensibilidad** (TP / (TP+FN)) * 100% | **Resultado** |
|---|---|---|
| 2 | (0/(0+10)) * 100% | 0% |
| 3 | (30/(30+0)) * 100% | 100% |
| 4 | (0/(0+11)) * 100 | 0% |
| 5 | (0/(0+1)) * 100 | 0% |
| Total | | 25% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una sensibilidad de 25% utilizando el algoritmo NB.

- **Logistic Regression**

**Tabla N° 42: Cálculo de la sensibilidad con el algoritmo LR**

| **Clases** | **Sensibilidad** (TP / (TP+FN)) * 100% | **Resultado** |
|---|---|---|
| 2 | (9/(9+0)) * 100% | 100% |
| 3 | (32/(32+0)) * 100% | 100% |
| 4 | (8/(8+1)) * 100 | 88.88% |
| 5 | (0/(0+2)) * 100 | 0% |
| Total | | 72.22% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una sensibilidad de 72.22% utilizando el algoritmo LR.

- **Ada Boosting**

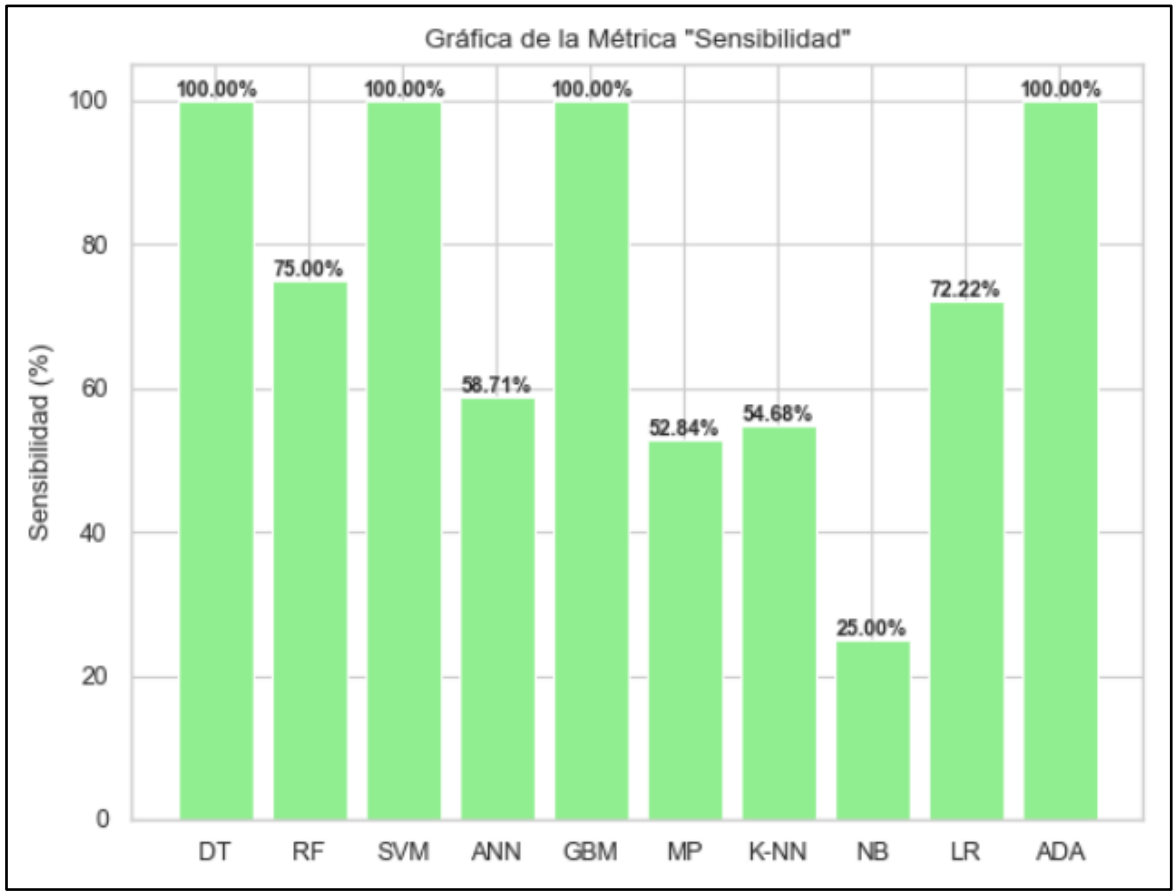
**Tabla N° 43: Cálculo de la sensibilidad con el algoritmo ADA**

| **Clases** | **Sensibilidad** (TP / (TP+FN)) * 100% | **Resultado** |
|---|---|---|
| 2 | (12/(12+0)) * 100% | 100% |
| 3 | (34/(34+0)) * 100% | 100% |
| 4 | (5/(5+0)) * 100 | 100% |
| 5 | (1/(1+0)) * 100 | 100% |
| Total | | 100% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una sensibilidad de 100% utilizando el algoritmo ADA.

**Figura N° 16: Resultados según la métrica de sensibilidad**




**Fuente: Elaboración Propia**

**Interpretación:** En este gráfico se puede observar que los algoritmos con mejores resultados respecto al indicador de sensibilidad fueron: DT, SVM, GBM y ADA con un 100% seguidos de RF con un 75% y LR con un 72.22%.

**Hipótesis 4:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables predecirá con exactitud el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana”

- **Decision Tree**

**Tabla N° 44: Cálculo de la exactitud con el algoritmo DT**

| **Combinación de Clases** | **Exactitud** (TP+TN)/(TP+TN+FP+FN) * 100 | **Resultado** |
|---|---|---|
| | (TP_2+TP_3+TP_4+TP_5)/ (TP_2+TP_3+TP_4+TP_5+FP_2+FP_3+FP_4 +FP_5) | |
| | (11+30+10+1)/(11+30+10+1+0+0+0+0) | 100% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una exactitud de 100% utilizando el algoritmo DT.

- **Random Forest**

**Tabla N° 45: Cálculo de la exactitud con el algoritmo RF**

| **Combinación de Clases** | **Exactitud** (TP+TN)/(TP+TN+FP+FN) * 100 | **Resultado** |
|---|---|---|
| | (TP_2+TP_3+TP_4+TP_5)/ (TP_2+TP_3+TP_4+TP_5+FP_2+FP_3+FP_4 +FP_5) | |
| | (12+26+13+0)/(12+26+13+0+0+1+0) | 98.08% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una exactitud de 98.08% utilizando el algoritmo RF.

- **Support Vector Machines**

**Tabla N° 46: Cálculo de la exactitud con el algoritmo SVM**

| **Combinación de Clases** | **Exactitud** (TP+TN)/(TP+TN+FP+FN) * 100 | **Resultado** |
|---|---|---|
| | (TP_2+TP_3+TP_4+TP_5)/ (TP_2+TP_3+TP_4+TP_5+FP_2+FP_3+FP_4 +FP_5) | |
| | (13+32+6+1)/(13+32+6+1+0+0+0+0) | 100% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una exactitud de 100% utilizando el algoritmo SVM.

- **Artificial Neural Network**

**Tabla N° 47: Cálculo de la exactitud con el algoritmo ANN**

| **Combinación de Clases** | **Exactitud** (TP+TN)/(TP+TN+FP+FN) * 100 | **Resultado** |
|---|---|---|
| | (TP_2+TP_3+TP_4+TP_5)/ (TP_2+TP_3+TP_4+TP_5+FP_2+FP_3+FP_4 +FP_5) | |
| | (5+23+12+0)/(5+23+12+0+3+4+5+0) | 76.92% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una exactitud de 76.92% utilizando el algoritmo ANN.

- **Gradient Boosting Machines**

**Tabla N° 48: Cálculo de la exactitud con el algoritmo GBM**

| **Combinación de Clases** | **Exactitud** (TP+TN)/(TP+TN+FP+FN) * 100 | **Resultado** |
|---|---|---|
| | (TP_2+TP_3+TP_4+TP_5)/ (TP_2+TP_3+TP_4+TP_5+FP_2+FP_3+FP_4 +FP_5) | |
| | (9+30+12+1)/(9+30+12+1+0+0+0+0) | 100% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una exactitud de 100% utilizando el algoritmo GBM.

- **Multi Perceptrón**

**Tabla N° 49: Cálculo de la exactitud con el algoritmo MP**

| **Combinación de Clases** | **Exactitud** (TP+TN)/(TP+TN+FP+FN) * 100 | **Resultado** |
|---|---|---|
| | (TP_2+TP_3+TP_4+TP_5)/ (TP_2+TP_3+TP_4+TP_5+FP_2+FP_3+FP_4 +FP_5) | |
| | (3+27+9+0)/(3+27+9+0+5+6+2+0) | 75% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una exactitud de 75% utilizando el algoritmo MP.

- **K-Nearest Neighboor**

**Tabla N° 50: Cálculo de la exactitud con el algoritmo KNN**

| **Combinación de Clases** | **Exactitud** (TP+TN)/(TP+TN+FP+FN) * 100 | **Resultado** |
|---|---|---|
| | (TP_2+TP_3+TP_4+TP_5)/ (TP_2+TP_3+TP_4+TP_5+FP_2+FP_3+FP_4 +FP_5) | |
| | (11+26+4+0)/(11+26+4+0+0+9+2+0) | 78.84% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una exactitud de 78.84% utilizando el algoritmo KNN.

- **Naïve Bayes**

**Tabla N° 51: Cálculo de la exactitud con el algoritmo NB**

| **Combinación de Clases** | **Exactitud** (TP+TN)/(TP+TN+FP+FN) * 100 | **Resultado** |
|---|---|---|
| | (TP_2+TP_3+TP_4+TP_5)/ (TP_2+TP_3+TP_4+TP_5+FP_2+FP_3+FP_4 +FP_5) | |
| | (0+30+0+0)/(0+30+0+0+0+22+0+0) | 57.69% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una exactitud de 57.69% utilizando el algoritmo NB.

- **Logistic Regression**

**Tabla N° 52: Cálculo de la exactitud con el algoritmo LR**

| **Combinación de Clases** | **Exactitud** (TP+TN)/(TP+TN+FP+FN) * 100 | **Resultado** |
|---|---|---|
| | (TP_2+TP_3+TP_4+TP_5)/ (TP_2+TP_3+TP_4+TP_5+FP_2+FP_3+FP_4 +FP_5) | |
| | (9+32+8+0)/( 9+32+8+0+0+1+2+0) | 94.23% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una exactitud de 94.23% utilizando el algoritmo LR.

- **Ada Boosting**

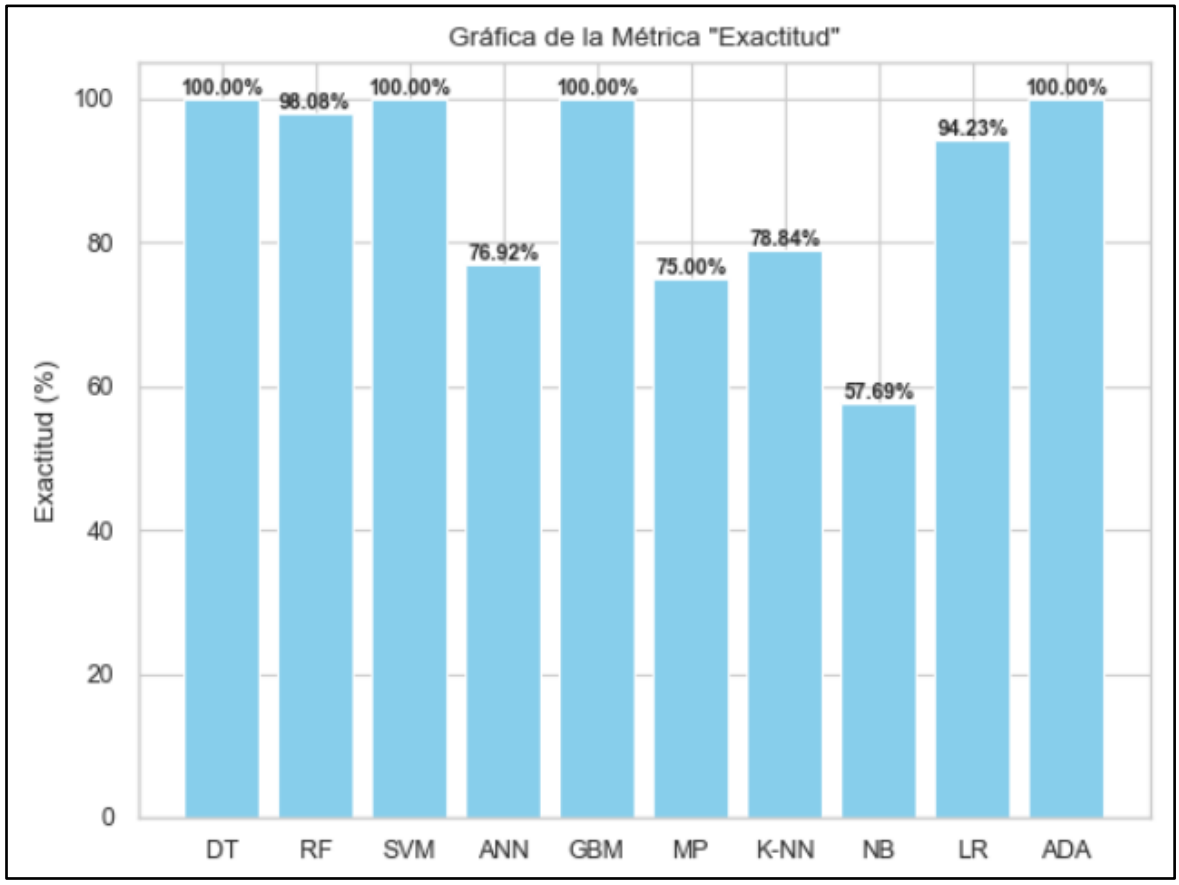
**Tabla N° 53: Cálculo de la exactitud con el algoritmo ADA**

| **Combinación de Clases** | **Exactitud** (TP+TN)/(TP+TN+FP+FN) * 100 | **Resultado** |
|---|---|---|
| | (TP_2+TP_3+TP_4+TP_5)/ (TP_2+TP_3+TP_4+TP_5+FP_2+FP_3+FP_4 +FP_5) | |
| | (12+34+5+1)/(12+34+5+1+0+0+0+0) | 100% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con una exactitud de 100% utilizando el algoritmo ADA.

**Figura N° 17: Resultados según la métrica de exactitud**




**Fuente: Elaboración Propia**

**Interpretación:** En este gráfico se puede observar que los algoritmos con mejores resultados respecto al indicador de sensibilidad fueron: DT, SVM, GBM y ADA con un 100% seguidos de RF con un 98.08% y LR con un 94.23%.

**Hipótesis 5:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables predecirá con F1-score el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana”

- **Decision Tree**

**Tabla N° 54: Cálculo de F1-score con el algoritmo DT**

| **Clases** | **F1-score**<br>2 * ((Precisión * Sensibilidad) / (Precisión + Sensibilidad)) * 100 | **Resultado** |
|---|---|---|
| 2 | 2 * ((1 * 1) / (1 + 1)) * 100 | 100% |
| 3 | 2 * ((1 * 1) / (1 + 1)) * 100 | 100% |
| 4 | 2 * ((1 * 1) / (1 + 1)) * 100 | 100% |
| 5 | 2 * ((1 * 1) / (1 + 1)) * 100 | 100% |
| Total | | 100% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con un F1-score de 100% utilizando el algoritmo DT.

- **Random Forest**

**Tabla N° 55: Cálculo de F1-score con el algoritmo RF**

| **Clases** | **F1-score**<br>2 * ((Precisión * Sensibilidad) / (Precisión + Sensibilidad)) * 100 | **Resultado** |
|---|---|---|
| 2 | 2 * ((1 * 1) / (1 + 1)) * 100 | 100% |
| 3 | 2 * ((1* 1) / (1 + 1)) * 100 | 100% |
| 4 | 2 * ((0.92* 1) / (0.92 + 1)) * 100 | 96.29% |
| 5 | 2 * ((0 * 0) / (0 + 0)) * 100 | 0% |

| | |
|---|---|
| Total | 74.07% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con un F1-score de 74.07% utilizando el algoritmo RF.

- **Support Vector Machines**

**Tabla N° 56: Cálculo de F1-score con el algoritmo SVM**

| **Clases** | **F1-score**<br>2 * ((Precisión * Sensibilidad) / (Precisión + Sensibilidad)) * 100 | **Resultado** |
|---|---|---|
| 2 | 2 * ((1 * 1) / (1 + 1)) * 100 | 100% |
| 3 | 2 * ((1 * 1) / (1 + 1)) * 100 | 100% |
| 4 | 2 * ((1 * 1) / (1 + 1)) * 100 | 100% |
| 5 | 2 * ((1 * 1) / (1 + 1)) * 100 | 100% |
| Total | | 100% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con un F1-score de 100% utilizando el algoritmo SVM.

- **Artificial Neural Network**

**Tabla N° 57: Cálculo de la sensibilidad con el algoritmo ANN**

| **Clases** | **F1-score**<br>2 * ((Precisión * Sensibilidad) / (Precisión + Sensibilidad)) * 100 | **Resultado** |
|---|---|---|
| 2 | 2 * ((0.62 * 0.55) / (0.62 + 0.55)) * 100 | 58.82% |

| | | |
|---|---|---|
| 3 | 2 * ((0.85 * 0.79) / (0.85 + 0.79)) * 100 | 82.14% |
| 4 | 2 * ((0.70 * 1.0) / (0.70 + 1.0)) * 100 | 82.75% |
| 5 | 2 * ((0 * 0) / (0 + 0)) * 100 | 0% |
| Total | | 55.93% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con un F1-score de 55.93% utilizando el algoritmo ANN.

- **Gradient Boosting Machines**

**Tabla N° 58: Cálculo de F1-score con el algoritmo GBM**

| **Clases** | **F1-score**<br>2 * ((Precisión * Sensibilidad) / (Precisión + Sensibilidad)) * 100 | **Resultado** |
|---|---|---|
| 2 | 2 * ((1 * 1) / (1 + 1)) * 100 | 100% |
| 3 | 2 * ((1 * 1) / (1 + 1)) * 100 | 100% |
| 4 | 2 * ((1 * 1) / (1 + 1)) * 100 | 100% |
| 5 | 2 * ((1 * 1) / (1 + 1)) * 100 | 100% |
| Total | | 100% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con un F1-score de 100% utilizando el algoritmo GBM.

- **Multi Perceptrón**

**Tabla N° 59: Cálculo de F1-score con el algoritmo MP**

| **Clases** | **F1-score**<br>2 * ((Precisión * Sensibilidad) / (Precisión + Sensibilidad)) * 100 | **Resultado** |
|---|---|---|
| 2 | 2 * ((0.37 * 0.60) / (0.37 + 0.60)) * 100 | 46.15% |
| 3 | 2 * ((0.81 * 0.87) / (0.81 + 0.87)) * 100 | 84.37% |
| 4 | 2 * ((0.81 * 0.64) / (0.81 + 0.64)) * 100 | 73.68% |
| 5 | 2 * ((0 * 0) / (0 + 0)) * 100 | 0% |
| Total | | 59.90% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con un F1-score de 59.90% utilizando el algoritmo MP.

- **K-Nearest Neighboor**

**Tabla N° 60: Cálculo de F1-score con el algoritmo KNN**

| **Clases** | **F1-score**<br>2 * ((Precisión * Sensibilidad) / (Precisión + Sensibilidad)) * 100 | **Resultado** |
|---|---|---|
| 2 | 2 * ((1.0 * 0.68) / (1.0+ 0.68)) * 100 | 81.48% |
| 3 | 2 * ((0.74 * 1.0) / (0.74 + 1.0)) * 100 | 85.24% |
| 4 | 2 * ((0.66 * 0.5) / (0.66 + 0.5)) * 100 | 57.14% |
| 5 | 2 * ((0 * 0) / (0 + 0)) * 100 | 0% |
| Total | | 55.96% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con un F1-score de 55.96% utilizando el algoritmo KNN.

- **Naïve Bayes**

**Tabla N° 61: Cálculo de F1-score con el algoritmo NB**

| **Clases** | **F1-score**<br>2 * ((Precisión * Sensibilidad) / (Precisión + Sensibilidad)) * 100 | **Resultado** |
|---|---|---|
| 2 | 2 * ((0 * 0) / (0+ 0)) * 100 | 0% |
| 3 | 2 * ((0.57 * 1) / (0.57 + 1)) * 100 | 73.17% |
| 4 | 2 * ((0 * 0) / (0 + 0)) * 100 | 0% |
| 5 | 2 * ((0 * 0) / (0 + 0)) * 100 | 0% |
| Total | | 18.29% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con un F1-score de 18.29% utilizando el algoritmo NB.

- **Logistic Regression**

**Tabla N° 62: Cálculo de F1-score con el algoritmo LR**

| **Clases** | **F1-score**<br>2 * ((Precisión * Sensibilidad) / (Precisión + Sensibilidad)) * 100 | **Resultado** |
|---|---|---|
| 2 | 2 * ((1 * 1) / (1+ 1)) * 100 | 100% |
| 3 | 2 * ((0.96 * 1) / (0.96+ 1)) * 100 | 98.46% |
| 4 | 2 * ((0.80 * 0.88) / (0.80 + 0.88)) * 100 | 84.21% |
| 5 | 2 * ((0 * 0) / (0 + 0)) * 100 | 0% |

| | |
|---|---|
| Total | 70.66% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con un F1-score de 70.66% utilizando el algoritmo LR.

- **Ada Boosting**

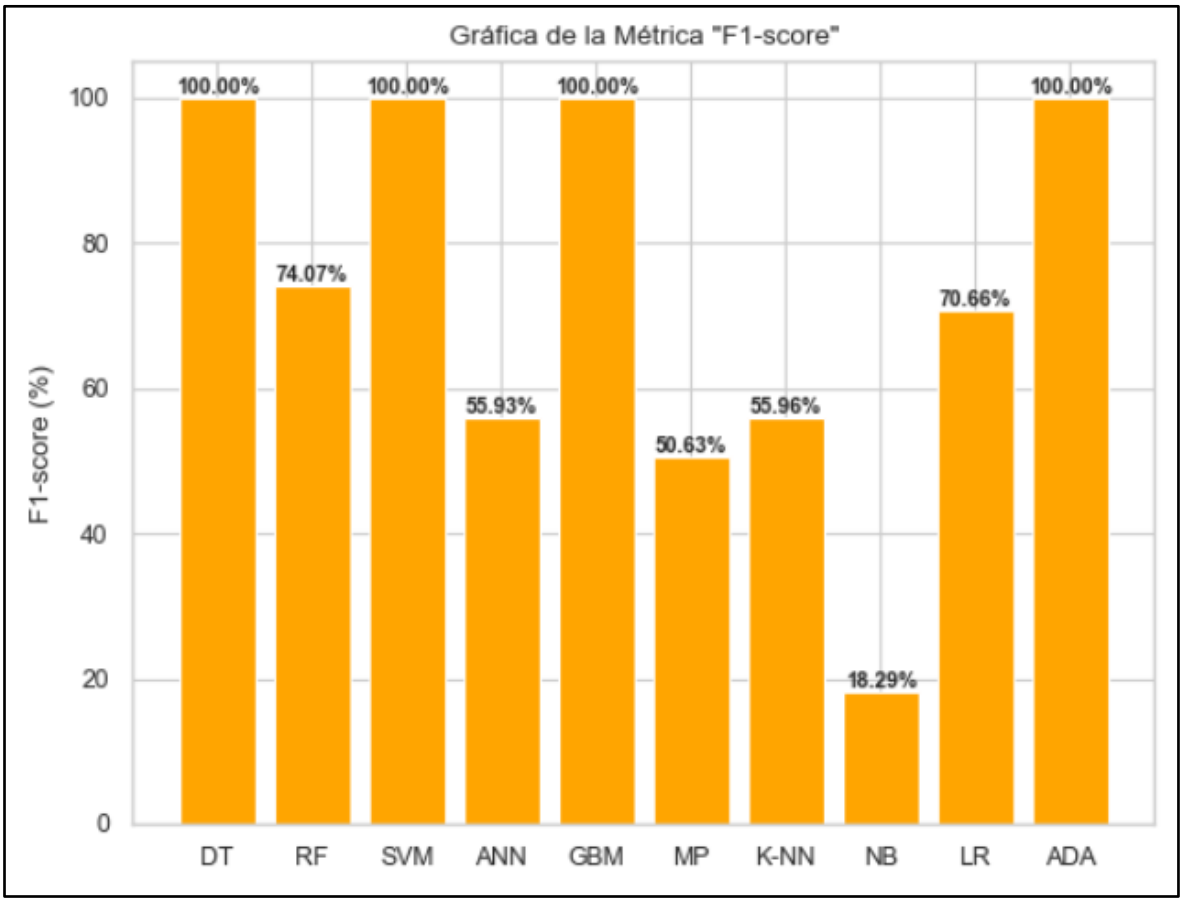
**Tabla N° 63: Cálculo de F1-score con el algoritmo ADA**

| **Clases** | **F1-score**<br>2 * ((Precisión * Sensibilidad) / (Precisión + Sensibilidad)) * 100 | **Resultado** |
|---|---|---|
| 2 | 2 * ((1 * 1) / (1+ 1)) * 100 | 100% |
| 3 | 2 * ((1 * 1) / (1+ 1)) * 100 | 100% |
| 4 | 2 * ((1 * 1) / (1+ 1)) * 100 | 100% |
| 5 | 2 * ((1 * 1) / (1+ 1)) * 100 | 100% |
| Total | | 100% |

**Fuente: Elaboración Propia**

**Interpretación:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables permitió predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana” con un F1-score de 100% utilizando el algoritmo ADA.

**Figura N° 18: Resultados según la métrica de F1-score**

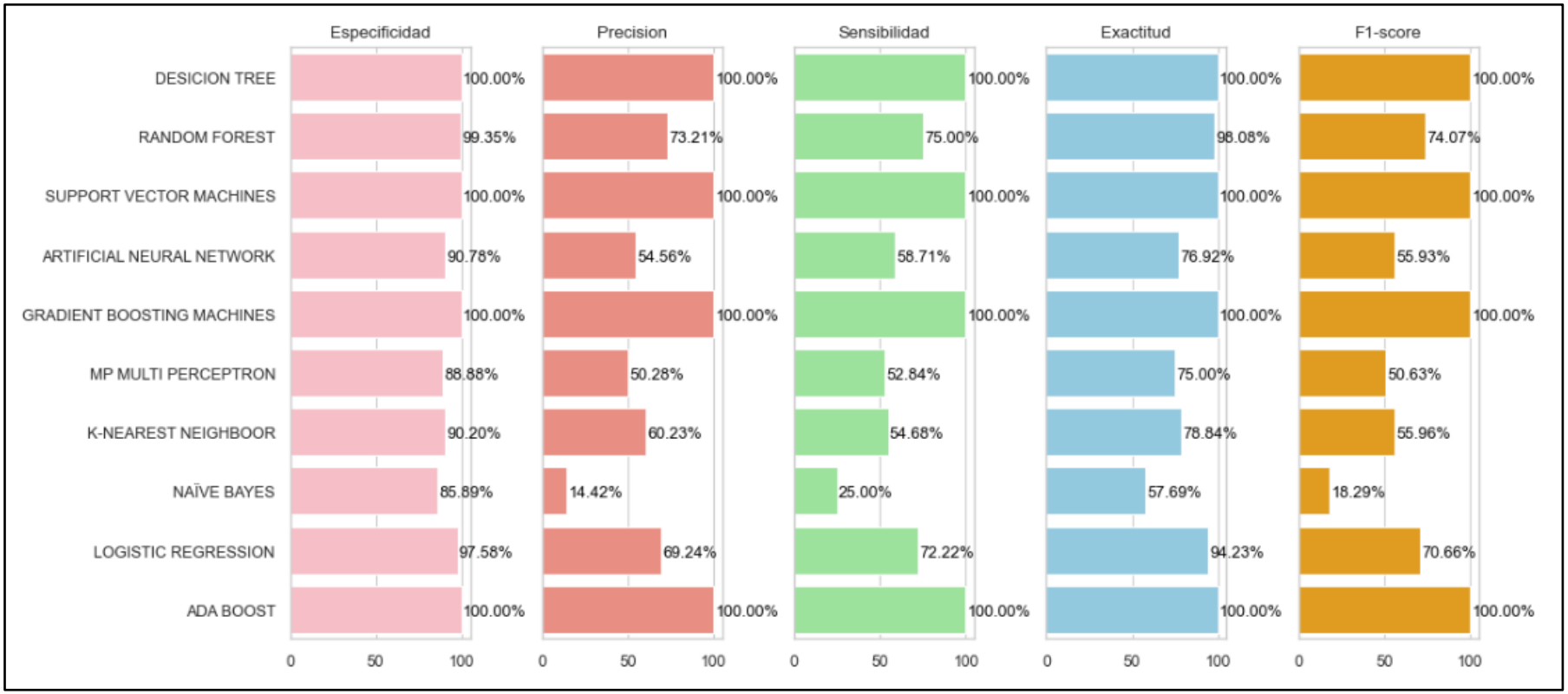



**Fuente: Elaboración Propia**

**Interpretación:** En este gráfico se puede observar que los algoritmos con mejores resultados respecto al indicador de F1-score fueron: DT, SVM, GBM y ADA con un 100% seguidos de RF con un 74.07% y LR con un 70.66%.

**Hipótesis General:** El desarrollo de un Sistema Inteligente con Machine Learning basado en selección de variables predecirá el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana”

**Figura N° 19: Resultados según todas las métricas**




**Fuente: Elaboración Propia**

**Interpretación:** En este gráfico se puede observar que los algoritmos con mejores resultados respecto todos los indicadores fueron: DT, SVM, GBM y ADA con un 100%.

Por otra parte, para validar la concordancia y medir la significancia entre dos clasificadores, se utiliza la Kappa de Cohen. Al respecto, Salas y Muñoz (2019) comentan que este coeficiente refleja la concordancia entre dos observadores y puede tomar los valores entre -1 y +1. Mientras más cercano sea a +1 existirá un mayor grado de concordancia (p. 2).

**Figura N° 20: Medición de Kappa de Cohen**

| Valores | Interpretación |
|---|---|
| < 0,01 | No acuerdo |
| 0,01 – 0,20 | Ninguna a escaso |
| 0,21 – 0,40 | Regular o razonable |
| 0,41 – 0,60 | Moderado |
| 0,61 – 0,80 | Substancial |
| 0,81 – 1,00 | Casi perfecto |

**Fuente: Manterola (2018)**

Los resultados obtenidos por el conjunto de validación de cada algoritmo fueron los siguientes:

**Tabla N° 64: Medida de Kappa de Cohen - DT**

| Medidas simétricas | | | | | |
|---|---|---|---|---|---|
| | | Valor | Error estándar asintótico | T aproximada | Significación aproximada |
| Medida de acuerdo | Kappa | 1.0 | 1.0 | 1.0 | 0.32 |
| Número de casos válidos | | 52 | | | |

**Fuente: Elaboración Propia**

**Interpretación**: Siendo el valor de Kappa igual a 1, se deduce que existe muy buena concordancia entre las etiquetas observadas y las etiquetas predichas por el algoritmo DT. De igual manera, muestra una significancia menor a 5%, es así como se aprobó la hipótesis al utilizar el algoritmo DT.

**Tabla N° 65: Medida de Kappa de Cohen – RF**

| Medidas simétricas | | | | |
|---|---|---|---|---|
| | Valor | Error estándar asintótico | T aproximada | Significación aproximada |

| | | | | | |
|---|---|---|---|---|---|
| Medida de acuerdo | Kappa | 0.97 | 0.97 | 0.99 | 0.32 |
| Número de casos válidos | | 52 | | | |

**Fuente: Elaboración Propia**

**Interpretación:** Siendo el valor de Kappa igual a 0.97, se deduce que existe una concordancia casi perfecta entre las etiquetas observadas y las etiquetas predichas por el algoritmo RF. De igual manera, muestra una significancia menor a 5%, es así como se aprobó la hipótesis al utilizar el algoritmo RF.

**Tabla N° 66: Medida de Kappa de Cohen - SVM**

| Medidas simétricas | | | | | |
|---|---|---|---|---|---|
| | | Valor | Error estándar asintótico | T aproximada | Significación aproximada |
| Medida de acuerdo | Kappa | 1.0 | 1.0 | 1.0 | 0.32 |
| Número de casos válidos | | 52 | | | |

**Fuente: Elaboración Propia**

**Interpretación:** Siendo el valor de Kappa igual a 1, se determina que hay muy buena concordancia entre las etiquetas observadas y las etiquetas predichas por el algoritmo SVM. Igualmente, muestra una significancia menor a 5%, es así como se aprobó la hipótesis al utilizar el algoritmo SVM.

**Tabla N° 67: Medida de Kappa de Cohen - ANN**

| Medidas simétricas | | | | | |
|---|---|---|---|---|---|
| | | Valor | Error estándar asintótico | T aproximada | Significación aproximada |
| Medida de acuerdo | Kappa | 0.62 | 0.68 | 0.91 | 0.36 |
| Número de casos válidos | | 52 | | | |

**Fuente: Elaboración Propia**

**Interpretación:** Siendo el valor de Kappa igual a 0.62, se deduce que existe una concordancia sustancial entre las etiquetas observadas y las etiquetas predichas

por el algoritmo ANN. Por otra parte, muestra una significancia menor a 5%, es así como se aprobó la hipótesis al utilizar el algoritmo ANN.

**Tabla N° 68: Medida de Kappa de Cohen - GBM**

| Medidas simétricas | | | | | |
|---|---|---|---|---|---|
| | | Valor | Error estándar asintótico | T aproximada | Significación aproximada |
| Medida de acuerdo | Kappa | 1.0 | 1.0 | 1.0 | 0.32 |
| Número de casos válidos | | 52 | | | |

**Fuente: Elaboración Propia**

**Interpretación:** Siendo el valor de Kappa igual a 1, se deduce que existe una concordancia perfecta entre las etiquetas observadas y las etiquetas predichas por el algoritmo GBM. Además, muestra una significancia menor a 5%, por lo que, se aprobó la hipótesis al utilizar el algoritmo GBM.

**Tabla N° 69: Medida de Kappa de Cohen - MP**

| Medidas simétricas | | | | | |
|---|---|---|---|---|---|
| | | Valor | Error estándar asintótico | T aproximada | Significación aproximada |
| Medida de acuerdo | Kappa | 0.54 | 0.58 | 0.93 | 0.35 |
| Número de casos válidos | | 52 | | | |

**Fuente: Elaboración Propia**

**Interpretación:** Siendo el valor de Kappa igual a 0.54, se deduce que existe una concordancia moderada entre las etiquetas observadas y las etiquetas predichas por el algoritmo MP. De igual manera, muestra una significancia menor a 5%, es así como se aprobó la hipótesis al utilizar el algoritmo MP.

**Tabla N° 70: Medida de Kappa de Cohen – KNN**

| Medidas simétricas | | | | | |
|---|---|---|---|---|---|
| | | Valor | Error estándar asintótico | T aproximada | Significación aproximada |

| | | | | | |
|---|---|---|---|---|---|
| Medida de acuerdo | Kappa | 0.64 | 0.69 | 0.92 | 0.36 |
| Número de casos válidos | | 52 | | | |

**Fuente: Elaboración Propia**

**Interpretación:** Siendo el valor de Kappa igual a 0.64, se deduce que existe una concordancia substancial entre las etiquetas observadas y las etiquetas predichas por el algoritmo KNN. De igual manera, muestra una significancia menor a 5%, es así como se aprobó la hipótesis al utilizar el algoritmo KNN.

**Tabla N° 71: Medida de Kappa de Cohen - NB**

| Medidas simétricas | | | | | |
|---|---|---|---|---|---|
| | | Valor | Error estándar asintótico | T aproximada | Significación aproximada |
| Medida de acuerdo | Kappa | 0 | 0 | | |
| Número de casos válidos | | 52 | | | |

**Fuente: Elaboración Propia**

**Interpretación:** Siendo el valor de Kappa igual a 0, se deduce que existe una concordancia no acuerdo entre las etiquetas observadas y las etiquetas predichas por el algoritmo NB. De igual manera, no logra mostrar una significancia aproximada, es así como se rechazó la hipótesis al utilizar el algoritmo NB.

**Tabla N° 72: Medida de Kappa de Cohen – LR**

| Medidas simétricas | | | | | |
|---|---|---|---|---|---|
| | | Valor | Error estándar asintótico | T aproximada | Significación aproximada |
| Medida de acuerdo | Kappa | 0.89 | 0.91 | 0.98 | 0.33 |
| Número de casos válidos | | 52 | 52 | | |

**Fuente: Elaboración Propia**

**Interpretación:** Siendo el valor de Kappa igual a 0.89, se deduce que existe una concordancia casi perfecta entre las etiquetas observadas y las etiquetas predichas

por el algoritmo LR. De igual manera, muestra una significancia menor a 5%, es así como se aprobó la hipótesis al utilizar el algoritmo LR.

**Tabla N° 73: Medida de Kappa de Cohen - ADA**

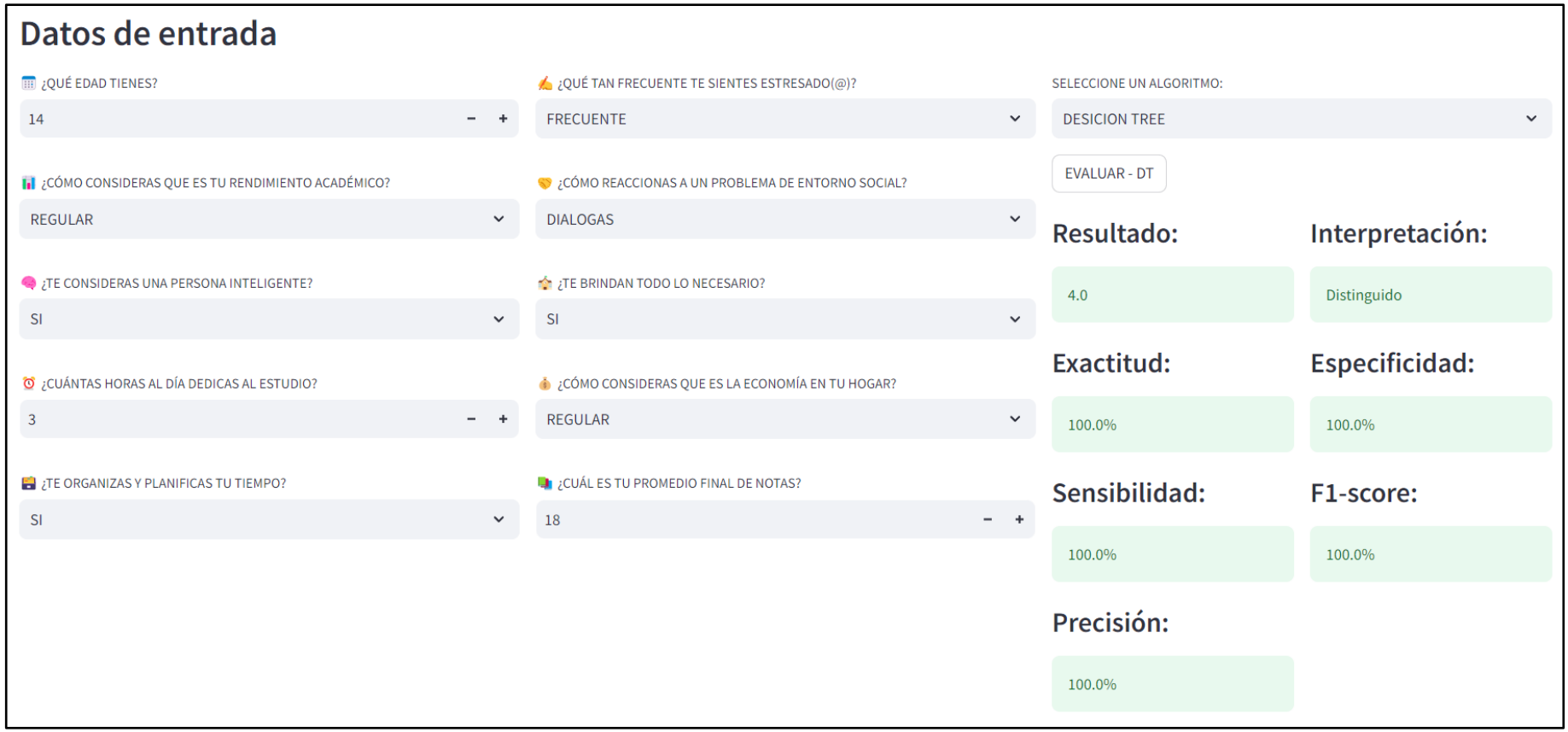
| Medidas simétricas | | | | | |
|---|---|---|---|---|---|
| | | Valor | Error estándar asintótico | T aproximada | Significación aproximada |
| Medida de acuerdo | Kappa | 1.0 | 1.0 | 1.0 | 0.32 |
| Número de casos válidos | | 52 | | | |

**Fuente: Elaboración Propia**

**Interpretación:** Siendo el valor de Kappa igual a 1, se deduce que existe una excelente concordancia entre las etiquetas observadas y las etiquetas predichas por el algoritmo ADA. También, muestra una significancia menor a 5%, es así como se aprobó la hipótesis al utilizar el algoritmo ADA.

