UNIVERSIDAD MAYOR DE SAN SIMÓN

FACULTAD DE CIENCIAS Y TECNOLOGÍA

DIRECCIÓN DE POSGRADO

DIPLOMADO CIENCIA DE DATOS

SEGUNDA VERSIÓN

ANÁLISIS Y PREDICCIÓN DEL BAJO RENDIMIENTO ACADÉMICO EN LA UNIDAD EDUCATIVA SAN JOSÉ OBRERO

PROYECTO PRESENTADO PARA OBTENER EL GRADO DE LICENCIATURA EN INGENIERÍA DE SISTEMAS
MODALIDAD DOBLE TITULACIÓN

POSTULANTE : LIMBERG VILLCA CORAITE
TUTOR : M.Sc. Ing. Ariel Mamani Nina

Cochabamba – Bolivia
2025

ANÁLISIS Y PREDICCIÓN DEL BAJO RENDIMIENTO ACADÉMICO EN LA UNIDAD EDUCATIVA SAN JOSÉ OBRERO

Por

Limberg Villca Coraite

El presente documento, Trabajo de Grado es presentado a la Dirección de Posgrado de la Facultad de Ciencias y Tecnología en cumplimiento parcial de los requisitos para la obtención del grado académico de Licenciatura (o sólo diplomado) en Ingeniería de Sistemas, modalidad Doble Titulación, habiendo cursado el Diplomado "Ciencia de datos V2" propuesta por el Centro de Estadística Aplicada (CESA) en su tercera versión.

ASESOR/TUTOR

M.Sc. Ing. Ariel Mamani Nina

COMITÉ DE EVALUACIÓN

Ing. M.Sc. Ronald Edgar Patiño Tito. (presidente)
Ing. M.Sc Guillen Salvador Roxana,. (Coordinador)
Ing. M. Sc Espinoza Orosco José (Tribunal)
Ing. por designar……., M.Sc. (Tribunal)

DIRECCIÓN DE POSGRADO, FACULTAD DE CIENCIAS Y TECNOLOGIA
Cochabamba, Bolivia

Aclaración

Este documento describe el trabajo realizado como parte del programa de estudios de Diplomado "Ciencia de Datos" en el Centro de Estadística Aplicada CESA y la Dirección de Posgrado de la Facultad de Ciencias y Tecnología. Todos los puntos de vista y opiniones expresadas en el mismo son responsabilidad exclusiva del autor y no representan necesariamente las de la institución.

Resumen

La calidad educativa en Bolivia, especialmente en zonas rurales, continúa siendo un desafío crítico para el sistema educativo nacional. En este contexto, el presente proyecto se enfoca en la Unidad educativa San José Obrero, ubicada en un área rural, con el objetivo de analizar el rendimiento académico de los estudiantes y desarrollar un modelo un modelo predictivo que permita anticipar que estudiantes reprobaran el año siguiente. Este trabajo parte del reconocimiento de que, a pesar de los avances de digitalización educativa, aún existe una marcada brecha entre estudiantes urbanos y rurales en términos de acceso, desempeño y oportunidades de mejora.

Para alcanzar el objetivo propuesto, se utilizó en un enfoque basado en la metodología CRISP-DM, iniciando con la recolección de datos históricos de calificaciones desde la gestión 2015 hasta 2024. Estos datos fueron proporcionados por la dirección del establecimiento educativo y comprendían información detallada de estudiantes de nivel primario y secundario. La etapa de preparación de datos incluyo la limpieza, codificación, transformación de variables y la creación de nuevas características como promedio histórico, numero de reprobaciones y tendencia del rendimiento, usando Python y jupyter notebook. Posteriormente, se aplicaron técnicas de machine learning con modelos como Random forest, XGBoost y CatBoost para construir clasificadores que permitan anticipar el riesgo de reprobación de los estudiantes.

Los resultados del análisis mostraron que las materias con mayor índice de reprobación fueron matemáticas, física y química, en especial en el nivel secundario. El modelo que obtuvo el mejor desempeño fue CatBoost, con un F1-Score ponderado superior a 0.84, permitiendo identificar con buena precisión a los estudiantes con mayor probabilidad de reprobar. Se generaron perfiles de riesgo individualizados y sugerencias con intervención basadas en resultados, lo que representa una herramienta útil para la planificación pedagógica y la toma de decisiones oportunas por parte de docentes y autoridades educativas.

Como conclusión, este proyecto demuestra que es factible aplicar técnicas avanzadas de ciencia de datos en entornos educativos, aprovechando datos existentes y herramientas de código abierto. La implementación de modelos predictivos puede transformar la gestión educativa, permitiendo la detección temprana de estudiantes en riesgo y facilitando estrategias de apoyo personalizadas. Además, se sienta un precedente para replicar esta iniciativa en otras unidades educativas del país, contribuyendo a la mejora continua del sistema educativo boliviano desde una perspectiva basada en evidencias.

Palabras clave

Machine learning, analítica de datos, predicción de reprobados, Random forest, CatBoost

Dedicatoria en Texto Garamond en cursiva tamaño 11.5, interlineado múltiple 1.2 con espaciado anterior y posterior de 6 puntos.

Se sugiere un máximo de 2 párrafos breves.

Agradecimientos

Quiero comenzar expresando mi gratitud a Dios, por haberme sostenido con su fuerza y luz durante todo este camino. En los momentos de duda, cansancio o incertidumbre, siempre encontré en Él una razón para seguir adelante, y por eso le dedico este logro con humildad y profundo agradecimiento.

A mi madre, por ser mi pilar más firme. Su sacrificio, constancia y amor incondicional me acompañaron en cada paso. Gracias por creer en mi cuando yo mismo dudaba, y por enseñarme que el verdadero éxito se construye con trabajo, paciencia y corazón.

A mi tutor, el M.Sc. Ing. Ariel Mamani Tito, por su orientación precisa, sus observaciones oportunas y su apoyo constante. Su acompañamiento fue fundamental para mantener la dirección académica de este trabajo, y agradezco sinceramente su compromiso con su ayuda incondicional.

A la profesora Patricia Maita, por brindarme todo su apoyo, la información necesaria y confiar en el propósito de este proyecto. Su apertura y colaboración hicieron posible que esta investigación se ancle a una realidad concreta y significativa para nuestro municipio.

Este trabajo no es solo un producto académico; es también el reflejo del apoyo, la fe y el acompañamiento de todas las personas que, de una forma u otra, han estado presentes en este proceso. A todos ellos, mil gracias de corazón.

Tabla de contenidos

UNIVERSIDAD MAYOR DE SAN SIMÓN 1

1. Introducción 1

1.1. Antecedentes 1

1.3. Justificación 2

1.4. Planteamiento del problema 2

1.5. Objetivo general 3

1.5.1. Objetivos específicos 3

1.6. Alcance 4

1.7. Limitaciones 4

2. Marco teórico 5

2.1. Conceptualización del rendimiento académico 5

2.2. Bajo rendimiento académico 5

2.3. Factores que influyen en el bajo rendimiento académico 5

2.4. Contexto educativo en Bolivia 6

2.5. Aprendizaje automático 6

2.6. Modelos predictivos aplicados en la educación 8

2.7. Aprendizaje supervisado 8

2.7.1. Regresión logística 8

2.7.2. XGBoost (Extreme Gradient Boosting) 9

2.7.3. Gradient Boosting 10

2.7.4. MLP (Multilayer Perceptron) 11

2.7.5. LightGBM (Light Gradient Boosting Machine) 11

2.7.6. CatBoost 12

2.7.7. Random forest 13

2.7.8. Máquinas de vectores de soporte (SVM) 14

2.8. Jupyter notebook 15

2.9. Python 15

2.10. Librerías de Python 15

2.10.1. Numpy 15
2.10.2. Pandas 15
2.10.3. Matplotlib 16
2.10.4. Scikit-learn 16
2.2. Mejor modelo de machine learning 16
2.2.1. F1-Score 16
2.2.2. F1-Score ponderado 16
3. Marco metodológico 18
3.1. Área de estudio 18
3.2. Flujograma metodológico 19
3.3. Fuentes de información 21
3.3.1. Fuentes de información secundaria 21
3.4. Adquirir datos de calificaciones de los estudiantes con datos actualizados. 23
3.5. Integración y limpieza de datos 23
3.5.1. Contando la cantidad de filas y columnas 25
3.5.2. Verificación de valores nulos 25
3.5.3. Borrando valores duplicados 25
3.5.4. Tratamiento de columnas categóricas y numéricas 26
3.5.5. Tratamiento de Numero CI 26
3.5.6. Cambio de tipo de dato a Fecha de nacimiento 26
3.6. Preparación de datos para análisis 27
3.6.1. Separando datos personales y calificaciones 27
3.6.2. Guardado de datos personales y calificaciones en archivos separados 27
3.6.3. Quitando a estudiantes con promedio = 0 27
3.6.4. Convirtiendo la gestión a tipo de dato entero 28
3.6.5. Examinar la evolución académica del estudiante 28
3.6.6. Identificación de las materias que representan mayor dificultad 28
3.7. Análisis exploratorio 29
3.7.1. Contando la cantidad de reprobados por año 29
3.7.2. Mapa de calor de correlaciones 29
3.7.3. Estadísticas descriptivas 30

3.7.4. Análisis con gráficos 31
3.8. Selección de características 32
3.8.1. Creación de características avanzadas 32
3.9. Modelado predictivo 33
3.9.1. Tabla minable 33
3.9.2. Importación de librerías 33
3.9.3. Creación de logging 34
3.9.4. Preprocesamiento de datos 34
3.9.5. División de datos por año 35
3.9.6. Preparación de datos para el modelado 36
3.10. Entrenamiento de modelos 37
3.10.1. Entrenar y evaluar múltiples modelos 37
3.10.2. Proyectar tendencias futuras 38
3.10.3. Identificando estudiantes en riesgo 39
3.10.4. Perfiles de intervención 39
3.10.5. Identificar factores de riesgo 40
3.10.6. Guardado de resultados 41
3.11. Evaluación y comunicación 41
3.12. Selección de herramientas 41
3.12.1. Jupyter notebook 41
3.12.2. Tableau public 41
4. Análisis de Resultados y Discusión 42
4.1. Resultados y análisis de Recolección y consolidación de datos académicos 42
4.2. Resultado y análisis de Organizar y limpiar datos de las calificaciones 43
4.3. Resultado y análisis de Examinar la evolución del rendimiento académico de los estudiantes 43
4.4. Resultado y análisis de Identificar las materias que representan mayor dificultad para estudiantes 48
4.5. Resultado y análisis de Construir un modelo con el fin de predecir e indicar a los estudiantes con mayor probabilidad de reprobar 50
5. Conclusiones 55
6. Recomendaciones 56
Referencias bibliográficas 57

Bibliografía 57

Anexos 61

Anexo 1. Fuente de datos original 61

Anexo 2. Código fuente 62

Anexo 3. Proyección de cantidad de estudiantes para los siguientes años 62

Anexo 4. Estudio de Hábitos de estudio 66

Anexo 5. Contenido del CD 67

Lista de figuras

Figura 3.1-1: Ubicación Geográfica de Bolivia 19
Figura 3.2-1: Flujograma metodológico 20
Figura 3.4-1: Muestra de archivos en formato Excel 23
Figura 3.5-1: Merge de los archivos 24
Figura 3.5-2: Juntando los archivos en jupyter 24
Figura 3.5-3: Importar el dataset completo 25
Figura 3.5-4: Importar el dataset completo 25
Figura 3.5-5: Verificando valores nulos 25
Figura 3.5-6: Borrado de filas vacías y duplicados 25
Figura 3.5-7: Reemplazando datos vacíos 26
Figura 3.5-8: Reemplazo de datos en Numero CI 26
Figura 3.5-9: Cambio de tipo de datos de Object a datetime 26
Figura 3.6-1: Ordenamiento de datos 27
Figura 3.6-2: Guardado de datos separados en archivos Excel 27
Figura 3.6-3: Descartando estudiantes con promedio=0 para la parte predictiva 27
Figura 3.6-4: Cambio de tipo de dato a la columna Gestion 28
Figura 3.6-5: Formula para obtener el nombre completo del estudiante 28
Figura 3.6-6: Haciendo pívot de las materias y notas 29
Figura 3.7-1: Explorando la cantidad de reprobados por gestión 29
Figura 3.7-2: Coeficiente de correlación de Pearson 30
Figura 3.7-3: Estadísticas descriptivas 31
Figura 3.7-4: Análisis univariado 31
Figura 3.7-5: Análisis bivariado 32
Figura 3.8-1: Creación de características avanzadas 33
Figura 3.9-1: Creación de la tabla minable 33
Figura 3.9-2: Instalación necesaria de las librerías 34
Figura 3.9-3: Importación de las librerías necesarias 34
Figura 3.9-4: Configuración del logging 34
Figura 3.9-5: Preprocesamiento de datos 35
Figura 3.9-6: División de datos 35
Figura 3.9-7: Preparación de datos para el modelado 36
Figura 3.10-1: Modelos seleccionados con sus parámetros 37
Figura 3.10-2: Entrenamiento implícito 38
Figura 3.10-3: Entrenamiento implícito 38
Figura 3.10-4: Predicción de reprobados 39
Figura 3.10-5: Probabilidades continuas 39
Figura 3.10-6: Factores de riesgo 40

Figura 3.11-1: Factores de riesgo 41
Figura 4.1-1: Cantidad de estudiantes por nivel y gestión educativa 42
Figura 4.2-1: Valores nulos por variables antes del procesamiento 43
Figura 4.3-1: Porcentaje de reprobados por año 44
Figura 4.3-2: Porcentaje total de reprobados 45
Figura 4.3-3: Distribución del resultado académico final de los estudiantes 45
Figura 4.3-4: Promedio por nivel educativo y gestión 46
Figura 4.3-5: Promedio general por año 46
Figura 4.3-6: Evolución del/la estudiante por año 47
Figura 4.3-7: Distribución del promedio de los estudiantes 47
Figura 4.4-1: Dificultad de materias de primaria según el promedio por año 48
Figura 4.4-2: Dificultad de materias de secundaria según el promedio por año 49
Figura 4.4-3: Materias con tasa de reprobación 50
Figura 4.5-1: Matriz de confusión del mejor modelo 52
Figura 4.5-2: Gráfico de estudiantes reprobados + predicción 53
Figura 4.5-3: Predicción de estudiantes con mayor riesgo de reprobación 53
Figura 4.5-4: Estudiantes con riesgo de reprobación 54
Figura 1-1: Contenido de los archivos originales 61
Figura 2-1: Código del proyecto 62
Figura 3-1: Selección de las comunas gestión y Codigo Rude 63
Figura 3-2: Dividiendo los datos en entrenamiento y prueba 63
Figura 3-3: Entrenamiento de los modelos seleccionados 64
Figura 3-4: Gráfico del histórico y las predicciones de los modelos 65
Figura 3-5: Grafico de cantidad de estudiantes hasta el año 2027 66
Figura 4-1: Gráfico de hábitos de estudio 66

Lista de tablas

Tabla 4.3-1: Variables/atributos de mayor influencia organizada por frecuencia de aparición 48
Tabla 4.5-1: Resultado del entrenamiento por modelo 51
Tabla 4.5-2: Resultado del entrenamiento de los modelos 51
Tabla 4.5-3: Matriz de confusión de Márquez Vera 52
Tabla 3-1: Métricas de los modelos para predecir la cantidad de inscritos en los próximos años 64
Tabla 3-2: Predicciones de los modelos usados 65

1. Introducción

La educación es un pilar fundamental para cualquier sociedad en el mundo, en un contexto nacional, las llamadas Unidades Educativas juegan un papel muy importante al proporcionar educación primaria y secundaria a nuestros niños y adolescentes. Las mismas van sentando las bases para una formación académica y profesional futura, sin embargo, la realidad en Bolivia presenta contrastes significativos, los estudios tienden a enfocarse más en entornos urbanos, dejando de lado las realidades que presentan los entornos rurales y provinciales del territorio nacional.

La calidad educativa es un factor importante en el desarrollo de un país. ("Reflexiones sobre la calidad educativa en las Instituciones de ...") El año 2021, Bolivia ocupaba el puesto 13 de 16 en América Latina en cuanto a la calidad educativa de la región (tiempos, 2021), esto hace evidente que se debe analizar, fortalecer y mejorar el sistema educativo. Es importante comprender que la base de la educación superior es la educación regular (Primaria y secundaria), si no se tienen sólidos conocimientos en esta etapa, es probable que se tengan problemas en el futuro.

Este estudio se basa en un análisis histórico de notas académicas de la unidad educativa San José Obrero desde la gestión 2015 a 2024, con lo cual se pretende identificar las materias en las que los estudiantes del nivel primario y secundario presentan mayor dificultad. Además, se busca desarrollar un modelo predictivo capaz de anticipar el bajo rendimiento académico, permitiendo así la implementación de estrategias para mejorar las falencias con el fin de brindar apoyo oportuno a estudiantes en riesgo. La identificación temprana de estudiantes con dificultades académicas es crucial para implementar intervenciones efectivas que mejoren su desempeño y reduzcan las tasas de deserción escolar.

El análisis a partir de datos históricos permitirá comprender mejor las tendencias y patrones de rendimiento académico de las distintas materias y cursos. A su vez, proporcionará información valiosa para diseñar programas de apoyo a estudiantes en situaciones desfavorables.

1.1. Antecedentes

En el mundo la educación es un pilar fundamental para el desarrollo de cualquier sociedad, Bolivia no es la excepción. El ministerio de educación mediante diversas maneras a implementado varios programas para mejorar el sistema educativo, se tiene la convicción de que la educación nos prepara para el futuro y ser más competitivos (HQEDGAR, 2023).

Las calificaciones por su parte son muy importantes, ya que mediante estas se indican las habilidades y áreas de interés de los estudiantes (Kichiuhua, 2024), por lo general en los centros educativos se destacan los promedios de los tres mejores estudiantes, ya sea como unidad educativa o curso, se deja de lado la

preocupación de los estudiantes que no logran buenas notas o los que reprueban. Entre el año 2006 – 2019, la tasa de abandono disminuyó de 5.51% a 3.82%, la tasa de reprobación por su parte tenía una proporción de 7.02% de reprobados para ese año, lo preocupante ocurre en el departamento de Potosí, con una alta tasa de reprobación (Laime, 2024)

Bolivia curso una etapa de digitalización de libretas escolares en el año 2015 (Ministerio de Educación del Estado plurinacional de Bolivia, 2015), esto nos indica que todo nuestro histórico académico desde entonces ya es 100% digital y que pueden ser usados para realizar estudios en esta área.

Con el paso del tiempo el crecimiento de y generación de datos en los últimos años ha sido sorprendente (Poulova & Mikulecká, 2019), desde luego incluyendo a los datos generados en el ámbito educativo. Con la digitalización, llegada de programas informáticos y dispositivo electrónicos de diversos tipos que generan millones de datos. Desde luego para analizar estos datos se debe realizar la limpieza y codificación, para posteriormente realizar la analítica de manera correcta o emplear algoritmos de machine learning y comparar métricas (Castillo Aráuz & Martínez, 2023).