Para concluir, los algoritmos que obtuvieron un mejor nivel de concordancia fueron: DT, SVM, GBM y ADA con un 1.0 según las medidas de acuerdo de la Kappa de Cohen, además de obtener un grado de significancia menor a 5%, por ende, se aprobó la hipótesis con estos algoritmos.

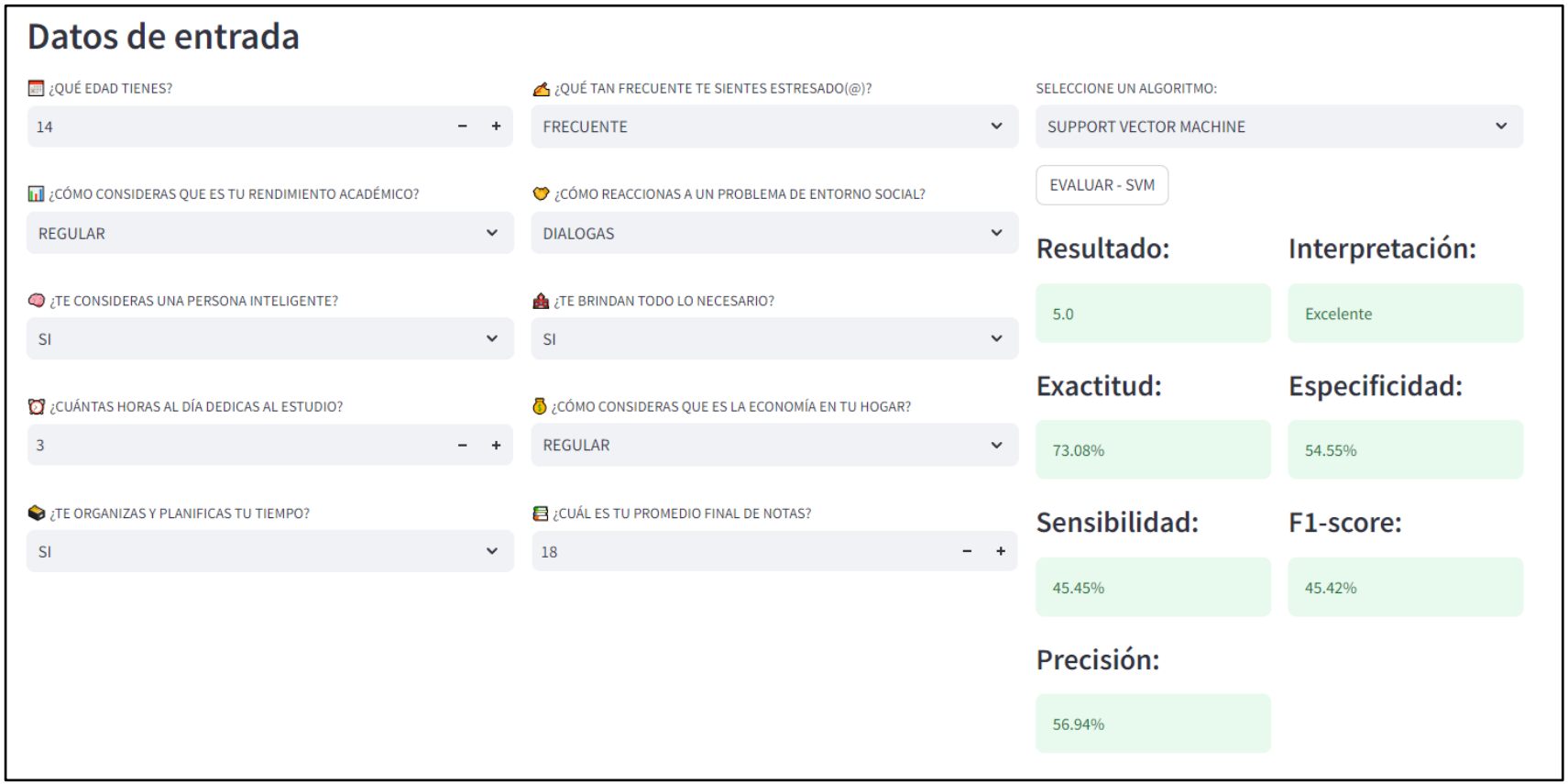
**Figura N° 21: Resultado del Sistema Inteligente con el modelo DT**




**Fuente: Elaboración Propia**

En esta figura, se visualiza que el resultado de una evaluación en específico con el modelo DT, tiene un valor de 4, es decir, el rendimiento académico es distinguido. Además, se obtuvo un 100% en todas las métricas de evaluación.

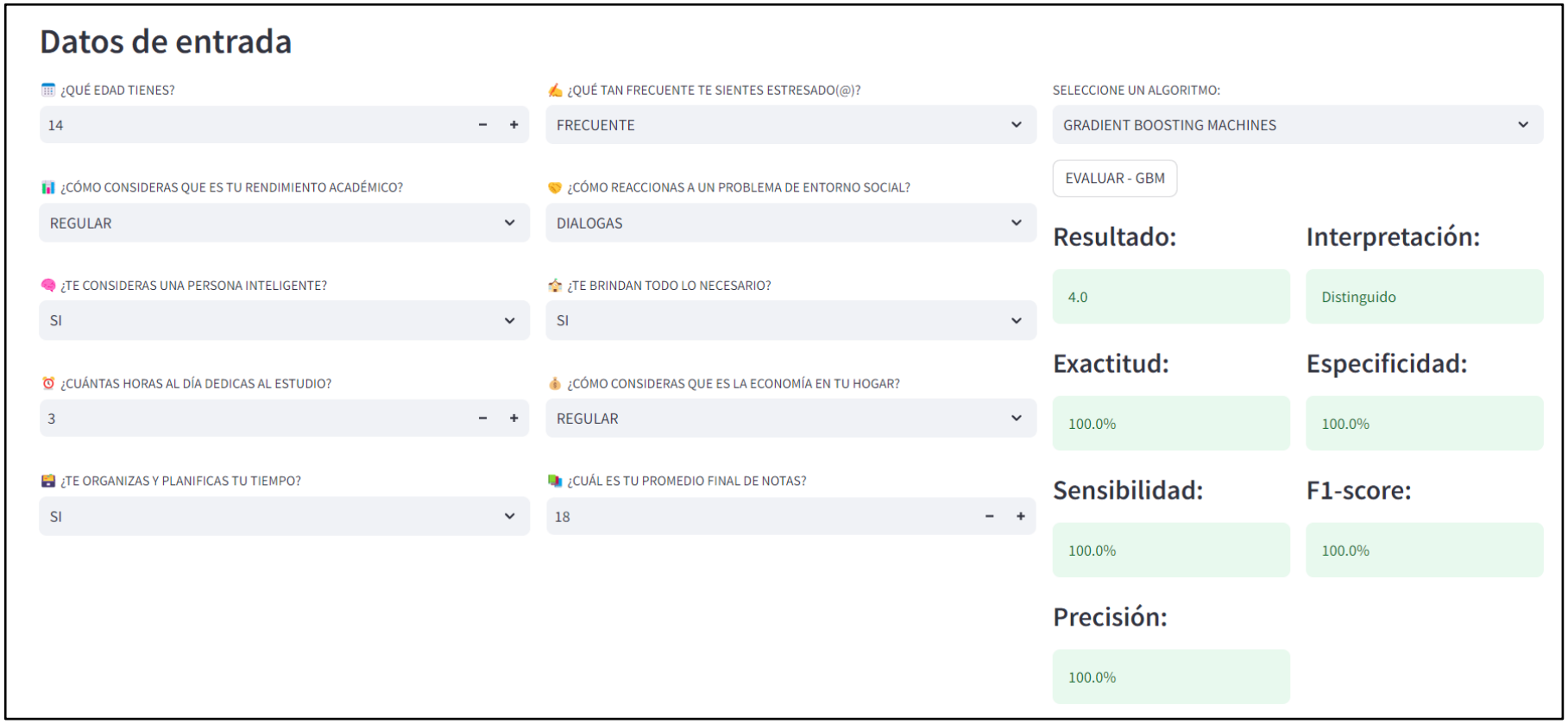
**Figura N° 22: Resultado del Sistema Inteligente con el modelo SVM**




**Fuente: Elaboración Propia**

Utilizando los mismos datos de entrada, se visualiza que el resultado con el modelo SVM, tiene un valor de 5, es decir, el rendimiento académico es excelente. No obstante, obtiene valores distintos al 100% en cada una de las métricas.

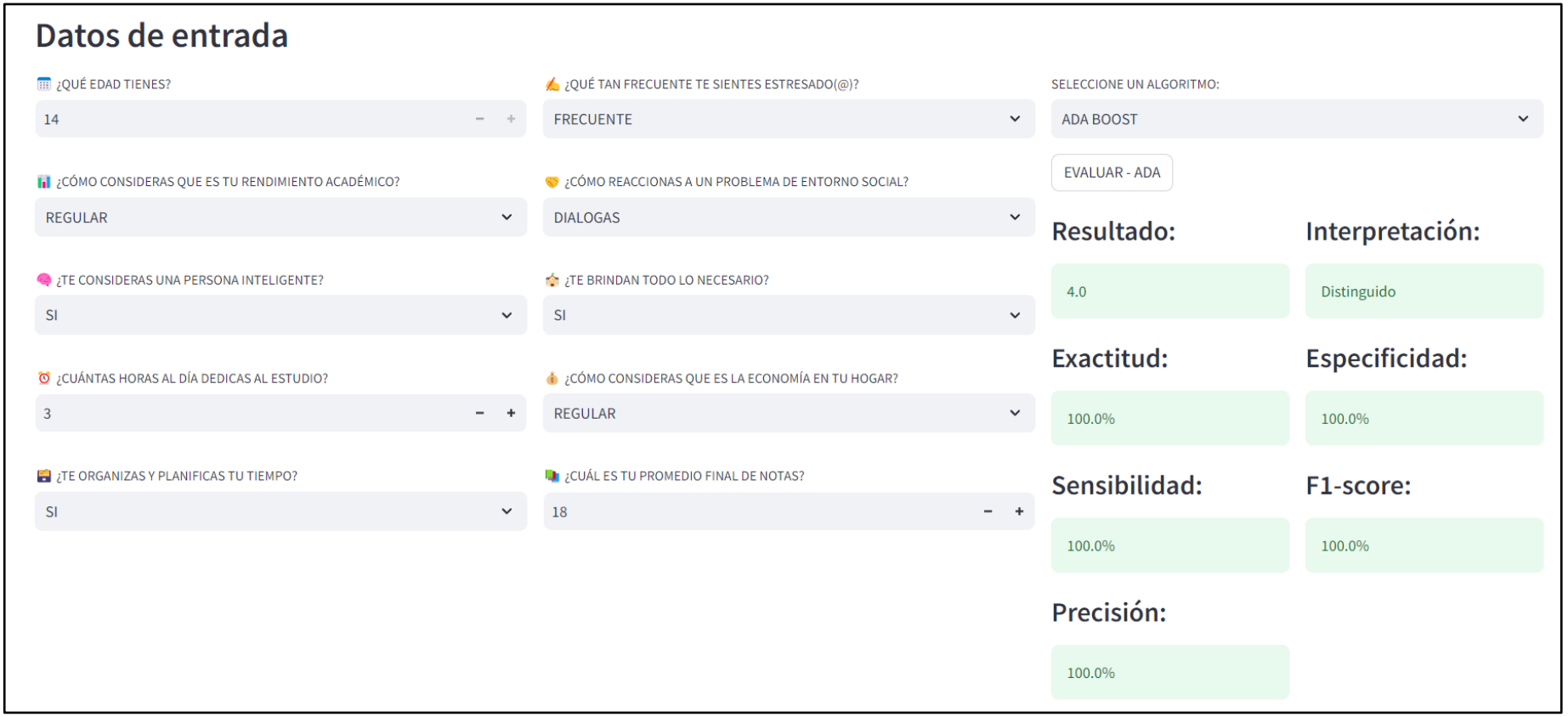
**Figura N° 23: Resultado del Sistema Inteligente con el modelo GBM**




**Fuente: Elaboración Propia**

De igual manera, se visualiza que el resultado con el modelo GBM, tiene un valor de 4, es decir, el rendimiento académico es distinguido, obteniendo el 100% en todas las métricas al igual que el modelo DT.

**Figura N° 24: Resultado del Sistema Inteligente con el modelo ADA**




**Fuente: Elaboración Propia**

Asimismo, se visualiza que el resultado con el modelo ADA, tiene un valor de 4, es decir, el rendimiento académico es distinguido. Este modelo también obtuvo un 100% en todas las métricas de evaluación.

**Figura N° 25: Resultados de evaluación con documento Excel**


| | EDAD | NIVEL_REND_ACAD | CONS_INTELIGENTE | HORAS_ESTUDIO | CONS_ORGANIZACION | CANT_ESTRES | REAC_PROB_SOCIAL | APOYO_ACAD | ECONOMIA_HOGAR | PROMEDIO_CALIF_NOTAS | RENDIMIENTO ACADEMICO | EXACTITUD |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 13 | 2 | 1 | 3 | 2 | 1 | 1 | 1 | 3 | 16 | 3 | 100.0% |
| 1 | 14 | 3 | 2 | 4 | 1 | 4 | 2 | 2 | 4 | 17 | 4 | 100.0% |
| 2 | 15 | 3 | 1 | 2 | 1 | 1 | 2 | 2 | 5 | 18 | 4 | 100.0% |
| 3 | 12 | 4 | 1 | 3 | 2 | 5 | 3 | 1 | 2 | 14 | 3 | 100.0% |
| 4 | 12 | 3 | 2 | 5 | 1 | 2 | 1 | 1 | 4 | 15 | 3 | 100.0% |
| 5 | 16 | 5 | 1 | 1 | 0 | 3 | 1 | 2 | 4 | 15 | 3 | 100.0% |
| 6 | 17 | 5 | 2 | 2 | 0 | 3 | 2 | 2 | 5 | 16 | 3 | 100.0% |
| 7 | 17 | 2 | 2 | 2 | 1 | 4 | 3 | 1 | 5 | 17 | 4 | 100.0% |
| 8 | 18 | 1 | 2 | 4 | 2 | 5 | 2 | 2 | 3 | 18 | 4 | 100.0% |
| 9 | 15 | 2 | 1 | 4 | 2 | 1 | 2 | 1 | 2 | 13 | 2 | 100.0% |

**Fuente: Elaboración Propia**

En este apartado del sistema, se evidencia como evaluar una cierta cantidad de datos a la vez mediante un documento Excel, dónde al procesarlo calcula el valor del rendimiento académico y el porcentaje de exactitud con el que fue calculado.

**Figura N° 26: Resultados exportados en documento Excel**

| EDAD | NIVEL_REND_ACAD | CONS_INTELIGENTE | HORAS_ESTUDIO | CONS_ORGANIZACION | CANT_ESTRES | REAC_PROB_SOCIAL | APOYO_ACAD | ECONOMIA_HOGAR | PROMEDIO_CALIF_NOTAS | RENDIMIENTO ACADEMICO | EXACTITUD |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 13 | 2 | 1 | 3 | 2 | 1 | 1 | 1 | 3 | 16 | 3 | 100.00% |
| 14 | 3 | 2 | 4 | 1 | 4 | 2 | 2 | 4 | 17 | 4 | 100.00% |
| 15 | 3 | 1 | 2 | 1 | 1 | 2 | 2 | 5 | 18 | 4 | 100.00% |
| 12 | 4 | 1 | 3 | 2 | 5 | 3 | 1 | 2 | 14 | 3 | 100.00% |
| 12 | 3 | 2 | 5 | 1 | 2 | 1 | 1 | 4 | 15 | 3 | 100.00% |
| 16 | 5 | 1 | 1 | 0 | 3 | 1 | 2 | 4 | 15 | 3 | 100.00% |
| 17 | 5 | 2 | 2 | 0 | 3 | 2 | 2 | 5 | 16 | 3 | 100.00% |
| 17 | 2 | 2 | 2 | 1 | 4 | 3 | 1 | 5 | 17 | 4 | 100.00% |
| 18 | 1 | 2 | 4 | 2 | 5 | 2 | 2 | 3 | 18 | 4 | 100.00% |
| 15 | 2 | 1 | 4 | 2 | 1 | 2 | 1 | 2 | 13 | 2 | 100.00% |
| 13 | 3 | 2 | 5 | 1 | 2 | 1 | 1 | 2 | 12 | 2 | 100.00% |
| 13 | 4 | 1 | 5 | 1 | 3 | 3 | 2 | 5 | 11 | 2 | 100.00% |
| 16 | 5 | 2 | 3 | 0 | 3 | 3 | 2 | 4 | 11 | 2 | 100.00% |
| 17 | 5 | 2 | 4 | 1 | 4 | 2 | 1 | 2 | 15 | 3 | 100.00% |
| 18 | 3 | 1 | 5 | 2 | 5 | 2 | 2 | 3 | 16 | 3 | 100.00% |
| 12 | 3 | 2 | 0 | 2 | 1 | 1 | 2 | 3 | 17 | 4 | 100.00% |
| 14 | 2 | 1 | 0 | 2 | 1 | 3 | 2 | 5 | 18 | 4 | 100.00% |
| 14 | 1 | 1 | 3 | 1 | 2 | 1 | 1 | 4 | 19 | 5 | 100.00% |
| 16 | 1 | 1 | 4 | 1 | 3 | 2 | 1 | 5 | 20 | 5 | 100.00% |
| 15 | 4 | 1 | 2 | 2 | 3 | 3 | 2 | 3 | 16 | 3 | 100.00% |

**Fuente: Elaboración Propia**

En esta figura, se puede evidenciar el documento Excel descargado de la anterior Figura N° 25. La finalidad de este flujo es poder evaluar una cantidad de datos mayor a 1 y tenerlo registrado en un documento, facilitando así la evaluación de múltiples alumnos en lugar de 1 a la vez.

# V. DISCUSIÓN

A continuación, se presentan las discusiones basadas en los resultados obtenidos durante el desarrollo del estudio en base a las evidencias recopiladas durante el estudio, así como la comparación e interpretación detallada de los hallazgos y su relevancia en el contexto de la investigación.

En cuanto al objetivo general, el cual se desarrolló un sistema inteligente con machine learning que se basa en la selección de variables para predecir el rendimiento académico de la I.E.P “Nuestra Señora de Copacabana”. Para ello, se aplicó la metodología KDD, que consiste en el descubrimiento de conocimientos útiles a partir de datos. Esta metodología simplifica el proceso, procesamiento y la construcción de modelos de aprendizaje automático para realizar predicciones en base a datos. Autores como Orsoni et al. (2023), Kannan, Abarna y Vairachilai (2023), Owusu-Boadu et al. (2021), García (2021), Urkude y Gupta (2019) y Mounika y Persis (2019), utilizaron esta metodología para la extracción de conocimiento. A diferencia de Ojajuni et al. (2021), utilizó una metodología similar con base en cinco procesos diferentes, del mismo modo Al-Alawi et al. (2022) que baso su investigación utilizando ocho fases: especificación del problema, obtención de recursos, limpieza de datos, preprocesamiento, minería de datos, evaluación, interpretación y explotación. Dinh et al. (2020) utilizo su propio modelo de predicción, que comparte similitudes con la metodología KDD en términos de procesos generales y fases de recopilación e integración, procesamiento, construcción, evaluación del modelo y finalmente Wang et al. (2022) se basó en el proceso de recopilación de datos y aplico la prueba de Chi-Cuadrado para el análisis, validación e identificación.

Para desarrollar del modelo predictivo se utilizaron las siguientes herramientas: Base de datos SQL, para el almacenamiento y manipulación de datos, Jupyter Notebook, como plataforma local web interactiva compatible con el lenguaje de programación Python y el análisis estadístico, como entorno de desarrollo Spyder, para el desarrollo del sistema. Autores como García (2021), Wang et al. (2022) utilizaron para el análisis estadístico y la minería de datos, la herramienta de modelado el SPSS y Ojajuni et al. (2021) utilizó las herramientas Python con las bibliotecas Scikit-learn y TensorFlow como entorno flexible y de alto

rendimiento para construir y entrenar modelos de aprendizaje automático a gran escala.

En lo que respecta al objetivo específico 1, a partir de la comparación entre los algoritmos de aprendizaje automático DT, RF, SVM, ANN, GBM, MP, KNN, NB, LR y ADA se observó que los mejores modelos se obtuvieron utilizando los algoritmos DT, SVM, GBM y AB, de las medidas utilizadas para evaluar el rendimiento y calidad del modelo en **especificidad** se obtuvieron resultados del 100% para la predicción del rendimiento académico. Con lo mencionado anteriormente, tiene similitud con la investigación de García (2021) utilizó los algoritmos de SVM con un 100%, DT obtuvo un 91.92% y K-NN con un resultado del 84.00%.

Al contrario, los siguientes autores Al-Alawi et al. (2022), Orsoni et al. (2023), Owusu-Boadu et al. (2021), Kannan, Abarna y Vairachilai (2023), Wang et al. (2022), Urkude y Gupta (2019), Mounika y Persis (2019) y Dinh et al. (2020), no consideraron esta métrica en sus investigaciones, ya que se enfocaron en las métricas de exactitud, sensibilidad, precisión y F1 - score.

En lo que respecta al objetivo específico 2, a partir de la comparación entre los algoritmos de aprendizaje automático DT, RF, SVM, ANN, GBM, MP, KNN, NB, LR y ADA se observó que los mejores modelos se obtuvieron utilizando los algoritmos DT, SVM, GBM y AB, de las medidas utilizadas para evaluar el rendimiento y calidad del modelo en **precisión** se llegó al 100% de los resultados obtenidos para la predicción del rendimiento académico. Con lo mencionado anteriormente, se obtuvieron los resultados similares en las siguientes investigaciones: Wang et al. (2022) con base en los algoritmos LR con un 81.61%, NB un 73.57%, RF un 78.42% y SVM con 80.96%.

Al contrario, los siguientes autores Orsoni et al. (2023), Kannan, Abarna y Vairachilai (2023), Al-Alawi et al. (2022), Owusu-Boadu, García (2021), Ojajuni et al. (2021), Dinh et al. (2020), Urkude y Gupta (2019) y Mounika y Persis (2019), no consideraron esta métrica en sus investigaciones, ya que se enfocaron en las métricas de exactitud, sensibilidad, especificidad y F1 - score.

En lo que respecta al objetivo específico 3, a partir de la comparación entre los algoritmos de aprendizaje automático DT, RF, SVM, ANN, GBM, MP, KNN, NB, LR y ADA se observó que los mejores modelos se obtuvieron utilizando los algoritmos DT, SVM, GBM y AB, de las medidas utilizadas para evaluar el rendimiento y calidad del modelo en **sensibilidad** se obtuvo el 100% para la predicción del rendimiento académico. Con lo mencionado anteriormente, en similitud con las investigaciones de García (2021) detalla que en los algoritmos SVM tuvo 100%, DT un 85.05% y KNN con un 72.41% y Wang et al. (2022) con los algoritmos LR con un 77.78%, NB un 76.27%, RF un 75.76% y SVM con 82.83%.

Al contrario, los siguientes autores Orsoni et al. (2023), Kannan, Abarna y Vairachilai (2023), Al-Alawi et al. (2022), Owusu-Boadu et al. (2021), Urkude y Gupta (2019), Mounika y Persis (2019), Ojajuni et al. (2021) y Dinh et al. (2020), no consideraron esta métrica en sus investigaciones, ya que se enfocaron en las métricas de exactitud, especificidad, precisión y F1 - score.

En lo que respecta al objetivo específico 4, a partir de la comparación entre los algoritmos de aprendizaje automático DT, RF, SVM, ANN, GBM, MP, K-NN, NB, LR y ADA se observó que los mejores modelos se obtuvieron utilizando los algoritmos DT, SVM, GBM y AB, de las medidas utilizadas para evaluar el rendimiento y calidad del modelo en **exactitud** obtuvo un 100% para la predicción del rendimiento académico. Con lo mencionado anteriormente, en similitud con las investigaciones de Al-Alawi et al. (2022) utilizó los algoritmos de DT que obtuvo un 82.4%, RF un 78.3%, RT un 76.1%, NB un 71.4%, MP un 80.9%, SVM un 70.8%, KNN un 76.1%, Vote un 81.3%, LB un 80.7% y Bagging con un 82.2%. Asimismo, Orsoni et al. (2023) del algoritmo ADA obtuvo una exactitud del 77.2% y ANN con un resultado del 91.7%. Del mismo modo, Owusu-Boadu et al. (2021) con base en los algoritmos RF con un 77.1%, LR un 77.9%, ANN un 76%, K-NN un 63.8%, SVM un 72.7% y ADA con 74.8%. Igualmente, Kannan, Abarna y Vairachilai (2023), detalla que en los algoritmos KNN tuvo 70.12%, DT un 74.17%, RF un 80.55%, SVM un 71.42%, ANN un 80.23%, NB un 66.23%, XGB un 79.05% y ADA con 75.74%. También, Wang et al. (2022) menciona que LR obtuvo un 85.04%, NB un 80.49% y RF con 83.71%. En cuanto García (2021) menciona que SVM tiene un 100%, DT un 91.92% y K-NN con 79.75%. Mounika y Persis (2019) detalla que

KNN tiene 78.66%, SVM un 98.60%, DT un 93.28%, RF un 91.61% y GBM con 99.66%. Dinh et al. (2020) menciona que los algoritmos NB y MP un 86.19%, SMO un 85.64%, DT un 73.48%, RF un 80.70%, RT un 77.90%, PART un 74.59% y OneR con 76.24%. Finalmente, Ojajuni et al. (2021) en el algoritmo DT alcanzó un 47.95%, RF un 92.60%, SVM un 42.88%, LR un 40.96%, ADA un 35.75%, SGD un 33.69%, XGBoost un 97.12% y CNN con 72.74%.

Al contrario, los siguientes autores Urkude y Gupta (2019), no consideraron esta métrica en su investigación, ya que se enfocaron en la métrica F1 - score.

En lo que respecta al objetivo específico 5, a partir de la comparación entre los algoritmos de aprendizaje automático DT, RF, SVM, ANN, GBM, MP, K-NN, NB, LR y ADA se observó que los mejores modelos se obtuvieron utilizando los algoritmos DT, SVM, GBM y AB, de las medidas utilizadas para evaluar el rendimiento y calidad del modelo en **F1 - Score** se obtuvo un 100% para la predicción del rendimiento académico. Con lo mencionado anteriormente, en similitud con las investigaciones de Orsoni et al. (2023) del algoritmo ADA un 89.9% y ANN un 91.6%. Finalmente, Urkude y Gupta (2019) con base en los algoritmos SVM con un 78.38%, DT un 66.13% y NV con 76.34%.

Al contrario, los siguientes autores Owusu-Boadu et al. (2021), Kannan, Abarna y Vairachilai (2023), Al-Alawi et al. (2022), Wang et al. (2022), García (2021), Mounika y Persis (2019), Dinh et al. (2020) y Ojajuni et al. (2021), no consideraron esta métrica en sus investigaciones, ya que se enfocaron en métricas de exactitud, especificidad, precisión y sensibilidad.

# VI. CONCLUSIONES

A partir de los resultados obtenidos en la investigación, presentamos las siguientes conclusiones:

**Primera:** Se desarrolló un sistema inteligente con ML para predecir el rendimiento académico de los estudiantes, haciendo uso de la metodología KDD, permitiendo así el descubrimiento de conocimientos útiles y simplificando el proceso mediante modelos de ML, dónde fueron los modelos de DT, SVM, GBM y ADA, los que obtuvieron un 100% en todas las métricas.

**Segunda:** Se desarrolló un sistema inteligente con ML para predecir con especificidad el rendimiento académico de los estudiantes. La comparación entre los algoritmos reveló que los modelos basados en DT, SVM, GBM y ADA obtuvieron un mejor resultado, alcanzando un 100%, es decir, se logró predecir en su totalidad los casos negativos reales que los modelos clasificaron correctamente.

**Tercera:** Se desarrolló un sistema inteligente con ML para predecir con precisión el rendimiento académico de los estudiantes. Las comparaciones de los algoritmos muestran que los modelos basados en DT, SVM, GBM y ADA lograron mejores resultados porque alcanzaron el 100%, es decir, son capaces de predecir completamente los casos clasificados como positivos y que realmente son positivos.

**Cuarta:** Se desarrolló un sistema inteligente con ML para predecir con sensibilidad el rendimiento académico de los estudiantes, lo cual demostró que, al realizar la comparación de los diferentes algoritmos, el DT, SVM, GBM y ADA, son aquellos que alcanzaron resultados al 100%, debido a que logran identificar la totalidad de los casos positivos de los modelos.

**Quinta:** Se desarrolló un sistema inteligente con ML para predecir con exactitud el rendimiento académico de los estudiantes. Esto, demostró que, al realizar la comparación de los algoritmos, destaca el de DT, SVM, GBM y ADA, con unos resultados del 100%, lo que indica que en la evaluación de los modelos realiza las predicciones correctas (tanto positivas como negativas) respecto al total de casos.

**Sexta:** Se desarrolló un sistema inteligente con ML para predecir con F1-score el rendimiento académico de los estudiantes. Las comparaciones de los diferentes algoritmos hacen que destaque el DT, SVM, GBM y ADA, donde sus resultados alcanzaron el 100%, lo que indica un rendimiento sólido en términos de equilibrio entre la precisión y la sensibilidad en la evaluación de los modelos que se combinan en una sola medida.

# VII. RECOMENDACIONES

Después de que se han establecido las conclusiones, se formulan las recomendaciones detallando las acciones sugeridas para mejorar y optimizar los aspectos identificados durante el estudio para futuras investigaciones:

**Primera:** Se recomienda explorar otras metodologías además de KDD, como CRISP-DM y SEMMA para obtener una visión más completa y considerar diferentes enfoques en la minería de datos, lo que podría resultar en una comprensión más profunda de los datos y generar resultados más detallados y específicos.

**Segunda:** Se recomienda automatizar el proceso de evaluación del rendimiento académico entre el sistema inteligente, la base de datos y JN. Es decir que, en el sistema inteligente, cuando se ingrese los datos de entrada de los alumnos, los resultados sean almacenados en la base de datos, y que al momento de realizar las consultas en JN, esta considere los nuevos datos ingresados y a su vez, genere nuevos modelos entrenados para la evaluación del rendimiento académico. De esta forma, el modelo entrenaría de manera automática y generaría resultados más específicos y completos.

**Tercera:** Se recomienda realizar la comparación de otros algoritmos: XGBosst, Principal Component Analysis (PCA), Support Vector Regression (SVR), Lasso Regression, K-Means Clustering y Gaussian Mixture Model (GMM), que no se han incluido en nuestra investigación con base en la métrica de especificidad, lo que ayudaría en la evaluación de una mejor capacidad del modelo para predecir el rendimiento académico, de esta manera se pueda establecer los alcances y posibles limitaciones, lo que podría proporcionar información adicional sobre enfoques alternativos y sus ventajas o desventajas en un modelo especifico.

**Cuarta:** Se recomienda llevar a cabo estudios comparativos con otras instituciones educativas que presenten o compartan características similares. Este enfoque no solo contribuirá a perfeccionar la precisión del modelo para la I.E.P. mencionada, sino que también posibilitará obtener patrones comunes que puedan aplicarse en diversos grados educativos.

# REFERENCIAS

ABDULKADIR, Oumer y Derbew, Getachew. Application of supervised machine learning algorithms for classifcation and prediction of type-2 diabetes disease status inAfar regional state, Northeastern Ethiopia 2021. Scientifc Reports, (13):1-12, 2023.
ISSN: 2045-2322.
Disponible en: https://n9.cl/atrjp

AL-ALAWI, L.; AL SHAQSI, J.; TARHINI, A.; AL-BUSAIDI, A.; Using machine learning to predict factors affecting academic performance: the case of college students on academic probation [en línea]. 2023, Education and Information Technologies.
ISSN: 1573-7608.
Disponible en: https://n9.cl/uaf7x

“ALARMANTE pero no sorprendente”: el rendimiento de los estudiantes de EE. UU. en matemáticas y lectura es el peor en dos décadas [en línea]. BBC News Mundo. 1 de septiembre de 2022. [Fecha de consulta: 30 de mayo de 2023]
Disponible en: https://goo.su/FAN3V5J

ALBÁN, Joselo y CALERO, José. EL RENDIMIENTO ACADÉMICO: APROXIMACIÓN NECESARIA A UN PROBLEMA PEDAGÓGICO ACTUAL. Revista pedagógica de la Universidad de Cienfuegos, 13(58):213-220, 2017.
ISSN: 1990-8644
Disponible en: https://n9.cl/vnoqz

AN Empirical Study for Student Academic Performance Prediction Using Machine Learning Techniques por Dinh Thi Ha [et al]. International Journal of Computer Science and Information Security, 18(3): 21 – 28, 2020.
ISSN: ISSN 1947-5500
Disponible en: https://n9.cl/lgmov

AWAD, Mariette y KHANNA, Rahul. Efficient Learning Machines. California: Apress Berkeley, CA. 2015, 268 pp.
ISBN: 9781430259893
Disponible en: https://n9.cl/3chgb

ARIAS, J., & Covino, M. diseño y metodología de la investigación. Primera edición digital, junio del 2021. Hecho el Depósito Legal en la Biblioteca Nacional del Perú Nª N° 2021-05553 2021. 68 pp.
ISBN: 978-612-48444-2-3.
Disponible en: https://n9.cl/3kqia

ARIAS, Fidias. El proyecto de investigación Introducción a la Metodología Científica. 6° ed. Caracas: EDITORIAL EPISTEME, 2012. 143 pp.
ISBN: 9800785299
Disponible en: https://n9.cl/gemjp

ARIAS, José. Guía para elaborar la operacionalización de variables. ESPACIO I+D, INNOVACIÓN MÁS DESARROLLO, 10(28):42-56, 2021.
ISSN: 2007-6703
Disponible en: https://n9.cl/2djf9f

BONACCORSO, Giuseppe. Machine Learning Algorithms. Birmingham:Packt Publishing, 2017. 353 pp.
ISBN: 9781785889622

BURMAN, I. y SOM, S. Predicting Students Academic Performance Using Support Vector Machine. Proceedings - 2019 Amity International Conference on 46 Artificial Intelligence, AICAI 2019, pp. 756-759.
Disponible en: https://n9.cl/l7au9

CABRERA, Eduardo y DÍAZ, Elena. GUIDE to jupyter notebooks for educational purposes. Madrid: Universidad Computlense, 2018. 38 pp.
Disponible en: https://n9.cl/1xbte

CHAUDHARI, K., & Thakkar, A. Neural network systems with an integrated coefficient of variation-based feature selection for stock price and trend prediction. Expert Systems with Applications, (2023).
Disponible en: https://n9.cl/j2ek4

CONTRERAS, Leonardo, FUENTES, Héctor y RODRIGUEZ, José. Predicción del rendimiento académico como indicador de éxito/fracaso de los estudiantes

de ingeniería, mediante aprendizaje automático. Form. Univ. [online]. 2020, vol.13, n.5 Fecha de consulta: 31 de mayo de 2023], pp.233-246.
ISSN: 0718-5006.
Disponible en: https://n9.cl/ygtx9

CORONA, Edgar, JIMÉNEZ, Abraham y CORTÉS, Griselda. Principales Metodologías en el Desarrollo de Proyectos de Minería de Datos. Revista Tecnocultura 51 [en línea]. 5 de enero de 2020. [Fecha de consulta: 2 de julio de 2023].
Disponible en: https://n9.cl/qfvcn
ISSN: 1870-7157

DALIANIS, Hercules. Clinical Text Mining. Sweden: Springer Open, 2018. 192 pp.
ISBN: 9783319785028

ESPINOZA, Eudaldo. LAS VARIABLES Y SU OPERACIONALIZACIÓN EN LA INVESTIGACIÓN EDUCATIVA. SEGUNDA PARTE. Revista pedagógica de la Universidad de Cienfuegos, 15(69):171-180, 2019.
ISSN: 1990-8644
Disponible en: https://n9.cl/hwo6q

ESTRADA, Alex. Estilos de aprendizaje y rendimiento académico. Revista Boletín Redipe [en línea], 7(7): 218-228, 4 de mayo - 28 de junio, 2018. [Fecha de consulta: 15 de abril de 2023]
Disponible en: https://goo.su/gXHey2
ISSN: 2266-1536

GARCÍA Dionisio, Jeancarlos. Machine Learning para predecir el rendimiento académico de los estudiantes universitarios. Tesis (Título profesional de Ingeniería de Sistemas). Lima: Universidad César Vallejo, Facultad de Ingeniería, 2021. 85pp.
Disponible en: https://n9.cl/jbpro

GONZÁLEZ Vidal, Ana. Selección de variables: Una revisión de métodos existentes. Tesis (Máster en Técnicas Estadísticas). España: Universidade da Coruña, 2015. 87 pp.
Disponible en: https://n9.cl/ut3p1

HALVORSEN, Hans. Python Programming. Noruega: Universidad del Sudeste, 2020. 140 pp.
ISBN: 9788269110647

HERNÁNDEZ, Roberto, FERNÁNDEZ, Carlos y BAPTISTA, Pilar. METODOLOGÍA DE LA INVESTIGACIÓN. 6° ed. México: McGRAW-HILL, 2014, 634 pp.
ISBN: 9781456223960
Disponible en: https://n9.cl/2i4

HOSSIN, M., Sulaiman, M.N. A review on evaluation metrics for data classification evaluations. International Journal of Data Mining & Knowledge Management Process, vol 5(2):1-11, March 2015.
ISSN: 2230-9608

INTRODUCTION to Anaconda and Python: Installation and setup por Damien Rolon Mérette [et al]. The Quantitative Methods for Psychology, 16(5): S3-S11, 2020.
DOI: 10.20982/tqmp.16.5.S003

INMACULADA Meca Sáez. Factorización y selección de variables para modelos predictivos en educación. (2011).
Disponible en: https://n9.cl/ye8k5

IMPACT of COVID-19 on the academic performance and mental health of HBCU pharmacy students por Carrión J., Antonio [et al]. Currents in Pharmacy Teaching and Learning, 1-7, 2023.
ISSN: 1877-1297

JHULISSA Isabel Villamagua Poma, & Sistemas, E. N. (2021). Minería de Datos Educativa aplicada al descubrimiento de patrones en registros académicos

de estudiantes de Ingeniería con asignaturas relacionadas a las matemáticas. Disponible en: https://n9.cl/rywqm

JOSÉ Ignacio Canto Gajardo. Selección de las variables psicosociales que caracterizan el desempeño de funciones ejecutivas en situaciones cotidianas, a través de un modelo de predicción etaria. (2023).
Disponible en: https://n9.cl/7sj1l

JOHN Frank Rivera Bardales. (2020). Determinación de la aceptación de un producto financiero basado en la gestión de llamadas a clientes potenciales en una campaña vigente usando algoritmos de aprendizaje automático.
Disponible en: https://n9.cl/qzbec

KANNAN, R.; ABARNA, K.; VAIRACHILAI, S.; et al. Student Academic Performance Prognosticative Using Optimized Hybrid Machine Learning Algorithms [en línea]. 2023.
Disponible en: https://n9.cl/xe97f

LOHR, Sharon. Sampling Design and Analysis. 3° ed. Abingdon: Taylor & Francis Group, 2022. 674 pp.
ISBN: 9780429298899
Disponible en: https://n9.cl/jl7bm

MANTEROLA, C., GRANDE, L., OTZEN, T., GARCÍA, N., SALAZAR, P. y QUIROZ, G., 2018. Confiabilidad, precisión o reproducibilidad de las mediciones. Métodos de valoración, utilidad y aplicaciones en la práctica clínica. Revista chilena de infectología, 35(6): 680-688.
ISSN: 0716-1018
Disponible en: https://n9.cl/rjb0i

MÉTRICAS de rendimiento para evaluar el aprendizaje automático en la clasificación de imágenes petroleras utilizando redes neuronales convolucionales por Zapeta Hernández, Adriana [et al.]. Ciencia Latina Revista Científica Multidisciplinar [en línea]. septiembre-octubre 2022, vol. 6, n.° 5. [Fecha de consulta: 01 de junio de 2023].