Para realizar este tipo de analítica y algoritmos, lo que se debe tener necesariamente son: datos personales, estado de calificación, cantidad de materias reprobadas y calificación como tal (Ramírez & Páez, 2024).

1.3. Justificación

El desempeño y aprovechamiento educativo son factores importantes para evaluar la calidad educativa en una institución del mismo tipo. El análisis del desempeño del estudiante permite identificar problemas que pueden estar afectando el aprendizaje, en consecuencia, dificultando la continuidad en su formación escolar o superior.

Comprender las causas y consecuencias del bajo rendimiento académico son esenciales para diseñar estrategias de apoyo que beneficien a los estudiantes con dificultades y fortalezcan el sistema educativo en las zonas rurales de Bolivia.

La importancia del análisis de datos radica en extraer conocimiento, comprender el pasado y predecir el futuro mediante modelos de machine learning. Al analizar datos desde un punto de vista estadístico podemos hallar correlaciones y sobre estos realizar gráficos interesantes que reflejan la realidad de lo que se tiene hoy en día. El empleo de algoritmos supervisados y no supervisados ayudaran a la identificación de patrones y predicciones relacionadas con el rendimiento académico.

1.4. Planteamiento del problema

La educación es fundamental en cualquier sociedad del mundo, ya que esta contribuye de forma invaluable al progreso y la mejora de cualquier entorno. En los últimos años, con el cambio de leyes y artículos en Bolivia, se lograron resultados de diversa índole según estudios realizados por expertos. Entre estos podemos mencionar algunos:

El desempeño de Bolivia en evaluaciones internacionales, como el ERCE 2019, indica que los estudiantes tienen un bajo rendimiento en lenguaje, matemáticas y ciencias en comparación con los promedios regionales (Gutierrez, 2022)

En un estudio presentado a principios de 2025 (en la prueba OPCE), solo el 3% de los estudiantes aprobaron la prueba de Química. Mientras la tasa de reprobados varía según el tipo de institución: en los colegios públicos, el 93.5% reprueba, en los colegios de convenio, el 90% no supera la prueba; y en los colegios privados, el 81.5% queda aplazado. Por territorio, el 90.5% de los estudiantes urbanos reprueban, mientras que en las áreas rurales. El 93.2% no aprueba (Amonzabel, 2025)

Analizando estos estudios, vemos que, al parecer los estudiantes en etapa escolar, tienden a tener más probabilidad de reprobar en áreas relacionadas con las áreas científicas, en especial, vemos mucha más negatividad en la educación que se imparte en las zonas rurales, los factores relacionados pueden ser diversos, la falta de equipamiento, infraestructura, poco acceso a la tecnología entre otros.

La identificación de estudiantes en peligro de reprobar materias en muchos casos es tardía, en este contexto, tener históricos de las calificaciones de los estudiantes se convierte en un recurso invaluable, ya que esto puede ayudar de manera activa a la toma de decisiones informada de manera anticipada y actuar en una etapa temprana para posteriormente llevar a cabo acciones concretas para prevenir que los estudiantes puedan reprobar. Es crucial realizar un análisis de las materias partiendo de los históricos de calificaciones (2015-2023), esto con el objetivo de identificar patrones de bajo rendimiento para luego desarrollar un modelo predictivo que permita la detección temprana de estudiantes con riesgo de aplazo. Los resultados de este modelo predictivo permitirán la intervención oportuna dentro de las unidades educativas, esto mismo podría servir para implementar nuevas políticas más efectivas a nivel nacional y una mejora en la calidad educativa.

1.5. Objetivo general

Analizar el rendimiento académico de los estudiantes de provincias y zonas alejadas de Bolivia, identificando factores relacionados al bajo desempeño y desarrollando un modelo de predicción basado en técnicas de machine learning para anticipar casos de estudiantes con riesgo de reprobación.

1.5.1. Objetivos específicos

- Recolectar y consolidar datos históricos de calificaciones en la Unidad Educativa San José Obrero, con el fin de establecer una base de datos

- Aplicar técnicas de limpieza y preprocesamiento de datos para garantizar su calidad y adecuación al análisis estadístico y al desarrollo de modelos de machine learning.

- Analizar la evolución del rendimiento académico de los estudiantes a lo largo del tiempo, identificando patrones, tendencias y variaciones significativas por nivel educativo.

- Determinar las materias con mayor índice de reprobación y explorar la relación del rendimiento académico con variables como el curso, el género y la gestión.

- Diseñar y evaluar modelos predictivos que identifiquen estudiantes con alta probabilidad de reprobar, utilizando algoritmos de aprendizaje supervisado y validando su aplicabilidad al contexto educativo local.

1.6. Alcance

Este proyecto tiene por objetivo estudiar el fenómeno del bajo rendimiento académico en la Unidad Educativa San José Obrero, ubicada en una comunidad rural de Bolivia. Se plantea trabajar con los registros escolares disponibles desde la gestión 2015 hasta la gestión 2024, centrando la atención en el estudio de los estudiantes de nivel primario y secundario.

El trabajo abarcará la recopilación , organización y análisis de datos académicos, con énfasis en detectar materias con mayores cantidades de reprobados y estudiantes que presenten patrones de bajo desempeño a lo largo de los años. Asimismo, se contempla la exploración inicial de técnicas de análisis de datos y herramientas de programación para sentar las bases de un modelo que, en una etapa posterior, permita anticipar casos de riesgo académico.

Todo el desarrollo se realizará con software libre, empleando entornos de trabajo adecuados para proyectos de ciencia de datos, y se buscará que los resultados obtenidos puedan ser usados por la unidad educativa, haciendo así que las intervenciones y la mejora dentro de la enseñanza de la unidad educativa mejoren.

1.7. Limitaciones

Se reconocen algunas limitaciones que podrían influir en el desarrollo y alcance de los resultados. En primer lugar, el análisis se enfocará únicamente en una unidad educativa especifica, lo cual restringe la posibilidad de generalizar las conclusiones a otras instituciones con contextos y entornos diferentes.

Además, el trabajo se limitará a los datos disponibles en los archivos escolares que contienen datos personales y calificaciones, estos en algunos casos podrían presentar errores al igual, valores ausentes o inconsistencias debido al cambio de materias que surgió la educación regular en el territorio nacional. Tampoco se contará con información detallada sobre los aspectos personales o familiares de los estudiantes, como situación económica o condiciones emocionales, factores que también inciden en el rendimiento académico, pero que no forman parte de los registros de la unidad educativa en cuestión.

Por último, dado que el enfoque principal es exploratorio y de diseño, no se contempla aun la aplicación directa de los resultados en decisiones pedagógicas ni su validación con intervenciones reales, aspectos que podrían abordarse en futuros trabajos.

2. Marco teórico

2.1. Conceptualización del rendimiento académico

El rendimiento académico es un indicador clave en todo proceso educativo, este mide el nivel de conocimiento adquirido por el estudiante en relación con la materia y sus objetivos establecidos (González, 2020). El rendimiento académico puede ser alto o bajo según el aprovechamiento del estudiante, desde luego, a esto van ligados valores y la habilidad que ellos demuestran (Navarro, 2003).

El aprendizaje es un proceso constructivo en el que individuos organizan y absorben información de manera activa (Vygotsky, 1978), es importante enfatizar la importancia del entorno social y la construcción del conocimiento de cada niño, joven o adolescente. No solo el estudiante es responsable de su rendimiento académico, también influyen factores externos como el contexto educativo y la metodología de enseñanza del profesor, por ejemplo.

En Bolivia, el rendimiento académico actual es objeto de muchos estudios, esto debido a las múltiples diferencias de desempeño en los estudiantes de zonas urbanas y rurales. Bolivia se ubica entre los últimos lugares en el ranking de calidad educativa (UNESCO, 2021), esto hace evidente que se debe analizar las condiciones de aprendizaje y su entorno real.

2.2. Bajo rendimiento académico

El bajo nivel académico hace referencia al desempeño por debajo del nivel deseado o esperado en entornos educativos. En Bolivia, se usa la escala calificativa de 0 – 100, para aprobar el estudiante debe obtener una calificación mayor o igual a 51, en cualquier otro caso se considera que el estudiante tiene bajo nivel académico (Cornejo, 2022).

2.3. Factores que influyen en el bajo rendimiento académico

Desde luego, el bajo rendimiento académico es resultado de múltiples factores que pueden agruparse en tres categorías principales:

- Factores institucionales

El entorno social y educativo es crucial para el rendimiento del estudiante. La infraestructura inadecuada, la escasez de docentes capacitados y la falta de materiales didácticos son algunos aspectos que afectan negativamente el aprendizaje (Fromero, 2021), en colegios alejados de ciudades grandes dichos factores aun nos más evidentes debido a la falta de inversión y apoyo económico.

- Factores socioeconómicos

El nivel socioeconómico de los estudiantes y cada una de sus familias tiene impacto directo con el desempeño escolar. Estudios señalan que los estudiantes provenientes de hogares con bajos recursos económico tienen menos acceso a recursos educativos, valga decir: libros, acceso a internet y computador. Muchos de estos estudiantes presentan más dificultad de aprendizaje y alta tasa de deserción encolar, la falta de apoyo familiar, falta de apoyo económico y las condiciones de vida precarias pueden generar la falta de motivación el estrés (CEPAL, 2020).

- Factores pedagógicos y psicológicos

La metodología de enseñanza aplicada por el profesor y la motivación también influyen en el buen o mal rendimiento académico, el usar materiales didácticos puede facilitar la compresión del contenido y mejorar el desempeño en la etapa escolar. Problemas emocionales como la ansiedad, baja autoestima y depresión también pueden influir negativamente en el mal rendimiento (Vygotsky, 1978).

2.4. Contexto educativo en Bolivia

Si bien es cierto, la educación pública en Bolivia es gratuita, está compuesta por instituciones educativas fiscales, instituciones educativas privadas y de convenio (Asamblea legislativa plurinacional, 2010), la educación es obligatoria hasta el bachillerato y en un contexto fiscal es gratuita en todos los niveles, incluyendo la educación superior.

Existe una clara brecha significativa en el sistema educativo boliviano entre áreas rurales y urbanas, tanto en infraestructura, apoyo económico, incentivos y acceso a una educación de calidad. Estos son algunas particularidades para que el analfabetismo rural haya permanecido alto, sumado a esto solo el 23% del presupuesto anual es destinado a la educación (Wikipedia, 2025) de forma general.

Para el año 2015, 1 de cada 7 niños no concluía la educación primaria, sumado a esto la resistencia de los sindicatos de docentes ha ralentizado la implementación de una educación intercultural bilingüe y la descentralización de la financiación (Binns, 2015), es cierto que hoy en día se implementaron varios cambios significativos, pero la calidad educativa ha sido decreciente con el pasar del tiempo.

2.5. Aprendizaje automático

Es un subconjunto de la inteligencia artificial, es mayor mente usado para realizar predicciones y tomar decisiones a partir de datos de entrenamiento. Dentro del mundo de ML se tienen muchas herramientas, tales como Python, Azure, Google cloud y muchas otras. En el contexto del ML los datos son algo muy importantes para que posteriormente la maquina aprenda y pueda ayudarnos a tomar decisiones futuras partiendo de los llamados dataset. El aprendizaje automático es muy ligado a las matemáticas ya que sus modelos tienen una fuerte relación con la misma (Bravo, Bermudez, & Cardona, 2021)., con el auge de la tecnología, no es de extrañar que machine learning haya tenido avances muy significativos.

A su vez, ML está centrado en los algoritmos computacionales especializados diseñados para emular la inteligencia humana, esta rama de la inteligencia artificial se ha aplicado a diversos campos, desde el reconocimiento de tendencias y patrones, visión artificial, finanzas, ventas, predicciones en las bolsas de valores, área deportiva por sitar algunas (Bravo, Bermudez, & Cardona, 2021).

Los datos históricos son muy interesantes a la hora de aplicar aprendizaje automático, ya que podemos partir analizando los datos de entrada para luego mediante modelos y algoritmos de ML producir datos de salida. Esta subdivisión de la IA no solo aprende mediante datos, sino que también se adapta a diversas situaciones que cambian de forma dinámica.

En cuando a su uso en la educación, debemos mencionar que aún se enfrenta diversos desafíos, entre estos resalta la poca o inexistente información escolar de estudiantes en países en desarrollo (Bravo, Bermudez, & Cardona, 2021); Los beneficios del uso de machine learning son muchos, entre estos podemos destacar los siguientes:

- Mejora la toma de decisiones

La información influye bastante en la toma de decisiones, esto debido a que se tiene datos sobre intuición (SAP Concur, 2021).

- Automatiza procesos

Con ML se pueden automatizar procesos y actividades que son repetitivos y complejos como ser: envío y respuestas de correo electrónico, toma de decisiones.

Es importante comprender que machine learning son capaces de adaptarse en tiempo real analizando grandes volúmenes de información para luego proponer ajustes y anticipar problemas cuando es empleada en líneas de producciones industriales (Improvitz, 2024).

- Incrementa la productividad

El aprendizaje automático es muy bueno para verificar la calidad del trabajo e identificar fallas productivas, errores en los procesos y fallas que se pueden evitar con entrenamiento o retroalimentación (SAP Concur, 2021).

- Prevención contra ciberataques

ML aprende de históricos, es decir, detecta actividades sospechosas, esto es provechoso ya que contribuye a la mitigación de ataques informáticos (SAP Concur, 2021).

2.6. Modelos predictivos aplicados en la educación

El aprendizaje automático ha demostrado ser una herramienta que es bastante efectiva para el uso de predicción en el rendimiento académico (López, 2022). Se destacan que la inteligencia artificial puede analizar grandes volúmenes de datos educativos con el fin de identificar los patrones y anticipar las materias con dificultades de aprendizaje. Entre los modelos más usado tenemos: Regresión logística, árbol de decisión y redes neuronales artificiales, podemos usar otros modelos además de estos, depende mucho del caso en el que nos encontremos.

2.7. Aprendizaje supervisado

El aprendizaje automático es una subcategoría del machine learning y la inteligencia artificial, su particularidad es que usa conjunto de datos que tienen etiquetas para entrenar algoritmos, estos clasifican los datos y predicen los resultados con mayor precisión (Ibm, 2025).

Algo primario que debemos saber es que el aprendizaje automático se basa en datos que incluyen tanto entradas como salidas correctas, esto mismo le enseña al modelo como llegar al resultado esperado. A medida que el modelo trabaja con estos datos, ajusta sus cálculos para reducir los errores y mejorar su precisión, guiándose por una función que mide qué tan bien está funcionando (Ibm, 2025)

Entre los modelos predictivos podemos identificar los modelos de clasificación y los modelos de regresión (Duc, Leiva, Casari, & Östberg, 2019).

- Los modelos de clasificación usan algoritmos para asignar con mayor precisión sus datos de prueba en categorías específicas (Duc, Leiva, Casari, & Östberg, 2019), es decir que son valores categóricos.
- Los modelos de regresión a diferencia de los modelos de clasificación usan valores del tipo continuo a diferencia de la clasificación que usa valores categóricos (Russo, 2019).

2.7.1. Regresión logística

La regresión logística es un modelo estadístico utilizado para predecir la probabilidad de un evento binario, un clásico ejemplo es: si un estudiante aprueba o no aprueba, desde luego esto a partir de variables independientes. Su ecuación matemática central transforma las probabilidades en una escala logarítmica mediante una función (Ibm, 2025):

$$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k$$

Esta función representa el modelo de regresión logística, utilizado para predecir la probabilidad de que ocurra un evento binario, como que un estudiante repruebe o no. En este contexto, p es la probabilidad de que el evento ocurra, y la expresión $\left(\frac{p}{1-p}\right)$ representa la razón de probabilidad de que

ocurra frente a que no ocurra. Al aplicar el algoritmo natural a estos, se obtiene el llamado logit, que es modelado como una combinación lineal de variables independientes como la cantidad de faltas, promedio anterior, entre otros.

Los coeficientes $\beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k$ indican la influencia de cada variable sobre el logit de la probabilidad.

Y mediante su forma de vector de predicciones:

$$\hat{y} = \frac{1}{1 + \exp(-z)}$$

En esta ecuación de la regresión logística, la variable $\hat{y}$ es una variable que depende de una respuesta, mientras que la variable -z es independiente (Ibm, 2025).

La naturalidad del algoritmo de regresión logística es predecir clases binarias, es decir dicotómicas con dos clases posibles, es usado por ejemplo para calcular la probabilidad de que ocurra un evento. Entre sus ventajas están que es fácil de usar y es base para cualquier problema de clasificación binaria (Ibm, 2025).

2.7.2. XGBoost (Extreme Gradient Boosting)

XGBoost es una técnica avanzada de aprendizaje automático basada en árboles de decisión, que ha ganado mucha popularidad en los últimos años debido a su alto rendimiento en competiciones y aplicaciones prácticas. Su nombre proviene de "Extreme Gradient Boosting" y representa una evolución optimizada del algoritmo tradicional de Gradient Boosting.

Una de sus principales fortalezas radica en su capacidad para construir modelos predictivos robustos mediante la suma secuencial de árboles que intentan corregir los errores de los árboles anteriores. Este enfoque permite que XGBoost logre una alta precisión sin perder eficiencia, ya que implementa técnicas como el paralelismo en el entrenamiento, el uso de estructuras de datos optimizadas y la regularización explícita.

La función objetivo que minimiza XGBoost puede expresarse como:

$$\mathcal{L}(\emptyset) = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t)}\right) + \sum_{k=1}^{t} \Omega(f_k)$$

Representa la función objetivo de modelos de aprendizaje ensamble como XGBoost. Esta función combina dos componentes: el error de predicción y la complejidad del modelo. El primer término, $\sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t)}\right)$, suma las pérdidas individuales entre los valores reales y_i y las predicciones del modelo

$\hat{y}_i^{(t)}$ para cada observación. Esta pérdida puede ser, por ejemplo, el error cuadrático en regresión o la log-loss en clasificación.

El segundo término $\sum_{k=1}^{t} \Omega(f_k)$, agrega una penalización por la complejidad de cada uno de los árboles f_k usados hasta la iteración t. Esta penalización busca evitar el sobreajuste, es decir, que el modelo aprenda demasiado bien los datos de entrenamiento y no generalice bien a nuevos datos. En conjunto, esta función objetivo balancea la precisión del modelo por su simplicidad, lo que permite construir predictores más robustos y eficientes.

XGBoost también maneja automáticamente los valores faltantes, permite realizar validación cruzada interna, y se adapta muy bien a datos heterogéneos (Chen & Guestrin, 2016).

2.7.3. Gradient Boosting

El método de Gradient Boosting es un algoritmo de aprendizaje supervisado que se basa en el concepto de ensamblado de modelos débiles —habitualmente árboles de decisión— para crear un modelo más fuerte y preciso. Su principio central consiste en construir el modelo de forma secuencial, de manera que cada nuevo modelo minimiza los errores cometidos por la suma de los modelos anteriores.

Este proceso se apoya en la optimización de una función de pérdida, utilizando técnicas similares al descenso por gradiente. Es decir, se calcula el gradiente de la función de error respecto a las predicciones, y ese gradiente se usa para ajustar el siguiente modelo.