ISSN: 2707-2215
Disponible en: https://n9.cl/sa2dh

MEZA Rodríguez, A. R., & Chue Gallardo, J. (2020). Uso del algoritmo Adaboost y la regresión logística para la predicción de fuga de clientes en una empresa de telefonía móvil. 102.
Disponible en: https://n9.cl/urm1q

MINISTERIO de Educación. Evaluación Muestral de Estudiantes 2022 presenta resultados más bajos que los de 2019 [en línea]. Lima: MINEDU, 2022 [fecha de consulta 30 de mayo de 2023].
Disponible en: https://goo.su/SfjZnP

MOUNIKA, B. y Persis, V. A Comparative Study of Machine Learning Algorithms for Student Academic Performance. International Journal of Computer Sciences and Engineering, 7(4):721-725, 2019.
ISSN: 2347-2693
Disponible en: https://n9.cl/qnh89

OLADIPUPO, Taiwo. Types of Machine Learning Algorithms. Reino Unido: University of Portsmouth, 2010. 32 pp.
ISBN: 9789533070346

ORSONI, M.; et al. Preliminary evidence on machine learning approaches for clusterizing students' cognitive profile [en línea]. 2023. Elsevier Ltd.
Disponible en: https://n9.cl/718zu

OWUSU-BOADU, B.; NTI, I.; NYARKO-BOATENG, O.; ANING, J.; et al. Academic Performance Modelling with Machine Learning Based on Cognitive and Non-Cognitive Features. [en línea]. 2021, Applied Computer Systems, Vol. 26, pp. 122-131.
Disponible en: https://n9.cl/jnxozk

PREDICTING Student Academic Performance Using Machine Learning por Ojajuni Opeyemi [et al]. Springer Nature Switzerland, 481 – 491, 2021.
Disponible en: https://n9.cl/e94t4

RELATIONSHIP between Creativity and Academic Performance in Spain and Colombia por López Fernández Verónica [et al]. Revista Colombiana de Educación [en línea], n.° 86: 31-52, 28 de agosto de 2020 – 25 de marzo de 2021, 2022. [Fecha de consulta: 15 de abril de 2023].
Disponible en: https://goo.su/QD2VdQ
ISSN: 0120-3916.

REVISTA UNIR. ¿Qué son los sistemas inteligentes? Importancia y aplicaciones. Perú. Marzo 2022.
Disponible en: https://n9.cl/u02hh

SALAS-DOMINGUEZ, M.I. y MUÑOZ-DÍAZ, I., 2019. Análisis de concordancia de atributos en color de piezas galvanizadas. Revista de Tecnologías en procesos Industriales, vol. 3, no. 6, pp.
Disponible en: https://goo.su/buaSy

SHALEV, Shwartz y BEN, David. Understanding Machine Learning: From Theory to Algorithms. United States: Cambridge University Press, 2014. 449 pp.
ISBN: 9781107057135

SEN, Jaydip. Machine Learning - Algorithms, Models and Applications. 7.° ed. London: Artificial Intelligence Series Editor, 2021. 154 pp.
ISBN: 9781839694868

SUN, T. H., Wang, C. C., Wu, Y. L., Hsu, K. C., & Lee, T. H. (2023). Machine learning approaches for biomarker discovery to predict large-artery atherosclerosis. Scientific Reports, 13(1). https://doi.org/10.1038/s41598-023-42338-0

URKUDE, S.; GUPTA, K. Student Intervention System using Machine Learning Techniques. International Journal of Engineering and Advanced Technology [en línea]. 2019, Vol. 8. ISSN: 2249 – 8958.
Disponible en: https://n9.cl/ormz9

WANG, D.; LIAN, D.; XING, Y.; DONG, S.; et al. Analysis and Prediction of Influencing Factors of College Student Achievement Based on Machine Learning. Frontiers in Psychology [en línea]. 2022, vol. 13.

ISSN: 1664-1078.
Disponible en: https://n9.cl/rjkwk

WEISSMAN, B., & van de Laar, E. (2020). SQL Server Big Data Clusters: Data Virtualization, Data Lake, and AI Platform. In SQL Server Big Data Clusters: Data Virtualization, Data Lake, and AI Platform. Apress Media LLC.
Disponible en: https://n9.cl/ibbjr

# ANEXOS

**ANEXO 1:**

**Tabla N° 74: Matriz de Operacionalización de Variables**

| Variable | Definición conceptual | Definición operacional | Dimensiones | Indicadores | Escalas de Medición | Metodología |
|---|---|---|---|---|---|---|
| **Independiente:** Sistema Inteligente con Machine Learning | Se define como Sistema inteligente a aquellos que utilizan inteligencia artificial como lo es el Machine Learning para optimizar y configurar los procesos de una organización (UNIR, 2022, párr. 2) | | | | | |
| **Dependiente:** | Según, Contreras, Fuentes y Rodríguez (2020) mencionan | Para la medición del rendimiento | Métricas de Evaluación de Modelo. | Especificidad (TN/(TN+FP)) * 100 Burman y Som (2019) | Razón | KDD |

| | | | | | | |
|---|---|---|---|---|---|---|
| Predecir el rendimiento académico | que el rendimiento académico es un indicador que determina el éxito o fracaso de un estudiante, a pesar de eso, no hay una forma de medición o predicción estandarizada que asegure una adecuada valoración (p. 234). | académico se empleará de manera concreta las métricas de especificidad, precisión, sensibilidad, exactitud y F1 Score, las cuáles serán obtenidas a partir de herramientas y técnicas del Machine Learning. | Abdulkadir y Derbew (2023) | Precisión<br>(TP/(TP+FP)) * 100<br>Zapeta et al. (2019) | | Corona, Jiménez y Cortés (2020) |
| | | | | Sensibilidad<br>(TP/(TP+FN)) * 100<br>Burman y Som (2019) | | |
| | | | | Exactitud<br>((TP+TN)/(TP+TN+FP+FN)) * 100<br>Hossin y Sulaiman (2015) | | |
| | | | | F1-score<br>2 * ((Precisión * Sensibilidad) / (Precisión + Sensibilidad)) * 100<br>Dalianis (2018) | | |

**Fuente: Elaboración propia**

## ANEXO 2:

### Tabla N° 75: Matriz de Consistencia

| Problema | Objetivo | Hipótesis | Variable | Dimensiones | Indicadores | Metodología |
|---|---|---|---|---|---|---|
| P. G: ¿Cómo un Sistema Inteligente con ML basado en selección de variables permitirá predecir el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana”? | **O. G:** ¿Desarrollar un Sistema Inteligente con ML basado en selección de variables para predecir el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana”? | **H. G:** La aplicación de un Sistema Inteligente con ML basado en selección de variables predecirá el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana” | **Independiente:** Sistema Inteligente con Machine Learning. (UNIR, 2022, párr. 2) | | | |
| **P. E. 1:** ¿Cómo un Sistema Inteligente con ML basado en selección de variables permitirá predecir con especificidad el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana”? | **O.E. 1:** Desarrollar un Sistema Inteligente con ML basado en selección de variables para predecir con especificidad el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana” | **H. E. 1:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables predecirá con especificidad el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana” | **Dependiente:** Predecir el rendimiento académico. Contreras, Fuentes y Rodríguez (2020) | Métricas de Evaluación de Modelo. Abdulkadir y Derbew (2023) | Especificidad (TN/(TN+FP)) * 100 Burman y Som (2019) | **Tipo de Investigación** Aplicada<br><br>**Diseño de investigación** Experimental de tipo pre - experimental |

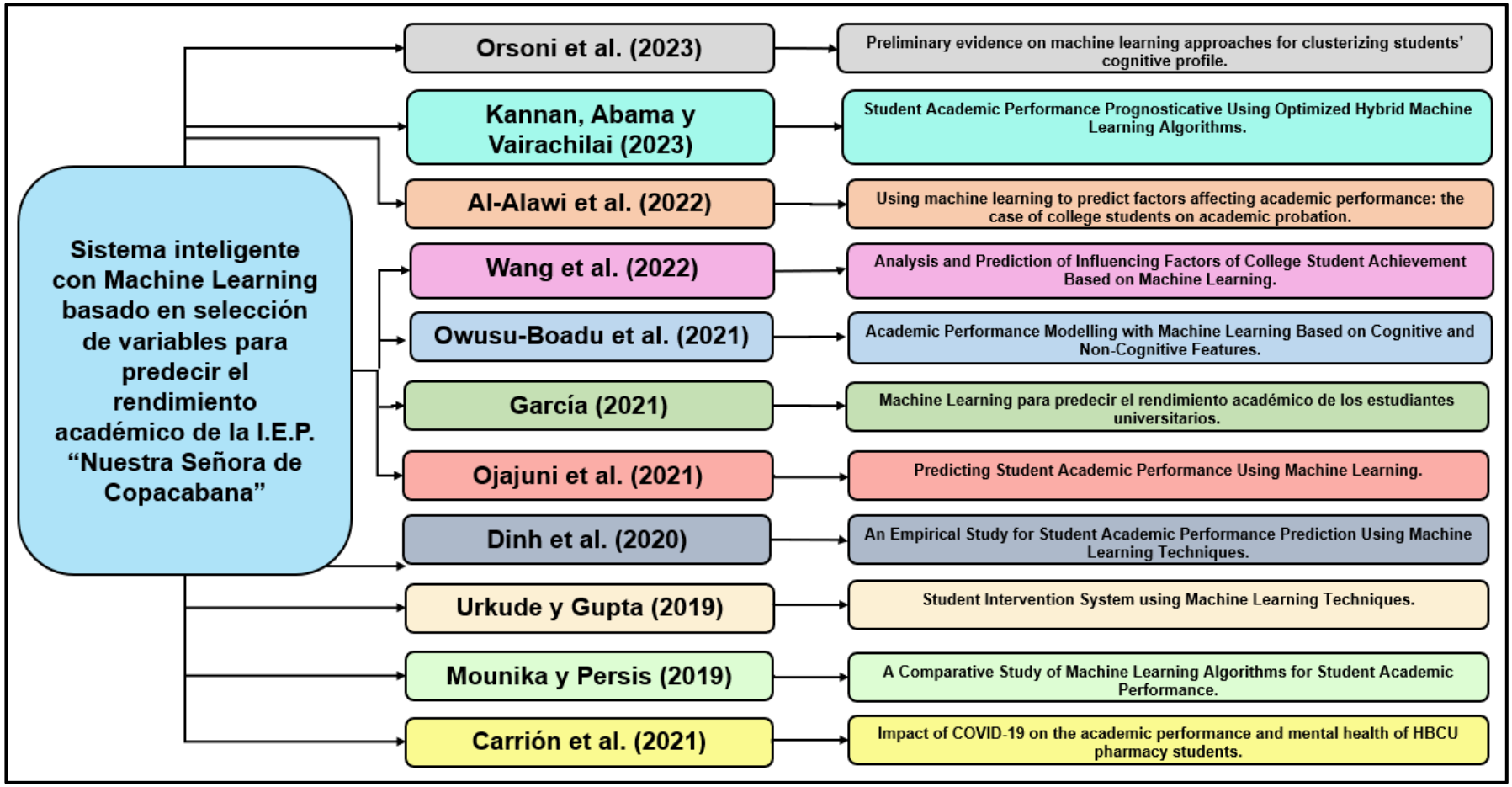
| | | | | | | |
|---|---|---|---|---|---|---|
| **P. E. 2:** ¿Cómo un Sistema Inteligente con ML basado en selección de variables permitirá predecir con precisión el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana”? | **O. E. 2:** Desarrollar un Sistema Inteligente con ML basado en selección de variables para predecir con precisión el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana” | **H. E. 2:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables predecirá con precisión el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana” | | | Precisión<br>(TP/(TP+FP)) * 100<br>Zapeta et al. (2019) | |
| **P. E. 3:** ¿Cómo un Sistema Inteligente con ML basado en selección de variables permitirá predecir con sensibilidad el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana”? | **O. E. 3:** Desarrollar un Sistema Inteligente con ML basado en selección de variables para predecir con sensibilidad el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana” | **H. E. 3:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables predecirá con sensibilidad el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana” | | | Sensibilidad<br>(TP/(TP+FN)) * 100<br>Burman y Som (2019) | |
| **P. E. 4:** ¿Cómo un Sistema Inteligente con ML basado en selección de variables permitirá predecir con exactitud el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana”? | **O. E. 4:** Desarrollar un Sistema Inteligente con ML basado en selección de variables para predecir con exactitud el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana” | **H. E. 4:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables predecirá con exactitud el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana” | | | Exactitud<br>((TP+TN)/(TP+TN+FP+FN)) * 100 | |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | | Hossin y Sulaiman (2015) | |
| **P. E. 5:** ¿Cómo un Sistema Inteligente con ML basado en selección de variables permitirá predecir con F1-score el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana”? | **O. E. 5:** Desarrollar un Sistema Inteligente con ML basado en selección de variables para predecir con F1-score el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana” | **H. E. 5:** El desarrollo de un Sistema Inteligente con ML basado en selección de variables predecirá con F1-score el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana” | | | F1-score<br>2 * ((Precisión * Sensibilidad) / (Precisión + Sensibilidad)) * 100<br>Dalianis (2018) | |

**Fuente: Elaboración Propia**

**ANEXO 3:**

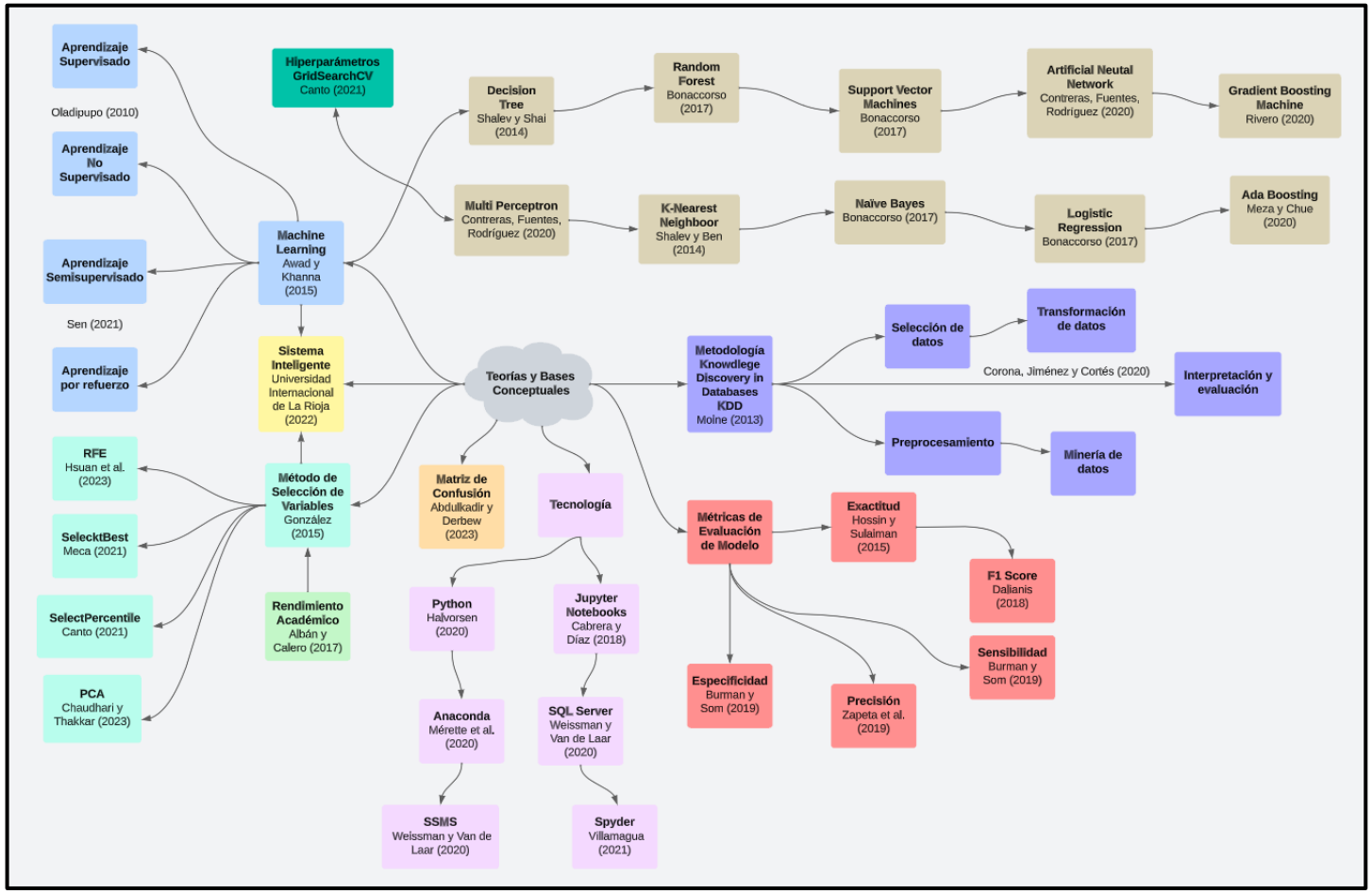
**Figura N° 27: Mapa Conceptual de los Antecedentes**




**Fuente: Elaboración Propia**

**ANEXO 4:**

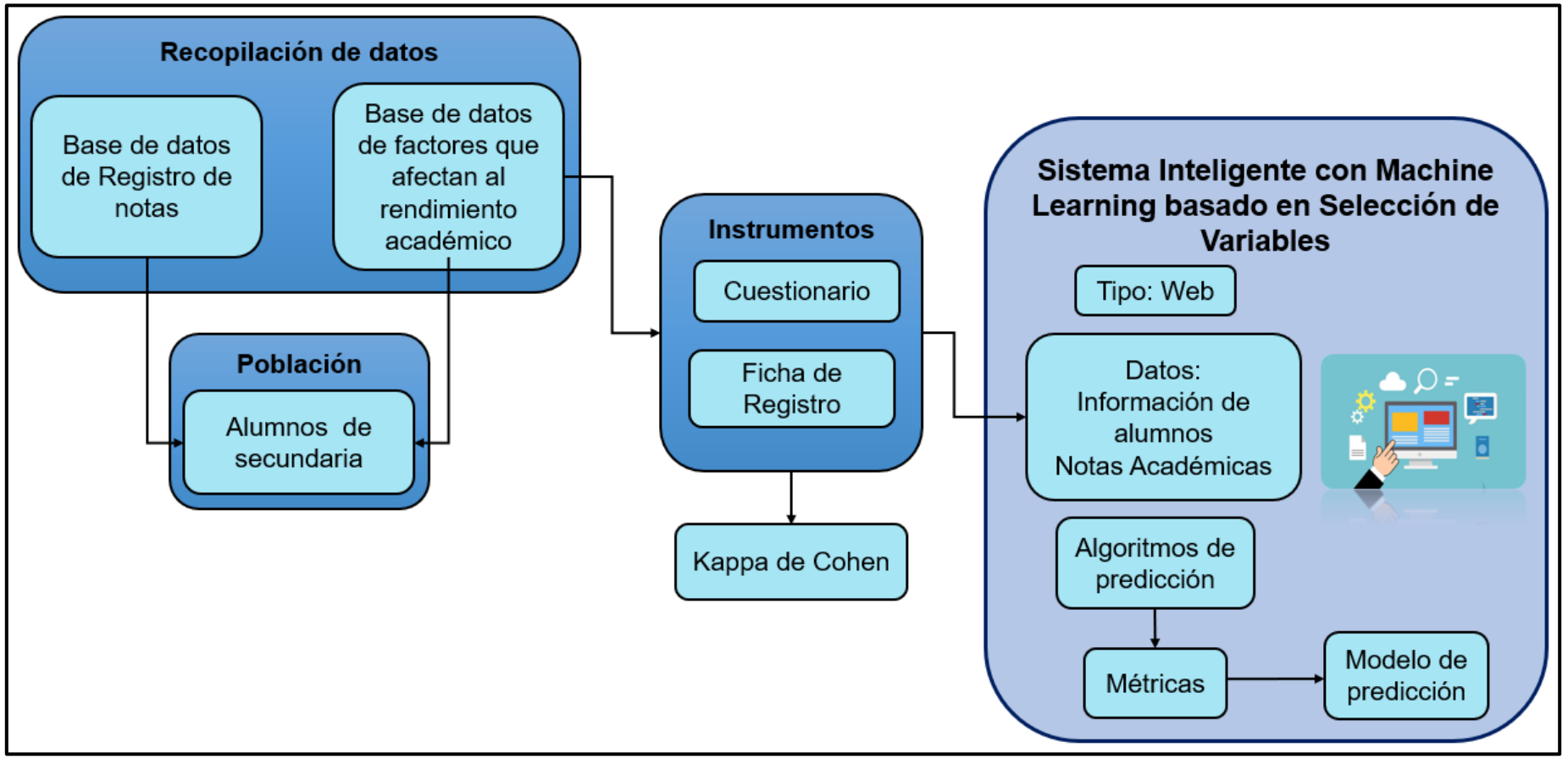
**Figura N° 28: Mapa Mental de las Teorías y Bases Conceptuales**




**Fuente: Elaboración Propia**

**ANEXO 5:**

**Figura N° 29: Mapa Conceptual del Procedimiento**




**Fuente: Elaboración Propia**

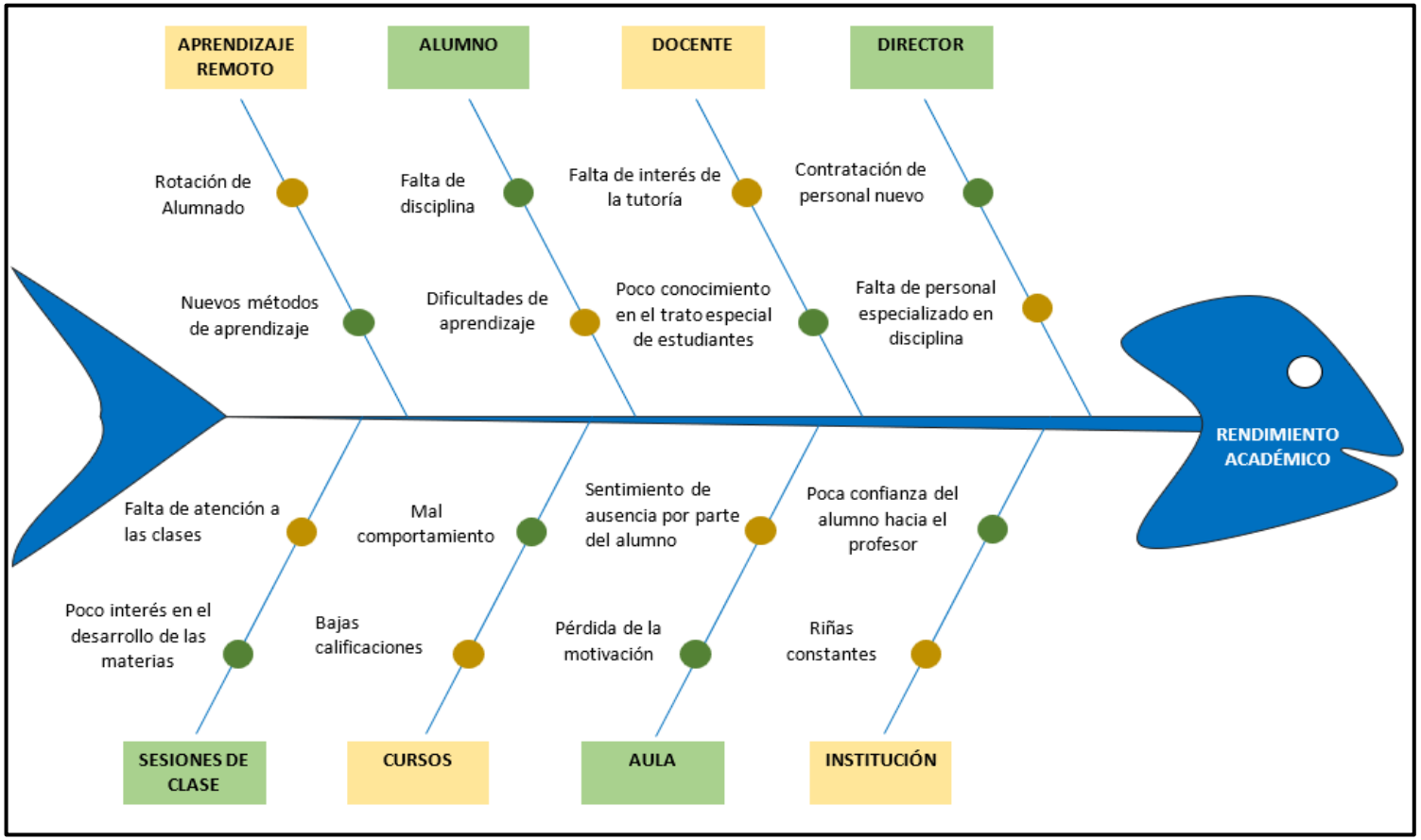
**ANEXO 6:**

**Figura N° 30: Propuesta tecnológica**


**Fuente: Elaboración Propia**

**ANEXO 7:**

**Figura N° 31: Diagrama Causa – Efecto**




**Fuente: Elaboración Propia**

**ANEXO 8:**

**CONSENTIMIENTO INFORMADO PARA PARTICIPANTES DE INVESTIGACIÓN**

**Lima 21 de junio de 2023**

**SEÑORES:**

**UNIVERSIDAD CÉSAR VALLEJO**

**Presente: Autorización de uso de datos y nombre de la empresa.**

Por medio del presente, yo, Victor Hugo Valverde Cardenas, con DNI N° 09402717, director general de la empresa NUESTRA SEÑORA DE COPACABANA S.A.C con RUC 20506992185, apruebo que se otorgue el acceso a la información necesaria, permisos de recolección y publicación de los datos a los estudiantes Nicole Valery Huertas Fabian, con DNI N° 72084487 y Juan Antonio Salcedo Zapata, con DNI N° 72762827, de la Escuela Profesional de Ingeniería de Sistemas de la Universidad César Vallejo, cuyo proyecto de investigación se titula "SISTEMA INTELIGENTE CON MACHINE LEARNING BASADO EN SELECCIÓN DE VARIABLES PARA PREDECIR EL RENDIMIENTO ACADÉMICO DE LA I.E.P. NUESTRA SEÑORA DE COPACABANA". Asimismo, manifiesto que puedo solicitar información y resultados de la investigación, una vez haya concluido.

**Atentamente,**

Dir.: Prolongación Andrés Avelino Cáceres N° 225 – 273, El Progreso – Carabayllo.
Telf.: 940381100

## ANEXO 9:

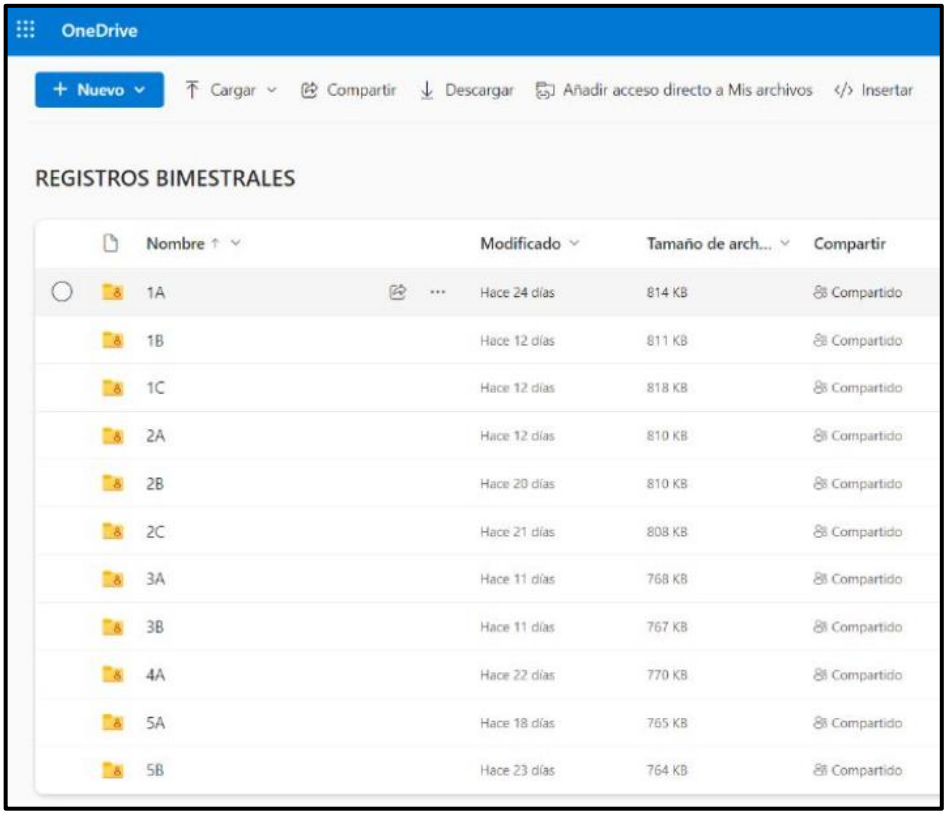
### Tabla N° 76: Cuadro de medición de rendimiento académico

| | | |
|---|---|---|
| **Examen Semanal de Avance Académico (ESAA)** | Exámenes realizados cada viernes para evaluar la retención de los temas trabajados en la semana. | 20% |
| **Simulacro** | Examen tomado cada dos meses para preparar a los alumnos a futuros exámenes de admisión. | 10% |
| **Examen Bimestral** | Examen tomado cada dos meses para evaluar la capacidad académica de los alumnos. | 20% |
| **Evaluación de progreso académico** | Resultado de todo el trabajo realizado durante las horas de clase, (Practicas, tareas, participación, etc.) | 50% |

**Fuente: Elaboración propia**

## ANEXO 10:

### Figura N° 32: Base de Datos de notas en Microsoft OneDrive




**Fuente: I.E.P. "Nuestra Señora de Copacabana"**

**ANEXO 11:**

## INTRUMENTO DE CUESTIONARIO PARA LA RECOLECCIÓN DE DATOS

| ÍTEM | FACTORES | PREGUNTAS | VALOR |
|---|---|---|---|
| **1** | **Datos Personales** | ¿Cuál es tu género biológico? | Masculino / Femenino |
| **2** | | ¿Qué edad tienes? | Dato Numérico |
| **3** | **Auto - percepción** | ¿Cómo consideras que es tu rendimiento académico? | Muy Malo / Malo / Regular / Bueno / Muy bueno |
| **4** | | ¿Te consideras una persona inteligente? | SI / NO |
| **5** | | ¿Te consideras una persona responsable? | SI / NO |
| **6** | | ¿Cuántas horas al día dedicas al estudio? | Dato Numérico |
| **7** | | ¿Te organizas y planificas tu tiempo antes de rendir las evaluaciones? | SI / NO |
| **8** | **Educación Familiar** | ¿Qué nivel de educación tiene tu madre? | Primaria / Secundaria / Técnico / Superior |
| **9** | | ¿Qué nivel de educación tiene tu padre? | Primaria / Secundaria / Técnico / Superior |
| **10** | | ¿Cuál es el nivel más alto de educación que existe en tu familia? | Primaria / Secundaria / Técnico / Superior |
| **11** | **Sociabilidad** | ¿Qué tan frecuente te sientes estresado(a) por los estudios? | Muy infrecuente / Infrecuente / Poco Frecuente / Frecuente / Muy Frecuente |
| **12** | | ¿Consideras que tienes una buena relación con tus compañeros de clases? | SI / NO |

| | | | |
|---|---|---|---|
| **13** | | ¿Cómo reaccionas ante un problema de entorno social? | Con violencia física o verbal / Dialogas / Con indiferencia |
| **14** | **Economía Familiar** | ¿Te proporcionan todo lo necesario para rendir bien académicamente? | SI / NO |
| **15** | | ¿Tus padres generan ingresos suficientes para cubrir tus gastos académicos? | SI / NO |
| **16** | | ¿Cómo consideras que es la economía en tu hogar? | Muy mala / Mala / Regular / Buena / Muy buena |

**ANEXO 12:**

## INTRUMENTO DE FICHA DE REGISTRO PARA EL ALGORITMO DT

| RESUMEN DE EVALUACIÓN | |
|---|---|
| **Tipo de Prueba** | Post Test |
| **Investigadores** | - Huertas Fabian Nicole Valery<br>- Salcedo Zapata, Juan Antonio |
| **Fecha de inicio** | 29/10/2023 |
| **Algoritmo** | DT |

| MATRIZ DE CONFUSIÓN | | | |
|---|---|---|---|
| | | **Estimación por el modelo** | |
| | | **Negativo (N)** | **Positivo (P)** |
| **Real** | **Negativo** | TN | FP |
| | **Positivo** | FN | TP |

| MÉTRICAS POR EVALUAR | | | | |
|---|---|---|---|---|
| **Ítem** | **Indicador** | **Medida** | **Fórmula** | **Precisión** |
| **1** | Especificidad | Razón | (TN/(TN+FP)) * 100 | 100% |
| **2** | Precisión | Razón | (TP/(TP+FP)) * 100 | 100% |
| **3** | Sensibilidad | Razón | (TP/(TP+FN)) * 100 | 100% |
| **4** | Exactitud | Razón | (TP+TN)/(TP+TN+FP+FN) | 100% |
| **5** | F1-score | Razón | 2 * ((Precisión * Sensibilidad) / (Precisión + Sensibilidad)) | 100% |

**ALFREDO DAZA VERGARAY**

**DNI 40466240**

**ANEXO 13:**

## INTRUMENTO DE FICHA DE REGISTRO PARA EL ALGORITMO RF

| RESUMEN DE EVALUACIÓN | |
|---|---|
| **Tipo de Prueba** | Post Test |
| **Investigadores** | - Huertas Fabian Nicole Valery<br>- Salcedo Zapata, Juan Antonio |
| **Fecha de inicio** | 29/10/2023 |
| **Algoritmo** | RF |

| MATRIZ DE CONFUSIÓN | | | |
|---|---|---|---|
| | | **Estimación por el modelo** | |
| | | **Negativo (N)** | **Positivo (P)** |
| **Real** | **Negativo** | TN | FP |
| | **Positivo** | FN | TP |

| MÉTRICAS POR EVALUAR | | | | |
|---|---|---|---|---|
| **Ítem** | **Indicador** | **Medida** | **Fórmula** | **Precisión** |
| **1** | Especificidad | Razón | (TN/(TN+FP)) * 100 | 99.35% |
| **2** | Precisión | Razón | (TP/(TP+FP)) * 100 | 73.21% |
| **3** | Sensibilidad | Razón | (TP/(TP+FN)) * 100 | 75% |
| **4** | Exactitud | Razón | (TP+TN)/(TP+TN+FP+FN) | 98.08% |
| **5** | F1-score | Razón | 2 * ((Precisión * Sensibilidad) / (Precisión + Sensibilidad)) | 74.07% |

**ALFREDO DAZA VERGARAY**

**DNI 40466240**

**ANEXO 14:**

## INTRUMENTO DE FICHA DE REGISTRO PARA EL ALGORITMO SVM

| RESUMEN DE EVALUACIÓN | |
|---|---|
| **Tipo de Prueba** | Post Test |
| **Investigadores** | - Huertas Fabian Nicole Valery<br>- Salcedo Zapata, Juan Antonio |
| **Fecha de inicio** | 29/10/2023 |
| **Algoritmo** | SVM |

| MATRIZ DE CONFUSIÓN | | | |
|---|---|---|---|
| | | **Estimación por el modelo** | |
| | | **Negativo (N)** | **Positivo (P)** |
| **Real** | **Negativo** | TN | FP |
| | **Positivo** | FN | TP |

| MÉTRICAS POR EVALUAR | | | | |
|---|---|---|---|---|
| **Ítem** | **Indicador** | **Medida** | **Fórmula** | **Precisión** |
| **1** | Especificidad | Razón | (TN/(TN+FP)) * 100 | 100% |
| **2** | Precisión | Razón | (TP/(TP+FP)) * 100 | 100% |
| **3** | Sensibilidad | Razón | (TP/(TP+FN)) * 100 | 100% |
| **4** | Exactitud | Razón | (TP+TN)/(TP+TN+FP+FN) | 100% |
| **5** | F1-score | Razón | 2 * ((Precisión * Sensibilidad) / (Precisión + Sensibilidad)) | 100% |

**ALFREDO DAZA VERGARAY**

**DNI 40466240**

**ANEXO 15:**

## INTRUMENTO DE FICHA DE REGISTRO PARA EL ALGORITMO ANN

| RESUMEN DE EVALUACIÓN | |
|---|---|
| **Tipo de Prueba** | Post Test |
| **Investigadores** | - Huertas Fabian Nicole Valery<br>- Salcedo Zapata, Juan Antonio |
| **Fecha de inicio** | 29/10/2023 |
| **Algoritmo** | ANN |

| MATRIZ DE CONFUSIÓN | | | |
|---|---|---|---|
| | | **Estimación por el modelo** | |
| | | **Negativo (N)** | **Positivo (P)** |
| **Real** | **Negativo** | TN | FP |
| | **Positivo** | FN | TP |

| MÉTRICAS POR EVALUAR | | | | |
|---|---|---|---|---|
| **Ítem** | **Indicador** | **Medida** | **Fórmula** | **Precisión** |
| **1** | Especificidad | Razón | (TN/(TN+FP)) * 100 | 90.78% |
| **2** | Precisión | Razón | (TP/(TP+FP)) * 100 | 54.56% |
| **3** | Sensibilidad | Razón | (TP/(TP+FN)) * 100 | 58.71% |
| **4** | Exactitud | Razón | (TP+TN)/(TP+TN+FP+FN) | 76.92% |
| **5** | F1-score | Razón | 2 * ((Precisión * Sensibilidad) / (Precisión + Sensibilidad)) | 55.93% |

**ALFREDO DAZA VERGARAY**

**DNI 40466240**

**ANEXO 16:**

**INTRUMENTO DE FICHA DE REGISTRO PARA EL ALGORITMO GBM**

| RESUMEN DE EVALUACIÓN | |
|---|---|
| **Tipo de Prueba** | Post Test |
| **Investigadores** | - Huertas Fabian Nicole Valery<br>- Salcedo Zapata, Juan Antonio |
| **Fecha de inicio** | 29/10/2023 |
| **Algoritmo** | GBM |

| MATRIZ DE CONFUSIÓN | | | |
|---|---|---|---|
| | | **Estimación por el modelo** | |
| | | **Negativo (N)** | **Positivo (P)** |
| **Real** | **Negativo** | TN | FP |
| | **Positivo** | FN | TP |

| MÉTRICAS POR EVALUAR | | | | |
|---|---|---|---|---|
| **Ítem** | **Indicador** | **Medida** | **Fórmula** | **Precisión** |
| **1** | Especificidad | Razón | (TN/(TN+FP)) * 100 | 100% |
| **2** | Precisión | Razón | (TP/(TP+FP)) * 100 | 100% |
| **3** | Sensibilidad | Razón | (TP/(TP+FN)) * 100 | 100% |
| **4** | Exactitud | Razón | (TP+TN)/(TP+TN+FP+FN) | 100% |
| **5** | F1-score | Razón | 2 * ((Precisión * Sensibilidad) / (Precisión + Sensibilidad)) | 100% |