El modelo predictivo final se define como:

$$F(x) = \sum_{m=1}^{M} \gamma_m \, h_m(x)$$

Representa la predicción final de un modelo de ensamble aditivo, como el boosting. En este contexto, el modelo completo $F_{(x)}$ se construye como una suma de modelos más simples o débiles, denotados por $h_m(x)$, que suelen ser árboles de decisión pequeños. Cada uno de estos modelos débiles contribuye con una parte de la predicción, ponderada por un coeficiente, que determina su influencia en el resultado final.

La idea principal es que cada nuevo modelo $h_m(x)$ se entrena para corregir los errores cometidos por la suma de modelos anteriores. Así, el modelo se va refinando paso a paso, y la predicción final es la acumulación ponderada de todos estos ajustes. Este enfoque permite construir modelos potentes a partir de componentes sencillos, logrando un alto rendimiento tanto en tareas de clasificación como de regresión.

Aunque su desempeño es alto, uno de sus retos principales es el tiempo de entrenamiento y la sensibilidad al sobreajuste, si no se aplican estrategias como la poda, regularización o ajuste de hiperparámetros (Friedman, 2001).

2.7.4. MLP (Multilayer Perceptron)

El Multilayer Perceptron o perceptrón multicapa, es una arquitectura fundamental dentro del campo de las redes neuronales artificiales. Está compuesto por múltiples capas de neuronas interconectadas: una capa de entrada, una o más capas ocultas, y una capa de salida. ("Perceptrón multicapa: definición, entrenamiento y aplicaciones") Cada neurona realiza una combinación lineal de sus entradas, seguida por una función de activación no lineal. ("Perceptrón Multicapas + NumPy")
El modelo aprende a través del proceso de retro propagación del error, ajustando los pesos de conexión para minimizar la diferencia entre la salida obtenida y la esperada. La función típica utilizada para ajustar los pesos es el descenso del gradiente.
La operación que realiza una neurona se expresa así:

$$z = \sum_{i=1}^{n} \omega_i \, x_i + b, \qquad a = \sigma(z)$$

La primera ecuación establece un proceso donde diversas señales de entrada (X_i) son consideradas según su relevancia ponderada (ωi), integrándose en un valor (z) al que se suma una constante de ajuste (b). Este valor intermedio (z) es posteriormente modificado por una función específica (σ) en la segunda ecuación, generando así la respuesta final (a) de esta etapa del procesamiento.
El MLP es capaz de modelar relaciones no lineales complejas, por lo que es ampliamente utilizado en problemas de clasificación, regresión y reconocimiento de patrones. Su entrenamiento requiere cuidado con aspectos como la normalización de datos, el número de capas y neuronas, y el uso de regularización para evitar el sobreajuste (Rumelhart, Hinton, & Williams, 1986).

2.7.5. LightGBM (Light Gradient Boosting Machine)

LightGBM es una variante moderna del Gradient Boosting, desarrollada por Microsoft, que se ha diseñado específicamente para ser más rápida y eficiente, especialmente cuando se trata de grandes volúmenes de datos. A diferencia de otros métodos, LightGBM utiliza histogramas discretizados para dividir los datos y construir los árboles, lo que reduce significativamente el consumo de memoria y acelera el proceso de entrenamiento.
Una característica distintiva de LightGBM es su estrategia de crecimiento de árboles leaf-wise (por hojas), en lugar del enfoque tradicional level-wise (por niveles). Este método busca dividir siempre la hoja que produce la mayor reducción en la pérdida, lo que puede mejorar el rendimiento del modelo, aunque también aumenta el riesgo de sobreajuste.
La función objetivo es similar a la de XGBoost, pero con una implementación más ligera y escalable. También soporta entrenamiento paralelo, manejo de valores faltantes y técnicas avanzadas como el Gradient-based One-Side Sampling (GOSS) (Ke, y otros, 2017).
Su fórmula principal representa la predicción como la suma de múltiples funciones, cada una correspondiente a un árbol de decisión:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), \quad f_k \in F$$

Aquí, cada función f_k representa un árbol que pertenece al conjunto de funciones F, y la predicción final para una entrada x_i se obtiene al sumar la salida de todos los árboles construidos hasta el paso K. Esta estructura refleja el enfoque iterativo del boosting, donde cada árbol intenta mejorar el rendimiento corrigiendo los errores cometidos por el conjunto anterior.
La función objetivo que optimiza LightGBM combina dos partes: la función de pérdida y el término de regularización. Esta se expresa como:

$$\mathcal{L}(\emptyset) = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t)}\right) + \sum_{k=1}^{t} \Omega(f_k)$$

En esta expresión, $l\left(y_i, \hat{y}_i^{(t)}\right)$ mide la diferencia entre la predicción del modelo en la iteración t y el valor real de salida, mientras que el segundo término $\Omega(f_k$ penaliza la complejidad de cada árbol para evitar el sobreajuste. LightGBM mejora la eficiencia al usar un crecimiento de árbol basado en hojas (leaf-wise) en lugar del crecimiento por niveles, lo que le permite encontrar divisiones más precisas y profundas. Además, emplea histogramas discretos para acelerar la selección de divisiones, lo que lo convierte en una herramienta potente y escalable, especialmente útil en escenarios con grandes volúmenes de datos.

2.7.6. CatBoost

CatBoost es una herramienta moderna de aprendizaje automático basada en gradient boosting, creada por Yandex, con un enfoque especializado en el manejo de variables categóricas. A diferencia de otros algoritmos que requieren codificar explícitamente las variables (por ejemplo, con one-hot encoding), CatBoost las trata internamente mediante técnicas como target encoding con ordenamiento aleatorio, reduciendo así el riesgo de leakage o filtrado de información.

CatBoost también introduce un método innovador para reducir el sesgo acumulado durante el entrenamiento secuencial, lo que mejora su capacidad de generalización, especialmente en conjuntos de datos pequeños o desbalanceados.

Al igual que otros métodos de boosting, CatBoost entrena modelos de árboles secuencialmente para minimizar una función de pérdida, pero su diseño interno permite un entrenamiento más estable y menos propenso al sobreajuste, sin necesidad de ajustes manuales extensivos.

Es especialmente valorado en competiciones de ciencia de datos debido a su rendimiento con pocos datos de preprocesamiento y su robustez frente a ruido y valores atípicos (Prokhorenkova, Gusev, Vorobev, Dorogush, & Gulin, 2018).

Su fórmula de predicción general se expresa de forma similar a otros métodos de boosting, como:

$$\hat{y}_i = \sum_{m=1}^{M} f_m(x_i)$$

Donde $\hat{y}_i$ es la predicción final para el ejemplo x_i, y cada f_m representa un árbol de decisión entrenado en la iteración m. Estos árboles se suman de manera secuencial para corregir los errores de predicción cometidos por los modelos anteriores. A diferencia de otros enfoques, CatBoost incorpora un ordenamiento especial y técnicas de procesamiento de datos categóricos mediante estadísticas de combinaciones y target encoding controlado, lo que mejora la generalización del modelo y reduce el sobreajuste.

La función objetivo que utiliza CatBoost para ser optimizada tiene la forma:

$$\mathcal{L}(\emptyset) = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t)}\right) + \sum_{k=1}^{t} \Omega(f_k)$$

En esta ecuación, $l\left(y_i, \hat{y}_i^{(t)}\right)$ representa la función de pérdida que mide cuán cerca está la predicción $\hat{y}_i$ del valor real y_i y $\Omega(f_k)$ es un término de regularización que penaliza la complejidad de los árboles, favoreciendo modelos más simples que generalicen mejor. Una de las innovaciones clave de CatBoost es el uso del algoritmo llamado "Ordered Boosting", el cual evita el leakage de información en el entrenamiento, especialmente útil en datasets pequeños o con muchas categorías. Además, CatBoost trabaja internamente con procesamiento simétrico de árboles, lo que lo hace más rápido en predicción y lo distingue estructuralmente de otros métodos de boosting como XGBoost o LightGBM.

2.7.7. Random forest

Random forest usa arboles de decisiones no correlacionados, que posteriormente de fusionan para reducir la varianza y crear predicciones más precisas, este tiene fines de uso en clasificación y regresión (Ibm, 2025). Este algoritmo aborda el problema de sobreajuste, cada árbol de decisión que conforma el random forest es ligeramente diferente al resto y la idea de esto es que cada árbol de decisión puede hacer predicciones relativamente buenas, aunque es probable que algunos sean mejor que otros (Müller & Guido, 2017), luego de esto se toma una decisión basada en votos de todos los árboles (geeksforgeeks, 2025), ´por ejemplo: si la mayoría de los árboles dicen que pertenecen a una categoría especifica, esa será su respuesta final.

Su fórmula general de predicción para regresión puede expresarse como:

$$\hat{y} = \frac{1}{T}\sum_{t=1}^{T} f_t(x)$$

Donde $\hat{y}$ es el valor predicho para una entrada x, $f_t(x)$ representa el valor predicho por el árbol de decisión número t, y T es el número total de árboles en el bosque. En lugar de construir un solo árbol, Random Forest genera muchos árboles entrenados sobre diferentes subconjuntos aleatorios de los datos (tanto en filas como en columnas) mediante una técnica conocida como bagging (bootstrap aggregating). Cada árbol toma decisiones de forma independiente, y al final sus predicciones se promedian (en tareas de regresión) o se votan (en clasificación).

Una de las grandes ventajas del Random Forest es que reduce el riesgo de errores que pueden ocurrir cuando solo usamos un árbol de decisión. Esto pasa porque cada árbol se entrena con una parte aleatoria de los datos, lo que ayuda a que no todos los árboles cometan los mismos errores. Además, esta técnica maneja muy bien grandes cantidades de información y puede trabajar con diferentes tipos de datos sin necesidad de hacer muchos ajustes previos (Breiman, 2001).

Por último, aunque el Random Forest es bastante poderoso, no es perfecto. A veces puede ser un poco lento si trabajamos con conjuntos de datos enormes o si creamos demasiados árboles, ya que tiene que calcular muchas cosas al mismo tiempo. Pero, en general, es una herramienta confiable y fácil de usar para muchos proyectos de machine learning (Brownlee, 2021).

2.7.8. Máquinas de vectores de soporte (SVM)

Máquinas de vector de soporte fue desarrollado en la década de los 90, es un algoritmo de clasificación y regresión. En sus inicios era usado como un método de clasificación binaria, con el paso del tiempo su aplicación se extendió a problemas de clasificación múltiple y regresión (Kowalczyk, 2025), si conocemos las clases o etiquetas podemos usar SVM en aprendizaje supervisado (es bastante usado), aunque también puede ser usado en aprendizaje no supervisado; Este algoritmo clasifica con una línea recta, de esto mismo depende del modelo (Cortes & Vapnik, 1995), esta línea la trazamos en los puntos más cercana a ellas, se debe continuar con varias líneas hasta encontrar un margen perfectamente balanceado.

Su fórmula fundamental es:

$$f(x) = sign(\omega^T x + b)$$

Donde x es el vector de características de entrada, ω es el vector de pesos que define la dirección del hiperplano, y b es el sesgo o término independiente. El modelo busca maximizar el margen, es decir, la distancia entre el hiperplano y los vectores de soporte (los puntos de datos más cercanos a dicho hiperplano), lo que se traduce en un problema de optimización convexa con restricciones.

Existe un parámetro que se puede usar en SVM cuando la línea no clasifica bien la información, este parámetro lo llamaremos C, este reajusta el hiperplano para ajustar de la mejor manera, este parámetro es ajustable y se puede adecuar al mejor valor y clasificar mejor los datos (codificandobits, 2025).

2.8. Jupyter notebook

Jupyter notebook es una herramienta esencial a la hora de trabajar con machine learning y datos como tal, es un entorno interactivo que permite escribir código Python, este usa bibliotecas especializadas como: numpy, pandas y demás, aparte de esto documenta paso a paso el proceso y su ejecución en un solo archivo. Esto es muy importante y útil para proyectos de ciencia de datos e inteligencia artificial, donde no solo es esencial analizar datos sino también como se llegó a las conclusiones.

Al trabajar con jupyter podemos exportar los resultados en formatos como PDF o HTML, esto hace que sea más fácil compartir los resultados o agregar en informes finales, vale destacar que se tienen también Mark Down para hacer el documento más narrativo y elegante (Perez & Granger, 2008).

2.9. Python

Python es un lenguaje de programación muy popular de código abierto, es conocido por su simplicidad y flexibilidad. Este es bastante usado en áreas como ciencia de datos, machine learning y automatización debido a su sintaxis clara y la amplia comunidad de desarrolladores que posee. En el contexto de ML. Python sirve como base para implementar algoritmos de aprendizaje supervisado, como KNN o SVM, su gran capacidad de integrarse con librerías especializadas hace que Python sea indispensable para proyectos de análisis de datos (VanderPlas, 2016)

2.10. Librerías de Python

Las librerías de Python son extensiones que amplían las capacidades del lenguaje para tareas específicas. En este estudio, utilizamos varias bibliotecas clave que facilitan el procesamiento y análisis de datos educativos, algunas de estas se detallan a continuación.

2.10.1. Numpy

Esta biblioteca es esencial para realizar cálculos numéricos eficientes. Proporciona estructuras de datos como arreglos multidimensionales y funciones optimizadas para operaciones matemáticas. Por ejemplo, si queremos calcular estadísticas básicas sobre el rendimiento académico de los estudiantes, Numpy puede manejar grandes volúmenes de datos de manera rápida y precisa (Oliphant, 2006).

2.10.2. Pandas

Pandas es una biblioteca diseñada para el análisis y manipulación de datos tabulares. ("Matrices y análisis de datos en Python con Pandas | Rootstack") Permite cargar, limpiar y transformar datos de manera sencilla, lo que es crucial cuando trabajamos con información heterogénea, como datos demográficos, asistencia escolar o resultados académicos. Por ejemplo, podemos usar Pandas para filtrar estudiantes con bajo rendimiento y analizar sus características comunes (Mckinney, 2012).

2.10.3. Matplotlib

Para visualizar patrones y tendencias en los datos, Matplotlib es una biblioteca fundamental. Nos permite crear gráficos como histogramas, diagramas de dispersión y líneas de tendencia, que ayudan a interpretar los resultados de manera más intuitiva. Por ejemplo, podemos graficar la relación entre la distancia al colegio y el rendimiento académico para identificar correlaciones visuales (Hunter, 2007)

2.10.4. Scikit-learn

Esta biblioteca es específica para machine learning y proporciona implementaciones listas para usar de algoritmos como KNN, SVM y Random Forest. En nuestro estudio, Scikit-learn nos permite entrenar modelos predictivos para identificar estudiantes en riesgo de bajo rendimiento basándonos en variables como el nivel socioeconómico o la asistencia escolar (Pedregosa, Scikit-learn: Machine Learning in Python, 2011).

2.2. Mejor modelo de machine learning

En tareas de clasificación supervisada, elegir el mejor modelo no siempre es una tarea sencilla, especialmente cuando las clases dentro del conjunto de datos están desbalanceadas. En tales escenarios, métricas tradicionales como la exactitud (accuracy) pueden resultar engañosas, ya que un modelo puede obtener una alta puntuación simplemente acertando en la clase mayoritaria, ignorando completamente las minoritarias. Por esta razón, es fundamental utilizar métricas que reflejen de manera más justa el rendimiento global del modelo. Una de las métricas más utilizadas y adecuadas en estos casos es el F1-score ponderado (Chicco & Jurman, 2020).

2.2.1. F1-Score

El F1-score es una medida que combina dos conceptos clave: la precisión, que evalúa cuántas de las predicciones positivas realizadas por el modelo fueron correctas, y la exhaustividad o recall, que indica cuántos de los casos positivos reales fueron identificados correctamente. Esta métrica se expresa como la media armónica entre ambos valores, lo que le da un enfoque equilibrado:

$$F1 = \frac{Precisión * Exhaustividad}{Precisión + Exhaustividad}$$

Esta fórmula permite tener una idea clara del rendimiento del modelo, especialmente en situaciones donde hay un coste alto tanto por falsos positivos como por falsos negativos (Chicco & Jurman, 2020).

2.2.2. F1-Score ponderado

Cuando se trabaja con múltiples clases, es común que algunas de ellas tengan muchas más muestras que otras. Para abordar esta desigualdad, se emplea el F1-score ponderado (también conocido como weighted F1-score), que tiene en cuenta la proporción de cada clase respecto al total del conjunto de datos. La fórmula para calcular esta métrica es la siguiente:

$$F1_{ponderado} = \sum_{k=1}^{k} (\frac{n_i}{N} * F1_i)$$

Donde:

- n_i es el número de muestras de la clase i.
- N es el número total de muestras.
- $F1_i$ es el F1-score correspondiente a la clase i.

Este enfoque garantiza que todas las clases, incluso aquellas con menor representación, contribuyan proporcionalmente a la evaluación general del modelo. De esta forma, se evita que el desempeño en clases mayoritarias o minoritarias se vea injustamente amplificado o minimizado (Sokolova & Lapalme, 2009).

Al comparar distintos algoritmos de aprendizaje automático como XGBoost, LightGBM, MLP, CatBoost, entre otros, es recomendable utilizar una métrica que proporcione una visión global del rendimiento. En ese sentido, el F1-score ponderado resulta ideal, ya que evalúa de forma balanceada la capacidad del modelo para predecir correctamente todas las clases, independientemente de su proporción.

En un proyecto práctico, como el análisis de rendimiento académico de estudiantes, este criterio puede ser aplicado para seleccionar el mejor modelo entre varios entrenados. Por ejemplo, al implementar una función que evalúe el rendimiento de cada modelo y retorne el que tenga el F1 ponderado más alto, se puede garantizar que el modelo elegido tenga un buen desempeño general y no esté sesgado hacia clases específicas (Pedregosa, Varaquaux, Gramfort, Michel, & Thirion, 2011).

3. Marco metodológico

3.1. Área de estudio

El área de estudio se centra en la Unidad educativa San José Obrero ubicado en el País de Bolivia.

Bolivia es llamada también como: "El corazón de América del Sur" por su posición en el centro-oeste dentro de la región Sur Americana, su ubicación exacta esta entre las latitudes 9° 38' N y 22° 38' S, y sus longitudes 57° 26' W y 69° 38' W (educabolivia, 2025). Bolivia limita con cinco países:

- Al norte y este: A lo largo de 3423 kilómetros con Brasil.
- Al sur: Por 753 kilómetros con Paraguay y a lo largo de 773 kilómetros con Argentina.
- Al oeste: Con Chile limita a lo largo de 742 km y Perú con 1047 km.

La extensión territorial de Bolivia es de 1098 kilómetros cuadrados, siendo el 5 país más grande de América del Sur (educabolivia, 2025), cuenta con 11.312.620 de habitantes (INE, 2024).

Figura 3.1-1: Ubicación Geográfica de Bolivia
Fuente: Google Earth (2025)

3.2. Flujograma metodológico

Incluir un diagrama de flujo con los pasos a seguir para el desarrollo del proyecto.

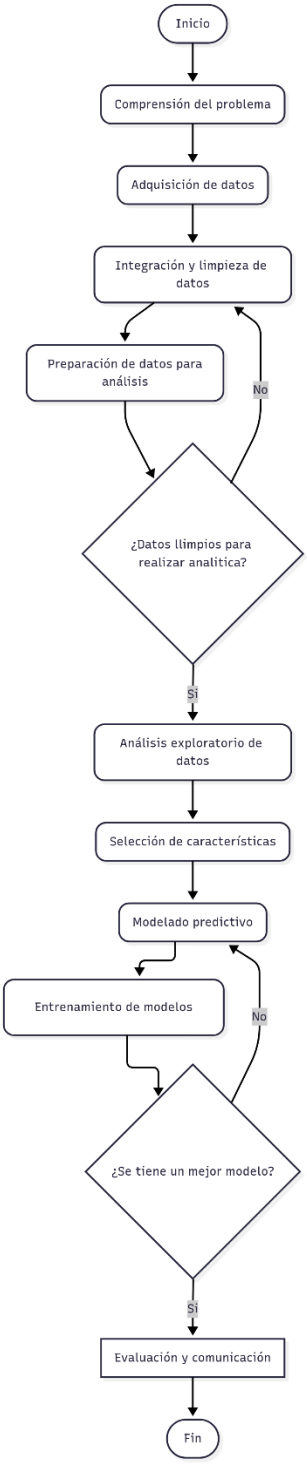


Figura 3.2-1: Flujograma metodológico

- Comprensión del problema: Se define claramente el objetivo y se contextualización del bajo rendimiento académico.

- Adquisición de datos: Recolección de archivos de calificaciones de todos los años posibles y convertirlos a formato Excel y separarlos por carpetas de primaria y secundaria.
- Integración y limpieza de datos: Juntar todos los datos en un solo documento, manejo de valores nulos, corrección de errores e inconsistencias, también se eliminan valores duplicados y transforman datos según en formatos adecuados.
- Preparación de datos para análisis: Realizar la separación de los datos en datos personales y calificación. En este punto realizar un análisis de correlación es importante para identificar relaciones entre variables.
- ¿Datos limpios para realizar analítica?: En este punto se verifica si los datos tienen la calidad suficiente para continuar con el análisis exploratorio y el modelado, en caso de que no se tenga la calidad suficiente, se regresa al paso de integración y limpieza de datos.
- Análisis exploratorio de datos (EDA): Se exploran los datos limpios para entender sus características principales. Se usa estadística descriptiva y visualizaciones pertinentes, el objeto es ganar intuición sobre los datos antes de modelarlos.
- Selección de características: Se seleccionan las variables que serán más útiles como entrada del modelo predictivo. Esto puede basarse en el análisis EDA.
- Modelado predictivo: Esta es la fase de construcción del modelo de machine learning. Se seleccionan los algoritmos que se van a probar, también se definen las estrategias de entrenamiento y evaluación.
- Entrenamiento de modelos: Se alimentan los datos de entrenamiento a los algoritmos elegidos en la fase de modelado predictivo para que aprendan patrones que relacionan las características con la variable objetivo. Esto incluye desde luego la optimización de hiperparámetros.
- ¿Se tiene un mejor modelo?: Se evalúa los modelos entrenados usando el conjunto de prueba y sus métricas definidas. Se comparan resultados de los diferentes modelos o configuración diferente del mismo. Si el criterio de rendimiento satisfactorio no es alcanzado se vuelve a las etapas anteriores, si el modelo cumple se avanza a la fase final.
- Evaluación y comunicación: Se realiza una evaluación detallada del mejor modelo seleccionado. Se interpretan los resultados, se preparan visualizaciones para comunicar los hallazgos, los perfiles estudiantes en riesgo y las conclusiones (estos son guardados).

3.3. Fuentes de información

3.3.1. Fuentes de información secundaria

Los datos obtenidos tienen 31 columnas y 2978 filas, representa una variedad de tipos de datos, entre los cuales se tiene calificaciones de estudiantes de nivel primario y secundario desde la gestión 2015 a la gestión 2023. A continuación, se detalla la naturaleza de los datos

- **Gestion –** Representa el año o período de gestión. Es un dato cuantitativo continuo, ya que puede tomar valores decimales.

- **Nivel –** Variable de tipo categórica, indica si el estudiante proviene del nivel primario o secundario.
- **Curso –** Variable de tipo categórica, especifica el curso de primaria o secundaria que aprobó o reprobó un estudiante.
- **A. Paterno –** Variable de tipo categórica, representa el apellido paterno del estudiante.
- **A. Materno-** Variable de tipo categórica, representa el apellido materno del estudiante.
- **Nombres –** Variable de tipo categórica que contiene el primer y segundo nombre del/la estudiante.
- **Codigo Rude –** Variable de tipo categórica nominal, representa el código de identificación del estudiante en el sistema nacional de educación, este se mantiene a lo largo de los años del estudiante en la educación regular.
- **Genero –** El género de un estudiante puede ser masculino o femenino, el tipo de dato es categórico nominal.
- **Fecha Nac –** Variable de tipo temporal, contine la fecha de nacimiento del estudiante.
- **Lug. Nac –** Lugar de nacimiento en el territorio nacional del estudiante, tipo de dato categórico nominal.
- **Numero CI –** Numero de cedula de identidad, su tipo de dato es categórico nominal.
- **Estado Matricula –** Estado del estudiante, Reprobado, aprobado o abandono. Tipo de dato categórico nominal.
- **Com. Lenguaje**: Calificaciones en Comunicación y Lenguaje. Cuantitativo continuo.
- **L. Extranjera**: Calificaciones en Lengua Extranjera. Cuantitativo continuo.
- **Ed. Civica**: Calificaciones en Educación Cívica. Cuantitativo continuo.
- **Geografia**: Calificaciones en Geografía. Cuantitativo continuo.
- **Cs. Sociales/Hist**: Calificaciones en Ciencias Sociales/Historia. Cuantitativo continuo.
- **Edu. Musical**: Calificaciones en Educación Musical. Cuantitativo continuo.
- **Art. Plasticas. V**: Calificaciones en Artes Plásticas. Cuantitativo continuo.
- **E. Fisica. D**: Calificaciones en Educación Física. Cuantitativo continuo.
- **Matematica**: Calificaciones en Matemáticas. Cuantitativo continuo.
- **T. General/Esp**: Calificaciones en Técnicas Generales/Especializadas. Cuantitativo continuo.
- **Fisica**: Calificaciones en Física. Cuantitativo continuo.
- **Quimica**: Calificaciones en Química. Cuantitativo continuo.
- **Bio./C.Nat**: Calificaciones en Biología/Ciencias Naturales. Cuantitativo continuo.
- **Cosmov y**: Calificaciones en Cosmovisiones y Filosofía. Cuantitativo continuo.
- **Val. Esp. Rel**: Calificaciones en Valores Espirituales y Religiones. Cuantitativo continuo.
- **Psicologia**: Calificaciones en Psicología. Cuantitativo continuo.
- **Promedio**: Promedio general de calificaciones. Cuantitativo continuo.

Este análisis será confirmado por un estudio previo realizado sobre los hábitos de estudio y las capacidades de los estudiantes.

3.4. Adquirir datos de calificaciones de los estudiantes con datos actualizados.

Para adquirir los archivos correspondientes para este proyecto, fue necesario hablar con la directora de la unidad educativa San José Obrero, la cual proporcionó archivos que contenían datos referentes a las calificaciones anuales de los estudiantes (desde el año 2015 al 2024), en un principio dichos archivos eran en formato pdf, luego de una conversión de archivo a tipo Excel se tenían archivos como se pueden apreciar en la figura 3.4-1

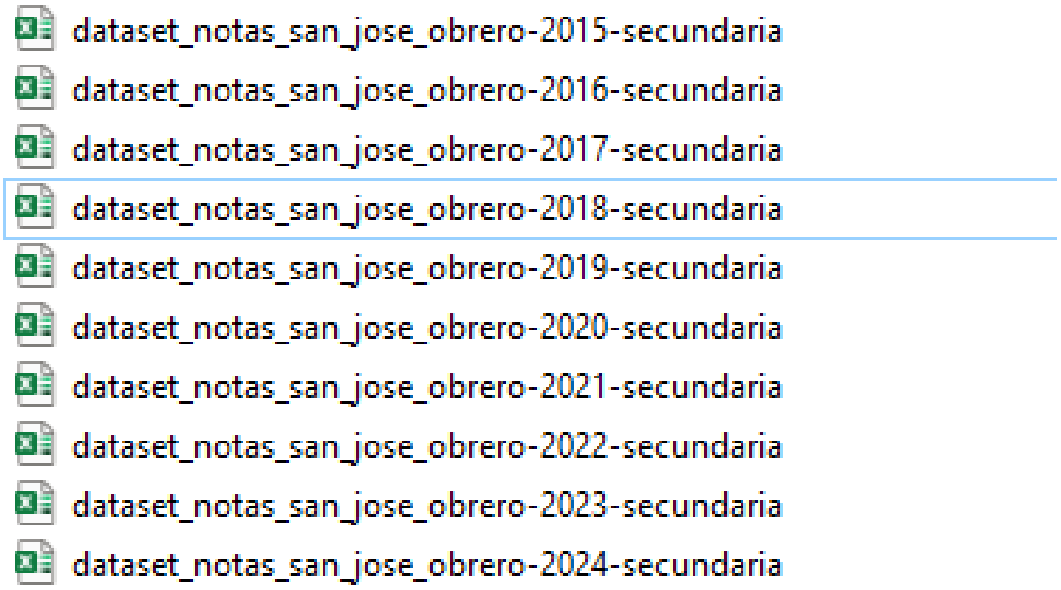


Figura 3.4-1: Muestra de archivos en formato Excel
Fuente: Google Earth (2025)

Los datos proporcionados fueron de nivel primario y secundario, en la figura 3.4-1, se puede apreciar los archivos Excel de secundaria, también se tiene de la misma manera y en el mismo formato para el nivel primario.

3.5. Integración y limpieza de datos

A lo largo de este paso se llevaron a cabo varios procesos para tener datos de calidad, al momento de iniciar el proceso se tenían 2 carpetas (Notas primarias y Notas secundaria), dentro de cada uno se tenía los archivos con las calificaciones por año, como se muestra en la figura 3.4-1

Una vez verificada la información se procedió a realizar una función para hacer un Merge de todos los archivos en un solo dataset desde jupyter como muestra la figura 3.5-1

```python
def merge_excel_files(folder_paths, output_file):
    dfs = []  # Lista para almacenar los DataFrames
    # Recorrer cada carpeta individualmente
    for folder_path in folder_paths:
        if not isinstance(folder_path, str):  # Verificar que la ruta es válida
            print(f"❌ Error: '{folder_path}' no es una ruta válida.")
            continue
        if not os.path.exists(folder_path):  # Verificar si la carpeta existe
            print(f"⚠️ Advertencia: La carpeta '{folder_path}' no existe. Se omitirá.")
            continue
        file_list = [f for f in os.listdir(folder_path) if f.endswith(".xlsx")]
        for file in file_list:
            file_path = os.path.join(folder_path, file)
            try:
                df = pd.read_excel(file_path, engine='openpyxl')  # Leer archivo Excel, especificando el engine
                print(f"Leyendo archivo: {file_path}") # Para verificar qué archivo se está leyendo

                # Extraer el año del nombre del archivo
                try:
                    df["Año"] = file.split("-")[1].split(".")[0] # Asegurarse de quitar la extensión
                except IndexError:
                    df["Año"] = "Desconocido"
                    print(f"⚠️ No se pudo extraer el año de '{file}'. Se asignará 'Desconocido'.")

                # Agregar la carpeta de origen como columna (opcional)
                df["Origen"] = os.path.basename(folder_path)

                dfs.append(df)  # Agregar DataFrame a la lista
            except ValueError as ve:
                print(f"❌ Error al leer el archivo '{file_path}': {ve}")
                print("   Asegúrate de que sea un archivo Excel válido (.xlsx) o considera especificar el 'engine' correcto (ej., 'xlrd' para .xls).")
            except Exception as e:
                print(f"❌ Error inesperado al procesar el archivo '{file_path}': {e}")
    if not dfs:
        print("❌ No se encontraron archivos para combinar. Verifica las rutas y los archivos .xlsx.")
        return None  # Retorna None para evitar otros errores
    # Concatenar todos los DataFrames
    merged_df = pd.concat(dfs, ignore_index=True)
    try:
        merged_df.to_excel(output_file, index=False)
        print(f"✅ Merge completado. Archivo guardado en: {output_file}")
    except Exception as e:
        print(f"❌ Error al guardar el archivo: {e}")

    return merged_df
```

Figura 3.5-1: Merge de los archivos
Fuente: Elaboración propia (2025)

Una vez realizo esto y definida la ruta donde se tienen los archivos (folder_paths), se define la ruta de salida (output_file), donde se indica el nombre de salida del archivo y su ruta definida como se muestra a continuación (ver figura 3.5-2)

```python
# Ejecutar la función
merged_df = merge_excel_files(folder_paths, output_file)
```

Figura 3.5-2: Juntando los archivos en jupyter
Fuente: Elaboración propia (2025)

Ya con un solo archivo con todos los datos, procedemos a importarlo y ver la cantidad de filas y columnas que posee nuestro set de datos (ver figura 3.5-3)

```
# Cargar el archivo Excel mergeado
file_path = "L:/Materiales cursos y diplomados/datascience/Notas San Jose Obrero/merged_dataset.xlsx"
```

```
# Leer el archivo Excel
df = pd.read_excel(file_path)
```

Figura 3.5-3: Importar el dataset completo
Fuente: Elaboración propia (2025)

3.5.1. Contando la cantidad de filas y columnas

Para iniciar este apartado se debe de contar la cantidad total de filas y columnas que tiene el dataset como se ve en la figura 3.5-4.

```
# Obtener el número de filas y columnas
num_filas, num_columnas = df.shape
print(f" El archivo tiene {num_filas} filas y {num_columnas} columnas.")
```

```
El archivo tiene 2978 filas y 31 columnas.
```

Figura 3.5-4: Importar el dataset completo
Fuente: Elaboración propia (2025)

3.5.2. Verificación de valores nulos

Seguido de esto, verificamos la cantidad de valores faltantes para luego abordarlos teniendo en cuenta la integridad de la información (ver figura 3.5-5)

```
df.isnull().sum()
```

Figura 3.5-5: Verificando valores nulos
Fuente: Elaboración propia (2025)

3.5.3. Borrando valores duplicados

Borramos los valores duplicados y las filas que tienen todos los campos vacíos, esto debido a que no contienen datos (ver figura 3.5-6)

```
df = df.dropna(how='all')
```

```
# Buscando valores duplicados
df.duplicated().sum()
```

Figura 3.5-6: Borrado de filas vacías y duplicados
Fuente: Elaboración propia (2025)

3.5.4. Tratamiento de columnas categóricas y numéricas

Es conveniente tener campos como nombre, apellidos, lugar de nacimiento con el valor de desconocido, mientras que las que son de tipo numeral (calificaciones) con el valor de 0 (ver figura 3.5-7), esto debido a que los estudiantes que tienen calificaciones con valor null, es debido a que no pasan dichas materias otro motivo es que el estudiante se haya retirado o cambiado de colegio, de poner alguna calificación diferente, el sistema de educación boliviano interpreta que el estudiante a concluido dicha materia o gestión escolar.

```
# Rellenar columnas categóricas con 'Desconocido'
categorical_cols = ["A. Paterno", "Lug. Nac"]
df[categorical_cols] = df[categorical_cols].fillna("Desconocido")
```

```
# Rellenar notas con 0
notas_cols = ["Com. Lenguaje", "L. Extrangera", "Ed. Civica", "Geografia", "Cs. Sociales/Hist", "Edu. Musical"
              , "Art. Plasticas. V.", "E. Fisica. D." , "Matematica", "T. General/Esp", "Fisica", "Quimica"
              , "Bio./C.Nat", "Cosmov y", "Val. Esp. Rel", "Psicologia", "Promedio"]
df[notas_cols] = df[notas_cols].fillna(0)
```

Figura 3.5-7: Reemplazando datos vacíos
Fuente: Elaboración propia (2025)

3.5.5. Tratamiento de Numero CI

Un tratamiento de dato que se realizó de manera independiente fue el número de CI, existen estudiantes que llegan a la unidad educativa desde el extranjero, en el sistema de registro de educación regular no permite ingresar en este campo este tipo de documentos (indicó lo directora de la unidad de educativa), en ese entendido se reemplazaron estos campos null por N/A (no aplica) como se puede ver en la figura 3.5-8.

```
# Rellenar columnas categóricas con 'Desconocido'
categorical_cols = ["Numero CI"]
df[categorical_cols] = df[categorical_cols].fillna("N/A")
```

Figura 3.5-8: Reemplazo de datos en Numero CI
Fuente: Elaboración propia (2025)

3.5.6. Cambio de tipo de dato a Fecha de nacimiento

Posteriormente, es importante cambiar a datetime la columna de Fecha Nac., aquí está contenida la fecha de nacimiento del estudiante (ver figura 3.5-9)

```
df["Fecha Nac."] = pd.to_datetime(df["Fecha Nac."], errors='coerce')
```

Figura 3.5-9: Cambio de tipo de datos de Object a datetime
Fuente: Elaboración propia (2025)

3.6. Preparación de datos para análisis

3.6.1. Separando datos personales y calificaciones

Para tener los datos más ordenados, se separaron estos en datos_personales y calificaciones (ver figura 3.6-1), también se cambió el tipo de dato de estos últimos, posteriormente veremos que esto nos será favorable a la hora de tener la narrativa de los descubrimientos.

```
datos_personales = ["A. Paterno", "A. Materno", "Nombres",
                    "Codigo Rude", "Genero", "Fecha Nac.", "Lug. Nac", "Numero CI"]

calificaciones = ["Codigo Rude", "Nivel", "Curso", "Estado Matricula", "Gestion", "Com. Lenguaje", "L. Extrangera", "Ed. Civica", "Geografia",
                  "Cs. Sociales/Hist", "Edu. Musical", "Art. Plasticas. V.", "E. Fisica. D.",
                  "Matematica", "T. General/Esp", "Fisica", "Quimica", "Bio./C.Nat", "Cosmov y",
                  "Val. Esp. Rel", "Psicologia", "Promedio"]

df["Codigo Rude"] = df["Codigo Rude"].astype(str)  # Mantener como texto
df[calificaciones[4:]] = df[calificaciones[4:]].fillna(0).astype(int)  # Convertir calificaciones a entero
```

Figura 3.6-1: Ordenamiento de datos
Fuente: Elaboración propia (2025)

3.6.2. Guardado de datos personales y calificaciones en archivos separados

Guardamos estos datos en archivos Excel separados para su posterior uso como se puede ver en la figura 3.6-2

```
df[datos_personales].to_excel("datos_personales.xlsx", index=True)
df[calificaciones].to_excel("calificaciones.xlsx", index=True)
```

Figura 3.6-2: Guardado de datos separados en archivos Excel
Fuente: Elaboración propia (2025)

3.6.3. Quitando a estudiantes con promedio = 0

Hasta este punto se tenían a estudiantes con promedio=0, ya que para realizar la analítica esta información es valiosa, los estudiantes que cuentan con este promedio son por el motivo de: traslado o abandono, para la parte posterior que es predictiva, tener estos datos era irrelevante, ya que si un estudiante ya no se encuentra en la unidad educativa no tiene sentido predecir si este reprobara el año escolar en la unidad educativa San José Obrero que es el objetivo de este proyecto.