**ALFREDO DAZA VERGARAY**

**DNI 40466240**

**ANEXO 17:**

## INTRUMENTO DE FICHA DE REGISTRO PARA EL ALGORITMO MP

| RESUMEN DE EVALUACIÓN | |
|---|---|
| **Tipo de Prueba** | Post Test |
| **Investigadores** | - Huertas Fabian Nicole Valery<br>- Salcedo Zapata, Juan Antonio |
| **Fecha de inicio** | 29/10/2023 |
| **Algoritmo** | MP |

| MATRIZ DE CONFUSIÓN | | | |
|---|---|---|---|
| | | **Estimación por el modelo** | |
| | | **Negativo (N)** | **Positivo (P)** |
| **Real** | **Negativo** | TN | FP |
| | **Positivo** | FN | TP |

| MÉTRICAS POR EVALUAR | | | | |
|---|---|---|---|---|
| **Ítem** | **Indicador** | **Medida** | **Fórmula** | **Precisión** |
| **1** | Especificidad | Razón | (TN/(TN+FP)) * 100 | 88.88% |
| **2** | Precisión | Razón | (TP/(TP+FP)) * 100 | 50.28% |
| **3** | Sensibilidad | Razón | (TP/(TP+FN)) * 100 | 52.84% |
| **4** | Exactitud | Razón | (TP+TN)/(TP+TN+FP+FN) | 75.00% |
| **5** | F1-score | Razón | 2 * ((Precisión * Sensibilidad) / (Precisión + Sensibilidad)) | 50.63% |

**ALFREDO DAZA VERGARAY**

**DNI 40466240**

**ANEXO 18:**

## INTRUMENTO DE FICHA DE REGISTRO PARA EL ALGORITMO KNN

| RESUMEN DE EVALUACIÓN | |
|---|---|
| **Tipo de Prueba** | Post Test |
| **Investigadores** | - Huertas Fabian Nicole Valery<br>- Salcedo Zapata, Juan Antonio |
| **Fecha de inicio** | 29/10/2023 |
| **Algoritmo** | KNN |

| MATRIZ DE CONFUSIÓN | | | |
|---|---|---|---|
| | | **Estimación por el modelo** | |
| | | **Negativo (N)** | **Positivo (P)** |
| **Real** | **Negativo** | TN | FP |
| | **Positivo** | FN | TP |

| MÉTRICAS POR EVALUAR | | | | |
|---|---|---|---|---|
| **Ítem** | **Indicador** | **Medida** | **Fórmula** | **Precisión** |
| **1** | Especificidad | Razón | (TN/(TN+FP)) * 100 | 90.20% |
| **2** | Precisión | Razón | (TP/(TP+FP)) * 100 | 60.23% |
| **3** | Sensibilidad | Razón | (TP/(TP+FN)) * 100 | 54.68% |
| **4** | Exactitud | Razón | (TP+TN)/(TP+TN+FP+FN) | 78.84% |
| **5** | F1-score | Razón | 2 * ((Precisión * Sensibilidad) / (Precisión + Sensibilidad)) | 55.96% |

**ALFREDO DAZA VERGARAY**

**DNI 40466240**

**ANEXO 19:**

## INTRUMENTO DE FICHA DE REGISTRO PARA EL ALGORITMO NB

| RESUMEN DE EVALUACIÓN | |
|---|---|
| **Tipo de Prueba** | Post Test |
| **Investigadores** | - Huertas Fabian Nicole Valery<br>- Salcedo Zapata, Juan Antonio |
| **Fecha de inicio** | 29/10/2023 |
| **Algoritmo** | NB |

| MATRIZ DE CONFUSIÓN | | | |
|---|---|---|---|
| | | **Estimación por el modelo** | |
| | | **Negativo (N)** | **Positivo (P)** |
| **Real** | **Negativo** | TN | FP |
| | **Positivo** | FN | TP |

| MÉTRICAS POR EVALUAR | | | | |
|---|---|---|---|---|
| **Ítem** | **Indicador** | **Medida** | **Fórmula** | **Precisión** |
| **1** | Especificidad | Razón | (TN/(TN+FP)) * 100 | 85.89% |
| **2** | Precisión | Razón | (TP/(TP+FP)) * 100 | 14.42% |
| **3** | Sensibilidad | Razón | (TP/(TP+FN)) * 100 | 25.00% |
| **4** | Exactitud | Razón | (TP+TN)/(TP+TN+FP+FN) | 57.69% |
| **5** | F1-score | Razón | 2 * ((Precisión * Sensibilidad) / (Precisión + Sensibilidad)) | 18.29% |

**ALFREDO DAZA VERGARAY**

**DNI 40466240**

**ANEXO 20:**

## INTRUMENTO DE FICHA DE REGISTRO PARA EL ALGORITMO LR

| RESUMEN DE EVALUACIÓN | |
|---|---|
| **Tipo de Prueba** | Post Test |
| **Investigadores** | - Huertas Fabian Nicole Valery<br>- Salcedo Zapata, Juan Antonio |
| **Fecha de inicio** | 29/10/2023 |
| **Algoritmo** | LR |

| MATRIZ DE CONFUSIÓN | | | |
|---|---|---|---|
| | | **Estimación por el modelo** | |
| | | **Negativo (N)** | **Positivo (P)** |
| **Real** | **Negativo** | TN | FP |
| | **Positivo** | FN | TP |

| MÉTRICAS POR EVALUAR | | | | |
|---|---|---|---|---|
| **Ítem** | **Indicador** | **Medida** | **Fórmula** | **Precisión** |
| **1** | Especificidad | Razón | (TN/(TN+FP)) * 100 | 97.58% |
| **2** | Precisión | Razón | (TP/(TP+FP)) * 100 | 69.24% |
| **3** | Sensibilidad | Razón | (TP/(TP+FN)) * 100 | 72.22% |
| **4** | Exactitud | Razón | (TP+TN)/(TP+TN+FP+FN) | 94.23% |
| **5** | F1-score | Razón | 2 * ((Precisión * Sensibilidad) / (Precisión + Sensibilidad)) | 70.66% |

**ALFREDO DAZA VERGARAY**

**DNI 40466240**

**ANEXO 21:**

## INTRUMENTO DE FICHA DE REGISTRO PARA EL ALGORITMO ADA

| RESUMEN DE EVALUACIÓN | |
|---|---|
| **Tipo de Prueba** | Post Test |
| **Investigadores** | - Huertas Fabian Nicole Valery<br>- Salcedo Zapata, Juan Antonio |
| **Fecha de inicio** | 29/10/2023 |
| **Algoritmo** | ADA |

| MATRIZ DE CONFUSIÓN | | | |
|---|---|---|---|
| | | **Estimación por el modelo** | |
| | | **Negativo (N)** | **Positivo (P)** |
| **Real** | **Negativo** | TN | FP |
| | **Positivo** | FN | TP |

| MÉTRICAS POR EVALUAR | | | | |
|---|---|---|---|---|
| **Ítem** | **Indicador** | **Medida** | **Fórmula** | **Precisión** |
| **1** | Especificidad | Razón | (TN/(TN+FP)) * 100 | 100% |
| **2** | Precisión | Razón | (TP/(TP+FP)) * 100 | 100% |
| **3** | Sensibilidad | Razón | (TP/(TP+FN)) * 100 | 100% |
| **4** | Exactitud | Razón | (TP+TN)/(TP+TN+FP+FN) * 100 | 100% |
| **5** | F1-score | Razón | 2 * ((Precisión * Sensibilidad) / (Precisión + Sensibilidad)) | 100% |

**ALFREDO DAZA VERGARAY**

**DNI 40466240**

**ANEXO 22:**

**Figura N° 33: Carta de Presentación**

UNIVERSIDAD CÉSAR VALLEJO

**Estimado Validador:**

Nos dirigimos hacia usted con el fin de solicitar su suma colaboración como experto para poder validar el cuestionario que será aplicado a los estudiantes de la I.E.P. "Nuestra Señora de Copacabana", agradeceríamos su evaluación junto con sus observaciones de ser requeridas.

Es de su consideración que el instrumento recaudará información directa para la investigación que se viene realizando, la cual se titula **"Sistema inteligente con Machine Learning basado en selección de variables para predecir el rendimiento académico de la I.E.P. "Nuestra Señora de Copacabana"**. El propósito es que pueda aplicarse de manera post-test.

Para la validación del instrumentos, usted deberá leer detalladamente cada enunciado y sus correspondientes alternativas en caso se especifiquen. Asimismo, es importante para nosotros poder recibir cualquier sugerencia en la redacción, contenido u otro aspecto con el fin de mejorar el trabajo.

**Gracias por su aporte.**

**Fuente: Elaboración Propia**

## ANEXO 23:

**VALIDACIÓN DEL INSTRUMENTO**

### I. DATOS GENERALES:

**Apellidos y Nombres del experto:** Daza Vergaray, Alfredo

**Título y/o grado:** Dr. Ingeniería de Sistemas

**Fecha:** 16/07/2023

**Instrumento**: Cuestionario

**Autores:** Huertas Fabian Nicole Valery, y Salcedo Zapata Juan Antonio

**Título de la investigación:**

Sistema inteligente con Machine Learning basado en selección de variables para predecir el rendimiento académico de la I.E.P. “Nuestra Señora de Copacabana”

### II. ASPECTOS DE VALIDACIÓN:

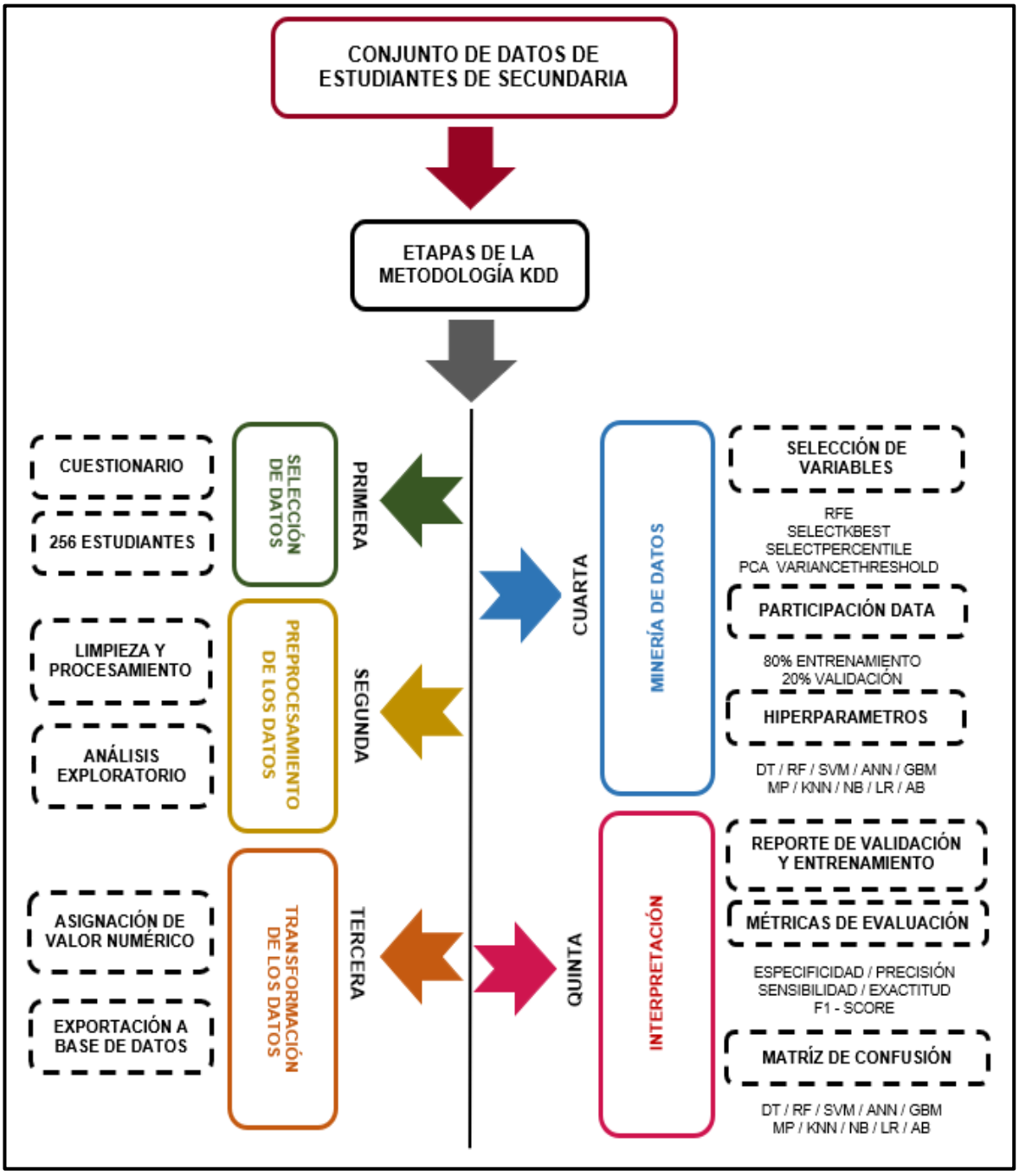
| INDICADORES | CRITERIOS | Deficiente 0 – 20% | Regular 21 – 50% | Bueno 51 – 70% | Muy bueno 71 – 80% | Excelente 81 – 100% |
|---|---|---|---|---|---|---|
| 1. Claridad | Está formulado con el lenguaje apropiado. | | | | | X |
| 2. Objetividad | Está expresado en conducta observable. | | | | | X |
| 3. Actualidad | Es adecuado al avance de la ciencia. | | | | | X |
| 4. Organización | Existe una organización lógica. | | | | | X |
| 5. Suficiencia | Comprende los aspectos de cantidad y calidad. | | | | | X |
| 6. Intencionalidad | Es adecuado para valorar aspectos del sistema metodológico y científico. | | | | | X |
| 7. Consistencia | Está basado en aspectos teóricos, científicos acordes a la tecnología educativa. | | | | | X |
| 8. Coherencia | Entre los índices, indicadores, dimensiones. | | | | | X |
| 9. Metodología | Responde al propósito del trabajo bajo los objetivos a lograr. | | | | | X |
| 10. Pertinencia | El instrumento es adecuado al tipo de investigación. | | | | | X |
| Promedio de Validación | | | | | | 90 % |

### III. Promedio de Valoración: 90%

### IV. Observaciones:

**ANEXO 24:**

**Figura N° 34: Enfoque empleado para el desarrollo del modelo**




**Fuente: Elaboración Propia**

**ANEXO 25:**

## Figura N° 35: Resolución del Consejo Universitario

UNIVERSIDAD CÉSAR VALLEJO

### RESOLUCIÓN DE CONSEJO UNIVERSITARIO N.° 0531-2021/UCV

Trujillo, 27 de julio de 2021

**VISTOS**: El Oficio N°291-2021-VI-UCV, que remite el Dr. Jorge Salas Ruiz, vicerrector de investigación de la Universidad César Vallejo y el acta de sesión ordinaria del Consejo Universitario, de fecha 27 de julio del presente año, que aprueba la actualización del **REGLAMENTO DE PROPIEDAD INTELECTUAL**; y

**CONSIDERANDO:**

Que junto al reconocimiento de las universidades como espacios de generación de conocimiento, recientemente empieza a cobrar fuerza la importancia que tiene la transferencia de los resultados de las actividades de investigación universitaria al sector productivo para lograr un mayor beneficio social. En este sentido, la propiedad intelectual y la transferencia de conocimientos y tecnologías, han sido entendidas como las herramientas indispensables para la promoción y el desarrollo de la economía basada en el conocimiento; esta percepción ha traído como consecuencia que se hayan desarrollado normas y herramientas de protección del conocimiento, así como de sus productos;

Que la protección jurídica al conocimiento que ofrece la Normatividad Nacional de Propiedad intelectual, tiene el propósito de estimular la investigación e intercambio de información frente al uso que terceros puedan hacer al conocimiento protegido;

Que mediante Resolución de Consejo Universitario N°0168-2020/UCV, de fecha 01 de julio del 2020, se aprobó el REGLAMENTO DE PROPIEDAD INTELECTUAL DE LA UNIVESIDAD CÉSAR VALLEJO; con la finalidad de establecer las normas de la Propiedad Intelectual que permiten regular todos los procesos que se generan como resultado de la actividad desarrollada por el personal docente, administrativo, estudiantes y egresados de la Universidad César Vallejo en el ejercicio de sus funciones con la universidad;

Que el Dr. Jorge Salas Ruiz, vicerrector de investigación, mediante Oficio N°291-2021-VI-UCV, ha solicitado al rectorado la aprobación de la actualización del **REGLAMENTO DE PROPIEDAD INTELECTUAL DE LA UNIVESIDAD CÉSAR VALLEJO,** elaborado por su área, con el objetivo de establecer las normas de la Propiedad Intelectual (PI) que permiten regular todos los procesos que se generan como resultado de la actividad desarrollada por el personal docente, administrativo, estudiantes y egresados en el ejercicio de sus funciones con la Universidad César Vallejo, las cuales están alineadas a la normativa vigente nacional e institucional;

Que, elevado el expediente al Consejo Universitario, en su sesión ordinaria del 27 de julio del año en curso, este órgano de gobierno ha evaluado el proyecto presentado y, encontrándolo conforme con los requerimientos técnicos básicos procedió a su aprobación con cargo a mejorar la redacción, encargándose al Dr. Jorge Salas Ruiz la presentación de la versión final del Reglamento de Propiedad Intelectual de la Universidad César Vallejo; documento que ya ha sido remitido; por lo cual es necesaria la emisión de la correspondiente resolución de consejo universitario;

Estando a lo expuesto y de conformidad con las normas legales y reglamentos vigentes.

**SE RESUELVE**:

UNIVERSIDAD CÉSAR VALLEJO

***Art. 1°.-*** **APROBAR** la actualización del **REGLAMENTO DE PROPIEDAD INTELECTUAL DE LA UNIVERSIDAD CÉSAR VALLEJO**, versión 03, presentado por el Vicerrectorado de Investigación, documento que como anexo 01 forma parte de la presente resolución de Consejo Universitario.

***Art. 2°.-*** **PRECISAR** que el reglamento actualizado que se aprueba en el artículo precedente será aplicado tanto en la Sede Institucional como en las filiales de la universidad César Vallejo y entrará en vigencia a partir de la emisión de la presente resolución.

***Art. 3°.-*** **DEJAR SIN EFECTO** la Resolución de Consejo Universitario N°0168-2020/UCV, de fecha 01 de julio del 2020, que aprobó el **REGLAMENTO DE PROPIEDAD INTELECTUAL DE LA UNIVERSIDAD CÉSAR VALLEJO.**

***Art. 4°.-*** **DISPONER** que los órganos académicos y administrativos de la universidad dicten las medidas y ejecuten las acciones necesarias para el cumplimiento de la presente resolución de Consejo Universitario.

Regístrese, comuníquese y cúmplase.

RECTORADO UNIVERSIDAD CÉSAR VALLEJO

**Dra. JEANNETTE TANTALEÁN RODRÍGUEZ**
*Rectora*

SECRETARÍA GENERAL UNIVERSIDAD CÉSAR VALLEJO

**Abog. ROSA LOMPARTE ROSALES**
*Secretaria General*

DISTRIBUCIÓN: Presidente de la JGA, presidenta del Directorio, rectora, presidenta ejecutiva, gerente general, V.A., VBU, V.I., directores generales, de la sede y filiales UCV, decanos, directores de escuela, coordinadores de carrera, DIT, Dir. Registros Académicos, Dir. Grados y Títulos, Centro de Información, archivo.

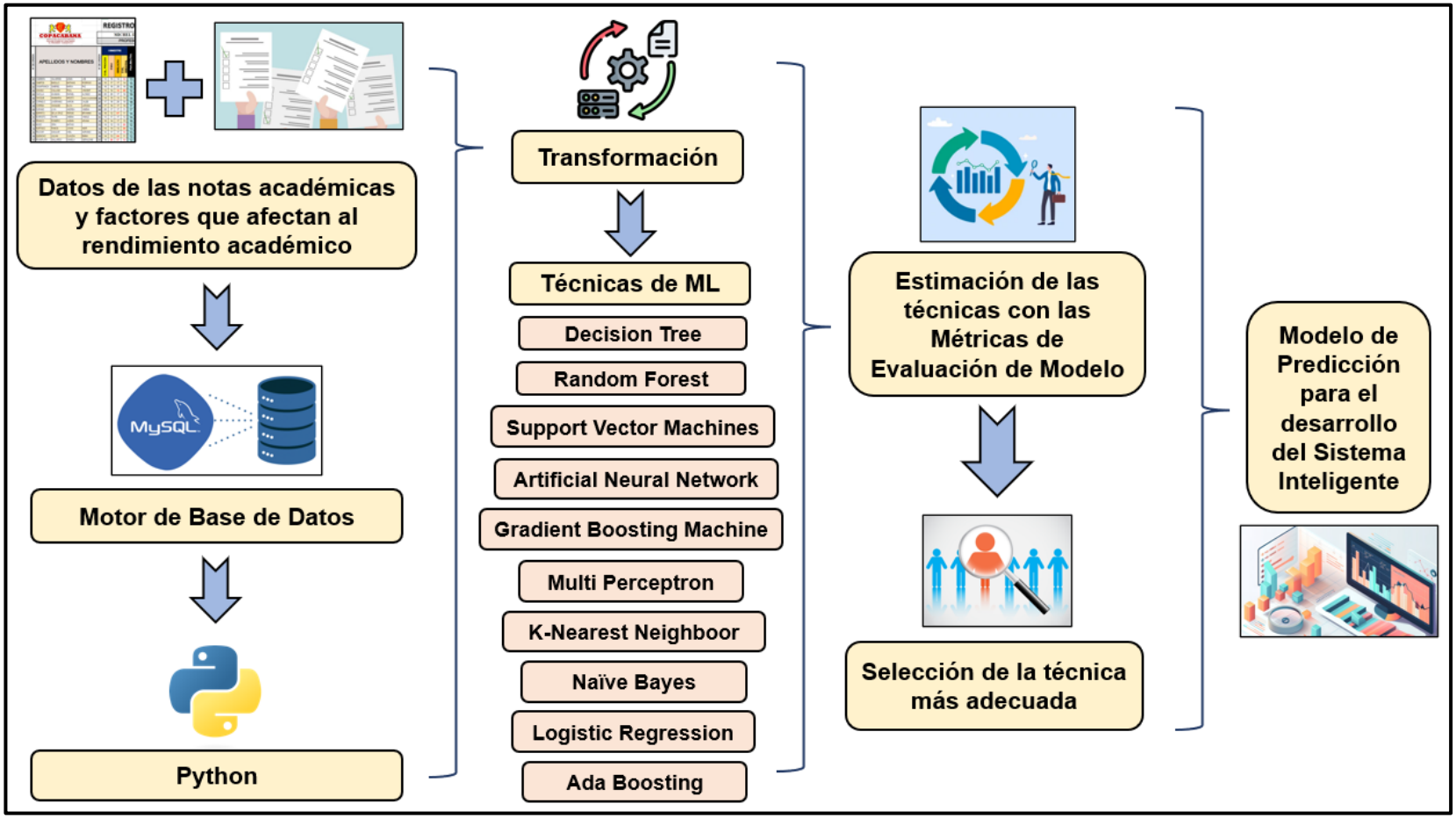
JCTR/rpach: asg



**Fuente: Universidad César Vallejo**

**ANEXO 27:**

**Figura N° 37: Prototipo**




**Fuente: Elaboración Propia**

**ANEXO 28:**

**Tabla N° 77: Innovación y aporte tecnológico**

| Antecedentes | Sistema Inteligente | Técnicas empleadas | Lenguaje de programación / Plataforma | Métricas usadas | Metodología de desarrollo |
|---|---|---|---|---|---|
| Preliminary evidence on machine learning approaches for clusterizing students' cognitive profile. | No | - ADA<br>- ANN | Python | - Exactitud<br>- F1-score | Metodología Propia<br>- Algoritmo de autoorganización de mapas de Kohonen y algoritmo de agrupación K-medias |
| Student Academic Performance Prognosticative Using Optimized Hybrid Machine Learning Algorithms. | No | - KNN, DT, RF,SVM, ANN, NB, GBM, ADA | No especifica | - Exactitud | Metodología Propia<br>- Recopilación de datos, selección de características, procesamiento, análisis de datos y la clasificación de alumnos |
| Using machine learning to predict factors affecting academic performance: the case of college students on academic probation. | No | - DT, RF, RT, NV, SVM, KNN, MP, LB<br>- Vote<br>- Bagging | No especifica | - Exactitud | Knowledge Discovery in Data bases (KDD) |
| Analysis and Prediction of Influencing Factors of College Student Achievement Based on Machine Learning. | No | - LR, SVM, NB, RF | No especifica | - Exactitud<br>- Precisión<br>- Sensibilidad | Metodología Propia<br>- Recopilación de datos, preprocesamiento, selección de características, construcción del modelo, evaluación del modelo y análisis de resultados |

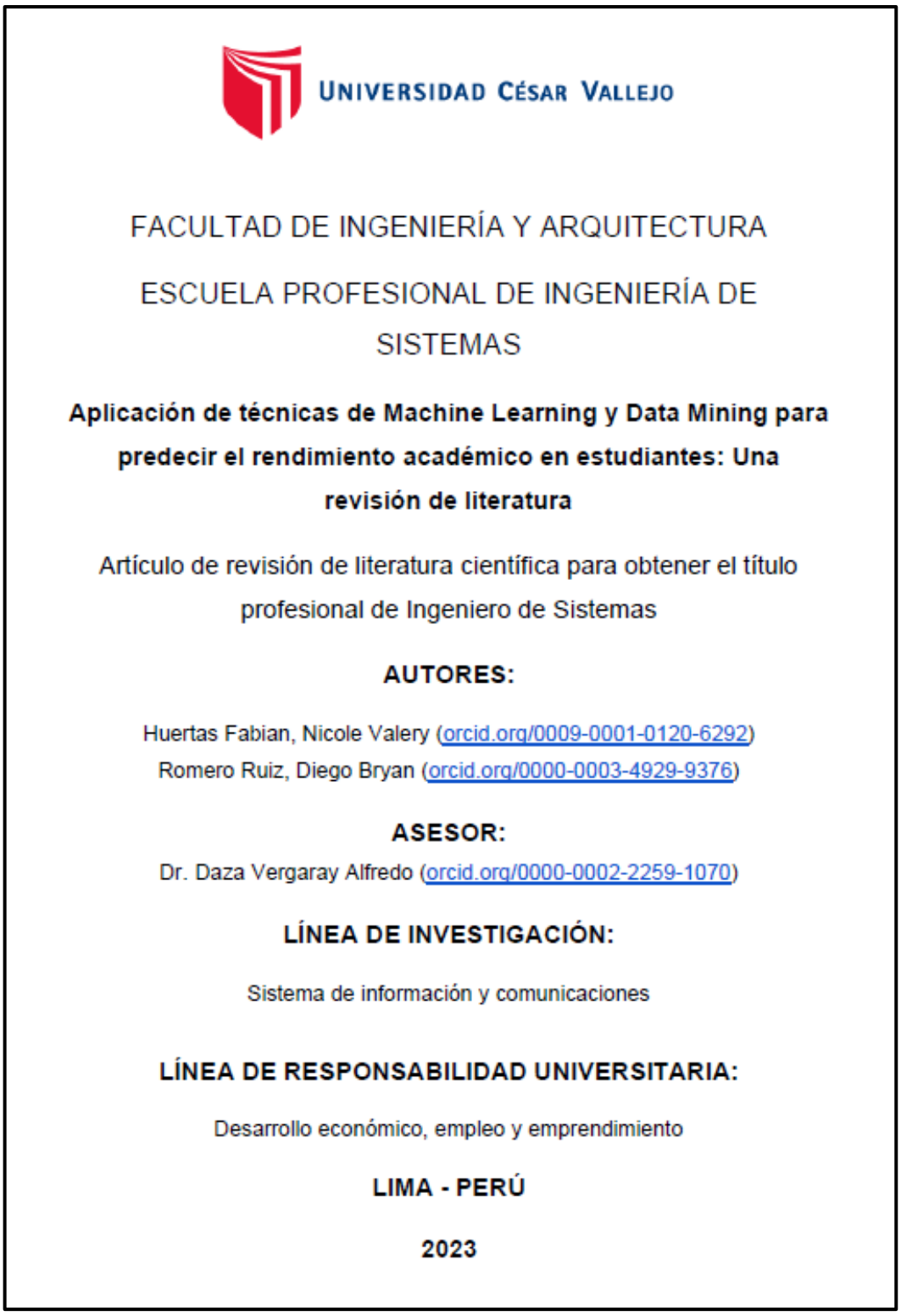
| | | | | | |
|---|---|---|---|---|---|
| Academic Performance Modelling with Machine Larning Based on Cognitive and Non-Cognitive Features. | No | - DT, KNN, ANN, LR, RF, ADA, SVM | Python | - AUC<br>- Exactitud | Metodología Propia<br>- Recopilación de datos, preprocesamiento y codificación numérica |
| Machine Learning para predecir el rendimiento académico de los estudiantes universitarios. | No | - SVM, DT, KNN | SPSS statistics y Modeler | - Exactitud<br>- Sensibilidad<br>- Precisión | Knowledge Discovery in Data bases (KDD) |
| Predicting Student Academic Performance Using Machine Learning. | No | - DT, RF, SVM, LR, ADA, XGBoost, CNN | Python | - Exactitud | Metodología Propia<br>- Definir herramientas, establecer el conjunto de datos, preprocesamiento de datos, ingeniería de funciones, modelado de ML y evaluación de desempeño. |
| An Empirical Study for Student Academic Performance Prediction Using Machine Learning Techniques. | No | - NB, SMO, MP, DT, RF, RT, PART, OneR | No especifica | - Exactitud | Metodología Propia<br>- Recopilación e integración de los datos, preprocesamiento, construcción y evaluación del modelo |
| Student Intervention System using Machine Learning Techniques. | No | - SVM, DT, NV | No especifica | - F1-score | Metodología Propia<br>- Recopilación de datos, preprocesamiento, construcción del modelo y evaluación |

| | | | | | |
|---|---|---|---|---|---|
| A Comparative Study of Machine Learning Algorithms for Student Academic Performance. | No | - KNN, SVM, DT, RF, GBM | No especifica | - Exactitud | Metodología Propia<br>- Recopilación, preprocesamiento, separación, resumen y predicciones |
| En la presente investigación, se plantea el desarrollo de un Sistema Inteligente con ML haciendo uso de la metodología KDD, considerando que los anteriores trabajos mencionados no han implementado un software para representar la información, además de utilizar dos fuentes de información, el cuestionario y la base de datos de notas académicas. Por tales motivos, este trabajo se considera de carácter innovador y original. De igual manera, se extraerá los aportes de los trabajos (técnicas y métricas) para la mejora de este mismo. | | | | | |

**Fuente: Elaboración propia**

**ANEXO 29:**

**Figura N° 38: Artículo de Revisión de Literatura**



UNIVERSIDAD CÉSAR VALLEJO

FACULTAD DE INGENIERÍA Y ARQUITECTURA

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

**Aplicación de técnicas de Machine Learning y Data Mining para predecir el rendimiento académico en estudiantes: Una revisión de literatura**

Artículo de revisión de literatura científica para obtener el título profesional de Ingeniero de Sistemas

**AUTORES:**

Huertas Fabian, Nicole Valery (orcid.org/0009-0001-0120-6292)
Romero Ruiz, Diego Bryan (orcid.org/0000-0003-4929-9376)

**ASESOR:**

Dr. Daza Vergaray Alfredo (orcid.org/0000-0002-2259-1070)

**LÍNEA DE INVESTIGACIÓN:**

Sistema de información y comunicaciones

**LÍNEA DE RESPONSABILIDAD UNIVERSITARIA:**

Desarrollo económico, empleo y emprendimiento

**LIMA - PERÚ**

**2023**

**Fuente: Elaboración Propia**

**Desarrollo del Sistema Inteligente con Machine Learning basado en selección de variables para predecir el rendimiento académico de la I.E.P. "Nuestra Señora de Copacabana"**

En esta sección se da a conocer las actividades que fueron aplicadas según la metodología KDD para el desarrollo de la propuesta tecnológica, ilustrada en el ANEXO 6, y el prototipo en el ANEXO 27. Dicha metodología, se dividió en 5 etapas, estas son las siguientes: selección de datos, preprocesamiento, transformación, minería de datos e interpretación. A continuación, se detalla lo elaborado en cada etapa:

**Primera etapa: Selección de datos**

El conjunto de datos fue recopilado aplicando el cuestionario plasmado anteriormente, véase el ANEXO 11, a la población de 256 alumnos de secundaria de la I.E.P. Nuestra Señora de Copacabana.

Dicho cuestionario estuvo compuesto por 5 apartados, los datos personales del estudiante (2 ítems), la autopercepción (5 ítems), la educación familiar (3 ítems), sociabilidad (3 ítems) y la economía familiar (3 ítems), con un total de 16 preguntas.

Respecto a las posibles respuestas, se debe tener en cuenta que, algunas son de tipo dicotómicas y otras con alternativas múltiples. En cuanto a su forma de aplicación, esta fue desarrollada utilizando Microsoft Word, tal como se muestra en la Figura N° 23, para posteriormente ser derivado de manera impresa a los alumnos de la institución educativa.

Una vez obtenida esta primera información, se pasó a extraer las notas académicas de los alumnos, de la base de datos brindada por la misma institución educativa, véase el ANEXO 10. Teniendo en cuenta las notas de los cursos de: Arte, Ciencias, Computación, Comunicación, Desarrollo Personal, Educación Física, Inglés, Matemática, Religión y Sociales, luego se procedió a calcular una nota promedio de todos los cursos.

**Segunda etapa: Preprocesamiento**

Es así como, se unieron ambas fuentes de información para recopilar un total de 256 datos que fueron traspasados de forma física a digital en un documento Excel de manera momentánea, tal como se muestra en la Figura N° 24.

**Figura N° 39: Cuestionario desarrollado mediante Microsoft Word**

## Cuestionario – Predicción del Rendimiento Académico

Queremos conocer tu opinión respecto a los factores que influyen en tu rendimiento académico, solo tomará unos minutos y es confidencial. Puedes marcar las respuestas con una X.

**1.- Datos Personales**

**¿Cuál es tu género biológico?** - Masculino ( ) - Femenino ( )
**¿Qué edad tienes?** _____

**2.- Autopercepción**

**¿Cómo consideras que es tu rendimiento académico?**
- Muy malo ( ) - Malo ( ) - Regular ( ) - Bueno ( ) - Muy bueno ( )
**¿Te consideras una persona inteligente?** - Si ( ) - No ( )
**¿Te consideras una persona responsable?** - Si ( ) - No ( )
**¿Cuántas horas al día dedicas al estudio?** _____
**¿Te organizas y planificas tu tiempo antes de rendir las evaluaciones?** - Si ( ) - No ( )

**3.- Educación Familiar**

| | Primaria | Secundaria | Técnico | Superior |
|---|---|---|---|---|
| ¿Qué nivel de educación tiene tu madre? | | | | |
| ¿Qué nivel de educación tiene tu padre? | | | | |
| ¿Cuál es el nivel más alto de educación que existe en tu familia? | | | | |

**4.- Sociabilidad**

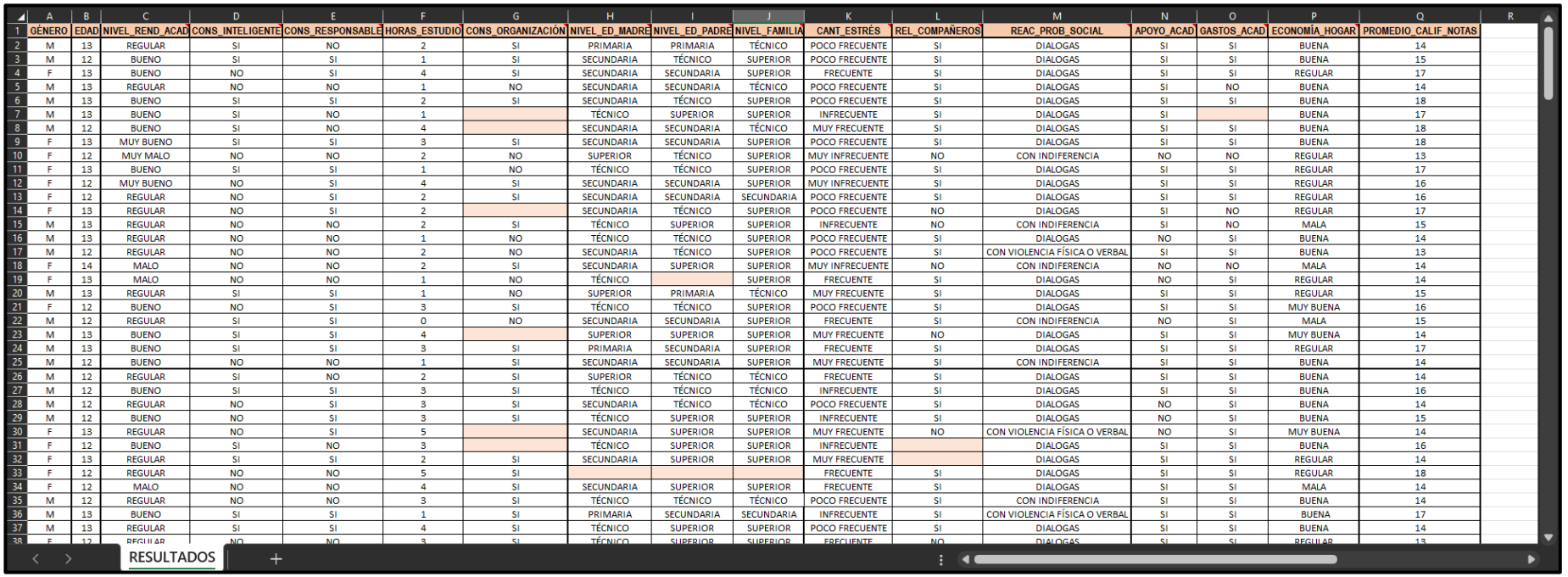
**¿Qué tan frecuente te sientes estresado(a) por los estudios?**
- Muy infrecuente ( ) - Infrecuente ( ) - Poco frecuente ( ) - Frecuente ( ) - Muy frecuente ( )
**¿Consideras que tienes una buena relación con tus compañeros de clase?**
- Si ( ) - No ( )
**¿Cómo reaccionas ante un problema de entorno social?**
- Con violencia física o verbal ( ) - Dialogas ( ) - Con indiferencia ( )

**5.- Economía Familiar**

**¿Te proporcionan todo lo necesario para rendir bien académicamente?** - Si ( ) - No ( )
**¿Tus padres generan ingresos suficientes para cubrir tus gastos académicos?**
- Si ( ) - No ( )
**¿Cómo consideras que es la economía en tu hogar?**
- Muy mala ( ) - Mala ( ) - Regular ( ) - Buena ( ) - Muy buena ( )

**Fuente: Elaboración Propia**

**Figura N° 40: Datos del cuestionario y la BD de notas académicas.**



| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | GÉNERO | EDAD | NIVEL_REND_ACAD | CONS_INTELIGENTE | CONS_RESPONSABLE | HORAS_ESTUDIO | CONS_ORGANIZACIÓN | NIVEL_ED_MADRE | NIVEL_ED_PADRE | NIVEL_FAMILIA | CANT_ESTRÉS | REL_COMPAÑEROS | REAC_PROB_SOCIAL | APOYO_ACAD | GASTOS_ACAD | ECONOMÍA_HOGAR | PROMEDIO_CALIF_NOTAS | |
| 2 | M | 13 | REGULAR | SI | NO | 2 | SI | PRIMARIA | PRIMARIA | TÉCNICO | POCO FRECUENTE | SI | DIALOGAS | SI | SI | BUENA | 14 | |
| 3 | M | 12 | BUENO | SI | SI | 1 | SI | SECUNDARIA | TÉCNICO | SUPERIOR | POCO FRECUENTE | SI | DIALOGAS | SI | SI | BUENA | 15 | |
| 4 | F | 13 | BUENO | NO | SI | 4 | SI | SECUNDARIA | SECUNDARIA | SUPERIOR | FRECUENTE | SI | DIALOGAS | SI | SI | REGULAR | 17 | |
| 5 | M | 13 | REGULAR | NO | NO | 1 | NO | SECUNDARIA | SECUNDARIA | TÉCNICO | POCO FRECUENTE | SI | DIALOGAS | SI | NO | BUENA | 14 | |
| 6 | M | 13 | BUENO | SI | SI | 2 | SI | SECUNDARIA | TÉCNICO | SUPERIOR | POCO FRECUENTE | SI | DIALOGAS | SI | SI | BUENA | 18 | |
| 7 | M | 13 | BUENO | SI | NO | 1 | | TÉCNICO | SUPERIOR | SUPERIOR | INFRECUENTE | SI | DIALOGAS | SI | | BUENA | 17 | |
| 8 | M | 12 | BUENO | SI | NO | 4 | | SECUNDARIA | SECUNDARIA | TÉCNICO | MUY FRECUENTE | SI | DIALOGAS | SI | SI | BUENA | 18 | |
| 9 | F | 13 | MUY BUENO | SI | SI | 3 | SI | SECUNDARIA | SECUNDARIA | SUPERIOR | POCO FRECUENTE | SI | DIALOGAS | SI | SI | BUENA | 18 | |
| 10 | F | 12 | MUY MALO | NO | NO | 2 | NO | SUPERIOR | TÉCNICO | SUPERIOR | MUY INFRECUENTE | NO | CON INDIFERENCIA | NO | NO | REGULAR | 13 | |
| 11 | F | 13 | BUENO | SI | SI | 1 | NO | TÉCNICO | TÉCNICO | SUPERIOR | POCO FRECUENTE | SI | DIALOGAS | SI | SI | REGULAR | 17 | |
| 12 | F | 12 | MUY BUENO | NO | SI | 4 | SI | SECUNDARIA | SECUNDARIA | SUPERIOR | MUY INFRECUENTE | SI | DIALOGAS | SI | SI | REGULAR | 16 | |
| 13 | F | 12 | REGULAR | NO | SI | 2 | SI | SECUNDARIA | SECUNDARIA | SECUNDARIA | POCO FRECUENTE | SI | DIALOGAS | SI | SI | REGULAR | 16 | |
| 14 | F | 13 | REGULAR | NO | SI | 2 | | SECUNDARIA | TÉCNICO | SUPERIOR | POCO FRECUENTE | NO | DIALOGAS | SI | NO | REGULAR | 17 | |
| 15 | M | 13 | REGULAR | NO | NO | 2 | SI | TÉCNICO | SUPERIOR | SUPERIOR | INFRECUENTE | NO | CON INDIFERENCIA | SI | NO | MALA | 15 | |
| 16 | M | 13 | REGULAR | NO | NO | 1 | NO | TÉCNICO | TÉCNICO | SUPERIOR | POCO FRECUENTE | SI | DIALOGAS | NO | SI | BUENA | 14 | |
| 17 | M | 12 | REGULAR | NO | NO | 2 | NO | SECUNDARIA | TÉCNICO | SUPERIOR | POCO FRECUENTE | SI | CON VIOLENCIA FÍSICA O VERBAL | SI | SI | BUENA | 13 | |
| 18 | F | 14 | MALO | NO | NO | 2 | SI | SECUNDARIA | SUPERIOR | SUPERIOR | MUY INFRECUENTE | NO | CON INDIFERENCIA | NO | NO | MALA | 14 | |
| 19 | F | 13 | MALO | NO | NO | 1 | NO | TÉCNICO | | SUPERIOR | FRECUENTE | SI | DIALOGAS | NO | SI | REGULAR | 14 | |
| 20 | M | 13 | REGULAR | SI | SI | 1 | NO | SUPERIOR | PRIMARIA | TÉCNICO | MUY FRECUENTE | SI | DIALOGAS | SI | SI | REGULAR | 15 | |
| 21 | F | 12 | BUENO | NO | SI | 3 | SI | TÉCNICO | TÉCNICO | SUPERIOR | POCO FRECUENTE | SI | DIALOGAS | SI | SI | MUY BUENA | 16 | |
| 22 | M | 12 | REGULAR | SI | SI | 0 | NO | SECUNDARIA | SECUNDARIA | SUPERIOR | FRECUENTE | SI | CON INDIFERENCIA | NO | SI | MALA | 15 | |
| 23 | M | 13 | BUENO | SI | SI | 4 | | SUPERIOR | SUPERIOR | SUPERIOR | MUY FRECUENTE | NO | DIALOGAS | SI | SI | MUY BUENA | 14 | |
| 24 | M | 13 | BUENO | SI | SI | 3 | SI | PRIMARIA | SECUNDARIA | SUPERIOR | FRECUENTE | SI | DIALOGAS | SI | SI | REGULAR | 17 | |
| 25 | M | 12 | BUENO | NO | NO | 1 | SI | SECUNDARIA | SECUNDARIA | SUPERIOR | MUY FRECUENTE | SI | CON INDIFERENCIA | SI | SI | BUENA | 14 | |
| 26 | M | 12 | REGULAR | SI | NO | 2 | SI | SUPERIOR | TÉCNICO | TÉCNICO | FRECUENTE | SI | DIALOGAS | SI | SI | BUENA | 14 | |
| 27 | M | 12 | BUENO | SI | SI | 3 | SI | TÉCNICO | TÉCNICO | TÉCNICO | INFRECUENTE | SI | DIALOGAS | SI | SI | BUENA | 16 | |
| 28 | M | 12 | REGULAR | NO | SI | 3 | SI | SECUNDARIA | TÉCNICO | TÉCNICO | POCO FRECUENTE | SI | DIALOGAS | NO | SI | BUENA | 14 | |
| 29 | M | 12 | BUENO | NO | SI | 3 | SI | TÉCNICO | SUPERIOR | SUPERIOR | INFRECUENTE | SI | DIALOGAS | NO | SI | BUENA | 15 | |
| 30 | F | 13 | REGULAR | NO | SI | 5 | | SECUNDARIA | SUPERIOR | SUPERIOR | MUY FRECUENTE | NO | CON VIOLENCIA FÍSICA O VERBAL | NO | SI | MUY BUENA | 14 | |
| 31 | F | 12 | BUENO | SI | NO | 3 | | TÉCNICO | SUPERIOR | SUPERIOR | INFRECUENTE | | DIALOGAS | SI | SI | BUENA | 16 | |
| 32 | F | 13 | REGULAR | SI | SI | 2 | SI | SECUNDARIA | SUPERIOR | SUPERIOR | MUY FRECUENTE | | DIALOGAS | SI | SI | REGULAR | 14 | |
| 33 | F | 12 | REGULAR | NO | NO | 5 | SI | | | | FRECUENTE | SI | DIALOGAS | SI | SI | REGULAR | 18 | |
| 34 | F | 12 | MALO | NO | NO | 4 | SI | SECUNDARIA | SUPERIOR | SUPERIOR | FRECUENTE | SI | DIALOGAS | SI | SI | MALA | 14 | |
| 35 | M | 12 | REGULAR | NO | NO | 3 | SI | TÉCNICO | TÉCNICO | TÉCNICO | POCO FRECUENTE | SI | CON INDIFERENCIA | SI | SI | BUENA | 14 | |
| 36 | M | 13 | BUENO | SI | SI | 1 | SI | PRIMARIA | SECUNDARIA | SECUNDARIA | INFRECUENTE | SI | CON VIOLENCIA FÍSICA O VERBAL | SI | SI | BUENA | 17 | |
| 37 | M | 13 | REGULAR | SI | SI | 4 | SI | TÉCNICO | SUPERIOR | SUPERIOR | POCO FRECUENTE | SI | DIALOGAS | SI | SI | BUENA | 14 | |
| 38 | F | 12 | REGULAR | NO | NO | 3 | SI | TÉCNICO | SUPERIOR | SUPERIOR | FRECUENTE | NO | DIALOGAS | SI | SI | REGULAR | 13 | |

RESULTADOS

**Fuente: Elaboración Propia**

**Tercera etapa: Transformación**

Luego de almacenar el conjunto de datos en un documento Excel, se procedió a asignarles un valor numérico a las respuestas para una mejor manipulación de los datos, teniendo en cuenta que las respuestas vacías tendrán el valor de 0 con un significado de “Sin especificar”. A continuación, se muestra el valor asignado por cada variable:

**Tabla N° 78: Valores numéricos por cada variable**

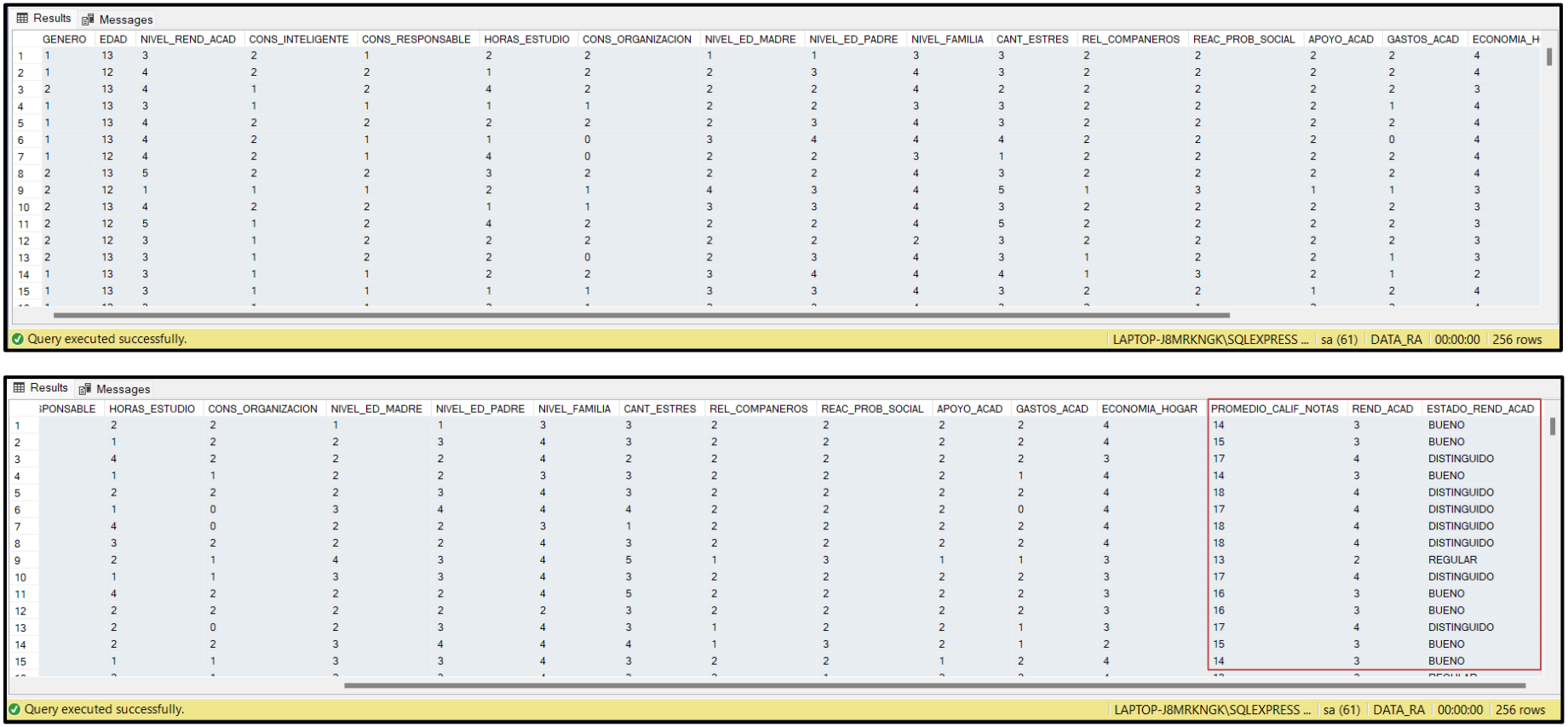
| | | |
|---|---|---|
| **GÉNERO** | Femenino | 2 |
| | Masculino | 1 |
| | | |
| **NIVEL_REND_ACAD** | Muy Malo | 1 |
| | Malo | 2 |
| | Regular | 3 |
| | Bueno | 4 |
| | Muy bueno | 5 |
| | | |
| **CONS_INTELIGENTE** | Si | 2 |
| | No | 1 |
| | | |
| **CONS_RESPONSABLE** | Si | 2 |
| | No | 1 |
| | | |
| **CONS_ORGANIZACIÓN** | Si | 2 |
| | No | 1 |
| | | |
| **NIVEL_ED_MADRE** | Primaria | 1 |
| | Secundaria | 2 |
| | Técnico | 3 |
| | Superior | 4 |
| | | |
| **NIVEL_ED_PADRE** | Primaria | 1 |
| | Secundaria | 2 |
| | Técnico | 3 |
| | Superior | 4 |
| | | |
| **NIVEL_FAMILIA** | Primaria | 1 |
| | Secundaria | 2 |
| | Técnico | 3 |
| | Superior | 4 |
| | | |
| **CANT_ESTRÉS** | Muy frecuente | 1 |
| | Frecuente | 2 |

| | | |
|---|---|---|
| | Poco Frecuente | 3 |
| | Infrecuente | 4 |
| | Muy infrecuente | 5 |
| | | |
| **REL_COMPAÑEROS** | Si | 2 |
| | No | 1 |
| | | |
| **REAC_PROB_SOCIAL** | Con violencia física o verbal | 1 |
| | Dialogas | 2 |
| | Con indiferencia | 3 |
| | | |
| **APOYO_ACAD** | Si | 2 |
| | No | 1 |
| | | |
| **GASTOS_ACAD** | Si | 2 |
| | No | 1 |
| | | |
| **ECONOMÍA_HOGAR** | Muy mala | 1 |
| | Mala | 2 |
| | Regular | 3 |
| | Buena | 4 |
| | Muy buena | 5 |

**Fuente: Elaboración propia**

Posteriormente, para poder establecer la data que se usará, se realizó el cálculo del rendimiento académico en base al promedio de notas del alumno. Es así como, partiendo de la variable **PROMEDIO_CALIF_NOTAS** se añadieron dos variables llamadas **REND_ACAD y ESTADO_REND_ACAD**, el cálculo de estas variables se muestra en la Tabla N° 3. Teniendo ya definido la data con los valores numéricos, se procedió a importar el documento Excel a SQL Server en una base de datos llamada “DATA_RA” con la tabla denominada “ALUMNOS”, véase la Figura N° 25.

**Figura N° 41: BD con valores numéricos exportados a SQL Server.**



Results Messages

| | GENERO | EDAD | NIVEL_REND_ACAD | CONS_INTELIGENTE | CONS_RESPONSABLE | HORAS_ESTUDIO | CONS_ORGANIZACION | NIVEL_ED_MADRE | NIVEL_ED_PADRE | NIVEL_FAMILIA | CANT_ESTRES | REL_COMPANEROS | REAC_PROB_SOCIAL | APOYO_ACAD | GASTOS_ACAD | ECONOMIA_H |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 13 | 3 | 2 | 1 | 2 | 2 | 1 | 1 | 3 | 3 | 2 | 2 | 2 | 2 | 4 |
| 2 | 1 | 12 | 4 | 2 | 2 | 1 | 2 | 2 | 3 | 4 | 3 | 2 | 2 | 2 | 2 | 4 |
| 3 | 2 | 13 | 4 | 1 | 2 | 4 | 2 | 2 | 2 | 4 | 2 | 2 | 2 | 2 | 2 | 3 |
| 4 | 1 | 13 | 3 | 1 | 1 | 1 | 1 | 2 | 2 | 3 | 3 | 2 | 2 | 2 | 1 | 4 |
| 5 | 1 | 13 | 4 | 2 | 2 | 2 | 2 | 2 | 3 | 4 | 3 | 2 | 2 | 2 | 2 | 4 |
| 6 | 1 | 13 | 4 | 2 | 1 | 1 | 0 | 3 | 4 | 4 | 4 | 2 | 2 | 2 | 0 | 4 |
| 7 | 1 | 12 | 4 | 2 | 1 | 4 | 0 | 2 | 2 | 3 | 1 | 2 | 2 | 2 | 2 | 4 |
| 8 | 2 | 13 | 5 | 2 | 2 | 3 | 2 | 2 | 2 | 4 | 3 | 2 | 2 | 2 | 2 | 4 |
| 9 | 2 | 12 | 1 | 1 | 1 | 2 | 1 | 4 | 3 | 4 | 5 | 1 | 3 | 1 | 1 | 3 |
| 10 | 2 | 13 | 4 | 2 | 2 | 1 | 1 | 3 | 3 | 4 | 3 | 2 | 2 | 2 | 2 | 3 |
| 11 | 2 | 12 | 5 | 1 | 2 | 4 | 2 | 2 | 2 | 4 | 5 | 2 | 2 | 2 | 2 | 3 |
| 12 | 2 | 12 | 3 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 2 | 2 | 2 | 2 | 3 |
| 13 | 2 | 13 | 3 | 1 | 2 | 2 | 0 | 2 | 3 | 4 | 3 | 1 | 2 | 2 | 1 | 3 |
| 14 | 1 | 13 | 3 | 1 | 1 | 2 | 2 | 3 | 4 | 4 | 4 | 1 | 3 | 2 | 1 | 2 |
| 15 | 1 | 13 | 3 | 1 | 1 | 1 | 1 | 3 | 3 | 4 | 3 | 2 | 2 | 1 | 2 | 4 |

Query executed successfully. LAPTOP-J8MRKNGK\SQLEXPRESS ... sa (61) DATA_RA 00:00:00 256 rows



Results Messages

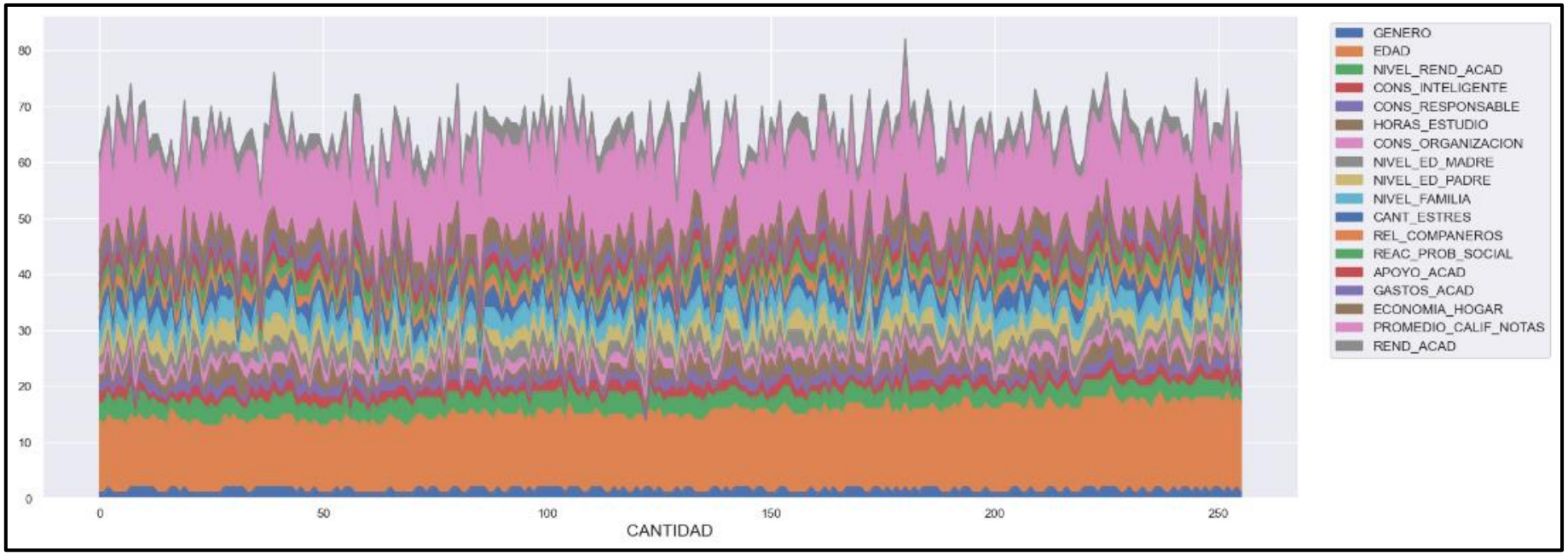
| | PONSABLE | HORAS_ESTUDIO | CONS_ORGANIZACION | NIVEL_ED_MADRE | NIVEL_ED_PADRE | NIVEL_FAMILIA | CANT_ESTRES | REL_COMPANEROS | REAC_PROB_SOCIAL | APOYO_ACAD | GASTOS_ACAD | ECONOMIA_HOGAR | PROMEDIO_CALIF_NOTAS | REND_ACAD | ESTADO_REND_ACAD |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | 2 | 2 | 1 | 1 | 3 | 3 | 2 | 2 | 2 | 2 | 4 | 14 | 3 | BUENO |
| 2 | | 1 | 2 | 2 | 3 | 4 | 3 | 2 | 2 | 2 | 2 | 4 | 15 | 3 | BUENO |
| 3 | | 4 | 2 | 2 | 2 | 4 | 2 | 2 | 2 | 2 | 2 | 3 | 17 | 4 | DISTINGUIDO |
| 4 | | 1 | 1 | 2 | 2 | 3 | 3 | 2 | 2 | 2 | 1 | 4 | 14 | 3 | BUENO |
| 5 | | 2 | 2 | 2 | 3 | 4 | 3 | 2 | 2 | 2 | 2 | 4 | 18 | 4 | DISTINGUIDO |
| 6 | | 1 | 0 | 3 | 4 | 4 | 4 | 2 | 2 | 2 | 0 | 4 | 17 | 4 | DISTINGUIDO |
| 7 | | 4 | 0 | 2 | 2 | 3 | 1 | 2 | 2 | 2 | 2 | 4 | 18 | 4 | DISTINGUIDO |
| 8 | | 3 | 2 | 2 | 2 | 4 | 3 | 2 | 2 | 2 | 2 | 4 | 18 | 4 | DISTINGUIDO |
| 9 | | 2 | 1 | 4 | 3 | 4 | 5 | 1 | 3 | 1 | 1 | 3 | 13 | 2 | REGULAR |
| 10 | | 1 | 1 | 3 | 3 | 4 | 3 | 2 | 2 | 2 | 2 | 3 | 17 | 4 | DISTINGUIDO |
| 11 | | 4 | 2 | 2 | 2 | 4 | 5 | 2 | 2 | 2 | 2 | 3 | 16 | 3 | BUENO |
| 12 | | 2 | 2 | 2 | 2 | 2 | 3 | 2 | 2 | 2 | 2 | 3 | 16 | 3 | BUENO |
| 13 | | 2 | 0 | 2 | 3 | 4 | 3 | 1 | 2 | 2 | 1 | 3 | 17 | 4 | DISTINGUIDO |
| 14 | | 2 | 2 | 3 | 4 | 4 | 4 | 1 | 3 | 2 | 1 | 2 | 15 | 3 | BUENO |
| 15 | | 1 | 1 | 3 | 3 | 4 | 3 | 2 | 2 | 1 | 2 | 4 | 14 | 3 | BUENO |

Query executed successfully. LAPTOP-J8MRKNGK\SQLEXPRESS ... sa (61) DATA_RA 00:00:00 256 rows

**Fuente: Elaboración Propia**

Continuando con el desarrollo, se dio inicio al uso de la herramienta Anaconda y Jupyter Notebooks para realizar el análisis exploratorio. Como vista general, se creó una gráfica que incluye todas las variables.

**Figura N° 42: Gráfico de Área de Calor**

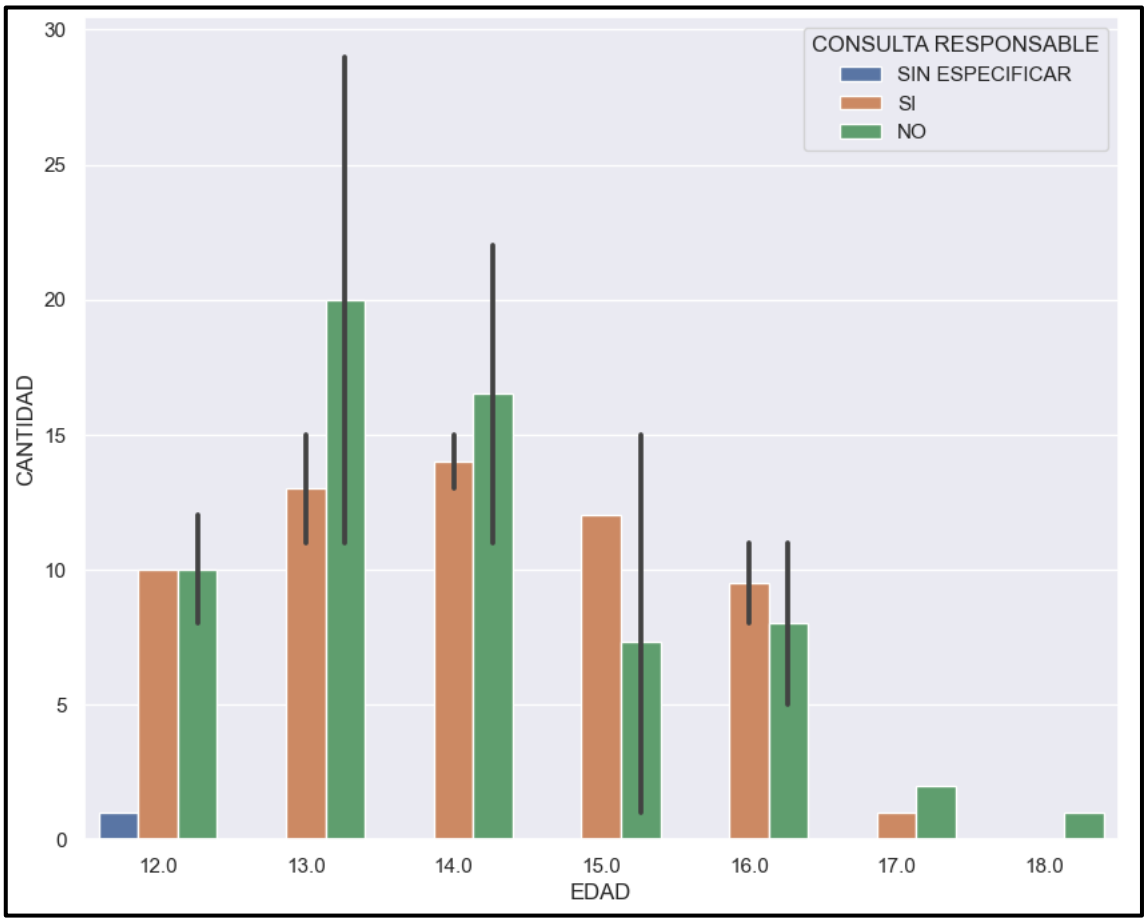



**Fuente: Elaboración propia**

Mediante la consulta a la base de datos “DATA_RA”, se realiza la selección y recupera los datos de la tabla “ALUMNOS”, luego mediante el gráfico de área o calor muestra los atributos de la tabla consultada, en el eje X se muestra la cantidad de alumnos y en el eje Y se muestra todos los atributos que pertenecen a dicha tabla.

Por otra parte, el siguiente gráfico muestra las respuestas de los 256 alumnos, en el que analiza, compara y agrupan en categorías específicas.

**Figura N° 43: Gráfico de Barras Agrupadas**

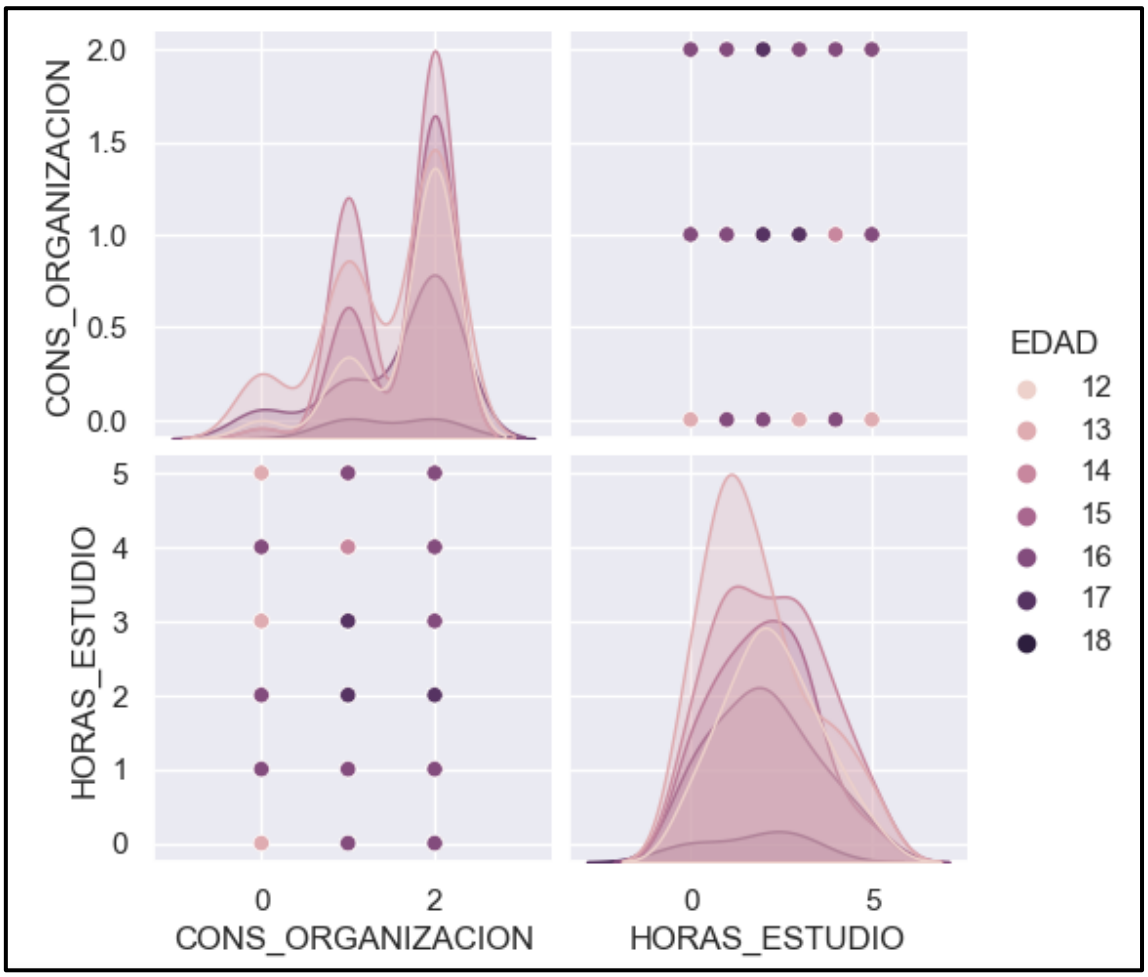



**Fuente: Elaboración propia**

Mediante la consulta realizada a la base de datos “DATA_RA”, realiza la selección y recupera los datos de la tabla “ALUMNOS”, luego mediante el gráfico de barras agrupadas muestra la distribución de la pregunta “¿Te consideras una persona responsable?” en función de la edad, en el eje X se muestra la edad de los alumnos y en el eje Y se muestra la cantidad de respuestas y como leyenda se visualiza las alternativas brindadas.

De acuerdo con el siguiente gráfico, se muestra las respuestas de los 256 alumnos, dónde cada fila y columna de la matriz representa una variable distinta, y cada celda muestra un gráfico de dispersión de la relación entre las variables correspondientes.

**Figura N° 44: Gráfico de Pares**

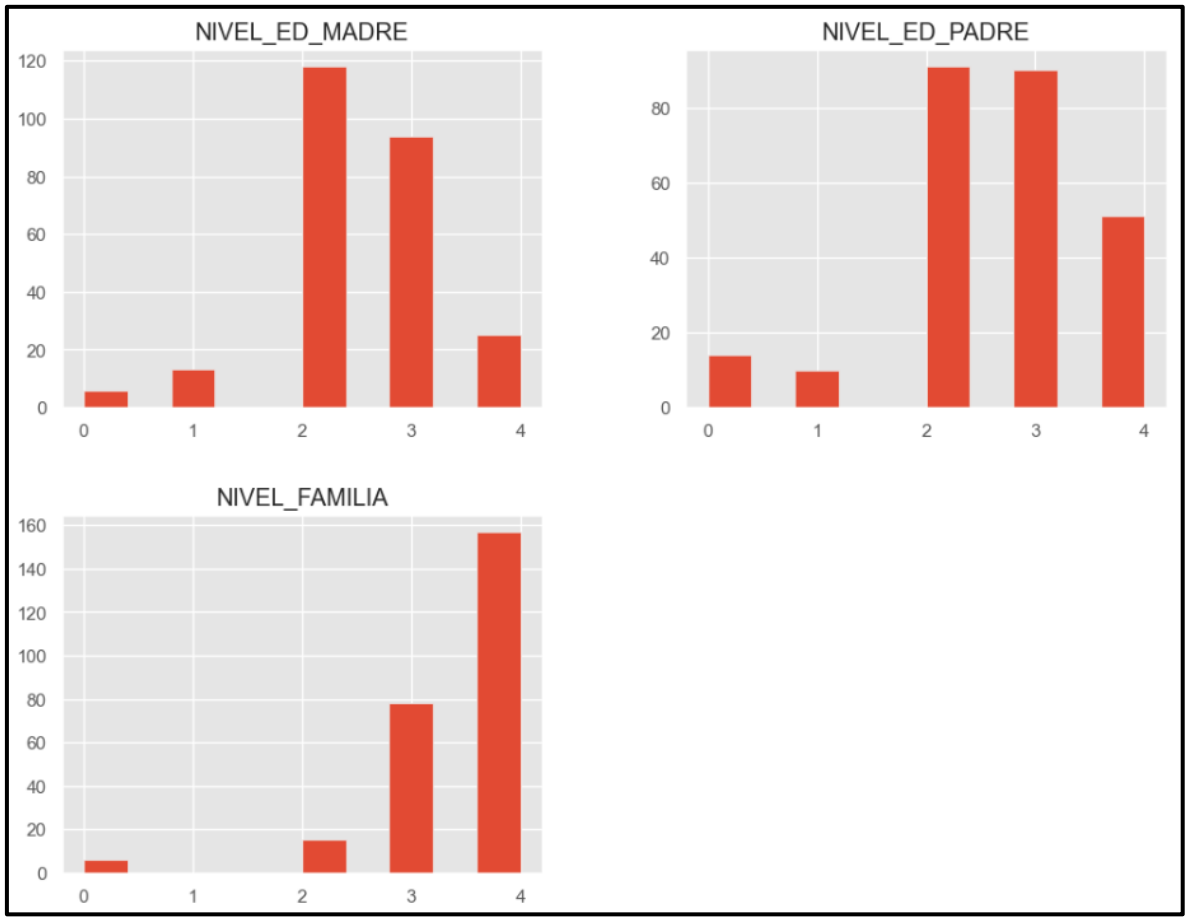



**Fuente: Elaboración propia**

Mediante la consulta realizada a la base de datos “DATA_RA”, realiza la selección y recupera los datos de la tabla “ALUMNOS”, luego mediante el gráfico de dispersión múltiple, muestra la relación entre las horas y la organización de estudio en función de la edad de los alumnos, identificando los patrones o tendencias en el comportamiento y sus hábitos de estudio.

Como se especifica en el siguiente gráfico, se muestra las respuestas de los 256 alumnos, en el que se identifica patrones de comportamiento y distribución en un conjunto de datos.

**Figura N° 45: Gráfico de Histogramas**

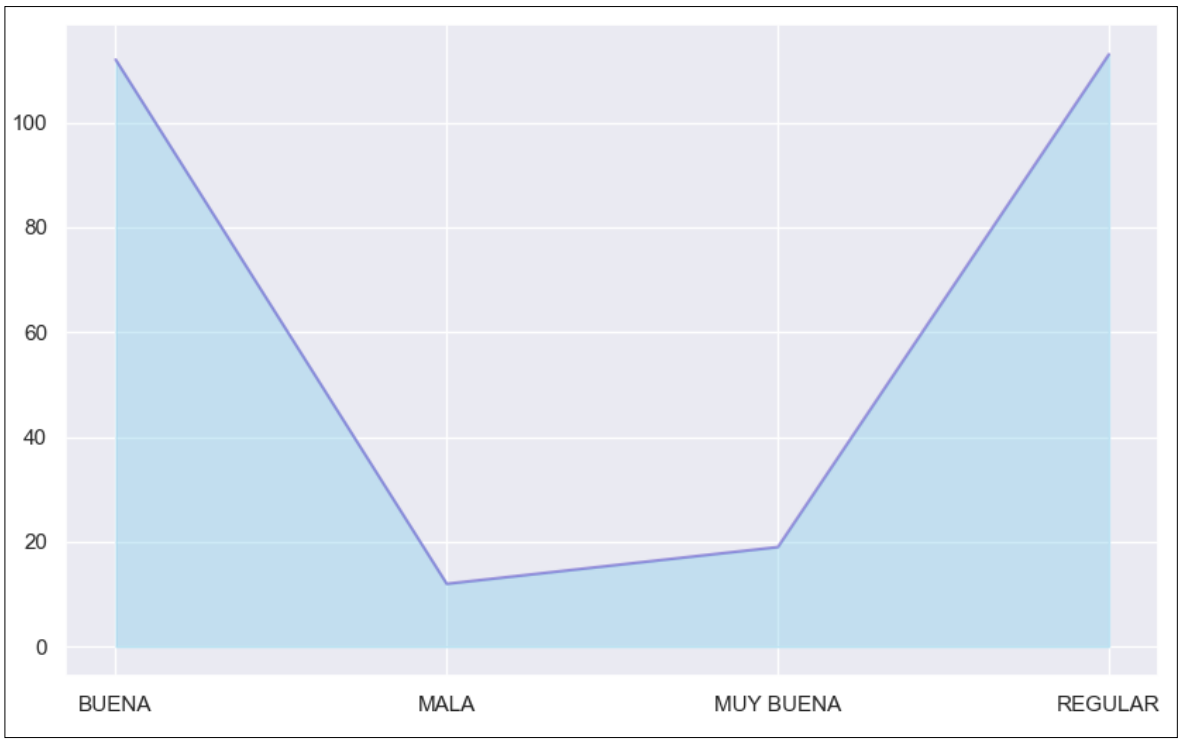



**Fuente: Elaboración propia**

Mediante la consulta realizada a la base de datos “DATA_RA, realiza la selección y recupera los datos de la tabla “ALUMNOS”, luego mediante el gráfico de histogramas, muestra la distribución de los respuestas de las siguientes preguntas: ¿Qué nivel de educación tiene tu madre?, ¿Qué nivel de educación tiene tu padre? y ¿Cuál es el nivel más alto de educación que existe en tu familia? en relación con la descripción de la Tabla N° 88.

De igual manera, el siguiente gráfico muestra las respuestas de los 256 alumnos, en el que muestra la evolución y compara las cantidades dentro de un conjunto de categorías.

**Figura N° 46: Gráfico de Área**

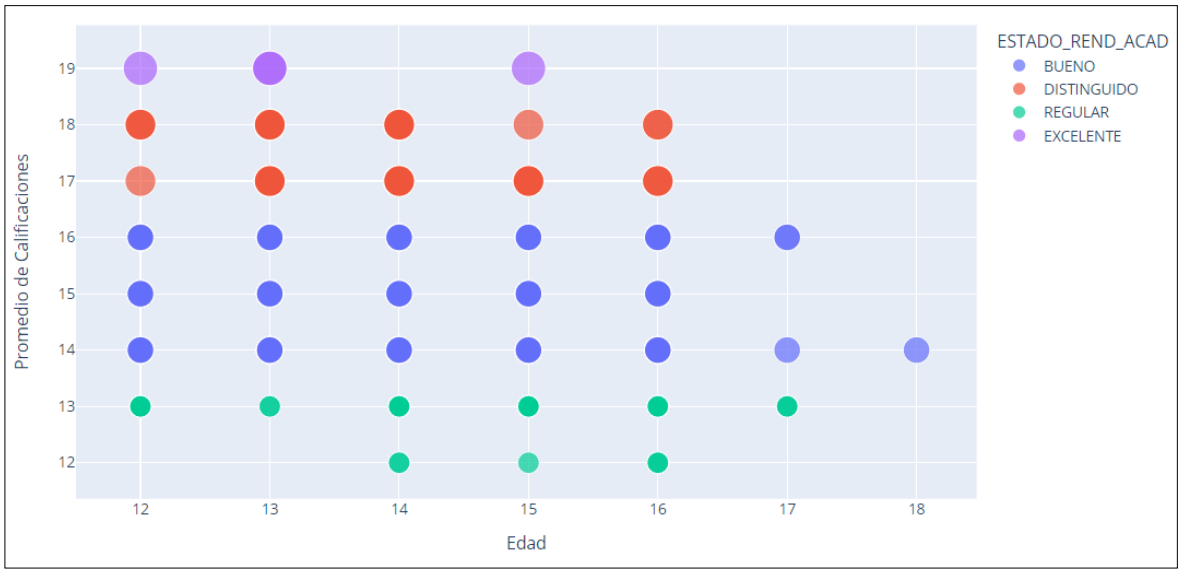



**Fuente: Elaboración propia**

Mediante la consulta realizada a la base de datos “DATA_RA”, realiza la selección y recupera los datos de la tabla “ALUMNOS”, luego mediante de gráfico de área muestra la distribución en función de la economía del hogar, en el eje X se muestra la descripción de los estados y en el eje Y se muestra la cantidad de respuestas.