Dado este motivo, se procedió a quitar del dataset a los estudiantes como promedio igual a cero (ver figura 3.6-3).

```
df = df[df['Promedio'] != 0]
```

Figura 3.6-3: Descartando estudiantes con promedio=0 para la parte predictiva
Fuente: Elaboración propia (2025)

3.6.4. Convirtiendo la gestión a tipo de dato entero

Además de esto, se cambió el tipo de dato a entero para la columna de Gestion como se puede ver en la figura 3.6-4

```
df["Gestion"] = df["Gestion"].astype(int)
```

Figura 3.6-4: Cambio de tipo de dato a la columna Gestion
Fuente: Elaboración propia (2025)

3.6.5. Examinar la evolución académica del estudiante

Para ver la evolución de cada estudiante a lo largo de los años, se usarán campos calculados, el primero será la combinación del nombre completo del estudiante (ver figura 3.6-5)

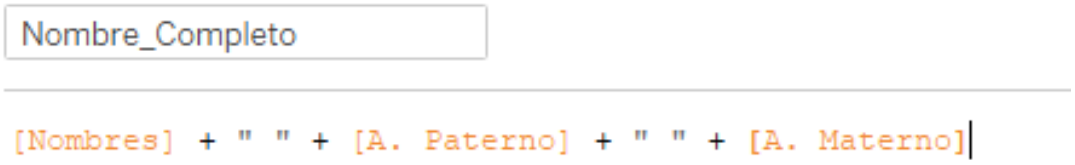


Figura 3.6-5: Formula para obtener el nombre completo del estudiante
Fuente: Elaboración propia (2025)

Los gráficos completos se generarán posteriormente con más detalle.

3.6.6. Identificación de las materias que representan mayor dificultad

Para identificar a las materias con mayor dificultad para los estudiantes, se realizó un ordenamiento de los promedios según su nivel, curso y materia, al tener el nombre de las materias en horizontal, lo primero que se realizó fue un pívot para tener una mejor comprensión y análisis por materia como se ve en la figura 3.6-6, desde luego se obviaron los promedios por materia que contenían 0, esto debido a que los estudiantes de primaria no pasan todas las materias del nivel secundario. También se omitieron los estudiantes que abandonaron la unidad educativa o realizaron su traslado, esto mismo era perjudicial para el promedio por materia y curso ya que algunos cursos tienen mayor cantidad de estos tipos de estudiantes, lo que hace que el promedio disminuya.

Abc Sheet11 Estado Matricula	# Sheet11 Gestion	# Sheet11 Promedio	Abc Pivot Materias	# Pivot Notas
PROMOVIDO	2,015	78	Art. Plasticas. V.	73
PROMOVIDO	2,015	78	Bio./C.Nat	75
PROMOVIDO	2,015	78	Com. Lenguaje	75
PROMOVIDO	2,015	78	Cosmov y	0
PROMOVIDO	2,015	78	Cs. Sociales/Hist	79

Figura 3.6-6: Haciendo pívot de las materias y notas
Fuente: Elaboración propia (2025)

3.7. Análisis exploratorio

Para este análisis exploratorio, veremos algunos puntos clave, para pasar a las siguientes etapas es necesario realizar una exploración de datos.

3.7.1. Contando la cantidad de reprobados por año

Ya que posteriormente realizaremos la predicción de estudiantes reprobados, es interesante ver la cantidad de estudiantes reprobados por gestión hasta el último año del cual se tienen datos (2024) como se muestra en la figura 3.7-1.

```
df["Reprobado"] = df["Estado Matricula"].str.contains("REPROBADO", case=False).astype(int)
```

```
cantidad_reprobados_por_anio = df.groupby("Gestion")["Reprobado"].sum()
print(cantidad_reprobados_por_anio)
```

```
Gestion
2015     6
2016     4
2017    20
2018    22
2019     6
2020     0
2021    27
2022    17
2023    30
2024    31
```

Figura 3.7-1: Explorando la cantidad de reprobados por gestión
Fuente: Elaboración propia (2025)

3.7.2. Mapa de calor de correlaciones

Es importante realizar también el análisis de mapa de calor de correlación para ver qué tan fuerte es la relación lineal entre variables, para nuestro caso, en las variables numéricas usaremos el coeficiente de correlación de Pearson como vemos en la figura 3.7-2.

```
correlacion = df.select_dtypes(include=["number"]).corr()
plt.figure(figsize=(12,8))
sns.heatmap(correlacion, annot=True, cmap="coolwarm", fmt=".2f", linewidths=0.5)
plt.title("Mapa de Calor de Correlación")
plt.show()
```

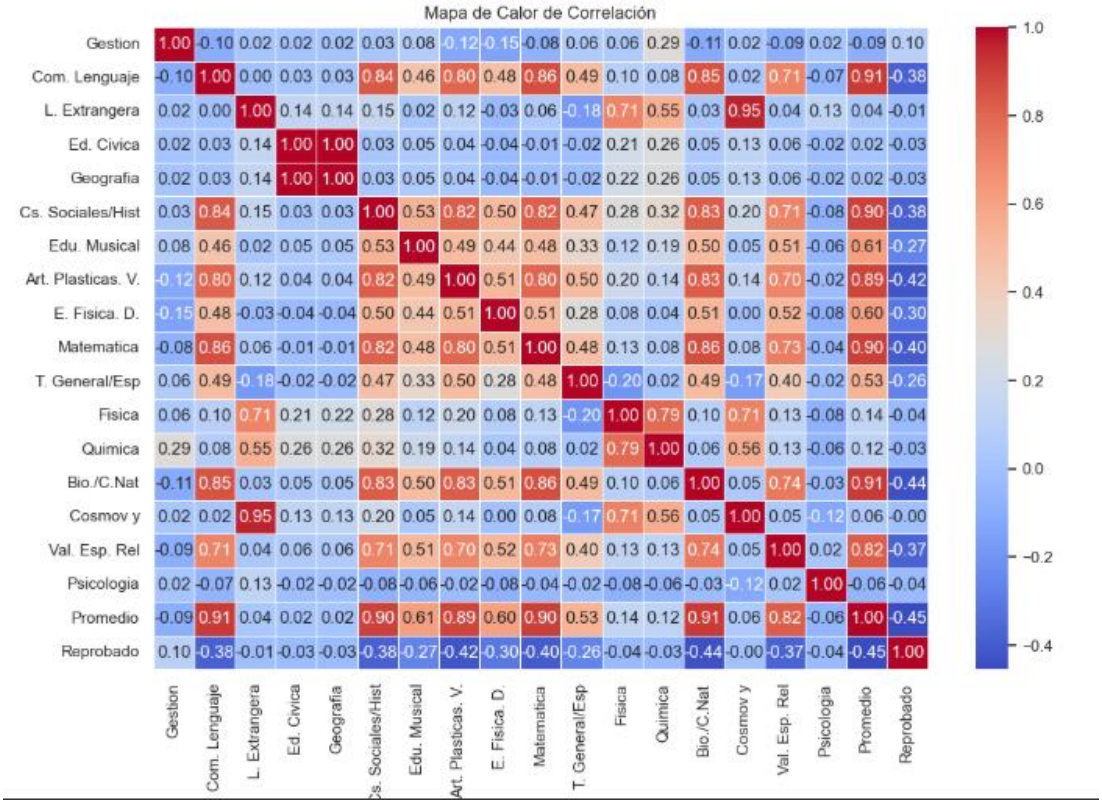


Figura 3.7-2: Coeficiente de correlación de Pearson
Fuente: Elaboración propia (2025)

3.7.3. Estadísticas descriptivas

El análisis de estadísticas descriptivas es clave para cualquier este estudio, ya que permite que conozcamos la información más a fondo (ver figura 3.7-3). Gracias a este paso, podemos identificar qué variables hay, cómo se comportan y si existen errores o valores fuera de lo normal. Además, ayuda a ver si tenemos datos faltantes y qué tan homogénea o dispersa está nuestra información. Por último, ofrece pistas importantes sobre posibles relaciones entre variables, lo que resulta útil para plantear hipótesis o construir modelos predictivos.

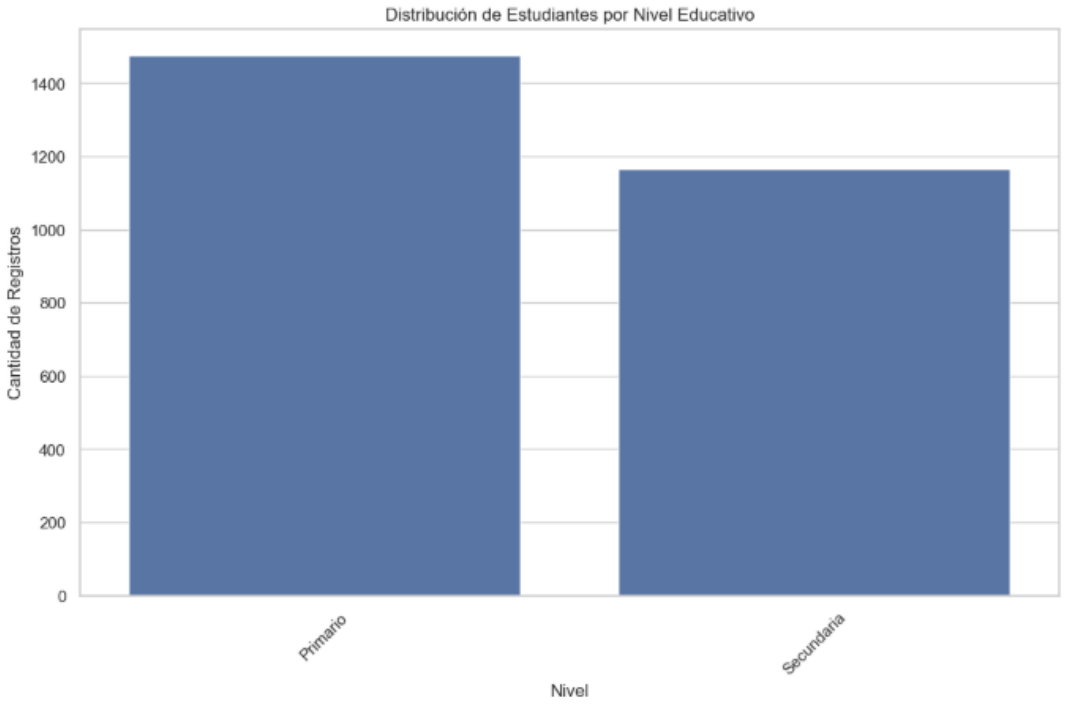
```
print("--- Estadísticas Descriptivas (Variables Numéricas) ---")
# Muestra estadísticas como media, std, min, max, cuartiles para columnas numéricas
print(df.describe())
print("-" * 50)
```

```
--- Estadísticas Descriptivas (Variables Numéricas) ---
           Gestion                      Fecha Nac.  Com. Lenguaje  \
count  2641.000000                            2641    2641.000000
mean   2019.644832  2008-03-24 17:53:35.600151296      66.925407
min    2015.000000            1995-03-18 00:00:00      35.000000
25%    2017.000000            2004-12-08 00:00:00      57.000000
50%    2020.000000            2008-04-02 00:00:00      66.000000
75%    2022.000000            2011-11-24 00:00:00      76.000000
max    2024.000000            2018-06-23 00:00:00      98.000000
std       2.854479                            NaN      12.636262
```

Figura 3.7-3: Estadísticas descriptivas
Fuente: Elaboración propia (2025)

3.7.4. Análisis con gráficos

Para este apartado, se usaron dos tipos de análisis: análisis univariado y bivariado, vemos que en la figura 3.7-4 podemos ver la cantidad de estudiantes registrados a lo largo de los años en cada nivel educativo.




```
Value Counts de Nivel:
Nivel
Primario      1475
Secundaria    1166
```

Figura 3.7-4: Análisis univariado
Fuente: Elaboración propia (2025)

Para el análisis bivariado, podemos analizar la distribución del promedio por género como podemos ven en la figura 3.7-5.

```
# Promedio vs. Género
if 'Genero' in df.columns:
    plt.figure()
    sns.boxplot(data=df, x='Genero', y='Promedio')
    plt.title('Distribución del Promedio por Género')
    plt.show()
```

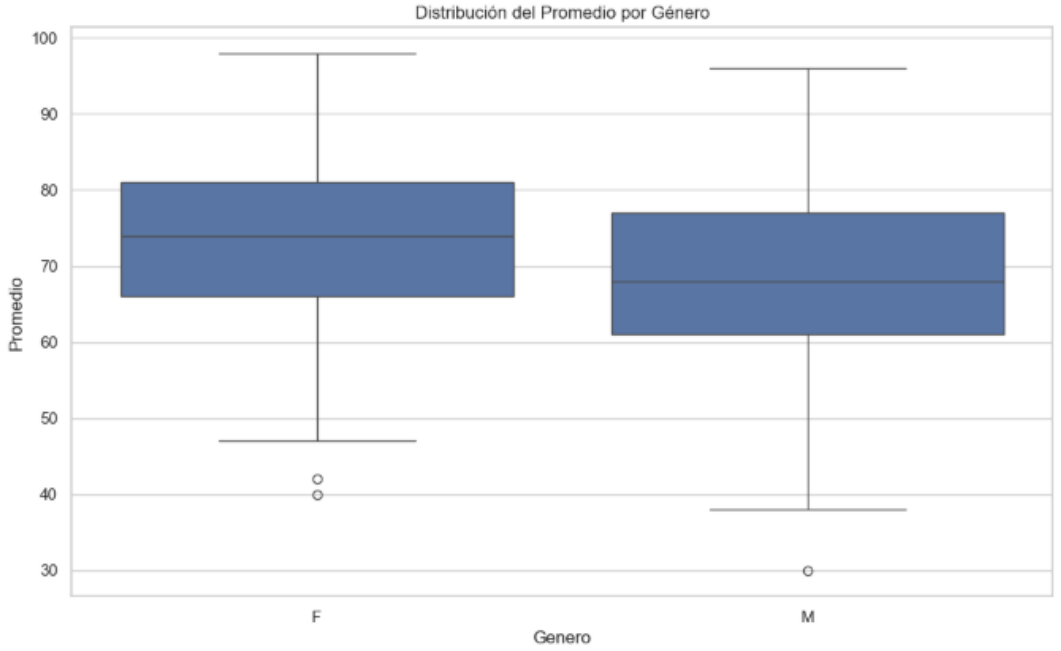


Figura 3.7-5: Análisis bivariado
Fuente: Elaboración propia (2025)

3.8. Selección de características

3.8.1. Creación de características avanzadas

Para enriquecer la información disponible para los modelos predictivos, se realiza ingeniería de características (feature engineering). A partir de los datos históricos de cada estudiante, se generan nuevas variables que resumen su trayectoria académica anual. Estas incluyen: el número acumulado de reprobaciones, el promedio de calificaciones de los últimos tres años, la tendencia del promedio (cambio respecto al año anterior), la variabilidad del promedio, los años que lleva el estudiante en el sistema educativo, el promedio global histórico, un indicador de si el rendimiento está mejorando, la diferencia entre el promedio actual y el global, la ratio de reprobación por año, la volatilidad reciente del rendimiento y la aceleración del promedio (ver figura 3.8-1). Estas características buscan capturar patrones dinámicos en el desempeño del estudiante.

```python
def create_features(df):
    """Crea características avanzadas a partir del historial académico."""
    logger.info("Iniciando creación de características...")

    # Agrupar por estudiante
    grouped = df.groupby("Codigo Rude")

    # Características basadas en historial académico
    df["Reprobaciones_acumuladas"] = grouped["Reprobado"].cumsum()
    df["Promedio_ultimos_3"] = grouped["Promedio"].rolling(3, min_periods=1).mean().reset_index(level=0, drop=True)
    df["Tendencia_promedio"] = grouped["Promedio"].diff().fillna(0)
    df["Variabilidad_promedio"] = grouped["Promedio"].rolling(3, min_periods=1).std().fillna(0).reset_index(level=0, drop=True)

    # Características de años en el sistema
    df["Años_en_sistema"] = df.groupby("Codigo Rude").cumcount() + 1

    # Calcular promedio global del estudiante hasta el momento
    df["Promedio_global"] = grouped["Promedio"].transform(lambda x: x.expanding().mean())

    # Calcular si el estudiante está mejorando o empeorando (tendencia)
    df["Mejorando"] = grouped["Promedio"].transform(lambda x: x.rolling(2, min_periods=2).apply(lambda y: 1 if y.iloc[1] > y.iloc[0] else 0)).fillna(0)

    # Características adicionales
    # Diferencia entre promedio actual y promedio global
    df["Delta_promedio"] = df["Promedio"] - df["Promedio_global"]

    # Ratio de reprobaciones por año
    df["Ratio_reprobacion"] = df["Reprobaciones_acumuladas"] / df["Años_en_sistema"]

    # Volatilidad del rendimiento en los últimos años (desviación estándar móvil)
    df["Volatilidad_rendimiento"] = grouped["Promedio"].transform(lambda x: x.rolling(3, min_periods=1).std()).fillna(0)

    # Aceleración del promedio (cambio en la tendencia)
    df["Aceleracion_promedio"] = grouped["Tendencia_promedio"].diff().fillna(0)

    logger.info("Creación de características completada.")
    return df
```

Figura 3.8-1: Creación de características avanzadas
Fuente: Elaboración propia (2025)

3.9. Modelado predictivo

3.9.1. Tabla minable

Ya con lo realizado en la limpieza y transformación de datos, procedemos a extraer nuestra tabla minable como se puede ver en la figura 3.9-1. Crear esta, representa una buena práctica de ingeniería de datos y estructuración de proyectos que mejora la claridad, modularidad, eficiencia y reproducibilidad para nuestro análisis y modelado predictivo. Es el puente bien definido entre la preparación de datos crudos y la aplicación de algoritmos de machine learning.

```python
# Crear una copia explícita para evitar el error
tabla_minable = df[["Codigo Rude", "Genero", "Nivel", "Curso", "Gestion", "Promedio", "Reprobado"]].copy()
```

```python
# Guardar la tabla
tabla_minable.to_csv("tabla_minable.csv", index=False)
print("✅ Tabla minable creada exitosamente.")
```

Figura 3.9-1: Creación de la tabla minable
Fuente: Elaboración propia (2025)

3.9.2. Importación de librerías

Existen librerías que nos necesarias instalar para continuar, dichas librerías no vienen con nuestro entorno de desarrollo, para esto se usa pip como se ve en la figura 3.9-2.

```
!pip install catboost optuna category_encoders imblearn shap xgboost
```

Figura 3.9-2: Instalación necesaria de las librerías
Fuente: Elaboración propia (2025)

El resto de las librerías no son necesarias descargarlas, estas vienen de manera nativa por lo cual solo necesitaríamos importarlas como se ve en la figura 3.9-3.

```
# Librerias
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import TimeSeriesSplit, cross_val_score
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from xgboost import XGBClassifier
from sklearn.svm import SVC
from sklearn.neural_network import MLPClassifier
from lightgbm import LGBMClassifier
from catboost import CatBoostClassifier
from sklearn.metrics import accuracy_score, classification_report, roc_curve, auc, confusion_matrix, precision_recall_curve
from sklearn.utils.class_weight import compute_class_weight
from sklearn.model_selection import GridSearchCV
from sklearn.pipeline import Pipeline
import logging
```

Figura 3.9-3: Importación de las librerías necesarias
Fuente: Elaboración propia (2025)

3.9.3. Creación de logging

Para tener un control detallado de lo que ocurre en la ejecución, se usará y configurará los logging (ver figura 3.9-4), estos no son necesariamente obligatorios para recrear este proyecto, pero ayuda bastante a la hora de ver lo que ocurre cuando ejecutamos nuestras celdas.

```
# --- Configuración de Logging ---
# Configura el logger para escribir en un archivo y en la consola
log_file = "student_analysis.log"
logging.basicConfig(
    level=logging.INFO, # Nivel mínimo de mensajes a registrar (DEBUG, INFO, WARNING, ERROR, CRITICAL)
    format='%(asctime)s - %(levelname)s - %(message)s', # Formato de los mensajes
    handlers=[
        logging.FileHandler(log_file), # Escribir logs en un archivo
        logging.StreamHandler() # Mostrar logs en la consola
    ]
)
# Obtener el objeto logger
logger = logging.getLogger(__name__)
# --- Fin Configuración de Logging ---
```

Figura 3.9-4: Configuración del logging
Fuente: Elaboración propia (2025)

3.9.4. Preprocesamiento de datos

Esta fase es crucial para estructurar los datos de manera adecuada para el análisis predictivo y temporal. Primero, los datos se ordenan cronológicamente por estudiante (Codigo Rude) y año (Gestion). Luego, se crea la variable objetivo clave, Reprobado_siguiente, que indica si un estudiante reprobó en el año posterior al registro actual; esto se logra desplazando (shift(-1)) el estado de reprobación dentro del

historial de cada estudiante. A continuación, el conjunto de datos se divide en dos partes: df_model, que contiene los datos históricos (excluyendo el último año) y para los cuales se conoce el resultado del año siguiente (la variable objetivo), y df_future, que contiene solo los datos del último año disponible y se reserva para realizar las predicciones finales una vez entrenado el modelo. Finalmente, se eliminan de df_model las filas correspondientes al último registro de cada estudiante, ya que no tienen un valor válido para Reprobado_siguiente, y se asegura que esta variable objetivo sea de tipo entero como se muestra en la figura 3.9-5.

```python
# Preprocesamiento de datos
def preprocess_data(df):
    """Preprocesa el dataframe para análisis y modelado."""
    logger.info("Iniciando preprocesamiento de datos...")
    # Ordenar por Código Rude y Gestión
    df = df.sort_values(by=["Codigo Rude", "Gestion"])

    # Crear la variable objetivo (si el estudiante reprueba el siguiente año)
    df["Reprobado_siguiente"] = df.groupby("Codigo Rude")["Reprobado"].shift(-1)

    # Eliminar la última gestión ya que no podemos saber si reprobaron después
    max_year = df["Gestion"].max()
    logger.info(f"Año máximo en los datos: {max_year}")
    df_model = df[df["Gestion"] < max_year].copy()

    # Guardar los datos del último año para predicciones futuras
    df_future = df[df["Gestion"] == max_year].copy()
    logger.info(f"Datos para modelado: {df_model.shape[0]} filas (hasta {max_year-1})")
    logger.info(f"Datos para predicción futura: {df_future.shape[0]} filas ({max_year})")

    # Eliminar filas con valores nulos en la variable objetivo
    initial_rows = df_model.shape[0]
    df_model = df_model.dropna(subset=["Reprobado_siguiente"])
    dropped_rows = initial_rows - df_model.shape[0]
    if dropped_rows > 0:
        logger.warning(f"Se eliminaron {dropped_rows} filas con 'Reprobado_siguiente' nulo (del último año por estudiante).")
    df_model["Reprobado_siguiente"] = df_model["Reprobado_siguiente"].astype(int)

    logger.info("Preprocesamiento de datos completado.")
    return df_model, df_future, max_year
```

Figura 3.9-5: Preprocesamiento de datos
Fuente: Elaboración propia (2025)

3.9.5. División de datos por año

Una vez preprocesados los datos históricos (df_model) y creadas las características avanzadas, es necesario dividirlos en un conjunto de entrenamiento (df_train) y un conjunto de prueba (df_test). Dado el carácter temporal de los datos (historial académico por año), se aplica una división temporal estricta. Se selecciona un año específico como punto de corte (test_year) como se ve en la figura 3.9-6.

```python
# Dividir datos según años
def temporal_split(df, test_year):
    """Divide los datos en entrenamiento y prueba basado en años."""
    logger.info(f"Realizando división temporal: entrenamiento < {test_year}, prueba = {test_year}")
    # Usar años anteriores para entrenamiento
    df_train = df[df["Gestion"] < test_year]
    # Usar el año especificado para prueba
    df_test = df[df["Gestion"] == test_year]

    logger.info(f"Tamaño entrenamiento: {df_train.shape[0]} filas")
    logger.info(f"Tamaño prueba: {df_test.shape[0]} filas")
    return df_train, df_test
```

Figura 3.9-6: División de datos
Fuente: Elaboración propia (2025)

Todos los datos anteriores a ese año se utilizan para entrenar el modelo, y los datos de ese año específico se reservan como conjunto de prueba para evaluar el rendimiento del modelo en datos no vistos, simulando un escenario de predicción real y evitando la fuga de información del futuro al pasado.

3.9.6. Preparación de datos para el modelado

Antes de alimentar los datos a los algoritmos de machine learning, se realizan dos pasos finales de preparación sobre los conjuntos de entrenamiento y prueba. Primero, las variables categóricas (Genero, Nivel, Curso) se convierten en representaciones numéricas utilizando la técnica de One-Hot Encoding; esto crea nuevas columnas binarias para cada categoría, permitiendo a los modelos procesarlas. Segundo, todas las características numéricas (incluyendo las recién creadas y las originales como Promedio o Gestion) se escalan utilizando StandardScaler. El escalado estandariza las características para que tengan media cero y desviación estándar uno, asegurando que las variables con rangos de valores más grandes no dominen indebidamente el proceso de aprendizaje del modelo. El escalador se ajusta (fit) solo con los datos de entrenamiento y luego se aplica (transform) tanto al conjunto de entrenamiento como al de prueba 3.9-7.

```python
# Preparar datos para modelado
def prepare_model_data(df_train, df_test):
    """Prepara caracteristicas X e y para entrenar y evaluar modelos."""
    logger.info("Preparando datos para modelado (codificación y escalado)...")
    drop_cols = ["Reprobado_siguiente", "Gestion", "Codigo Rude"]
    categorical_cols = ["Genero", "Nivel", "Curso"]
    logger.info(f"Aplicando One-Hot Encoding a: {categorical_cols}")
    df_combined = pd.concat([df_train, df_test])
    cols_to_encode = [col for col in categorical_cols if col in df_combined.columns]
    if len(cols_to_encode) < len(categorical_cols):
        missing_cols = set(categorical_cols) - set(cols_to_encode)
        logger.warning(f"Columnas categóricas no encontradas y no se codificarán: {missing_cols}")
    if cols_to_encode:
        df_encoded = pd.get_dummies(df_combined, columns=cols_to_encode, drop_first=True)
    else:
        df_encoded = df_combined # No hacer nada si no hay columnas que codificar
    train_idx = df_encoded.index.isin(df_train.index)
    df_train_encoded = df_encoded[train_idx]
    df_test_encoded = df_encoded[~train_idx]
    logger.info(f"Columnas a eliminar antes de escalar: {drop_cols}")
    existing_drop_cols_train = [col for col in drop_cols if col in df_train_encoded.columns]
    existing_drop_cols_test = [col for col in drop_cols if col in df_test_encoded.columns]
    X_train = df_train_encoded.drop(columns=existing_drop_cols_train)
    y_train = df_train_encoded["Reprobado_siguiente"]
    X_test = df_test_encoded.drop(columns=existing_drop_cols_test)
    y_test = df_test_encoded["Reprobado_siguiente"]
    feature_names = X_train.columns
    logger.info(f"Número de caracteristicas para el modelo: {len(feature_names)}")
    logger.debug(f"Nombres de caracteristicas: {feature_names.tolist()}") # Log detallado opcional
    logger.info("Aplicando StandardScaler a las caracteristicas.")
    scaler = StandardScaler()
    X_train_scaled = scaler.fit_transform(X_train)
    X_test_scaled = scaler.transform(X_test)
    logger.info("Preparación de datos para modelado completada.")
    return X_train_scaled, y_train, X_test_scaled, y_test, scaler, feature_names
```

Figura 3.9-7: Preparación de datos para el modelado
Fuente: Elaboración propia (2025)

3.10. Entrenamiento de modelos

Para determinar el algoritmo más efectivo para predecir la reprobación, se adoptó un enfoque comparativo. Se definió un conjunto diverso de modelos candidatos, incluyendo Regresión Logística, Random Forest, XGBoost, Gradient Boosting, Máquinas de Soporte Vectorial (SVM), Perceptrón Multicapa (MLP), LightGBM y CatBoost. Para cada modelo, se especificó una parrilla de hiperparámetros clave a explorar. Se utilizó la técnica de Búsqueda en Parrilla (GridSearchCV) en combinación con Validación Cruzada específica para Series Temporales (TimeSeriesSplit) para encontrar la mejor combinación de hiperparámetros para cada algoritmo como se puede apreciar en la figura 3.10-1, optimizando según el F1-score ponderado.

```
models = {
    "Logistic Regression": LogisticRegression(max_iter=1000, class_weight=class_weight_dict, random_state=42),
    "Random Forest": RandomForestClassifier(n_estimators=100, class_weight=class_weight_dict, random_state=42),
    "XGBoost": XGBClassifier(scale_pos_weight=scale_pos_weight_xgb, eval_metric='logloss', use_label_encoder=False, random_state=42),
    "Gradient Boosting": GradientBoostingClassifier(random_state=42),
    "SVM": SVC(probability=True, class_weight=class_weight_dict, random_state=42),
    "MLP": MLPClassifier(max_iter=500, random_state=42, early_stopping=True),
    "LightGBM": LGBMClassifier(random_state=42, class_weight=class_weight_dict),
    "CatBoost": CatBoostClassifier(random_state=42, verbose=0, auto_class_weights='Balanced')
}

# Parámetros para GridSearch por modelo
param_grids = {
    "Logistic Regression": {'C': [0.01, 0.1, 1, 10]},
    "Random Forest": {'max_depth': [5, 10, None], 'min_samples_split': [2, 10]},
    "XGBoost": {'max_depth': [3, 5, 7], 'learning_rate': [0.05, 0.1]},
    "Gradient Boosting": {'n_estimators': [100, 200], 'learning_rate': [0.05, 0.1]},
    "SVM": {'C': [0.1, 1, 10], 'gamma': ['scale', 0.1]},
    "MLP": {'hidden_layer_sizes': [(50,), (100,)], 'alpha': [0.0001, 0.001]},
    "LightGBM": {'num_leaves': [31, 50], 'learning_rate': [0.05, 0.1]},
    "CatBoost": {'depth': [4, 6], 'learning_rate': [0.05, 0.1]}
}

# Resultados para cada modelo
results = {}
best_models = {}

# Entrenar y evaluar cada modelo
for name, model in models.items():
    logger.info(f"--- Optimizando hiperparámetros para {name} ---")
    # Usar TimeSeriesSplit para CV si es apropiado, aunque 5-fold estándar también es común
    # cv_strategy = 5 # CV estándar
    cv_strategy = TimeSeriesSplit(n_splits=5)  # Usar validación cruzada para series temporales
    logger.info(f"Usando TimeSeriesSplit con {cv_strategy.n_splits} divisiones para validación cruzada")
    grid = GridSearchCV(model, param_grids[name], cv=cv_strategy, scoring='f1_weighted', n_jobs=-1) # Usar f1_weighted
```

Figura 3.10-1: Modelos seleccionados con sus parámetros
Fuente: Elaboración propia (2025)

Finalmente, el modelo con el mejor F1-score ponderado validado en el conjunto de prueba fue seleccionado como el modelo final para las predicciones.

3.10.1. Entrenar y evaluar múltiples modelos

Luego cada modelo candidato, con sus hiperparámetros optimizados por GridSearchCV, fue entrenado utilizando el conjunto de datos de entrenamiento (X_train_scaled, y_train). Posteriormente, el rendimiento de cada modelo entrenado se evaluó rigurosamente sobre el conjunto de prueba (X_test_scaled, y_test), que representa datos no vistos durante el entrenamiento. Se calcularon métricas clave de clasificación, incluyendo Accuracy, Precisión, Recall y F1-score (utilizando las versiones

ponderadas para manejar adecuadamente el posible desbalance de clases entre estudiantes aprobados y reprobados como se puede ver en la figura 3.10-2).

```
grid.fit(X_train, y_train)
best_models[name] = grid.best_estimator_
logger.info(f"Mejores parámetros para {name}: {grid.best_params_}")

# Predecir en conjunto de prueba
y_pred = grid.best_estimator_.predict(X_test)
y_prob = grid.best_estimator_.predict_proba(X_test)[:, 1]

# Evaluar y almacenar resultados
accuracy = accuracy_score(y_test, y_pred)
# Usar zero_division=0 para evitar warnings si una clase no tiene predicciones
report = classification_report(y_test, y_pred, output_dict=True, zero_division=0)
report_str = classification_report(y_test, y_pred, zero_division=0) # Para loggear

results[name] = {
    'model': grid.best_estimator_,
    'params': grid.best_params_,
    'accuracy': accuracy,
    'f1_weighted': report['weighted avg']['f1-score'], # Guardar f1_weighted consistentemente
    'precision_weighted': report['weighted avg']['precision'],
    'recall_weighted': report['weighted avg']['recall'],
    'predictions': y_pred,
    'probabilities': y_prob
}
```

Figura 3.10-2: Entrenamiento implícito
Fuente: Elaboración propia (2025)

Adicionalmente, se realizó una validación cruzada final (cross_val_score con TimeSeriesSplit) sobre el modelo seleccionado como el mejor para verificar su estabilidad y generalización. Los resultados de la evaluación se visualizaron mediante gráficos comparativos, curvas ROC, curvas Precisión-Recall y matrices de confusión (ver figura 3.10-3).

```
logger.info(f"\nEvaluando robustez de {best_model_name} con validación cruzada...")
cv_scores = {}
tscv = TimeSeriesSplit(n_splits=5)  # Validación cruzada de series temporales
for metric_name, scoring in [('accuracy', 'accuracy'), ('f1', 'f1_weighted'),
                             ('precision', 'precision_weighted'), ('recall', 'recall_weighted')]:
    try:
        from sklearn.model_selection import cross_val_score
        scores = cross_val_score(best_model, X_train, y_train, cv=tscv, scoring=scoring)
        cv_scores[metric_name] = scores
        logger.info(f"CV {metric_name.capitalize()}: {scores.mean():.4f} (±{scores.std():.4f})")
    except Exception as e:
        logger.error(f"Error en validación cruzada para {metric_name}: {e}")
```

Figura 3.10-3: Entrenamiento implícito
Fuente: Elaboración propia (2025)

3.10.2. Proyectar tendencias futuras

El modelo predictivo entrenado se utiliza para generar una perspectiva sobre el riesgo de reprobación en el futuro inmediato. Específicamente, se aplica el mejor modelo seleccionado y el escalador ajustado a los datos del último año disponible (df_future). El modelo calcula la probabilidad de que cada

estudiante en ese conjunto de datos repruebe en el siguiente periodo académico (ver figura 3.10-4). Si bien esto no es una proyección de tendencias a largo plazo de la tasa de reprobación general, sí proporciona una "proyección" individualizada del riesgo basada en los patrones aprendidos del pasado y la situación más reciente de cada estudiante.

```
logger.debug("Prediciendo probabilidades de reprobación...")
try:
    risk_probs = best_model.predict_proba(X_risk_scaled)[:, 1]

    # Ajustar el nivel de confianza para estudiantes nuevos
    if 'Es_Estudiante_Nuevo' in df_risk.columns:
        nuevos_indices = df_risk['Es_Estudiante_Nuevo'] == 1
        logger.info(f"Ajustando predicciones para {nuevos_indices.sum()} estudiantes nuevos")

        # Aplicar un enfoque más conservador para estudiantes nuevos
        # Usar el promedio de probabilidad como base para estudiantes nuevos
        if nuevos_indices.any():
            promedio_prob = risk_probs[~nuevos_indices].mean() if (~nuevos_indices).any() else 0.5
            # Ajustar hacia el valor medio para reflejar mayor incertidumbre
            risk_probs[nuevos_indices] = (risk_probs[nuevos_indices] + promedio_prob) / 2
except Exception as e:
    logger.error(f"Error prediciendo probabilidades para identificación de riesgo: {e}")
    return pd.DataFrame(), pd.DataFrame()
```

Figura 3.10-4: Predicción de reprobados
Fuente: Elaboración propia (2025)

3.10.3. Identificando estudiantes en riesgo

Utilizando las probabilidades de reprobación proyectadas para cada estudiante en el último año de datos, se procede a identificar aquellos con mayor riesgo. Las probabilidades continuas (entre 0 y 1) se segmentan en categorías discretas de riesgo: 'Bajo', 'Medio', 'Alto' y 'Muy Alto' (esta manera de segmentar es bastante usada en estudios sobre estudios de rendimiento académico), utilizando umbrales predefinidos. Esto facilita la priorización de intervenciones. Se genera un listado (risk_df) que contiene el identificador del estudiante (Codigo Rude), su probabilidad de reprobación predicha y su nivel de riesgo asignado, ordenado de mayor a menor probabilidad como se puede ver en la figura 3.10-5. Adicionalmente, se generan gráficos para visualizar la distribución de estudiantes entre los diferentes niveles de riesgo, y opcionalmente, desglosados por género o nivel educativo.

```
# Clasificar nivel de riesgo
risk_bins = [0, 0.25, 0.5, 0.75, 1.0]
risk_labels = ['Bajo', 'Medio', 'Alto', 'Muy Alto']
risk_df['Nivel_Riesgo'] = pd.cut(
    risk_df['Probabilidad_Reprobacion'],
    bins=risk_bins,
    labels=risk_labels,
    right=True # Incluir el límite derecho (ej. 1.0 en 'Muy Alto')
)
```

Figura 3.10-5: Probabilidades continuas
Fuente: Elaboración propia (2025)

3.10.4. Perfiles de intervención

Para facilitar acciones pedagógicas proactivas, se generan perfiles más detallados para un número determinado (top 5) de los estudiantes identificados con riesgo 'Alto' o 'Muy Alto'. Estos perfiles van más allá de la simple probabilidad de riesgo, incorporando información contextual relevante del último

año (como el promedio obtenido), métricas históricas clave (reprobaciones acumuladas, años en sistema), una comparación de su rendimiento con el promedio de su curso/nivel, una lista de posibles factores de riesgo identificados heurísticamente y un conjunto de intervenciones recomendadas específicas para esos factores (esto mismo acordado con la directora y con la revisión de la psicóloga de la defensoría de la niñez y adolescencia de la comunidad). El objetivo es proporcionar a los profesores una visión más completa de la situación del estudiante para guiar la intervención.