Asimismo, el siguiente grafico muestra las respuestas de los 256 alumnos, en el que muestra la evaluación del rendimiento académico en función de la edad y el promedio de calificación de los alumnos.

**Figura N° 47: Gráfico Interactivo**




**Fuente: Elaboración propia**

Mediante la consulta realizada a la base de datos “DATA_RA”, realiza la selección y recupera los datos de la tabla “ALUMNOS”, luego mediante el gráfico interactivo muestra la el promedio de las calificaciones por edad y el estado del rendimiento académico, en el eje X se muestra la edad de los alumnos y en el eje Y se muestra el rango de las notas, el color de los puntos representa el estado de rendimiento académico de los alumnos y como leyenda cada estado de rendimiento académico que se puede distinguir por un color diferente.

Una vez finalizado el análisis exploratorio, se procedió a realizar la selección de variables en base a 4 algoritmos: RFE, Selectkbest, Selectpercentile, y Puntuación PCA y Variancethreshold. Para esto, se incluyeron todas las variables a excepción de **ESTADO_REND_ACAD**, ya que sólo define el valor de la variable de **REND_ACAD.**

Asimismo, previamente se hizo la importación de las librerías para la conexión a la base de datos y de los cuatro algoritmos de selección de variables en el entorno interactivo Jupyter Notebooks. A continuación, se detalla:

- **Conexión a la base de datos (SQL)**

  **Importación de librerías:**

**Figura N° 48: Conexión a la base de datos desde Jupyter**

```
#IMPORTAR LIBRERIAS
#LIBRERIAS
import pyodbc
import pandas as pd

#DATOS DEL SERVIDOR
server = 'LAPTOP-J8MRKNGK\SQLEXPRESS'
bd= 'DATA_RA'

#CONEXION AL SERVIDOR (SQL)
conexion = pyodbc.connect(driver='{SQL server}', host = server, database = bd)
usuario = 'sa'
contrasena = 'botis123'

consulta_data14 = pd.read_sql("""SELECT GENERO,EDAD,NIVEL_REND_ACAD,CONS_INTELIGENTE,CONS_RESPONSABLE,
HORAS_ESTUDIO,CONS_ORGANIZACION,NIVEL_ED_MADRE,NIVEL_ED_PADRE,
NIVEL_FAMILIA,CANT_ESTRES,REL_COMPANEROS,REAC_PROB_SOCIAL,
APOYO_ACAD,GASTOS_ACAD,ECONOMIA_HOGAR,PROMEDIO_CALIF_NOTAS,REND_ACAD
FROM ALUMNOS""", conexion)
#conexion.close()
```

**Fuente: Elaboración propia**

**import pyodbc ()**

Proporciona una interfaz para acceder a bases de datos utilizando el estándar ODBC.

**import pandas as pd**

Proporciona estructuras de datos y herramientas de análisis de datos de alto rendimiento y se asigna el alias pd.

**Datos y conexión al servidor**

Proporciona el nombre del servidor, base de datos, conexión y como credenciales el usuario y la contraseña.

**Consulta (SQL)**

Es la consulta definida con todas las variables de la base de datos SQL.

- **Algoritmos de selección de variables**

  - **Método de filtro basado en puntuación RFE**

    **(**Eliminación de funciones recursivas**)**

**Figura N° 49: Puntuación RFE**

```
from sklearn.linear_model import LinearRegression
from sklearn.feature_selection import RFE

# Se elimina la columna objetivo del DataFrame para obtener las características
X = consulta_data14.drop(columns=["REND_ACAD"])

# Crear el modelo de regresión lineal
model = LinearRegression()

# Crear el algoritmo RFE
rfe = RFE(model, n_features_to_select= 13)

# Seleccionar las variables
rfe.fit(X, consulta_data14["REND_ACAD"])

# Obtener las variables seleccionadas
selected_features = rfe.support_

# Etiquetas de las variables seleccionadas
selected_labels = X.columns[selected_features].tolist()

# Imprimir las etiquetas de las variables seleccionadas
print(selected_labels)
```

```
['EDAD', 'CONS_INTELIGENTE', 'HORAS_ESTUDIO', 'NIVEL_ED_MADRE', 'NIVEL_ED_PADRE',
'NIVEL_FAMILIA', 'CANT_ESTRES', 'REL_COMPANEROS', 'REAC_PROB_SOCIAL', 'APOYO_ACAD',
'GASTOS_ACAD', 'ECONOMIA_HOGAR', 'PROMEDIO_CALIF_NOTAS']
```

**Fuente: Elaboración propia**

**sklearn.linear_model, sklearn.feature_selection y LinearRegression**

Importa la clase LinearRegression del módulo sklearn.linear_model, la clase RFE del módulo sklearn.feature_selection.

**consulta_data14.drop(columns=["REND_ACAD"])**

De la consulta realizada, no considera la variable objetivo.

**model = LinearRegression()**

**rfe.fit(X, consulta_data14["REND_ACAD"])**

Crea una instancia del propio algoritmo y aplica al conjunto de datos de las características.

**selected_labels = X.columns[selected_features].tolist()**

Obtiene las etiquetas de las variables en una lista.

**print(selected_labels)**

Imprimes las etiquetas de las variables seleccionadas.

- **Método de filtro basado en puntuación Selectkbest**

  **(**Calcula un puntaje para cada característica en función de ciertos criterios**)**

**Figura N° 50: Puntuación Selectkbest**

```
from sklearn.feature_selection import SelectKBest, f_regression
import pandas as pd

# Definir las variables independientes
X = consulta_data14.drop("REND_ACAD", axis=1)

# Definir la variable objetivo
y = consulta_data14["REND_ACAD"]

# Seleccionar las 14 variables más importantes
selector = SelectKBest(score_func=f_regression,k=13)

# Seleccionar las variables
X_new = selector.fit_transform(X, y)

# Obtener las variables seleccionadas
selected_features = selector.get_support()

# Imprimir las variables seleccionadas
print(X.columns[selected_features])
```

```
Index(['GENERO', 'EDAD', 'NIVEL_REND_ACAD', 'CONS_INTELIGENTE',
       'CONS_RESPONSABLE', 'HORAS_ESTUDIO', 'CONS_ORGANIZACION', 'CANT_ESTRES',
       'REL_COMPANEROS', 'REAC_PROB_SOCIAL', 'APOYO_ACAD', 'ECONOMIA_HOGAR',
       'PROMEDIO_CALIF_NOTAS'],
      dtype='object')
```

**Fuente: Elaboración propia**

**sklearn.linear_model y SelectKBest, f_regression**

Importa la clase f_regression del módulo sklearn.linear_model y la clase SelectKBest del módulo sklearn.feature_selection.

**X = consulta_data14.drop("REND_ACAD", axis=1)**

**y = consulta_data14["REND_ACAD"]**

Define la variable objetivo y no la considera para la selección de variables.

**selector = SelectKBest(score_func=f_regression,k=13)**

**selected_features = selector.get_support()**

Se ajusta el selector a los datos “X” e “Y”, luego indica las características seleccionadas.

- **Método de filtro basado en puntuación Selectpercentile**

  **(**Selecciona las características más importantes de un conjunto de datos**)**

**Figura N° 51: Puntuación Selectpercentile**

```
from sklearn.feature_selection import SelectPercentile

# Separar las variables predictoras y la variable objetivo
X = consulta_data14.drop("REND_ACAD", axis=1)
y = consulta_data14["REND_ACAD"]

# Seleccionar las variables
selector = SelectPercentile(percentile=60)
selector.fit(X, y)

# Obtener las variables seleccionadas
selected_features = X.columns[selector.get_support()]

# Imprimir las variables seleccionadas
print(selected_features)
```

```
Index(['GENERO', 'EDAD', 'NIVEL_REND_ACAD', 'CONS_INTELIGENTE',
       'HORAS_ESTUDIO', 'CONS_ORGANIZACION', 'CANT_ESTRES', 'APOYO_ACAD',
       'ECONOMIA_HOGAR', 'PROMEDIO_CALIF_NOTAS'],
      dtype='object')
```

**Fuente: Elaboración propia**

**sklearn.feature_selection y SelectPercentile**

Importa la clase feature_selection del módulo sklearn.linear_model y la clase SelectPercentile del módulo sklearn.feature_selection.

**selected_features = X.columns[selector.get_support()]**

Se ajusta el selector a los datos “X” e “Y”, luego devuelve las características seleccionadas.

- **Método de transformación PCA combinado con el método de filtro Variancethreshold**

  **(**Reduce la dimensionalidad de los datos y mejorar la eficiencia y rendimiento de los modelos**)**

**Figura N° 52: Puntuación PCA y Variancethreshold**

```
from sklearn.decomposition import PCA
from sklearn.feature_selection import VarianceThreshold

# Realizar PCA
pca = PCA(n_components=10)
pca.fit(consulta_data14)

# Obtener las variables principales
principal_components = pca.components_

# Seleccionar las variables usando VarianceThreshold
selector = VarianceThreshold(threshold=0.01)
selector.fit(principal_components)

# Obtener las variables seleccionadas
selected_features = selector.get_support()

# Imprimir las variables seleccionadas
print(consulta_data14.columns[selected_features])
```

```
Index(['EDAD', 'NIVEL_REND_ACAD', 'CONS_INTELIGENTE', 'HORAS_ESTUDIO',
       'CONS_ORGANIZACION', 'NIVEL_ED_MADRE', 'NIVEL_ED_PADRE',
       'NIVEL_FAMILIA', 'CANT_ESTRES', 'REAC_PROB_SOCIAL', 'ECONOMIA_HOGAR',
       'PROMEDIO_CALIF_NOTAS', 'REND_ACAD'],
      dtype='object')
```

**Fuente: Elaboración propia**

**sklearn.decomposition PCA y from sklearn.feature_selection Variance Threshold**

Importa la clase PCA y VarianceThreshold del módulo sklearn.decomposition y sklearn.feature_selection.

**pca = PCA(n_components=10)**

**pca.fit(consulta_data14)**

Selecciona las 10 variables más importantes de consulta_data14.

**selector = VarianceThreshold(threshold=0.01)**

Se inicializa un objeto de VarianceThreshold con un umbral de varianza de 0.01.

**selected_features = X.columns[selector.get_support()]**

Recupera un array booleano que indica qué variables seleccionadas cumplen con el umbral de varianza.

## Tabla N° 79: Comparación de resultados de selección de Variables

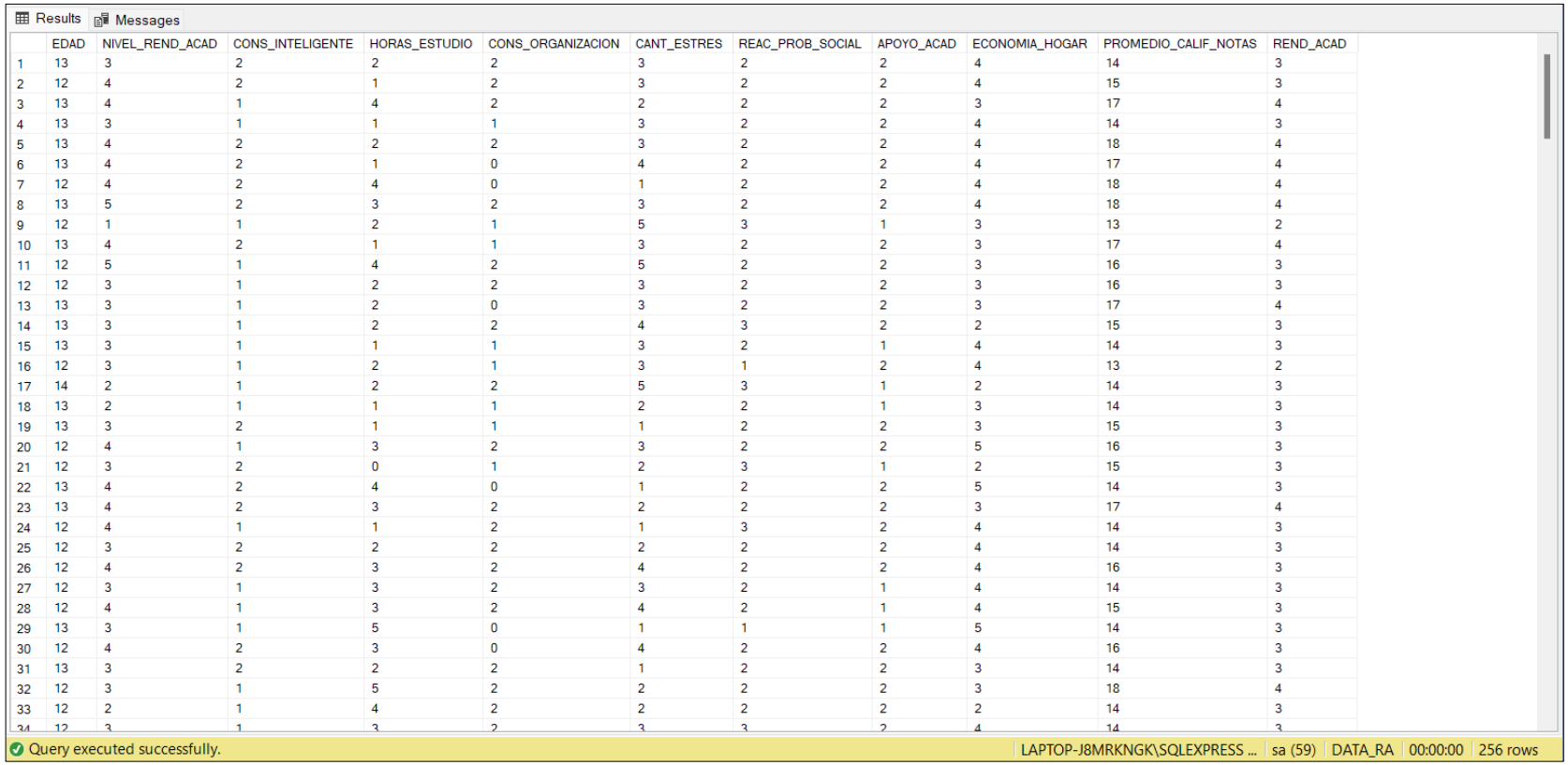
| ALGORITMOS | VARIABLES | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | GENERO | EDAD | NIVEL_REND_ACAD | CONS_INTELIGENTE | CONS_RESPONSABLE | HORAS_ESTUDIO | CONS_ORGANIZACION | NIVEL_ED_MADRE | NIVEL_ED_PADRE |
| RFE | | EDAD | | CONS_INTELIGENTE | | HORAS_ESTUDIO | | NIVEL_ED_MADRE | NIVEL_ED_PADRE |
| SELECTKBEST | GENERO | EDAD | NIVEL_REND_ACAD | CONS_INTELIGENTE | CONS_RESPONSABLE | HORAS_ESTUDIO | CONS_ORGANIZACION | | |
| SELECTPERCENTILE | GENERO | EDAD | NIVEL_REND_ACAD | CONS_INTELIGENTE | | HORAS_ESTUDIO | CONS_ORGANIZACION | | |
| PCA y VARIANCETHRESHOLD | | EDAD | NIVEL_REND_ACAD | CONS_INTELIGENTE | | HORAS_ESTUDIO | CONS_ORGANIZACION | NIVEL_ED_MADRE | NIVEL_ED_PADRE |

| ALGORITMOS | VARIABLES | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | NIVEL_FAMILIA | CANT_ESTRES | REL_COMPANEROS | REAC_PROB_SOCIAL | APOYO_ACAD | GASTOS_ACAD | ECONOMIA_HOGAR | PROMEDIO_CALIF_NOTAS |
| RFE | NIVEL_FAMILIA | CANT_ESTRES | REL_COMPANEROS | REAC_PROB_SOCIAL | APOYO_ACAD | GASTOS_ACAD | ECONOMIA_HOGAR | PROMEDIO_CALIF_NOTAS |
| SELECTKBEST | | CANT_ESTRES | REL_COMPANEROS | REAC_PROB_SOCIAL | APOYO_ACAD | | ECONOMIA_HOGAR | PROMEDIO_CALIF_NOTAS |
| SELECTPERCENTILE | | CANT_ESTRES | | | APOYO_ACAD | | ECONOMIA_HOGAR | PROMEDIO_CALIF_NOTAS |
| PCA y VARIANCETHRESHOLD | NIVEL_FAMILIA | CANT_ESTRES | | REAC_PROB_SOCIAL | | | ECONOMIA_HOGAR | PROMEDIO_CALIF_NOTAS |

**Fuente: Elaboración propia**

Luego de obtener los resultados de los algoritmos de selección de variables se realizó la siguiente tabla con el consolidado y se seleccionó las variables en común, siendo estas las siguientes:

**Figura N° 53: Modelo definido luego de la selección de variables**



Results Messages

| | EDAD | NIVEL_REND_ACAD | CONS_INTELIGENTE | HORAS_ESTUDIO | CONS_ORGANIZACION | CANT_ESTRES | REAC_PROB_SOCIAL | APOYO_ACAD | ECONOMIA_HOGAR | PROMEDIO_CALIF_NOTAS | REND_ACAD |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 13 | 3 | 2 | 2 | 2 | 3 | 2 | 2 | 4 | 14 | 3 |
| 2 | 12 | 4 | 2 | 1 | 2 | 3 | 2 | 2 | 4 | 15 | 3 |
| 3 | 13 | 4 | 1 | 4 | 2 | 2 | 2 | 2 | 3 | 17 | 4 |
| 4 | 13 | 3 | 1 | 1 | 1 | 3 | 2 | 2 | 4 | 14 | 3 |
| 5 | 13 | 4 | 2 | 2 | 2 | 3 | 2 | 2 | 4 | 18 | 4 |
| 6 | 13 | 4 | 2 | 1 | 0 | 4 | 2 | 2 | 4 | 17 | 4 |
| 7 | 12 | 4 | 2 | 4 | 0 | 1 | 2 | 2 | 4 | 18 | 4 |
| 8 | 13 | 5 | 2 | 3 | 2 | 3 | 2 | 2 | 4 | 18 | 4 |
| 9 | 12 | 1 | 1 | 2 | 1 | 5 | 3 | 1 | 3 | 13 | 2 |
| 10 | 13 | 4 | 2 | 1 | 1 | 3 | 2 | 2 | 3 | 17 | 4 |
| 11 | 12 | 5 | 1 | 4 | 2 | 5 | 2 | 2 | 3 | 16 | 3 |
| 12 | 12 | 3 | 1 | 2 | 2 | 3 | 2 | 2 | 3 | 16 | 3 |
| 13 | 13 | 3 | 1 | 2 | 0 | 3 | 2 | 2 | 3 | 17 | 4 |
| 14 | 13 | 3 | 1 | 2 | 2 | 4 | 3 | 2 | 2 | 15 | 3 |
| 15 | 13 | 3 | 1 | 1 | 1 | 3 | 2 | 1 | 4 | 14 | 3 |
| 16 | 12 | 3 | 1 | 2 | 1 | 3 | 1 | 2 | 4 | 13 | 2 |
| 17 | 14 | 2 | 1 | 2 | 2 | 5 | 3 | 1 | 2 | 14 | 3 |
| 18 | 13 | 2 | 1 | 1 | 1 | 2 | 2 | 1 | 3 | 14 | 3 |
| 19 | 13 | 3 | 2 | 1 | 1 | 1 | 2 | 2 | 3 | 15 | 3 |
| 20 | 12 | 4 | 1 | 3 | 2 | 3 | 2 | 2 | 5 | 16 | 3 |
| 21 | 12 | 3 | 2 | 0 | 1 | 2 | 3 | 1 | 2 | 15 | 3 |
| 22 | 13 | 4 | 2 | 4 | 0 | 1 | 2 | 2 | 5 | 14 | 3 |
| 23 | 13 | 4 | 2 | 3 | 2 | 2 | 2 | 2 | 3 | 17 | 4 |
| 24 | 12 | 4 | 1 | 1 | 2 | 1 | 3 | 2 | 4 | 14 | 3 |
| 25 | 12 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 4 | 14 | 3 |
| 26 | 12 | 4 | 2 | 3 | 2 | 4 | 2 | 2 | 4 | 16 | 3 |
| 27 | 12 | 3 | 1 | 3 | 2 | 3 | 2 | 1 | 4 | 14 | 3 |
| 28 | 12 | 4 | 1 | 3 | 2 | 4 | 2 | 1 | 4 | 15 | 3 |
| 29 | 13 | 3 | 1 | 5 | 0 | 1 | 1 | 1 | 5 | 14 | 3 |
| 30 | 12 | 4 | 2 | 3 | 0 | 4 | 2 | 2 | 4 | 16 | 3 |
| 31 | 13 | 3 | 2 | 2 | 2 | 1 | 2 | 2 | 3 | 14 | 3 |
| 32 | 12 | 3 | 1 | 5 | 2 | 2 | 2 | 2 | 3 | 18 | 4 |
| 33 | 12 | 2 | 1 | 4 | 2 | 2 | 2 | 2 | 2 | 14 | 3 |

Query executed successfully. LAPTOP-J8MRKNGK\SQLEXPRESS ... sa (59) DATA_RA 00:00:00 256 rows

**Fuente: Elaboración propia**

Una vez definida las 10 variables en común más la variable objetivo, se estableció el modelo de datos para la aplicación de los algoritmos de ML.

**Cuarta etapa: Minería de datos**

Para identificar las relaciones y el reconocimiento de patrones de datos del modelo rendimiento académico con las variables seleccionadas se aplicó en 10 algoritmos de aprendizaje automático para realizar el entrenamiento y la validación en el entorno interactivo Jupyter.

Para los 10 algoritmos se realizó la importación en general de las siguientes librerías y algoritmos específicos.

- **Conexión a la base de datos (SQL) e importación de las librerías (**Referencia Figura N° 32**)**

  **Importación de librerías:**

**Figura N° 54: Librerías Generales**

```
#Librerias - Gráficos
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix
import plotly.express as ax
sns.set(style="whitegrid")

#Selección de mejores Hiperparametros
from sklearn.model_selection import GridSearchCV
```

**Fuente: Elaboración propia**

**import matplotlib.pyplot as plt**

Se utiliza para crear gráficos y visualizaciones se asigna el alias plt.

**import seaborn as sns**

Proporciona una interfaz de alto nivel para crear gráficos más atractivos se asigna el alias sns.

**from sklearn.metrics import confusion_matrix**

Se utiliza para evaluar el rendimiento de un modelo de clasificación al comparar las etiquetas verdaderas con las predicciones del modelo.

**import plotly.express as ax**

Permite crear gráficos interactivos y gráficos en línea se asigna el alias ax.

**sns.set(style="whitegrid")**

Establece el estilo predeterminado de seaborn como "whitegrid".

**from sklearn.model_selection import GridSearchCV**

Se utiliza para la búsqueda exhaustiva para encontrar los mejores hiperparámetros para un modelo de ML específico.

**Figura N° 55: Librerías de algoritmos**

```
#Libreria de Algoritmos
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.neural_network import MLPClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.neural_network import MLPRegressor
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import AdaBoostClassifier
```

**Fuente: Elaboración propia**

**from sklearn.tree import DecisionTreeClassifier**

Se utiliza para construir un modelo de árbol de decisión para la clasificación de datos.

**from sklearn.ensemble import RandomForestClassifier**

Se utilizan para tareas de clasificación y regresión, y se basan en la construcción de múltiples árboles de decisión durante el proceso de entrenamiento.

**from sklearn.svm import SVC**

Se utiliza para tareas de clasificación y regresión.

**from sklearn.neural_network import MLPClassifier**

Es un tipo de red neuronal artificial que se utiliza para tareas de clasificación.

**from sklearn.ensemble import GradientBoostingClassifier**

Utiliza el concepto de impulso para construir un modelo de ensamblaje a partir de modelos más débiles.

**from sklearn.preprocessing import StandardScaler**

Se utiliza para estandarizar características eliminando la media y escalando a la varianza unitaria.

**from sklearn.neural_network import MLPRegressor**

Se utiliza para tareas de regresión utilizando redes neuronales artificiales.

**from sklearn.neighbors import KNeighborsClassifier**

Se utiliza para tareas de clasificación basadas en la cercanía de los puntos de datos en el espacio de características.

**from sklearn.naive_bayes import MultinomialNB**

Se utiliza para tareas de clasificación basadas en el teorema de Bayes con la suposición de independencia condicional entre las características.

**from sklearn.linear_model import LogisticRegression**

Se utiliza para problemas de clasificación binaria y multiclase modelando la probabilidad de pertenecer a una clase específica.

**from sklearn.ensemble import AdaBoostClassifier**

Construye un modelo fuerte combinando varios modelos más débiles.

**Figura N° 56: Librerías de métricas**

```
#Librerias accuracy/precision/recall/f1 score
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_recall_fscore_support
from sklearn.metrics import recall_score
from sklearn.metrics import classification_report
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import train_test_split
```

**Fuente: Elaboración propia**

**from sklearn.metrics import accuracy_score**

Se utiliza para calcular la precisión del modelo.

**from sklearn.metrics import precision_recall_fscore_support**

Se utiliza para calcular la precisión, la recuperación y la puntuación F1 del modelo.

**from sklearn.metrics import recall_score**

Se utiliza para calcular la tasa de verdaderos positivos dividida por la suma de verdaderos positivos y falsos negativos.

**from sklearn.metrics import classification_report**

Se utiliza para mostrar un informe detallado que incluye la precisión, la recuperación, la puntuación F1 y el soporte para cada clase en un problema de clasificación.

**from sklearn.model_selection import cross_val_score**

Se utiliza para realizar validación cruzada en un modelo.

**from sklearn.model_selection import train_test_split**

Se utiliza para dividir un conjunto de datos en conjuntos de entrenamiento y pruebas.

**Figura N° 57: Librerías de Kappa de Cohen**

```
#Libreria Kappa de Cohen
import scipy
from sklearn.metrics import cohen_kappa_score
```

**Fuente: Elaboración propia**

**scipy**

Proporciona diversas rutinas numéricas como la integración, optimización, álgebra lineal, entre otras.

**from sklearn.metrics cohen_kappa_score**

Se utiliza comúnmente para evaluar la confiabilidad inter-anotador de un sistema de clasificación.

- **Particionamiento de la data**

**Figura N° 58: Particionamiento de la data**

```
# Seleccionar las variables
selected_features = ['EDAD', 'NIVEL_REND_ACAD', 'CONS_INTELIGENTE', 'HORAS_ESTUDIO',
                     'CONS_ORGANIZACION', 'CANT_ESTRES', 'REAC_PROB_SOCIAL',
                     'APOYO_ACAD','ECONOMIA_HOGAR', 'PROMEDIO_CALIF_NOTAS']

# Definir las variables
X = consulta_data14[selected_features]
y = consulta_data14["REND_ACAD"]

# Dividir los datos en un conjunto de entrenamiento y un conjunto de validación
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.20)
```

**Fuente: Elaboración propia**

**Selección de las variables**

Contiene las 10 variables establecidas para el modelo.

**Definición de las variables**

Se asigna a la variable “X” y representa la matriz de características.

Se asigna a la variable “Y” y representa la variable objetivo.

**División del conjunto de entrenamiento y validación**

Se dividen los datos en un 80% y 20% para el conjunto de entrenamiento y validación, respectivamente.

- **Decisión Tree**

**Figura N° 59: Hiperparámetros - DT**

```
# Definir los hiperparámetros a optimizar
param_grid = {
    'max_depth': [3, 5, 7, 10],
    'min_samples_split': [2, 5, 10]
}

# Instanciar el algoritmo de optimización de hiperparámetros con Grid Search
grid_search = GridSearchCV(estimator=DecisionTreeClassifier(), param_grid=param_grid, cv=10, scoring='accuracy')

# Ajustar el algoritmo a los datos de entrenamiento
grid_search.fit(X_train, y_train)

# Imprimir los mejores hiperparámetros encontrados por Grid Search
print("Mejores hiperparámetros encontrados:", grid_search.best_params_)
```

```
Mejores hiperparámetros encontrados: {'max_depth': 3, 'min_samples_split': 2}
```

**Fuente: Elaboración propia**

**Hiperparámetros a optimizar**

max_depth: Controla la profundidad máxima del DT.

min_samples_split: Controla el número mínimo de muestras necesarias para dividir un nodo en el DT.

**Algoritmo de optimización de Hiperparámetros**

Realiza la búsqueda de combinaciones de hiperparámetros, utilizando la clase GridSearchCV de la biblioteca scikit-learn.

**Evaluación de las métricas**

**Figura N° 60: Evaluación de las métricas – DT**

```
# Crear un modelo Decision Tree con los hiperparámetros dados
dt = DecisionTreeClassifier(max_depth=3, min_samples_split=2)

# Entrenar el modelo Decision Tree
dt.fit(X_train, y_train)

# Predecir los valores de y para el conjunto de datos de entrenamiento y validación
y_pred_train = dt.predict(X_train)
y_pred_val = dt.predict(X_val)

# Calcular las métricas de rendimiento para el conjunto de entrenamiento
accuracy_train = accuracy_score(y_train, y_pred_train)
precision_train, recall_train, f1_train, support_train = precision_recall_fscore_support(y_train, y_pred_train, average='macro')

# Calcular las métricas de rendimiento para el conjunto de validación
accuracy_val = accuracy_score(y_val, y_pred_val)
precision_val, recall_val, f1_val, support_val = precision_recall_fscore_support(y_val, y_pred_val, average='macro')

# Imprimir los resultados
print("RESULTADOS DEL CONJUNTO DE ENTRENAMIENTO:")
print("Accuracy:", accuracy_train)
print("Precision:", precision_train)
print("Recall:", recall_train)
print("F1-score:", f1_train)

print("-----Reporte de Entrenamiento:-----")
print(classification_report(y_train, y_pred_train))
print("Matriz de confusión:")
print(confusion_matrix(y_train, y_pred_train))

print("////////////////////////////////////////////////////")

print("RESULTADOS DEL CONJUNTO DE VALIDACIÓN:")
print("Accuracy:", accuracy_val)
print("Precision:", precision_val)
print("Recall:", recall_val)
print("F1-score:", f1_val)

print("-----Reporte de Validación:-----")
print(classification_report(y_val, y_pred_val))
print("Matriz de confusión:")
print(confusion_matrix(y_val, y_pred_val))
```

**Fuente: Elaboración propia**

- **Random Forest**

**Figura N° 61: Hiperparámetros – RF**

```
# Definir los hiperparámetros a optimizar
param_grid = {
    'n_estimators': [10, 50, 100, 200],
    'max_depth': [3, 5, 7, 10],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 5]
}

# Instanciar el algoritmo de optimización de hiperparámetros con Grid Search
grid_search = GridSearchCV(estimator=RandomForestClassifier(), param_grid=param_grid, scoring='accuracy', cv=10)

# Ajustar el algoritmo a los datos de entrenamiento
grid_search.fit(X_train, y_train)

# Imprimir los mejores hiperparámetros encontrados por Grid Search
print("Mejores hiperparámetros encontrados:", grid_search.best_params_)
```

```
Mejores hiperparámetros encontrados: {'max_depth': 5, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}
```

**Fuente: Elaboración propia**

**Hiperparámetros a optimizar**

n_estimators: Son el número de DT que se utilizan en el RF.

max_depth: Son la profundidad máxima de cada DT en el RF.

min_samples_split: Son el número mínimo de muestras necesarias para dividir un nodo en un DT.

min_samples_leaf: Son un número mínimo de muestras necesarias para que un nodo sea considerado una hoja en un DT.

**Figura N° 62: Evaluación de las métricas – RF**

```
best_params = grid_search.best_params_
best_n_estimator = best_params['n_estimators']
best_max_depth = best_params['max_depth']
best_min_samples_split = best_params['min_samples_split']
best_min_samples_leaf = best_params['min_samples_leaf']

# Crea un modelo random forest con la mejor combinación de hiperparámetros
rf = RandomForestClassifier(n_estimators=best_n_estimator, max_depth=best_max_depth,
                            min_samples_split=best_min_samples_split, min_samples_leaf=best_min_samples_leaf)
# Entrenamos el modelo random forest
rf.fit(X_train, y_train)

# Predecir los valores de y para el conjunto de datos de entrenamiento y validación
y_pred_train = rf.predict(X_train)
y_pred_val = rf.predict(X_val)

# Calcular las métricas de rendimiento para el conjunto de entrenamiento
accuracy_train = accuracy_score(y_train, y_pred_train)
precision_train, recall_train, f1_train, support_train = precision_recall_fscore_support(y_train, y_pred_train, average='macro')

# Calcular las métricas de rendimiento para el conjunto de validación
accuracy_val = accuracy_score(y_val, y_pred_val)
precision_val, recall_val, f1_val, support_val = precision_recall_fscore_support(y_val, y_pred_val, average='macro')
```

**Fuente: Elaboración propia**

- **Support Vector Machines**

**Figura N° 63: Hiperparámetros – SVM**

```
# Definir los hiperparámetros a optimizar
param_grid = {'C': [0.1, 1, 10, 100],
              'kernel': ['linear', 'rbf']}

# Instanciar el algoritmo de optimización de hiperparámetros
grid_search = GridSearchCV(SVC(), param_grid, cv=10, scoring='accuracy')

# Entrenar el modelo con la búsqueda de cuadrícula
grid_search.fit(X_train, y_train)

# Imprimir los mejores hiperparámetros encontrados
print("Mejores hiperparámetros encontrados:")
print(grid_search.best_params_)

# Obtener el mejor modelo
best_svm = grid_search.best_estimator_
```

```
Mejores hiperparámetros encontrados:
{'C': 1, 'kernel': 'linear'}
```

**Fuente: Elaboración propia**

**Hiperparámetros a optimizar**

c: Controla la regularización del modelo SVM

kernel: Controla la función de kernel utilizada por el modelo SVM. Las funciones de kernel más comunes son la función lineal y la función RBF. La función lineal es más simple y rápida de calcular, pero la función RBF puede ser más precisa en algunos casos.

**Figura N° 64: Evaluación de las métricas – SVM**

```
# Entrenar el modelo SVM con los mejores hiperparámetros encontrados en el conjunto de entrenamiento
best_svm.fit(X_train, y_train)

# Predecir los valores de y para el conjunto de datos de entrenamiento y validación
y_pred_train = best_svm.predict(X_train)
y_pred_val = best_svm.predict(X_val)

# Calcular las métricas de rendimiento para el conjunto de entrenamiento
accuracy_train = accuracy_score(y_train, y_pred_train)
precision_train, recall_train, f1_train, support_train = precision_recall_fscore_support(y_train, y_pred_train, average='macro')

# Calcular las métricas de rendimiento para el conjunto de validación
accuracy_val = accuracy_score(y_val, y_pred_val)
precision_val, recall_val, f1_val, support_val = precision_recall_fscore_support(y_val, y_pred_val, average='macro')
```

**Fuente: Elaboración propia**

- **Artificial Neural Network**

**Figura N° 65: Hiperparámetros – ANN**

```
# Definimos los parámetros a optimizar
parameters = {
    'hidden_layer_sizes': [(10), (20), (30)],
    'learning_rate_init': [0.001, 0.01, 0.1]
}

# Instanciamos el modelo
mlp = MLPClassifier()

# Instanciamos el algoritmo de optimización de hiperparámetros GridSearchCV
clf = GridSearchCV(mlp, parameters, cv=10)

# Ajustamos el modelo a los datos de entrenamiento
clf.fit(X_train, y_train)

# Imprimimos los mejores hiperparámetros encontrados
print("Mejores hiperparámetros encontrados:")
print(clf.best_params_)
```

```
Mejores hiperparámetros encontrados:
{'hidden_layer_sizes': 30, 'learning_rate_init': 0.01}
```

**Fuente: Elaboración propia**

**Hiperparámetros a optimizar**

hidden_layer_sizes: Controla el número de neuronas en cada capa oculta de la red neuronal.

learning_rate_int: Controla el tamaño del paso utilizado durante la optimización del descenso del gradiente.

**Figura N° 66: Evaluación de las métricas – ANN**

```
# Utilizamos los mejores hiperparámetros para entrenar el modelo final
best_hidden_layer_sizes = clf.best_params_['hidden_layer_sizes']
best_learning_rate_init = clf.best_params_['learning_rate_init']

# Crea un modelo MLPClassifier con la mejor combinación de hiperparámetros
model = MLPClassifier(hidden_layer_sizes=best_hidden_layer_sizes, learning_rate_init=best_learning_rate_init)

# Entrenamos el modelo MLPClassifier
model.fit(X_train, y_train)

# Predecir los valores de y para el conjunto de datos de entrenamiento y validación
y_pred_train = model.predict(X_train)
y_pred_val = model.predict(X_val)

# Calcular las métricas de rendimiento para el conjunto de entrenamiento
accuracy_train = accuracy_score(y_train, y_pred_train)
precision_train, recall_train, f1_train, support_train = precision_recall_fscore_support(y_train, y_pred_train, average='macro')

# Calcular las métricas de rendimiento para el conjunto de validación
accuracy_val = accuracy_score(y_val, y_pred_val)
precision_val, recall_val, f1_val, support_val = precision_recall_fscore_support(y_val, y_pred_val, average='macro')
```

**Fuente: Elaboración propia**

- **Gradient Boosting Machines**

**Figura N° 67: Hiperparámetros – GBM**

```
# Definir los hiperparámetros a optimizar
param_grid = {
    'n_estimators': [100, 200, 300],
    'learning_rate': [0.1, 0.01, 0.001],
    'max_depth': [3, 5, 10]
}

# Instanciar el clasificador Gradient Boosting Machines
gbr = GradientBoostingClassifier()

# Instanciar GridSearchCV
grid_search = GridSearchCV(estimator=gbr, param_grid=param_grid, cv=10, scoring='accuracy')

# Ajustar GridSearchCV a los datos
grid_search.fit(X_train, y_train)

# Imprimir los mejores hiperparámetros encontrados
print("Mejores hiperparámetros:", grid_search.best_params_)
```

```
Mejores hiperparámetros: {'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 100}
```

**Fuente: Elaboración propia**

**Hiperparámetros a optimizar**

n_estimators: Controla el número de DT que se utilizan en el GBM.

learning_rate: Controla la cantidad de aprendizaje que se realiza en cada iteración del GBM.

max_depth: Controla la profundidad máxima de cada DT en el GBM.