3.10.5. Identificar factores de riesgo

Para comprender mejor por qué un estudiante es clasificado como de alto riesgo, se implementó un sistema heurístico que analiza su perfil individual. Esta función (_identify_risk_factors) revisa características clave como el historial de reprobaciones acumuladas, si su promedio del último año está por debajo de un umbral crítico (ej., 70), si su rendimiento ha sido muy inestable (alta volatilidad), o si su desempeño está significativamente por debajo del promedio de su curso como se puede apreciar en la figura 3.10-6. Se genera una lista legible de estos factores identificados para cada estudiante de alto riesgo. Complementariamente, el análisis de importancia de características (feature importance) derivado de los modelos de árbol (como Random Forest, XGBoost) proporciona una visión global de qué variables tienen mayor poder predictivo sobre la reprobación en general.

```python
def _identify_risk_factors(student_row):
    """Identifica los factores que contribuyen al riesgo del estudiante (heurístico)."""
    risk_factors = []

    # Considerar si es estudiante nuevo
    if student_row.get('Es_Estudiante_Nuevo', 0) == 1:
        risk_factors.append("Estudiante nuevo (sin historial académico previo)")

    # Usar .get(key, default) para evitar KeyError si la columna no existe
    if student_row.get('Reprobaciones_Acumuladas', 0) > 0:
        risk_factors.append("Historial de reprobaciones")

    # Usar el promedio del último año si existe, si no, no usar esta regla
    promedio_check = student_row.get('Promedio_Ultimo_Año', 100) # Default alto para no activar si no existe
    if promedio_check < 70: # Umbral hardcodeado
        risk_factors.append("Promedio bajo (<70)")

    # Usar Volatilidad si existe
    volatilidad_check = student_row.get('Volatilidad_Rendimiento', 0) # Default bajo si no existe
    if volatilidad_check > 10: # Umbral hardcodeado
        risk_factors.append("Rendimiento inestable (>10 std dev)")

    # Usar diferencia vs curso si existe y es válida
    diff_check = student_row.get('Diferencia_vs_Promedio_Curso', 0) # Default 0 si no existe o es NaN
    if pd.notna(diff_check) and diff_check < -5: # Umbral hardcodeado
        risk_factors.append("Rendimiento bajo vs curso (<-5 pts)")

    # Si no hay factores identificados, agregar uno genérico
    if not risk_factors:
        risk_factors.append("Riesgo multifactorial (probabilidad alta sin factor dominante claro)")

    return ", ".join(risk_factors)
```

Figura 3.10-6: Factores de riesgo
Fuente: Elaboración propia (2025)

3.10.6. Guardado de resultados

Todos los artefactos y resultados importantes generados durante el pipeline se guardan para su posterior análisis, reporte y uso. Esto incluye: un archivo de log (student_analysis.log) que registra el flujo de ejecución, advertencias y errores; las diversas visualizaciones generadas (comparación de modelos, curvas ROC/PR, matrices de confusión, distribución de riesgo) guardadas como archivos de imagen (PNG); la tabla de datos preprocesada y lista para modelar (tabla_minable.csv, si se sigue ese paso del flujo completo); y fundamentalmente, los resultados finales de la predicción, como el listado de estudiantes en riesgo con sus probabilidades y niveles (risk_df) y las predicciones completas (full_predictions_df), típicamente guardados en formato Excel o CSV. Adicionalmente, se contempla la posibilidad de guardar el objeto del modelo entrenado final y el objeto escalador (scaler) para poder reutilizarlos en futuras predicciones sin necesidad de reentrenar.

3.11. Evaluación y comunicación

Para esta parte, usaremos el método main(), este ejecuta todas las funciones creadas para realizar las predicciones como podemos apreciar en la figura 3.11-1.

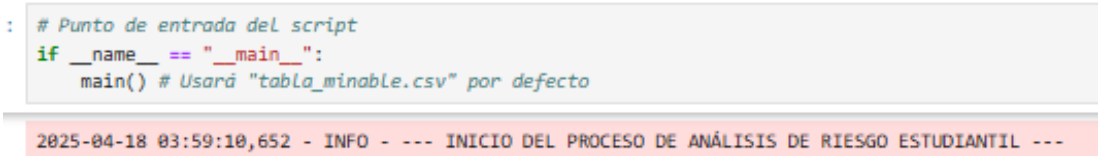


Figura 3.11-1: Factores de riesgo
Fuente: Elaboración propia (2025)

3.12. Selección de herramientas

3.12.1. Jupyter notebook

Jupyter notebook proporciona un manejo de uso fácil en su interfaz, es gratuita y funciona en una maquina local, para este proyecto se tiene la versión 3.13.3 de Python y jupyter en su versión 7.2.2.

Es sabido que existen muchas herramientas para realizar analítica de datos y machine learning, está por su parte hace que inspeccionar, limpiar y transformar sea a un nivel más profundo y con el control total de lo que se realiza, desde luego ya viene preparada con librerías que solo se necesitan importar para empezar a usar, también vemos que existen una gran posibilidad de librerías compatibles y descargables para su uso posterior.

3.12.2. Tableau public

Para este proyecto se usó la versión 2024.3.4 de tableau public, esta herramienta es la versión libre y gratuita de tableau desktop. Esta proporciona visualizaciones interactivas con narraciones visuales, además de esto tiene una interfaz muy agradable y fácil de manejar. Esta herramienta es bastante usada e integral en conjunto con jupyter notebook.

4. Análisis de Resultados y Discusión

En esta etapa se evalúa la efectividad del mejor modelo frente a otros estudios, se hace énfasis en diferentes analíticas como ser: analítica descriptiva, analítica diagnostica y analítica predictiva, esto con el fin de cumplir los objetivos específicos.

Estos resultados servirán para actuar tempranamente en casos de riesgo de reprobación del/la estudiante, teniendo en cuenta su historial calificativo.

4.1. Resultados y análisis de Recolección y consolidación de datos académicos

Se recolectaron registros de calificaciones anuales finales de estudiantes de la Unidad Educativa San José Obrero, con un total de 2978 filas y 31 columnas correspondientes a 10 gestiones académicas como se ve en el gráfico 4.1-1, y fue estructurada con atributos como nivel, curso, materia, promedio, gestión y estado (Aprobado/Reprobado), según este gráfico podemos indicar que el nivel primario es el que a lo largo de los años tiene mayor cantidad de estudiantes inscritos con excepción 2020. Vemos que a partir del año 2022 existe una tendencia minima al aumento de la cantidad de estos estudiantes, por otro lado, el nivel secundario los últimos tres años parece aumentar en cantidad, mientras que no los años anteriores sube y baja de forma abrupta.

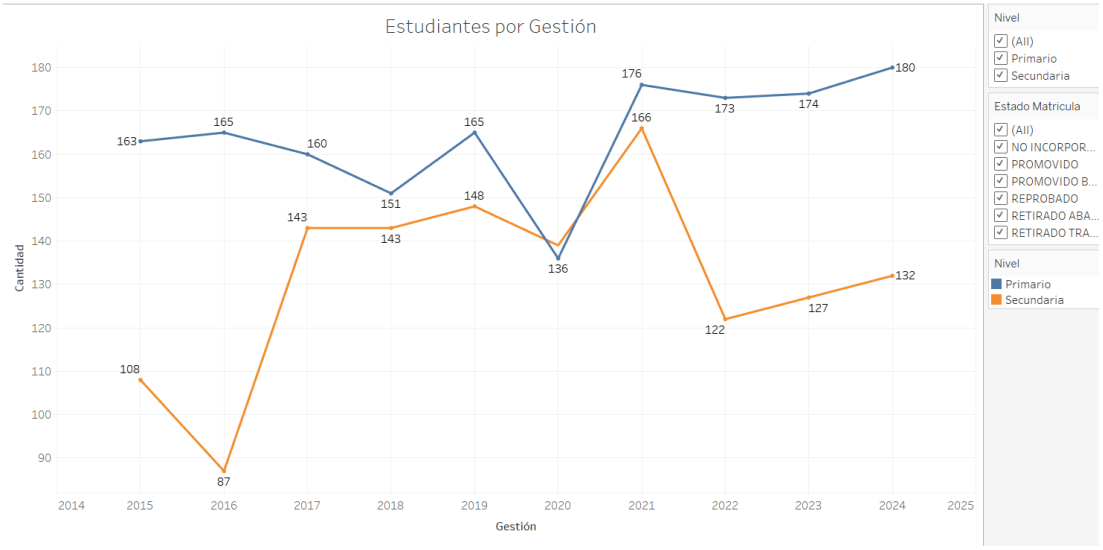


Figura 4.1-1: Cantidad de estudiantes por nivel y gestión educativa
Fuente: Elaboración propia (2025)

A diferencia del estudio de (Marquez Vera, 2015), quien incorporó también variables conductuales, contextuales y con 419 datos de estudiantes inscritos en el programa II de la UAPUAZ del primer semestre del año 2012, este estudio demuestra que es posible obtener modelos predictivos efectivos aun con un conjunto de variables puramente académicas.

4.2. Resultado y análisis de Organizar y limpiar datos de las calificaciones

Se detectaron múltiples inconsistencias (ver figura 4.2-1): valores nulos en calificación de materias, promedios, apellidos de demás, esto ocurrió debido a que no todos los estudiantes pasan las clases de todas las materias, también debido a que existen estudiantes que se retiran y realizan el traspaso a otras unidades educativas. Se aplicó limpieza mediante la codificación en Python, eliminando duplicados y filas vacías, creando posteriormente nuevas variables a partir de estas.

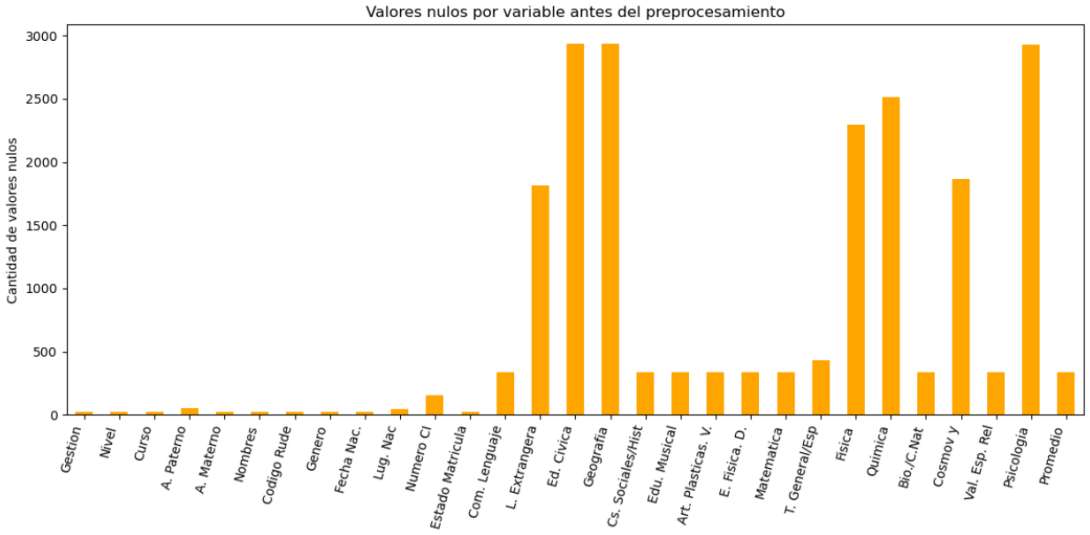


Figura 4.2-1: Valores nulos por variables antes del procesamiento
Fuente: Elaboración propia (2025)

Márquez Vera, destaca en su estudio la importancia de este paso, señalando que la calidad de los datos influye directamente en la precisión de los modelos.

4.3. Resultado y análisis de Examinar la evolución del rendimiento académico de los estudiantes

Se observó un aumento significativo a 31 reprobados en el año 2023, siendo de estos 15 estudiantes de secundaria y 15 de primaria, la figura 4.3-1 muestra el porcentaje de estudiantes reprobados por año, siendo la taza de este 9.97%.

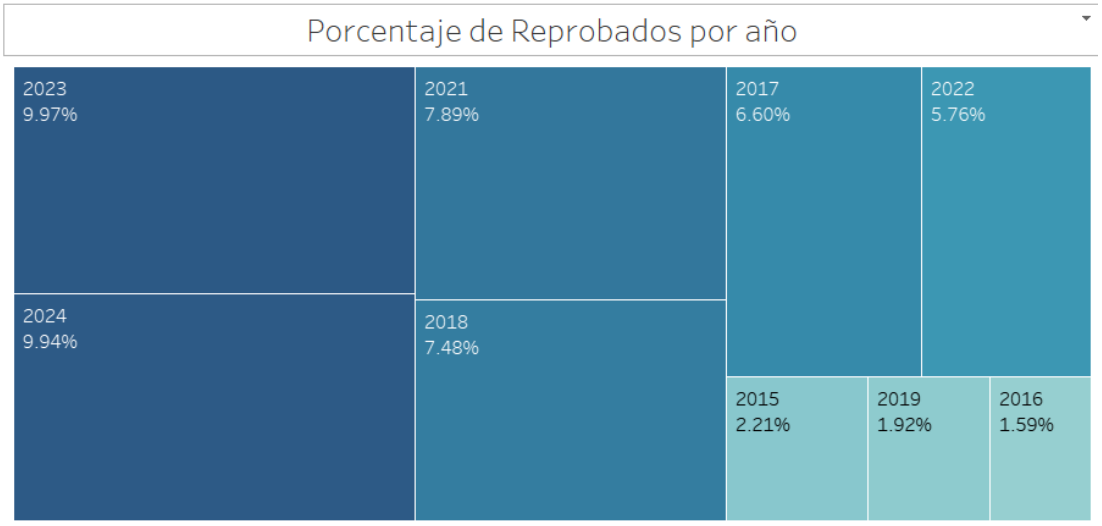


Figura 4.3-1: Porcentaje de reprobados por año
Fuente: Elaboración propia (2025)

Vemos que no aparece el año 2020 en la figura 4.3-1, esto fue debido a que este año por el motivo de la pandemia del COVID-19 se aprobó de forma automática a todos los estudiantes, los dos últimos años apreciamos como las tendencias de estudiantes reprobados aumentan en 0.03%, en las unidades educativas de educación regular se deben aprobar todas las materias, en caso de reprobar alguna se reprueba el año escolar. Si comparamos la cantidad de estudiantes del año 2023 (301 estudiantes) y 2024 (312 estudiantes) vemos que el primero tiene menos estudiantes inscritos como veos en la figura 4.1-1, pero solo uno menos como reprobado.

En cuanto a la tasa de reprobación general, se tiene el 5.51% como se puede ver en la figura 4.3-2, para el caso de los estudiantes del nivel primario un total de 4.5% estudiantes, para el nivel secundario el 6.77% siendo este el mayor.

Figura 4.3-2: Porcentaje total de reprobados
Fuente: Elaboración propia (2025)

A nivel de unidades educativas del sistema regular, este porcentaje de reprobados es enorme ya que indica que 6 de cada 100 estudiantes reprueban algún año escolar.

En comparativa, el proyecto de Márquez obtiene una mayor tasa de reprobados, es decir un 9% como se puede ver en la figura 4.3-3.

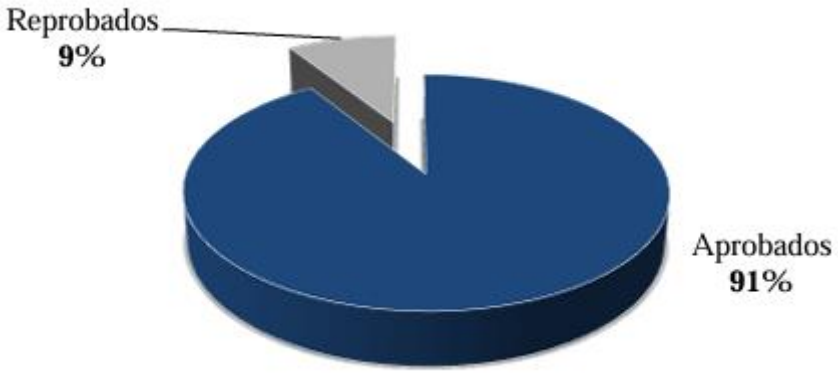


Figura 4.3-3: Distribución del resultado académico final de los estudiantes
Fuente: Carlos Márquez Vera (2015)

Márquez Vera también identificó variaciones en el rendimiento académico asociadas a contextos institucionales, aunque su enfoque se centró más en el abandono, algo que no se puede dejar de lado es que las calificaciones en México son entre 0 – 10, en Bolivia el rango es desde 0 - 100.

En el caso del promedio, para el caso de primaria el año con el más bajo promedio fue 2019, para secundaria el año 2021 como se puede ver en la figura 4.3-4.

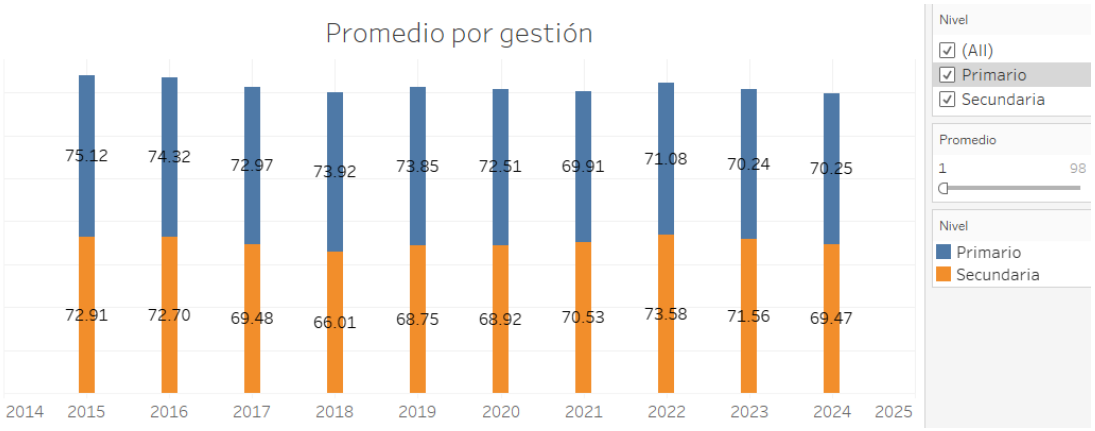


Figura 4.3-4: Promedio por nivel educativo y gestión
Fuente: Elaboración propia (2025)

Según el análisis de promedios generales por año, podemos apreciar que el año con mejor rendimiento académico a nivel general fue el año 2015 con un promedio general de 74.24 puntos como podemos ver en el gráfico 4.3-5.