**Figura N° 68: Evaluación de las métricas – GBM**

```
best_params = grid_search.best_params_
best_n_estimators = best_params['n_estimators']
best_learning_rate = best_params['learning_rate']
best_max_depth = best_params['max_depth']

# Crear un modelo GBM con la mejor combinación de hiperparámetros
gbr = GradientBoostingClassifier(n_estimators=best_n_estimators, learning_rate=best_learning_rate, max_depth=best_max_depth)

# Entrenar el modelo GBM
gbr.fit(X_train, y_train)

# Predecir los valores de y para el conjunto de datos de entrenamiento y validación
y_pred_train = gbr.predict(X_train)
y_pred_val = gbr.predict(X_val)

# Calcular las métricas de rendimiento para el conjunto de entrenamiento
accuracy_train = accuracy_score(y_train, y_pred_train)
precision_train, recall_train, f1_train, support_train = precision_recall_fscore_support(y_train, y_pred_train, average='macro')

# Calcular las métricas de rendimiento para el conjunto de validación
accuracy_val = accuracy_score(y_val, y_pred_val)
precision_val, recall_val, f1_val, support_val = precision_recall_fscore_support(y_val, y_pred_val, average='macro')
```

**Fuente: Elaboración propia**

- **Multi Perceptrón**

**Figura N° 69: Hiperparámetros - MP**

```
# Escalar los datos para mejorar el rendimiento del modelo
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_val = scaler.transform(X_val)

# Definir los parámetros a optimizar
parameters = {
    'hidden_layer_sizes': [(50,50,50), (50,100,50), (100)],
    'activation': ['tanh', 'relu'],
    'solver': ['sgd', 'adam'],
    'alpha': [0.0001, 0.05],
    'learning_rate': ['constant','adaptive'],
}

# Instanciar el algoritmo de optimización de hiperparámetros
grid_search = GridSearchCV(MLPRegressor(max_iter=500), parameters, n_jobs=-1)

# Entrenar el modelo con los datos de entrenamiento
grid_search.fit(X_train, y_train)

# Imprimir los mejores hiperparámetros
print("Mejores hiperparámetros:", grid_search.best_params_)
```

```
Mejores hiperparámetros: {'activation': 'tanh', 'alpha': 0.0001, 'hidden_layer_sizes': 100, 'learning_rate': 'constant', 'solver': 'adam'}
```

**Fuente: Elaboración propia**

**Hiperparámetros a optimizar**

hidden_layer_sizes: Controla la arquitectura de la red neuronal.

activation: Determina la función de activación que se aplica a las salidas de cada neurona.

solver: Define el algoritmo de optimización.

alpha: Controla la regularización L2, para evitar el sobreajuste.

learning_rate: Determina la tasa de aprendizaje.

**Figura N° 70: Evaluación de las métricas – MP**

```
best_params = grid_search.best_params_
best_hidden_layer_sizes = best_params['hidden_layer_sizes']
best_activation = best_params['activation']
best_solver = best_params['solver']
best_alpha = best_params['alpha']
best_learning_rate = best_params['learning_rate']

# Crear un modelo con la mejor combinación de hiperparámetros
model = MLPClassifier(hidden_layer_sizes=best_hidden_layer_sizes, activation=best_activation,
                      solver=best_solver, alpha=best_alpha, learning_rate=best_learning_rate)
# Entrenar el modelo con los datos de entrenamiento
model.fit(X_train, y_train)

# Predecir los valores de y para el conjunto de datos de entrenamiento y validación
y_pred_train = model.predict(X_train)
y_pred_val = model.predict(X_val)

# Calcular las métricas de rendimiento para el conjunto de entrenamiento
accuracy_train = accuracy_score(y_train, y_pred_train)
precision_train, recall_train, f1_train, support_train = precision_recall_fscore_support(y_train, y_pred_train, average='macro')

# Calcular las métricas de rendimiento para el conjunto de validación
accuracy_val = accuracy_score(y_val, y_pred_val)
precision_val, recall_val, f1_val, support_val = precision_recall_fscore_support(y_val, y_pred_val, average='macro')
```

**Fuente: Elaboración propia**

- **K-Nearest Neighboor**

**Figura N° 71: Hiperparámetros – KNN**

```
# Definir la cuadrícula de parámetros a buscar
param_grid = {
    'n_neighbors': [1, 3, 5, 7, 9],
    'metric': ['euclidean', 'manhattan', 'minkowski']
}

# Instanciar el modelo de KNN
knn = KNeighborsClassifier()

# Instanciar el algoritmo de optimización de hiperparámetros
grid_search = GridSearchCV(knn, param_grid, cv=10, scoring='accuracy')

# Ajustar el modelo con los datos de entrenamiento
grid_search.fit(X_train, y_train)

# Imprimir los mejores parámetros encontrados
print("Mejores parámetros:", grid_search.best_params_)
```

```
Mejores parámetros: {'metric': 'manhattan', 'n_neighbors': 3}
```

**Fuente: Elaboración propia**

**Hiperparámetros a optimizar**

n_neighbors: Controla el número de vecinos que se consideran al predecir una nueva muestra.

metric: Determina la distancia que se utiliza para medir la similitud entre dos muestras.

**Figura N° 72: Evaluación de las métricas – KNN**

```
best_params = grid_search.best_params_
best_metric = best_params['metric']
best_n_neighbors = best_params['n_neighbors']

# Utilizar los mejores parámetros para crear un nuevo modelo KNN
knn_best_model = KNeighborsClassifier(n_neighbors=best_n_neighbors, metric=best_metric)

# Entrenar el modelo KNeighbors con los datos de entrenamiento
knn_best_model.fit(X_train, y_train)

# Predecir los valores de y para el conjunto de datos de entrenamiento y validación
y_pred_train = knn_best_model.predict(X_train)
y_pred_val = knn_best_model.predict(X_val)

# Calcular las métricas de rendimiento para el conjunto de entrenamiento
accuracy_train = accuracy_score(y_train, y_pred_train)
precision_train, recall_train, f1_train, support_train = precision_recall_fscore_support(y_train, y_pred_train, average='macro')

# Calcular las métricas de rendimiento para el conjunto de validación
accuracy_val = accuracy_score(y_val, y_pred_val)
precision_val, recall_val, f1_val, support_val = precision_recall_fscore_support(y_val, y_pred_val, average='macro')
```

**Fuente: Elaboración propia**

- **Naïve Bayes**

**Figura N° 73: Hiperparámetros – NB**

```
# Definir los valores de alpha a probar
param_grid = {'alpha': [0.1, 10, 1, 100]}

# Crear un objeto MultinomialNB
clf = MultinomialNB()

# Instanciar GridSearchCV
grid_search = GridSearchCV(estimator=clf, param_grid=param_grid, cv=10, scoring='accuracy')

# Ajustar GridSearchCV con los datos de entrenamiento
grid_search.fit(X_train, y_train)

# Imprimir los mejores parámetros y puntuaciones
print("Mejor valor de hiperparámetro alpha:", grid_search.best_params_)
```

```
Mejor valor de hiperparámetro alpha: {'alpha': 0.1}
```

**Fuente: Elaboración propia**

**Hiperparámetros a optimizar**

alpha: Controla la regularización del modelo NB.

**Figura N° 74: Evaluación de las métricas – NB**

```
best_alpha = grid_search.best_params_['alpha']

# Crear un nuevo modelo con los mejores parámetros
clf_best = MultinomialNB(alpha=best_alpha)

# Entrenar el modelo con los datos de entrenamiento
clf_best.fit(X_train, y_train)

# Predecir los valores de y para el conjunto de datos de entrenamiento y validación
y_pred_train = clf_best.predict(X_train)
y_pred_val = clf_best.predict(X_val)

# Calcular las métricas de rendimiento para el conjunto de entrenamiento
accuracy_train = accuracy_score(y_train, y_pred_train)
precision_train, recall_train, f1_train, support_train = precision_recall_fscore_support(y_train, y_pred_train, average='macro')

# Calcular las métricas de rendimiento para el conjunto de validación
accuracy_val = accuracy_score(y_val, y_pred_val)
precision_val, recall_val, f1_val, support_val = precision_recall_fscore_support(y_val, y_pred_val, average='macro')
```

**Fuente: Elaboración propia**

- **Logistic Regression**

**Figura N° 75: Hiperparámetros – LR**

```
# Hiperparámetros a optimizar
parameters = {
    'C': [0.1, 1, 10, 100],
    'penalty': ['l1', 'l2'],
    'solver': ['liblinear']
}

# Instancia el algoritmo de optimización de hiperparámetros
grid_search = GridSearchCV(estimator=LogisticRegression(max_iter=1000), param_grid=parameters, cv=10, scoring='accuracy')

# Ajusta el modelo
grid_search.fit(X_train, y_train)

# Imprime los mejores parámetros encontrados
print("Mejores parámetros:", grid_search.best_params_)
```

```
Mejores parámetros: {'C': 100, 'penalty': 'l1', 'solver': 'liblinear'}
```

**Fuente: Elaboración propia**

**Hiperparámetros a optimizar**

c: Controla la fuerza de la regularización.

penalty: Determina el tipo de regularización que se utiliza.

solver: Determina el algoritmo de optimización utilizado para entrenar el modelo LR.

**Figura N° 76: Evaluación de las métricas – LR**

```
best_params = grid_search.best_params_

# Utilizar los mejores parámetros para crear un nuevo modelo
best_lr = LogisticRegression(**best_params, max_iter=1000)

# Entrenar el modelo
best_lr.fit(X_train, y_train)

# Predecir los valores de y para el conjunto de datos de entrenamiento y validación
y_pred_train = best_lr.predict(X_train)
y_pred_val = best_lr.predict(X_val)

# Calcular las métricas de rendimiento para el conjunto de entrenamiento
accuracy_train = accuracy_score(y_train, y_pred_train)
precision_train, recall_train, f1_train, support_train = precision_recall_fscore_support(y_train, y_pred_train, average='macro')

# Calcular las métricas de rendimiento para el conjunto de validación
accuracy_val = accuracy_score(y_val, y_pred_val)
precision_val, recall_val, f1_val, support_val = precision_recall_fscore_support(y_val, y_pred_val, average='macro')
```

**Fuente: Elaboración propia**

- **AdaBoost**

**Figura N° 77: Hiperparámetros – ADA**

```
# Definir el espacio de búsqueda de hiperparámetros
param_grid = {
    "n_estimators": [10, 50, 100, 200],
    "learning_rate": [0.1, 0.01, 0.001],
}

# Instanciar el algoritmo de optimización de hiperparámetros
grid_search = GridSearchCV(AdaBoostClassifier(), param_grid, scoring="accuracy", cv=10)
grid_search.fit(X_train, y_train)

# Obtener los mejores hiperparámetros
best_params = grid_search.best_params_
print("Los mejores hiperparámetros son:", best_params)
```

```
Los mejores hiperparámetros son: {'learning_rate': 0.1, 'n_estimators': 100}
```

**Fuente: Elaboración propia**

**Hiperparámetros a optimizar**

n_estimators: Controla el número de DT utilizados en el conjunto de ADA.

learning_rate: Controla la contribución de cada aprendiz débil a la predicción final.

**Figura N° 78: Evaluación de las métricas – ADA**

```
best_params = grid_search.best_params_

# Utilizar los mejores parámetros para crear un nuevo modelo
best_ada = AdaBoostClassifier(**best_params)

# Entrenar el modelo
best_ada.fit(X_train, y_train)

# Predecir los valores de y para el conjunto de datos de entrenamiento y validación
y_pred_train = best_ada.predict(X_train)
y_pred_val = best_ada.predict(X_val)

# Calcular las métricas de rendimiento para el conjunto de entrenamiento
accuracy_train = accuracy_score(y_train, y_pred_train)
precision_train, recall_train, f1_train, support_train = precision_recall_fscore_support(y_train, y_pred_train, average='macro')

# Calcular las métricas de rendimiento para el conjunto de validación
accuracy_val = accuracy_score(y_val, y_pred_val)
precision_val, recall_val, f1_val, support_val = precision_recall_fscore_support(y_val, y_pred_val, average='macro')
```

**Fuente: Elaboración propia**

**Quinta etapa: Interpretación**

- **Decision Tree**

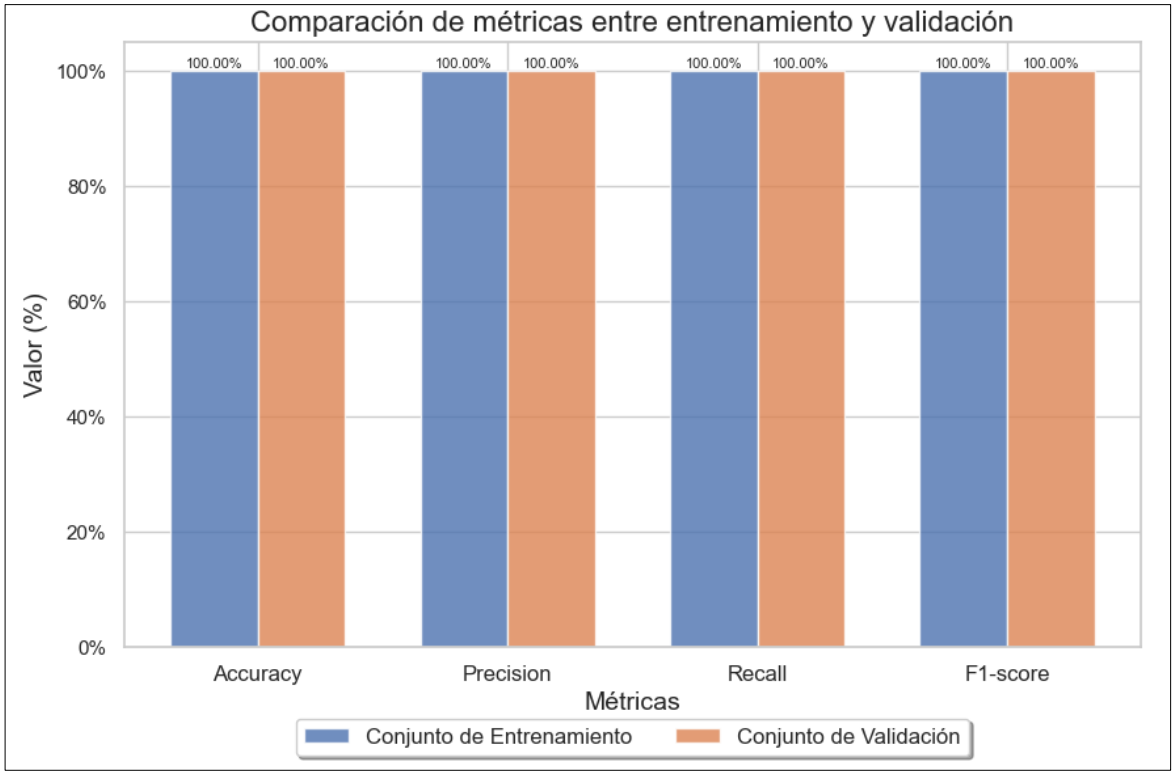
**Figura N° 79: Reporte de entrenamiento y validación – DT**

```
RESULTADOS DEL CONJUNTO DE ENTRENAMIENTO:
Accuracy: 1.0
Precision: 1.0
Recall: 1.0
F1-score: 1.0
-----Reporte de Entrenamiento:-----
              precision    recall  f1-score   support

         2.0       1.00      1.00      1.00        37
         3.0       1.00      1.00      1.00       125
         4.0       1.00      1.00      1.00        39
         5.0       1.00      1.00      1.00         3

    accuracy                           1.00       204
   macro avg       1.00      1.00      1.00       204
weighted avg       1.00      1.00      1.00       204

Matriz de confusión:
[[ 37   0   0   0]
 [  0 125   0   0]
 [  0   0  39   0]
 [  0   0   0   3]]
//////////////////////////////////////////////////
RESULTADOS DEL CONJUNTO DE VALIDACIÓN:
Accuracy: 1.0
Precision: 1.0
Recall: 1.0
F1-score: 1.0
-----Reporte de Validación:-----
              precision    recall  f1-score   support

         2.0       1.00      1.00      1.00        11
         3.0       1.00      1.00      1.00        30
         4.0       1.00      1.00      1.00        10
         5.0       1.00      1.00      1.00         1

    accuracy                           1.00        52
   macro avg       1.00      1.00      1.00        52
weighted avg       1.00      1.00      1.00        52

Matriz de confusión:
[[11  0  0  0]
 [ 0 30  0  0]
 [ 0  0 10  0]
 [ 0  0  0  1]]
```

**Fuente: Elaboración propia**

En el reporte muestra la partición del conjunto de datos del 80% para el entrenamiento y 20% para la validación, de los resultados obtenidos en el conjunto de validación en las métricas de exactitud, precisión, sensibilidad y F1-score indica un rendimiento del 100% para cada clase, demostrando que el modelo es muy preciso. La "Matriz de confusión" respalda esta afirmación al revelar que no se han cometido errores de clasificación en ninguna de las clases en ambos conjuntos de datos.

**Figura N° 80: Gráfica de comparación – DT**




**Fuente: Elaboración propia**

El análisis comparativo de “Entrenamiento y Validación” muestran que, en las métricas de exactitud, precisión, sensibilidad y F1-score se obtuvo un 100%, lo cual denota un rendimiento muy preciso en el desempeño del modelo en ambos conjuntos de datos.

- **Random Forest**

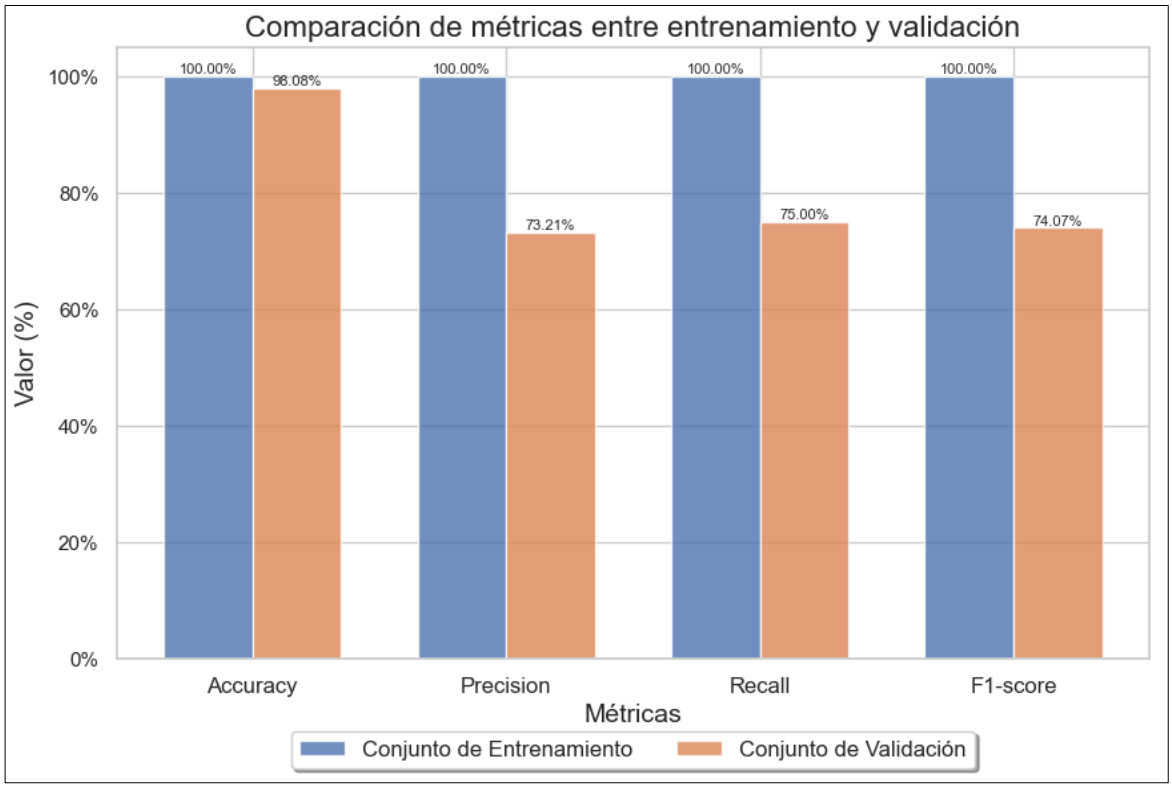
**Figura N° 81: Reporte de entrenamiento y validación – RF**

```
RESULTADOS DEL CONJUNTO DE ENTRENAMIENTO:
Accuracy: 1.0
Precision: 1.0
Recall: 1.0
F1-score: 1.0
-----Reporte de Entrenamiento:-----
              precision    recall  f1-score   support

         2.0       1.00      1.00      1.00        36
         3.0       1.00      1.00      1.00       129
         4.0       1.00      1.00      1.00        36
         5.0       1.00      1.00      1.00         3

    accuracy                           1.00       204
   macro avg       1.00      1.00      1.00       204
weighted avg       1.00      1.00      1.00       204

Matriz de confusión:
[[ 36   0   0   0]
 [  0 129   0   0]
 [  0   0  36   0]
 [  0   0   0   3]]
////////////////////////////////////////////////////
RESULTADOS DEL CONJUNTO DE VALIDACIÓN:
Accuracy: 0.9807692307692307
Precision: 0.7321428571428572
Recall: 0.75
F1-score: 0.7407407407407407
-----Reporte de Validación:-----
              precision    recall  f1-score   support

         2.0       1.00      1.00      1.00        12
         3.0       1.00      1.00      1.00        26
         4.0       0.93      1.00      0.96        13
         5.0       0.00      0.00      0.00         1

    accuracy                           0.98        52
   macro avg       0.73      0.75      0.74        52
weighted avg       0.96      0.98      0.97        52

Matriz de confusión:
[[12  0  0  0]
 [ 0 26  0  0]
 [ 0  0 13  0]
 [ 0  0  1  0]]
```

**Fuente: Elaboración propia**

En el reporte muestra la partición del conjunto de datos del 80% para el entrenamiento y 20% para la validación, de los resultados en el conjunto de validación muestran los valores obtenidos en las siguientes métricas: exactitud 98.08%, precisión 73.21%, sensibilidad 75.00% y F1-score 74.07%, lo que indica que el rendimiento es del 80% para cada clase, demostrando que el modelo es moderado.

**Figura N° 82: Gráfica de comparación – RF**




**Fuente: Elaboración propia**

El análisis comparativo de “Entrenamiento y Validación” muestran que, en los resultados obtenidos en las siguientes métricas: exactitud 98.08%, precisión 73.21%, sensibilidad 75.00% y F1-score 74.07%, denota que el rendimiento en el desempeño del modelo es moderado en ambos conjuntos de datos.

- **Support Vector Machines**

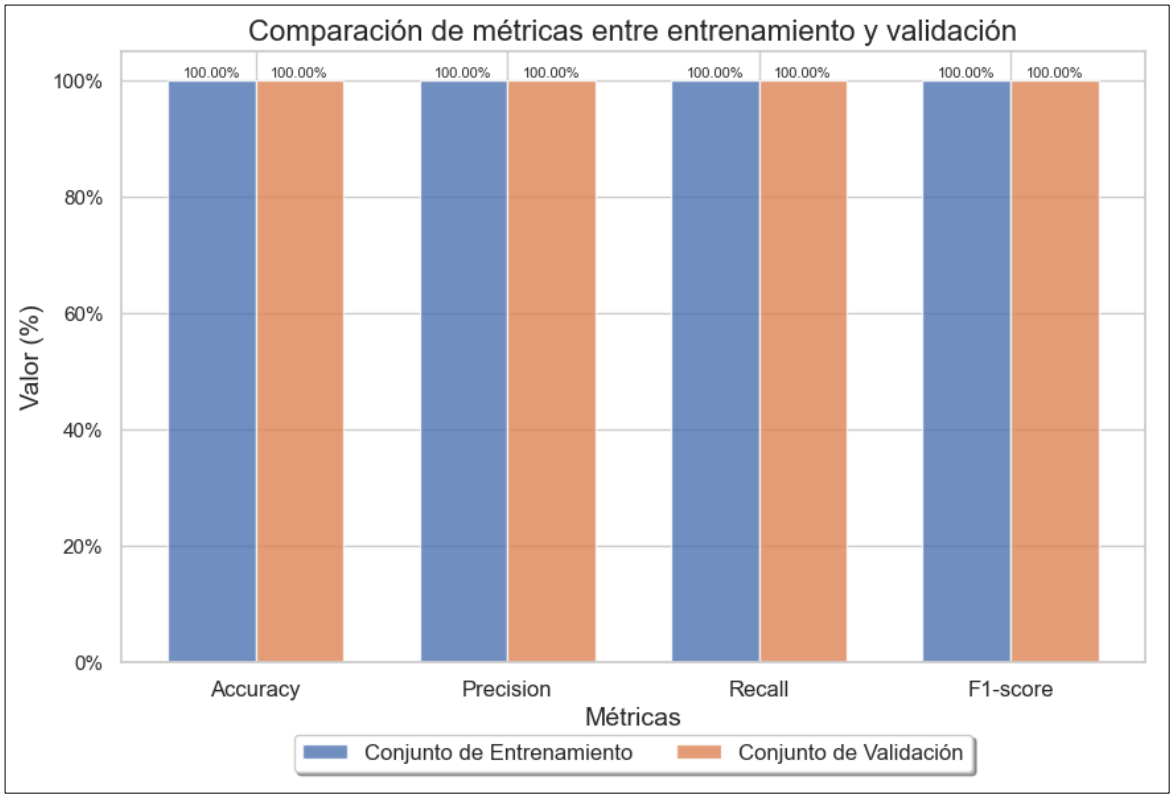
**Figura N° 83: Reporte de entrenamiento y validación – SVM**

```
RESULTADOS DEL CONJUNTO DE ENTRENAMIENTO:
Accuracy: 1.0
Precision: 1.0
Recall: 1.0
F1-score: 1.0
-----Reporte de Entrenamiento:-----
              precision    recall  f1-score   support

         2.0       1.00      1.00      1.00        35
         3.0       1.00      1.00      1.00       123
         4.0       1.00      1.00      1.00        43
         5.0       1.00      1.00      1.00         3

    accuracy                           1.00       204
   macro avg       1.00      1.00      1.00       204
weighted avg       1.00      1.00      1.00       204

Matriz de confusión:
[[ 35   0   0   0]
 [  0 123   0   0]
 [  0   0  43   0]
 [  0   0   0   3]]
//////////////////////////////////////////////////
RESULTADOS DEL CONJUNTO DE VALIDACIÓN:
Accuracy: 1.0
Precision: 1.0
Recall: 1.0
F1-score: 1.0
-----Reporte de Validación:-----
              precision    recall  f1-score   support

         2.0       1.00      1.00      1.00        13
         3.0       1.00      1.00      1.00        32
         4.0       1.00      1.00      1.00         6
         5.0       1.00      1.00      1.00         1

    accuracy                           1.00        52
   macro avg       1.00      1.00      1.00        52
weighted avg       1.00      1.00      1.00        52

Matriz de confusión:
[[13  0  0  0]
 [ 0 32  0  0]
 [ 0  0  6  0]
 [ 0  0  0  1]]
```

**Fuente: Elaboración propia**

En el reporte muestra la partición del conjunto de datos del 80% para el entrenamiento y 20% para la validación, de los resultados obtenidos en el conjunto de validación en las métricas de exactitud, precisión, sensibilidad y F1-score indica un rendimiento del 100% para cada clase, demostrando que el modelo es muy preciso. La "Matriz de confusión" respalda esta afirmación al revelar que no se han cometido errores de clasificación en ninguna de las clases en ambos conjuntos de datos.

**Figura N° 84: Gráfica de comparación – SVM**




**Fuente: Elaboración propia**

El análisis comparativo de “Entrenamiento y Validación” muestran que, en las métricas de exactitud, precisión, sensibilidad y F1-score se obtuvo un 100%, lo cual denota un rendimiento muy preciso en el desempeño del modelo en ambos conjuntos de datos.

- **Artificial Neural Network**

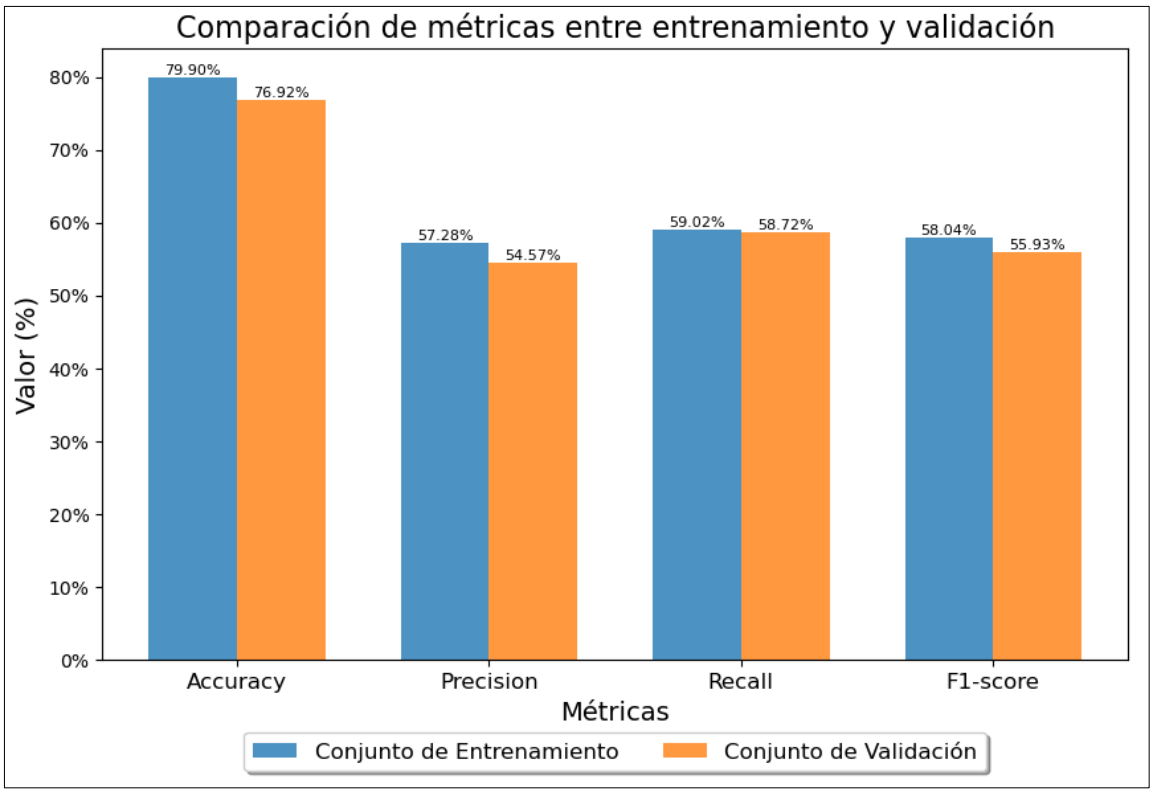
**Figura N° 85: Reporte de entrenamiento y validación – ANN**

```
RESULTADOS DEL CONJUNTO DE ENTRENAMIENTO:
Accuracy: 0.7990196078431373
Precision: 0.5728087589933427
Recall: 0.5901767151767152
F1-score: 0.5804050618056643
-----Reporte de Entrenamiento:-----
              precision    recall  f1-score   support

         2.0       0.76      0.74      0.75        39
         3.0       0.85      0.83      0.84       126
         4.0       0.67      0.78      0.72        37
         5.0       0.00      0.00      0.00         2

    accuracy                           0.80       204
   macro avg       0.57      0.59      0.58       204
weighted avg       0.80      0.80      0.80       204

Matriz de confusión:
[[ 29  10   0   0]
 [  9 105  12   0]
 [  0   8  29   0]
 [  0   0   2   0]]
////////////////////////////////////////////////////
RESULTADOS DEL CONJUNTO DE VALIDACIÓN:
Accuracy: 0.7692307692307693
Precision: 0.5456835511982571
Recall: 0.5871647509578544
F1-score: 0.5593125181106926
-----Reporte de Validación:-----
              precision    recall  f1-score   support

         2.0       0.62      0.56      0.59         9
         3.0       0.85      0.79      0.82        29
         4.0       0.71      1.00      0.83        12
         5.0       0.00      0.00      0.00         2

    accuracy                           0.77        52
   macro avg       0.55      0.59      0.56        52
weighted avg       0.75      0.77      0.75        52

Matriz de confusión:
[[ 5  4  0  0]
 [ 3 23  3  0]
 [ 0  0 12  0]
 [ 0  0  2  0]]
```

**Fuente: Elaboración propia**

En el reporte muestra la partición del conjunto de datos del 80% para el entrenamiento y 20% para la validación, de los resultados en el conjunto de validación muestran los valores obtenidos en las siguientes métricas: exactitud 76.92%, precisión 54.57%, sensibilidad 58.72% y F1-score 55.93%, lo que indica que el rendimiento es del 61.54% para cada clase, demostrando que el modelo es moderado.

**Figura N° 86: Gráfica de comparación – ANN**




**Fuente: Elaboración propia**

El análisis comparativo de “Entrenamiento y Validación” muestran que, en los resultados obtenidos en las siguientes métricas: exactitud 76.92%, precisión 54.57%, sensibilidad 58.72% y F1-score 55.93%, denota que el rendimiento en el desempeño del modelo es moderado en ambos conjuntos de datos.

- **Gradient Boosting Machines**

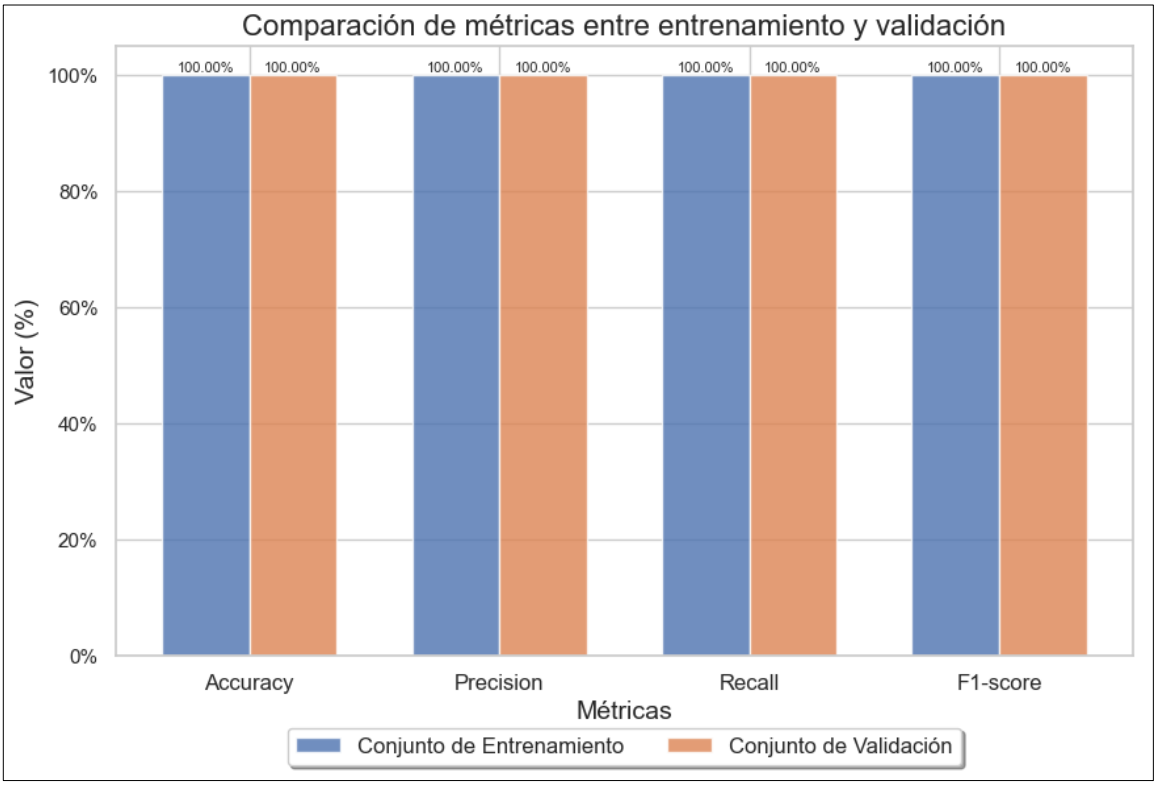
**Figura N° 87: Reporte de entrenamiento y validación – GBM**

```
RESULTADOS DEL CONJUNTO DE ENTRENAMIENTO:
Accuracy: 1.0
Precision: 1.0
Recall: 1.0
F1-score: 1.0
-----Reporte de Entrenamiento:-----
              precision    recall  f1-score   support

         2.0       1.00      1.00      1.00        39
         3.0       1.00      1.00      1.00       125
         4.0       1.00      1.00      1.00        37
         5.0       1.00      1.00      1.00         3

    accuracy                           1.00       204
   macro avg       1.00      1.00      1.00       204
weighted avg       1.00      1.00      1.00       204

Matriz de confusión:
[[ 39   0   0   0]
 [  0 125   0   0]
 [  0   0  37   0]
 [  0   0   0   3]]
//////////////////////////////////////////////////
RESULTADOS DEL CONJUNTO DE VALIDACIÓN:
Accuracy: 1.0
Precision: 1.0
Recall: 1.0
F1-score: 1.0
-----Reporte de Validación:-----
              precision    recall  f1-score   support

         2.0       1.00      1.00      1.00         9
         3.0       1.00      1.00      1.00        30
         4.0       1.00      1.00      1.00        12
         5.0       1.00      1.00      1.00         1

    accuracy                           1.00        52
   macro avg       1.00      1.00      1.00        52
weighted avg       1.00      1.00      1.00        52

Matriz de confusión:
[[ 9  0  0  0]
 [ 0 30  0  0]
 [ 0  0 12  0]
 [ 0  0  0  1]]
```

**Fuente: Elaboración propia**

En el reporte muestra la partición del conjunto de datos del 80% para el entrenamiento y 20% para la validación, de los resultados obtenidos en el conjunto de validación en las métricas de exactitud, precisión, sensibilidad y F1-score indica un rendimiento del 100% para cada clase, demostrando que el modelo es muy preciso. La "Matriz de confusión" respalda esta afirmación al revelar que no se han cometido errores de clasificación en ninguna de las clases en ambos conjuntos de datos.

**Figura N° 88: Gráfica de comparación – GBM**




**Fuente: Elaboración propia**

El análisis comparativo de “Entrenamiento y Validación” muestran que, en las métricas de exactitud, precisión, sensibilidad y F1-score se obtuvo un 100%, lo cual denota un rendimiento muy preciso en el desempeño del modelo en ambos conjuntos de datos.

- **Multi Perceptrón**

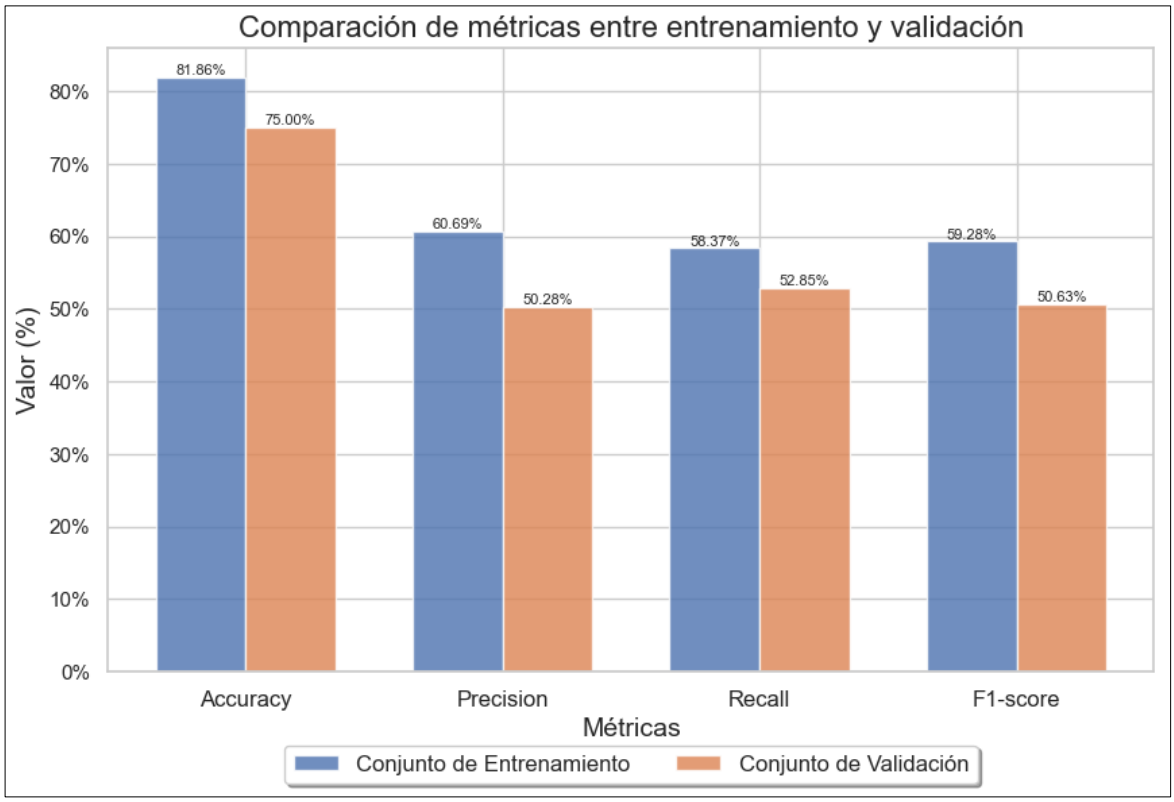
**Figura N° 89: Reporte de entrenamiento y validación – MP**

```
RESULTADOS DEL CONJUNTO DE ENTRENAMIENTO:
Accuracy: 0.8186274509803921
Precision: 0.6068722943722944
Recall: 0.5837343264387526
F1-score: 0.5927857313902571
-----Reporte de Entrenamiento:-----
              precision    recall  f1-score   support

         2.0       0.77      0.79      0.78        43
         3.0       0.83      0.89      0.86       124
         4.0       0.82      0.66      0.73        35
         5.0       0.00      0.00      0.00         2

    accuracy                           0.82       204
   macro avg       0.61      0.58      0.59       204
weighted avg       0.81      0.82      0.81       204

Matriz de confusión:
[[ 34   9   0   0]
 [ 10 110   4   0]
 [  0  12  23   0]
 [  0   1   1   0]]
//////////////////////////////////////////////////
RESULTADOS DEL CONJUNTO DE VALIDACIÓN:
Accuracy: 0.75
Precision: 0.5028409090909092
Recall: 0.5284562211981567
F1-score: 0.5063221153846154
-----Reporte de Validación:-----
              precision    recall  f1-score   support

         2.0       0.38      0.60      0.46         5
         3.0       0.82      0.87      0.84        31
         4.0       0.82      0.64      0.72        14
         5.0       0.00      0.00      0.00         2

    accuracy                           0.75       52
   macro avg       0.50      0.53      0.51       52
weighted avg       0.74      0.75      0.74       52

Matriz de confusión:
[[ 3  2  0  0]
 [ 4 27  0  0]
 [ 1  4  9  0]
 [ 0  0  2  0]]
```

**Fuente: Elaboración propia**

En el reporte muestra la partición del conjunto de datos del 80% para el entrenamiento y 20% para la validación, de los resultados en el conjunto de validación muestran los valores obtenidos en las siguientes métricas: exactitud 75.00%, precisión 50.28%, sensibilidad 52.85% y F1-score 50.63%, lo que indica que el rendimiento es del 57.44% para cada clase, demostrando que el modelo es bajo.

**Figura N° 90: Gráfica de comparación – MP**




**Fuente: Elaboración propia**

El análisis comparativo de “Entrenamiento y Validación” muestran que, en los resultados obtenidos en las siguientes métricas: exactitud 75.00%, precisión 50.28%, sensibilidad 52.85% y F1-score 50.63%, denota que el rendimiento en el desempeño del modelo es moderado en ambos conjuntos de datos.

- **K-Nearest Neighboor**

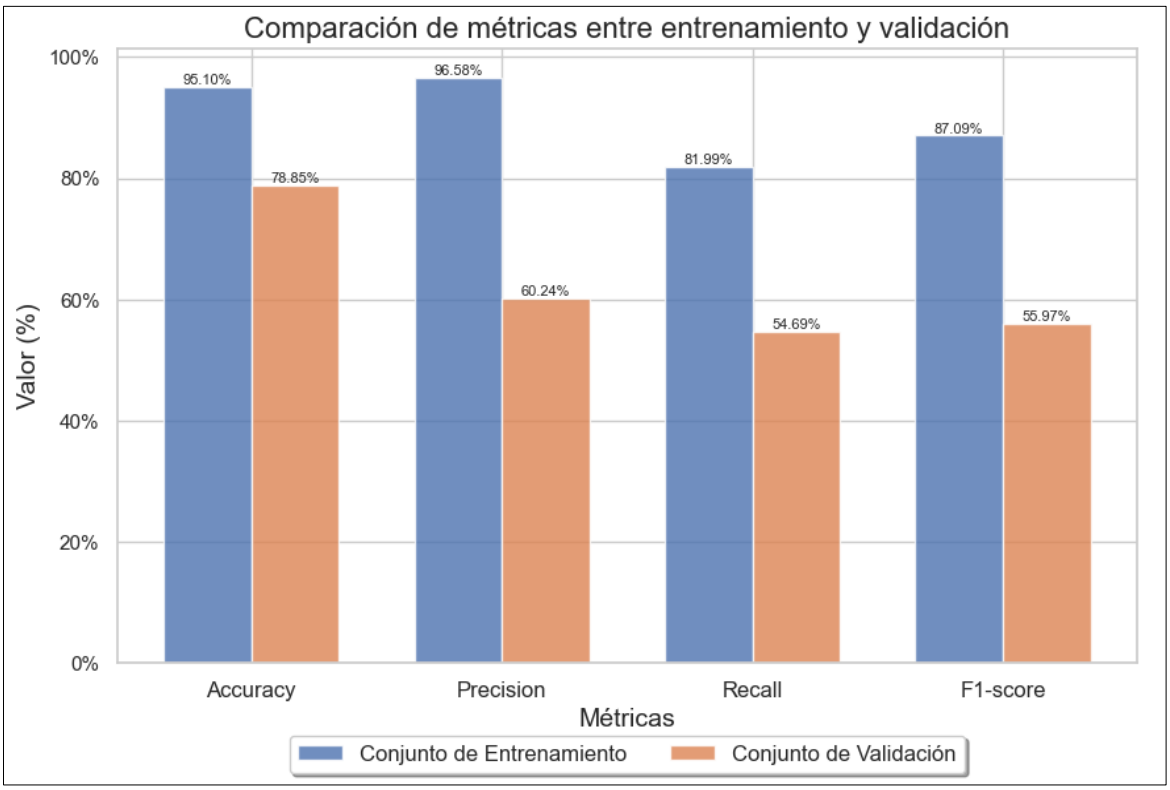
**Figura N° 91: Reporte de entrenamiento y validación – KNN**

```
RESULTADOS DEL CONJUNTO DE ENTRENAMIENTO:
Accuracy: 0.9509803921568627
Precision: 0.9658166051276718
Recall: 0.8198664090565324
F1-score: 0.8709164116355992
-----Reporte de Entrenamiento:-----
              precision    recall  f1-score   support

         2.0       0.96      0.84      0.90        32
         3.0       0.95      0.98      0.97       129
         4.0       0.95      0.95      0.95        41
         5.0       1.00      0.50      0.67         2

    accuracy                           0.95       204
   macro avg       0.97      0.82      0.87       204
weighted avg       0.95      0.95      0.95       204

Matriz de confusión:
[[ 27   5   0   0]
 [  1 127   1   0]
 [  0   2  39   0]
 [  0   0   1   1]]
//////////////////////////////////////////////////
RESULTADOS DEL CONJUNTO DE VALIDACIÓN:
Accuracy: 0.7884615384615384
Precision: 0.6023809523809524
Recall: 0.546875
F1-score: 0.5596756006592073
-----Reporte de Validación:-----
              precision    recall  f1-score   support

         2.0       1.00      0.69      0.81        16
         3.0       0.74      1.00      0.85        26
         4.0       0.67      0.50      0.57         8
         5.0       0.00      0.00      0.00         2

    accuracy                           0.79        52
   macro avg       0.60      0.55      0.56        52
weighted avg       0.78      0.79      0.76        52

Matriz de confusión:
[[11  5  0  0]
 [ 0 26  0  0]
 [ 0  4  4  0]
 [ 0  0  2  0]]
```

**Fuente: Elaboración propia**

En el reporte muestra la partición del conjunto de datos del 80% para el entrenamiento y 20% para la validación, de los resultados en el conjunto de validación muestran los valores obtenidos en las siguientes métricas: exactitud 78.85%, precisión 60.24%, sensibilidad 54.69% y F1-score 56.97%, lo que indica que el rendimiento es del 62.69% para cada clase, demostrando que el modelo es moderado.

**Figura N° 92: Gráfica de comparación – KNN**




**Fuente: Elaboración propia**

El análisis comparativo de “Entrenamiento y Validación” muestran que, en los resultados obtenidos en las siguientes métricas: exactitud 78.85%, precisión 60.24%, sensibilidad 54.69% y F1-score 56.97%, denota que el rendimiento en el desempeño del modelo es moderado en ambos conjuntos de datos.

- **Naïve Bayes**

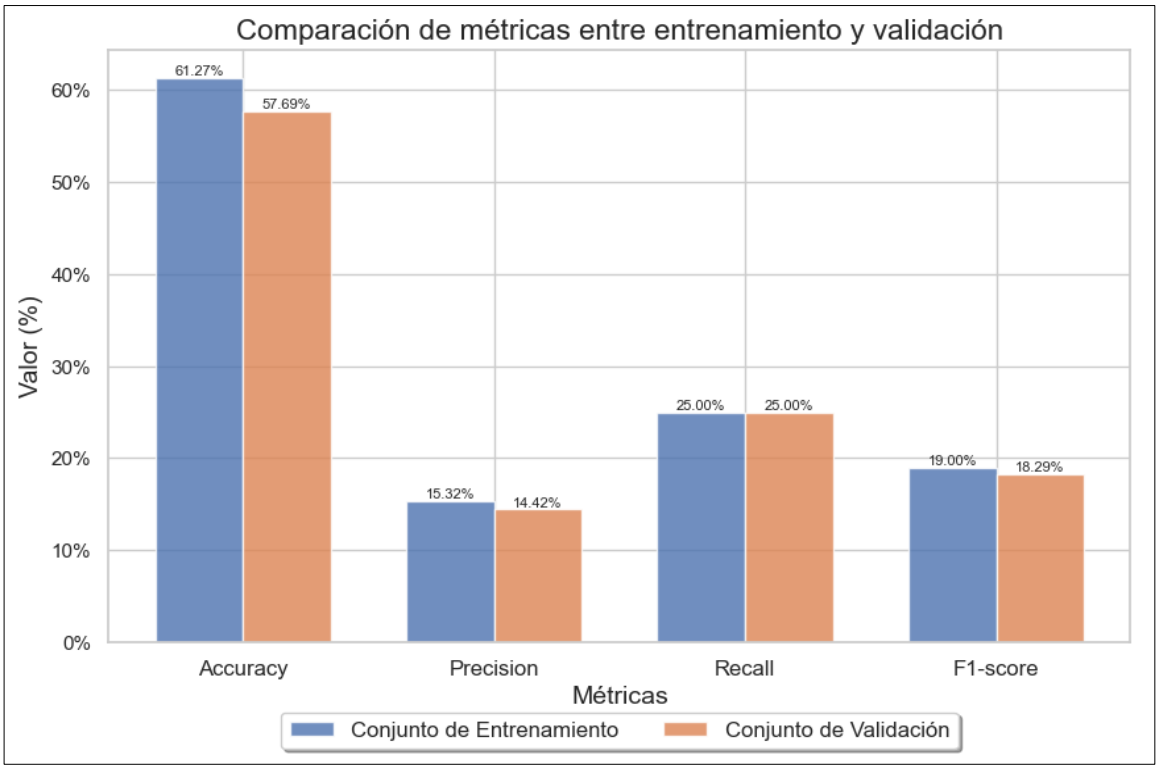
**Figura N° 93: Reporte de entrenamiento y validación – NB**

```
RESULTADOS DEL CONJUNTO DE ENTRENAMIENTO:
Accuracy: 0.6127450980392157
Precision: 0.15318627450980393
Recall: 0.25
F1-score: 0.1899696048632219
-----Reporte de Entrenamiento:-----
              precision    recall  f1-score   support

         2.0       0.00      0.00      0.00        38
         3.0       0.61      1.00      0.76       125
         4.0       0.00      0.00      0.00        38
         5.0       0.00      0.00      0.00         3

    accuracy                           0.61       204
   macro avg       0.15      0.25      0.19       204
weighted avg       0.38      0.61      0.47       204

Matriz de confusión:
[[  0  38   0   0]
 [  0 125   0   0]
 [  0  38   0   0]
 [  0   3   0   0]]
//////////////////////////////////////////////////
RESULTADOS DEL CONJUNTO DE VALIDACIÓN:
Accuracy: 0.5769230769230769
Precision: 0.14423076923076922
Recall: 0.25
F1-score: 0.18292682926829268
-----Reporte de Validación:-----
              precision    recall  f1-score   support

         2.0       0.00      0.00      0.00        10
         3.0       0.58      1.00      0.73        30
         4.0       0.00      0.00      0.00        11
         5.0       0.00      0.00      0.00         1

    accuracy                           0.58        52
   macro avg       0.14      0.25      0.18        52
weighted avg       0.33      0.58      0.42        52

Matriz de confusión:
[[ 0 10  0  0]
 [ 0 30  0  0]
 [ 0 11  0  0]
 [ 0  1  0  0]]
```

**Fuente: Elaboración propia**

En el reporte muestra la partición del conjunto de datos del 80% para el entrenamiento y 20% para la validación, de los resultados en el conjunto de validación muestran los valores obtenidos en las siguientes métricas: exactitud 57.69%, precisión 14.42%, sensibilidad 25.00% y F1-score 18.29%, lo que indica que el rendimiento es del 28.85% para cada clase, demostrando que el modelo es deficiente.

**Figura N° 94: Gráfica de comparación – NB**




**Fuente: Elaboración propia**

El análisis co mparativo de “Entrenamiento y Validación” muestran que, en los resultados obtenidos en las siguientes métricas: exactitud 57.69%, precisión 14.42%, sensibilidad 25.00% y F1-score 18.29%, denota que el rendimiento en el desempeño del modelo es moderado en ambos conjuntos de datos.

- **Logistic Regression**

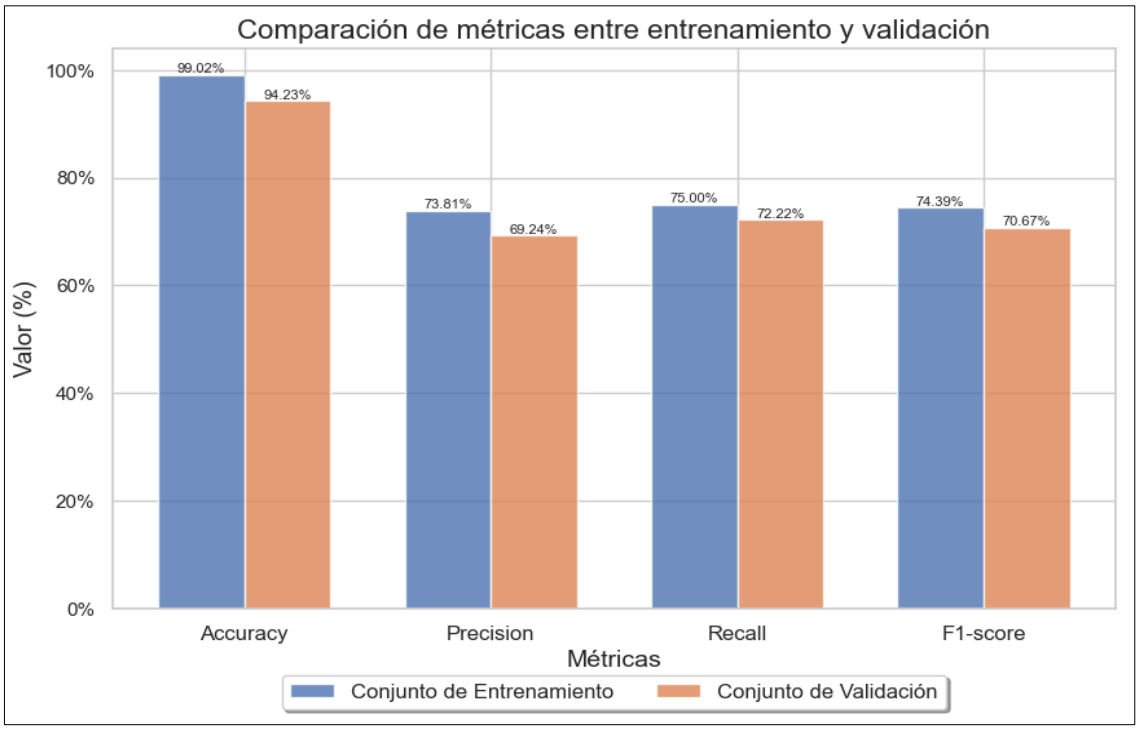
**Figura N° 95: Reporte de entrenamiento y validación - LR**

```
RESULTADOS DEL CONJUNTO DE ENTRENAMIENTO:
Accuracy: 0.9901960784313726
Precision: 0.7380952380952381
Recall: 0.75
F1-score: 0.7439024390243902
-----Reporte de Entrenamiento:-----
              precision    recall  f1-score   support

         2.0       1.00      1.00      1.00        39
         3.0       1.00      1.00      1.00       123
         4.0       0.95      1.00      0.98        40
         5.0       0.00      0.00      0.00         2

    accuracy                           0.99       204
   macro avg       0.74      0.75      0.74       204
weighted avg       0.98      0.99      0.99       204

Matriz de confusión:
[[ 39   0   0   0]
 [  0 123   0   0]
 [  0   0  40   0]
 [  0   0   2   0]]
//////////////////////////////////////////////////////
RESULTADOS DEL CONJUNTO DE VALIDACIÓN:
Accuracy: 0.9423076923076923
Precision: 0.6924242424242424
Recall: 0.7222222222222222
F1-score: 0.7066801619433198
-----Reporte de Validación:-----
              precision    recall  f1-score   support

         2.0       1.00      1.00      1.00         9
         3.0       0.97      1.00      0.98        32
         4.0       0.80      0.89      0.84         9
         5.0       0.00      0.00      0.00         2

    accuracy                           0.94        52
   macro avg       0.69      0.72      0.71        52
weighted avg       0.91      0.94      0.92        52

Matriz de confusión:
[[ 9  0  0  0]
 [ 0 32  0  0]
 [ 0  1  8  0]
 [ 0  0  2  0]]
```

**Fuente: Elaboración propia**

En el reporte muestra la partición del conjunto de datos del 80% para el entrenamiento y 20% para la validación, de los resultados en el conjunto de validación muestran los valores obtenidos en las siguientes métricas: exactitud 94.23%, precisión 69.24%, sensibilidad 72.22% y F1-score 70.67%, lo que indica que el rendimiento es del 76.59% para cada clase, demostrando que el modelo es moderado.

**Figura N° 96: Gráfica de comparación – LR**




**Fuente: Elaboración propia**

El análisis comparativo de “Entrenamiento y Validación” muestran que, en los resultados obtenidos en las siguientes métricas: exactitud 94.23%, precisión 69.24%, sensibilidad 72.22% y F1-score 70.67%, denota que el rendimiento en el desempeño del modelo es moderado en ambos conjuntos de datos.

- **Ada Boosting**

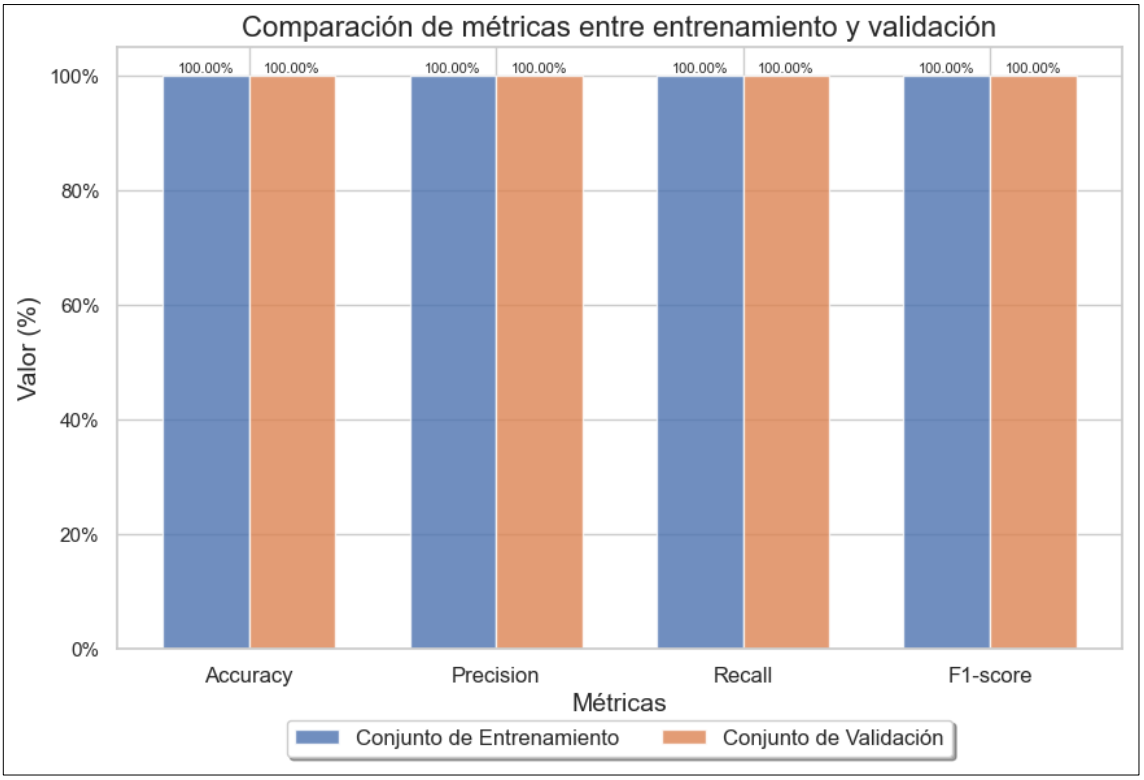
**Figura N° 97: Reporte de entrenamiento y validación – ADA**

```
RESULTADOS DEL CONJUNTO DE ENTRENAMIENTO:
Accuracy: 1.0
Precision: 1.0
Recall: 1.0
F1-score: 1.0
-----Reporte de Entrenamiento:-----
              precision    recall  f1-score   support

         2.0       1.00      1.00      1.00        36
         3.0       1.00      1.00      1.00       121
         4.0       1.00      1.00      1.00        44
         5.0       1.00      1.00      1.00         3

    accuracy                           1.00       204
   macro avg       1.00      1.00      1.00       204
weighted avg       1.00      1.00      1.00       204

Matriz de confusión:
[[ 36   0   0   0]
 [  0 121   0   0]
 [  0   0  44   0]
 [  0   0   0   3]]
//////////////////////////////////////////////////
RESULTADOS DEL CONJUNTO DE VALIDACIÓN:
Accuracy: 1.0
Precision: 1.0
Recall: 1.0
F1-score: 1.0
-----Reporte de Validación:-----
              precision    recall  f1-score   support

         2.0       1.00      1.00      1.00        12
         3.0       1.00      1.00      1.00        34
         4.0       1.00      1.00      1.00         5
         5.0       1.00      1.00      1.00         1

    accuracy                           1.00        52
   macro avg       1.00      1.00      1.00        52
weighted avg       1.00      1.00      1.00        52

Matriz de confusión:
[[12  0  0  0]
 [ 0 34  0  0]
 [ 0  0  5  0]
 [ 0  0  0  1]]
```

**Fuente: Elaboración propia**

En el reporte muestra la partición del conjunto de datos del 80% para el entrenamiento y 20% para la validación, de los resultados obtenidos en el conjunto de validación en las métricas de exactitud, precisión, sensibilidad y F1-score indica un rendimiento del 100% para cada clase, demostrando que el modelo es muy preciso. La "Matriz de confusión" respalda esta afirmación al revelar que no se han cometido errores de clasificación en ninguna de las clases en ambos conjuntos de datos.

**Figura N° 98: Gráfica de comparación – ADA**




**Fuente: Elaboración propia**

El análisis comparativo de “Entrenamiento y Validación” muestran que, en las métricas de exactitud, precisión, sensibilidad y F1-score se obtuvo un 100%, lo cual denota un rendimiento muy preciso en el desempeño del modelo en ambos conjuntos de datos.

**Figura N° 99: Modelo entrenado - DT**

```
import pickle
# Guardar el modelo entrenado en un archivo pickle
with open('decision_tree_model_trained.pkl', 'wb') as model_file:
    pickle.dump(grid_search.best_estimator_, model_file)
```

**Fuente: Elaboración propia**

Guarda el modelo entrenado del algoritmo de aprendizaje automático DT, mediante el uso de la biblioteca “pickle” en base a los hiperparámetros y al particionamiento de entrenamiento y validación para llevar a cabo las predicciones.

**Figura N° 100: Modelo entrenado - SVM**

```
import pickle
# Guardar el modelo entrenado en un archivo pickle
with open('support_vector_machine_trained.pkl', 'wb') as model_file:
    pickle.dump(grid_search.best_estimator_, model_file)
```

**Fuente: Elaboración propia**

Guarda el modelo entrenado del algoritmo de aprendizaje automático SVM, mediante el uso de la biblioteca “pickle” en base a los hiperparámetros y al particionamiento de entrenamiento y validación para llevar a cabo las predicciones.

**Figura N° 101: Modelo entrenado - GBM**

```
import pickle
# Guardar el modelo entrenado en un archivo pickle
with open('gradient_boosting_machine.pkl', 'wb') as model_file:
    pickle.dump(grid_search.best_estimator_, model_file)
```

**Fuente: Elaboración propia**

Guarda el modelo entrenado del algoritmo de aprendizaje automático GBM, mediante el uso de la biblioteca “pickle” en base a los hiperparámetros y al particionamiento de entrenamiento y validación para llevar a cabo las predicciones.

**Figura N° 102: Modelo entrenado - ADA**

```
import pickle
# Guardar el modelo entrenado en un archivo pickle
with open('ada_boosting.pkl', 'wb') as model_file:
    pickle.dump(grid_search.best_estimator_, model_file)
```

**Fuente: Elaboración propia**

Guarda el modelo entrenado del algoritmo de aprendizaje automático ADA,mediante el uso de la biblioteca “pickle” en base a los hiperparámetros y al particionamiento de entrenamiento y validación para llevar a cabo las predicciones.

- **Sistema Inteligente con ML**

**Figura N° 103: Librerías usadas para el Sistema Inteligente**

```
1
2   import streamlit_option_menu as som
3   import streamlit as st
4   import pandas as pd
5   import numpy as np
6   import pickle
7
8   #ALGORITMOS-----------------------------------------------------------
9   from sklearn.model_selection import train_test_split
10  from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix
11  import seaborn as sns
12  import matplotlib.pyplot as plt
13  import pyodbc
```

**Fuente: Elaboración propia**

Se importan las librerías “streamlit” y “streamlit_option” para el diseño del sistema, para la conexión a la BD “pyodbc” y “pickle”, “sklearm.model_selection”, “sklearn.metrics” para las métricas y los algoritmos de aprendizaje automático.

**Figura N° 104: Conexión a la base de datos desde Spyder**

```
15  #CONEXIÓN A BASE DE DATOS-------------------------------------------------------
16
17  #DATOS DEL SERVIDOR
18  server = 'LAPTOP-J8MRKNGK\SQLEXPRESS'
19  bd= 'DATA_RA'
20
21  #CONEXION AL SERVIDOR (SQL)
22  conexion = pyodbc.connect(driver='{SQL server}', host = server, database = bd)
23  usuario = 'sa'
24  contrasena = 'botis123'
```

**Fuente: Elaboración propia**

Se importan las librerías “server” y “bd” para la conexión al servidor y para la conexión a la base de datos se establece los parámetros de “conexión”, “usuario” y “contraseña”.

**Figura N° 105: Modelos importados al Sistema Inteligente**

```
#Modelo DT
model_dt = pickle.load(open('C:\\Users\\valer\\Desktop\\sistema_academic\\models\\decision_tree_model_trained.pkl', 'rb'))

#Modelo SVM
model_svm = pickle.load(open('C:\\Users\\valer\\Desktop\\sistema_academic\\models\\svm_model_trained.pkl', 'rb'))

#Modelo GBM
model_gbm = pickle.load(open('C:\\Users\\valer\\Desktop\\sistema_academic\\models\\gbm_model_trained.pkl', 'rb'))

#Modelo ADA
model_ada = pickle.load(open('C:\\Users\\valer\\Desktop\\sistema_academic\\models\\ada_model_trained.pkl', 'rb'))
```

**Fuente: Elaboración propia**

Mediante el uso de la biblioteca “pickle” y estableciendo la ruta en las que se almacenaron, se carga los modelos entrenados (DT, SVM, GBM y ADA).

**Figura N° 106: Codificación del modelo DT**

```
        if selected_algorithm == 'DESICION TREE':
            if st.button("EVALUAR - DT"):
                prediction = model_dt.predict(user_input)
                probability = model_dt.predict_proba(user_input)
                argmax = np.argmax(probability)
                probability = probability[0]

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

                # Agregar interpretación
                interpretation = ""
                if classification_result == 1:
                    interpretation = "Desaprobado"
                elif classification_result == 2:
                    interpretation = "Regular"
                elif classification_result == 3:
                    interpretation = "Bueno"
                elif classification_result == 4:
                    interpretation = "Distinguido"
                elif classification_result == 5:
                    interpretation = "Excelente"
                st.subheader('Interpretación:')
                st.success(interpretation)
```

**Fuente: Elaboración propia**

Se establece una condición para seleccionar, evaluar, realizar la predicción en base al modelo entrenado e interpretación de los resultados del algoritmo DT.

**Figura N° 107: Codificación del modelo SVM**

```
379         elif selected_algorithm == 'SUPPORT VECTOR MACHINE':
380             if st.button("EVALUAR - SVM"):
381                 prediction = model_svm.predict(user_input)
382                 probability = model_svm.predict_proba(user_input)
383                 argmax = np.argmax(probability)
384                 probability = probability[0]
385 
386                 st.subheader('Resultado:')
387                 classification_result = prediction[0]
388                 st.success(classification_result)
389 
390                 # Agregar interpretación
391                 interpretation = ""
392                 if classification_result == 1:
393                     interpretation = "Desaprobado"
394                 elif classification_result == 2:
395                     interpretation = "Regular"
396                 elif classification_result == 3:
397                     interpretation = "Bueno"
398                 elif classification_result == 4:
399                     interpretation = "Distinguido"
400                 elif classification_result == 5:
401                     interpretation = "Excelente"
402                 st.subheader('Interpretación:')
403                 st.success(interpretation)
```

**Fuente: Elaboración propia**

Se establece una condición para seleccionar, evaluar, realizar la predicción en base al modelo entrenado e interpretación de los resultados del algoritmo SVM.

**Figura N° 108: Codificación del modelo GBM**

```
408         elif selected_algorithm == 'GRADIENT BOOSTING MACHINES':
409             if st.button("EVALUAR - GBM"):
410                 prediction = model_gbm.predict(user_input)
411                 probability = model_gbm.predict_proba(user_input)
412                 argmax = np.argmax(probability)
413                 probability = probability[0]
414 
415                 st.subheader('Resultado:')
416                 classification_result = prediction[0]
417                 st.success(classification_result)
418 
419                 # Agregar interpretación
420                 interpretation = ""
421                 if classification_result == 1:
422                     interpretation = "Desaprobado"
423                 elif classification_result == 2:
424                     interpretation = "Regular"
425                 elif classification_result == 3:
426                     interpretation = "Bueno"
427                 elif classification_result == 4:
428                     interpretation = "Distinguido"
429                 elif classification_result == 5:
430                     interpretation = "Excelente"
431                 st.subheader('Interpretación:')
432                 st.success(interpretation)
```

**Fuente: Elaboración propia**

Se establece una condición para seleccionar, evaluar, realizar la predicción en base al modelo entrenado e interpretación de los resultados del algoritmo GBM.

**Figura N° 109: Codificación del modelo ADA**

```
437         elif selected_algorithm == 'ADA BOOST':
438             if st.button("EVALUAR - ADA"):
439                 prediction = model_ada.predict(user_input)
440                 probability = model_ada.predict_proba(user_input)
441                 argmax = np.argmax(probability)
442                 probability = probability[0]
443 
444                 st.subheader('Resultado:')
445                 classification_result = prediction[0]
446                 st.success(classification_result)
447 
448                 # Agregar interpretación
449                 interpretation = ""
450                 if classification_result == 1:
451                     interpretation = "Desaprobado"
452                 elif classification_result == 2:
453                     interpretation = "Regular"
454                 elif classification_result == 3:
455                     interpretation = "Bueno"
456                 elif classification_result == 4:
457                     interpretation = "Distinguido"
458                 elif classification_result == 5:
459                     interpretation = "Excelente"
460                 st.subheader('Interpretación:')
461                 st.success(interpretation)
```

**Fuente: Elaboración propia**

Se establece una condición para seleccionar, evaluar, realizar la predicción en base al modelo entrenado e interpretación de los resultados del algoritmo ADA.

**Figura N° 110: Vista principal del Sistema Inteligente**


**Fuente: Elaboración Propia**

En la opción de "Inicio" se muestra la pantalla principal del sistema inteligente en el cual detalla los datos y logo de la institución educativa.

**Figura N° 111: Visualización de los datos del modelo**


| | EDAD | NIVEL_REND_ACAD | CONS_INTELIGENTE | HORAS_ESTUDIO | CONS_ORGANIZACION | CANT_ESTRES | REAC_PROB_SOCIAL | APOYO_ACAD | ECO |
|---|---|---|---|---|---|---|---|---|---|
| 32 | 12 | 4 | 2 | 1 | 2 | 4 | 2 | 2 | |
| 33 | 12 | 3 | 1 | 0 | 2 | 2 | 2 | 2 | |
| 34 | 12 | 4 | 2 | 2 | 2 | 3 | 2 | 2 | |
| 35 | 12 | 5 | 2 | 2 | 1 | 1 | 2 | 2 | |
| 36 | 12 | 4 | 2 | 2 | 2 | 3 | 2 | 2 | |
| 37 | 12 | 2 | 1 | 2 | 2 | 3 | 3 | 2 | |
| 38 | 12 | 4 | 1 | 3 | 2 | 2 | 2 | 2 | |
| 39 | 12 | 4 | 1 | 1 | 1 | 2 | 1 | 2 | |
| 40 | 12 | 0 | 0 | 2 | 2 | 3 | 2 | 2 | |
| 41 | 13 | 3 | 1 | 2 | 2 | 4 | 2 | 2 | |

| | EDAD | NIVEL_REND_ACAD | CONS_INTELIGENTE | HORAS_ESTUDIO | CONS_ORGANIZACION | CANT_ESTRES | REAC_PROB_SOCIAL | APOYO_ACA |
|---|---|---|---|---|---|---|---|---|
| 0 | 12 | Bueno | Si | 1 | Si | Poco Frecuente | Dialogas | Si |
| 1 | 12 | Bueno | Si | 4 | Sin especificar | Muy Frecuente | Dialogas | Si |
| 2 | 12 | Muy malo | No | 2 | No | Muy Infrecuente | Con Indiferencia | No |
| 3 | 12 | Muy Bueno | No | 4 | Si | Muy Infrecuente | Dialogas | Si |
| 4 | 12 | Regular | No | 2 | Si | Poco Frecuente | Dialogas | Si |

**Fuente: Elaboración Propia**

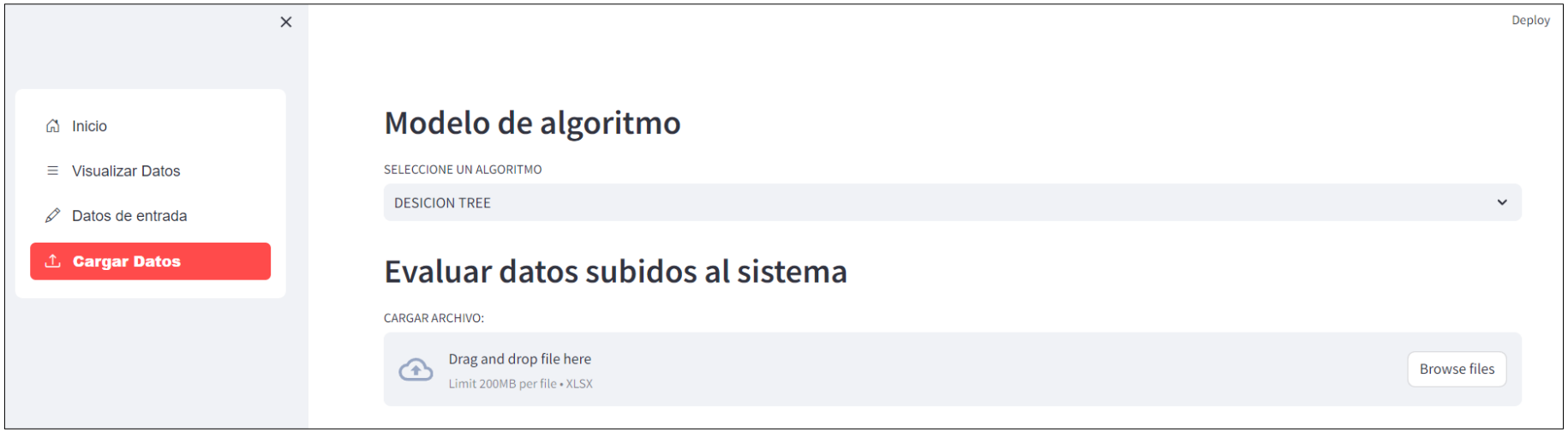
En la opción de “Visualizar Datos” se muestra los datos de los alumnos en base a 2 consultas realizadas a la base de datos principal.

**Figura N° 112: Datos de entrada del Sistema Inteligente**


**Fuente: Elaboración Propia**

En la opción de “Datos de entrada” muestra los campos que se deben de completar para luego realizar predicción en base a las métricas y los modelos entrenados (DT, SVM, GBM y ADA).

**Figura N° 113: Evaluación de los datos mediante archivos Excel**




**Fuente: Elaboración Propia**

En la opción de “Cargar Datos” se realiza la carga de un archivo en excel en base a las variables de entrada para luego realizar la predicción de los nuevos datos en base a los modelos entrenados (DT, SVM, GBM y ADA).

Finalmente, se demuestra las evidencias fotográficas de la Institución educativa y las pruebas realizadas con el sistema.

**Figura N° 114: I.E.P. "Nuestra Señora de Copacabana"**


**Fuente: Elaboración Propia**

**Figura N° 115: Participantes en las pruebas del sistema**

**Fuente: Elaboración Propia**

**Figura N° 116: Pruebas del sistema**


**Fuente: Elaboración Propia**

UNIVERSIDAD CÉSAR VALLEJO

**FACULTAD DE INGENIERÍA Y ARQUITECTURA**

**ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**

# Declaratoria de Autenticidad del Asesor

Yo, DAZA VERGARAY ALFREDO, docente de la FACULTAD DE INGENIERÍA Y ARQUITECTURA de la escuela profesional de INGENIERÍA DE SISTEMAS de la UNIVERSIDAD CÉSAR VALLEJO SAC - LIMA NORTE, asesor de Tesis titulada: "SISTEMA INTELIGENTE CON MACHINE LEARNING BASADO EN SELECCIÓN DE VARIABLES PARA PREDECIR EL RENDIMIENTO ACADÉMICO DE LA I.E.P. "NUESTRA SEÑORA DE COPACABANA", cuyos autores son SALCEDO ZAPATA JUAN ANTONIO, HUERTAS FABIAN NICOLE VALERY, constato que la investigación tiene un índice de similitud de 11.00%, verificable en el reporte de originalidad del programa Turnitin, el cual ha sido realizado sin filtros, ni exclusiones.

He revisado dicho reporte y concluyo que cada una de las coincidencias detectadas no constituyen plagio. A mi leal saber y entender la Tesis cumple con todas las normas para el uso de citas y referencias establecidas por la Universidad César Vallejo.

En tal sentido, asumo la responsabilidad que corresponda ante cualquier falsedad, ocultamiento u omisión tanto de los documentos como de información aportada, por lo cual me someto a lo dispuesto en las normas académicas vigentes de la Universidad César Vallejo.

LIMA, 26 de Diciembre del 2023

| **Apellidos y Nombres del Asesor:** | **Firma** |
|---|---|
| DAZA VERGARAY ALFREDO<br>**DNI:** 40466240<br>**ORCID:** 0000-0002-2259-1070 | Firmado electrónicamente por: ADAZAVE el 26-12-2023 12:24:58 |

Código documento Trilce: TRI - 0708288

INVESTIGA UCV