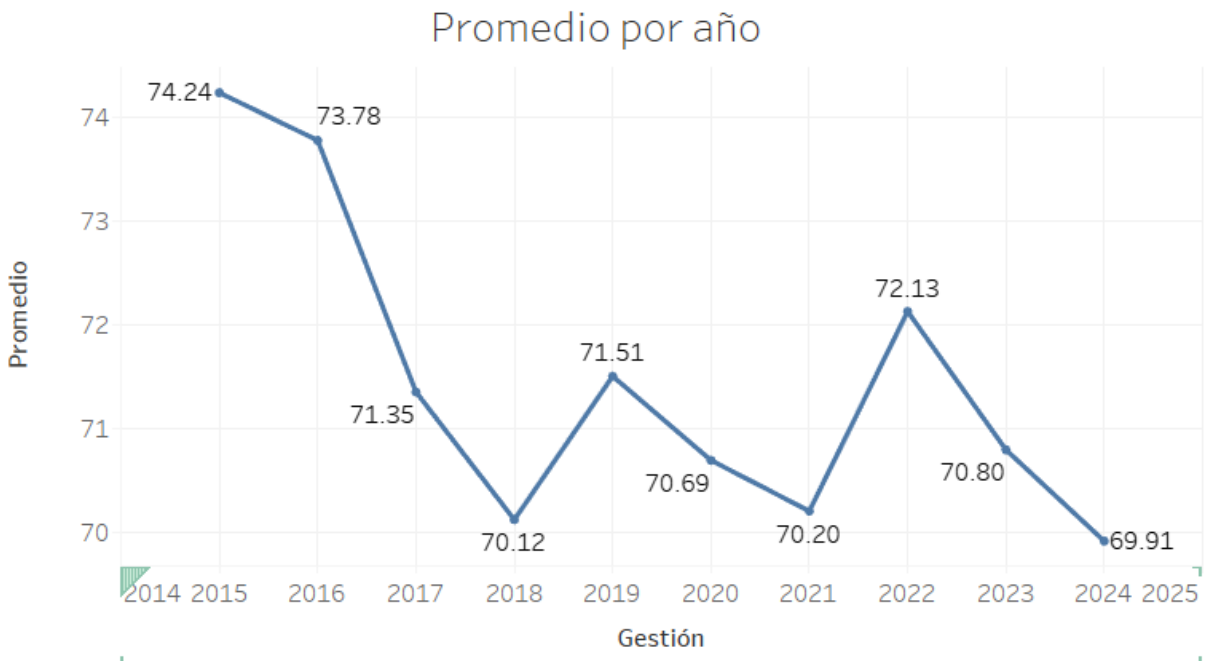


Figura 4.3-5: Promedio general por año
Fuente: Elaboración propia (2025)

Como parte importante de este apartado tenemos el seguimiento anual y por curso de cada estudiante, viendo los años y el promedio en dicha gestión, con este gráfico podemos apreciar si un estudiante

mejora o empeora si nivel educativo basado en la calificación final, en este ejemplo se muestra el histórico de promedios de una estudiante como se ve en la figura 4.3-6.

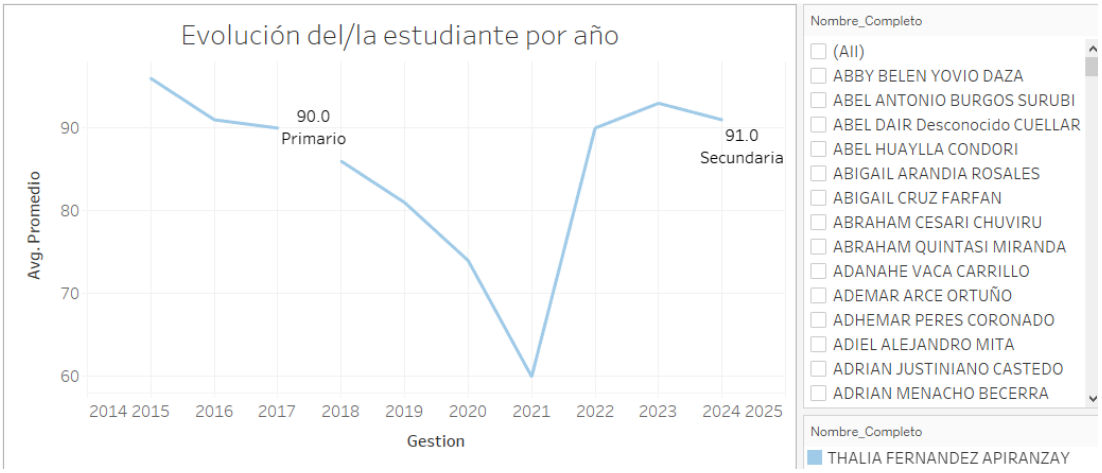


Figura 4.3-6: Evolución del/la estudiante por año
Fuente: Elaboración propia (2025)

Podemos apreciar en la figura 4.3-7 que los promedios están concentrados entre 60 – 80 puntos.

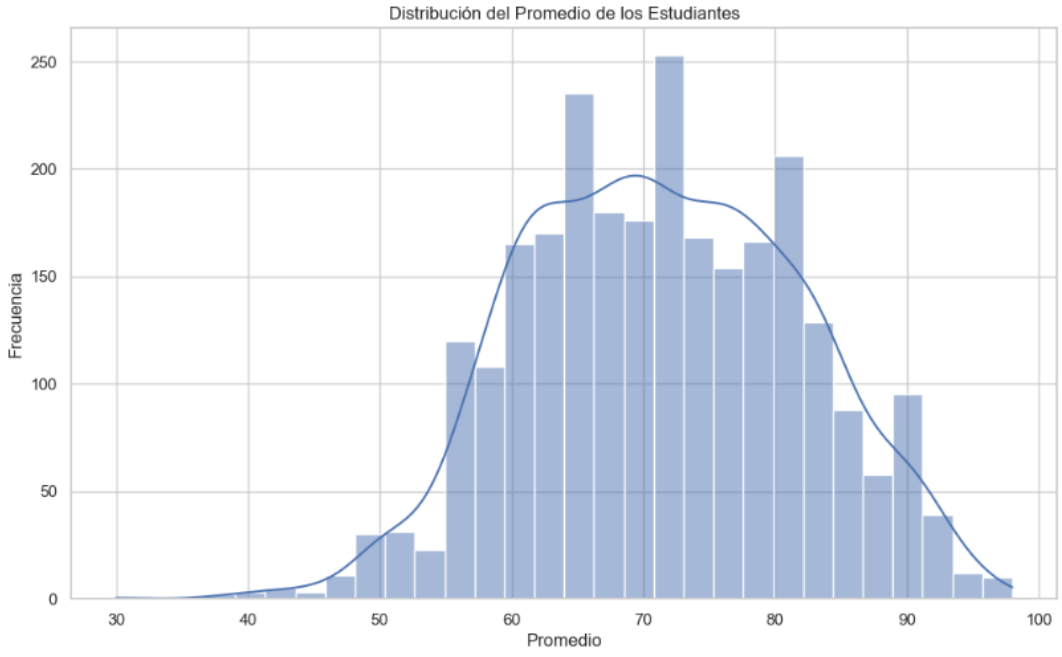


Figura 4.3-7: Distribución del promedio de los estudiantes
Fuente: Elaboración propia (2025)

Vemos que la tendencia del promedio general de la unidad educativa tiene una tendencia que en los últimos años ha descendido, siendo el año 2024 el más bajo de todos los años. Los promedios de los

estudiantes están aglomerados en su mayoría entre 60 y 80 puntos respectivamente y son pocos los estudiantes que tienen calificados promedios anuales mayores a 90 puntos.

En el análisis de Márquez Vera podemos ver que el promedio general es de 3, lo que a simple vista se distingue como bajo (ver tabla 4.3-1), si bien es cierto que este autor toma también otros aspectos, estas frecuencias solo salen a partir de un corto tiempo, mientras que en el caso de los resultados de este proyecto son el resultado de 10 gestiones o años, lo que hace que sea mucho más real que el contraste.

Variable/atributo	Frecuencia
Nota en Humanidades 1, Nota en Inglés 1.	10
Nota en Ciencias Sociales 1, Nota en Matemáticas 1, Nota en Taller de lectura y redacción 1, Nota en Física 1, Nota en Computación 1.	9
Nivel de motivación.	5
Promedio en la secundaria	3
Edad, Número de hermanos, Grupo, Fumas, Promedio EXANI I.	2
Estudia en grupo, Estado civil, Tiempo de ejercicio, Nota en Historia.	1

Tabla 4.3-1: Variables/atributos de mayor influencia organizada por frecuencia de aparición
Fuente: Carlos Márquez Vera (2015)

4.4. Resultado y análisis de Identificar las materias que representan mayor dificultad para estudiantes

Podemos determinar que para el caso de primaria las materias que representan promedios más bajos son: comunicación y leguaje con 68.04 de promedio y Matemática con 68.34 respectivamente, por lo contrario, podemos notar que los estudiantes tienen un mejor desempeño en Educación Musical y Educación Física como podemos ver en la figura 4.4-1.

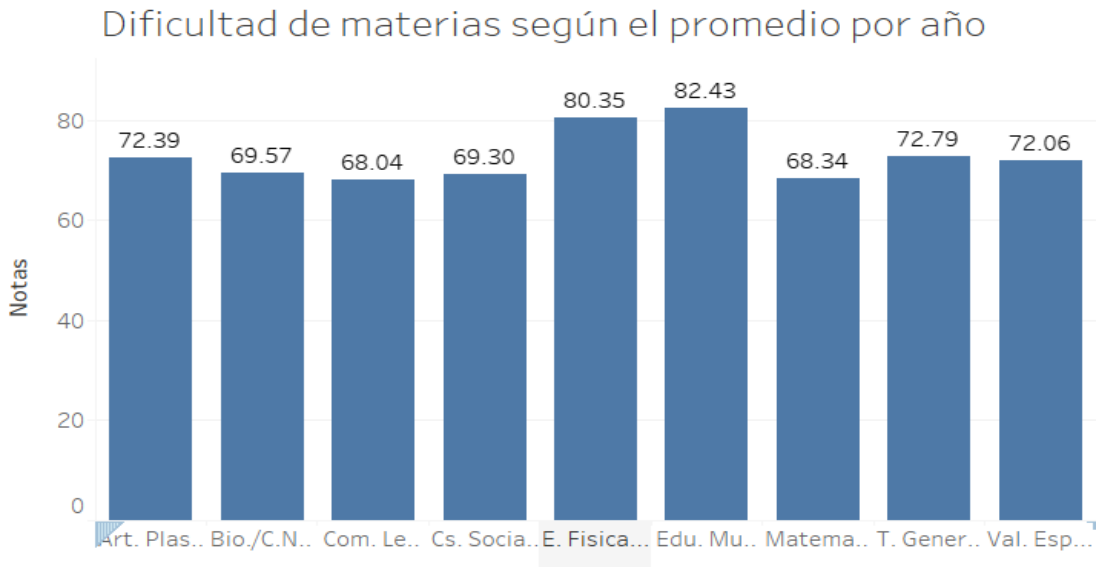


Figura 4.4-1: Dificultad de materias de primaria según el promedio por año

Fuente: Elaboración propia (2025)

Para el caso del nivel secundario, la figura 4.4-2 nos muestra que las materias con menor promedio son: Psicología con 63.25 puntos y Comunicación y lenguaje con 65.52 puntos de promedio, por contraste las que mayor promedio tienen son: Educación musical y física con 81.01 el primero y 78.18 el segundo.

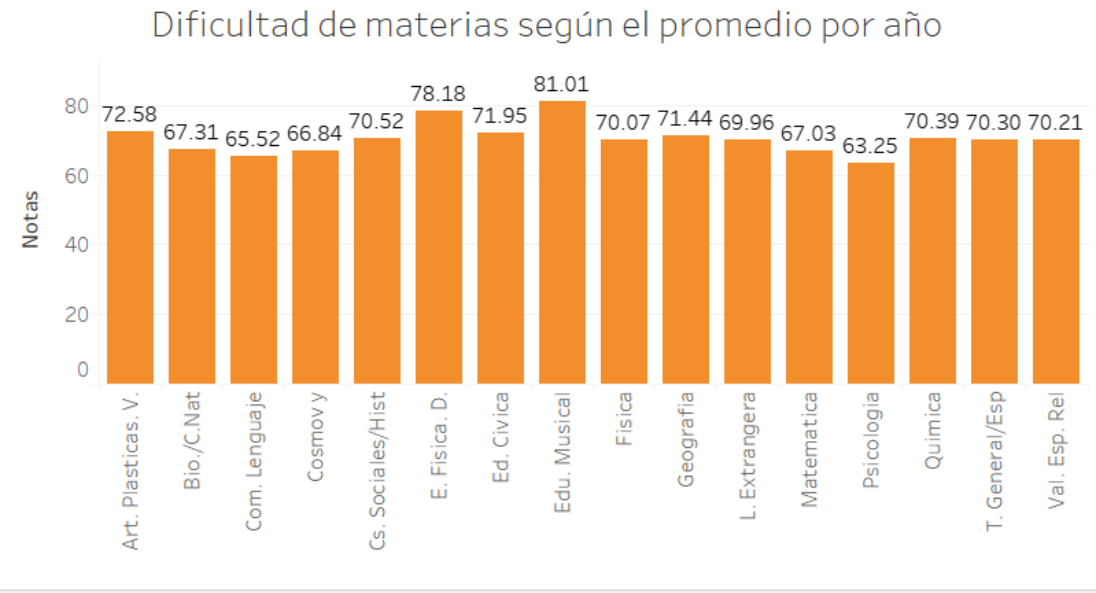


Figura 4.4-2: Dificultad de materias de secundaria según el promedio por año
Fuente: Elaboración propia (2025)

En general se puede tener como resultado que los estudiantes de la U.E. tienen dificultad en la materia de Comunicación y lenguajes. Las materias que contienen mayor promedio son Educación física y musical. Matemáticas, física y química también representan promedios relativamente bajos, esto se alinea bastante con estudios internacionales como el ERCE (2019) donde Bolivia obtuvo puntuaciones bajas en áreas científicas, Una fuerte posible causa de esto puede llegar a ser las barreras de comprensión acumuladas desde el nivel inicial y primario, lo que llega a repercutir en el nivel secundario y posteriormente en estudios superiores. Márquez Vera por su lado, no examina de forma específica, más bien aglomera las materias y variables por frecuencias.

Según el porcentaje total de reprobados a lo largo de los últimos 10 años, podemos ver que el 17.24% de los estudiantes reprobó Biogeografía y Ciencias naturales, el 16.55% matemáticas, siendo estas dos las que han tenido la mayor cantidad de reprobados como podemos ver en el gráfico 4.4-3.

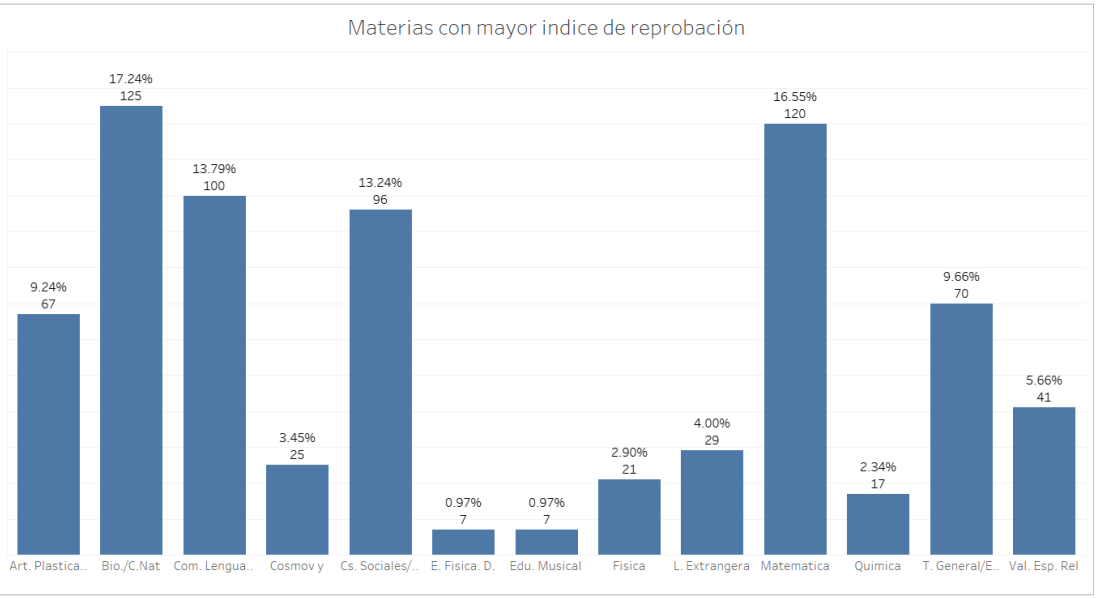


Figura 4.4-3: Materias con tasa de reprobación
Fuente: Elaboración propia (2024)

Como vimos anteriormente, Comunicación y lenguaje tiene el promedio más bajo tanto en primaria y secundaria, esto repercute en la tasa de reprobación ya que la comprensión lectora, análisis y puntos de vista son importantes para materias como Biogeografía/Ciencias Naturales y matemáticas. Esto confirma que estas falencias se originan en el nivel primario y repercuten en el nivel secundario.

4.5. Resultado y análisis de Construir un modelo con el fin de predecir e indicar a los estudiantes con mayor probabilidad de reprobar

Se entrenaron varios modelos de machine learning, entre estos tenemos a: XGBoost, MLP, CatBoost y demás, para el caso del dataset usado actualmente el mejor modelo fue CatBoost con 0.8408, esta elección basada en F1-Score (Weighted), en la tabla 4.5-1 se detallan los resultados de todos los modelos.

Modelo	Accuracy	F1_Weighted	Precision_Weighted	Recall_Weighted
Logistic Regression	0.701923	0.752104	0.889228	0.701923
Random Forest	0.870192	0.814267	0.765097	0.870192
XGBoost	0.807692	0.807692	0.807692	0.807692
Gradient Boosting	0.870192	0.829765	0.821552	0.870192

SVM	0.807692	0.816028	0.825639	0.807692
MLP	0.865385	0.811856	0.764563	0.865385
LighGBM	0.831731	0.823920	0.817131	0.831731
CatBoost	0.850962	0.840776	0.833090	0.850962

Tabla 4.5-1: Resultado del entrenamiento por modelo
Fuente: Elaboración propia (2025)

En cuanto a la validación cruzada, la robustez de CatBoost (mejor modelo) obtuvo un CV – Accuracy de 0.9182, FI: 0.9149, CV Precision: 0.9149 y CV Recall: 0.9182.

El proyecto de Márquez Vera, prueba también con muchos modelos de machine learning, en el caso de este su métrica de evaluación principal son las TP (verdaderos positivos) como se muestra en la tabla 4.5-2.

Algoritmo	TP_{rate}	TN_{rate}	*Acc*	*GM*	*#Reglas*	*#Condiciones por regla*	*#Condiciones*
JRip	97.0	81.7	95.7	89.0	5.7	1.5	8.7
NNge	98.0	76.7	96.1	86.7	22.2	14.0	310.8
OneR	98.9	41.7	93.7	64.2	2.0	0.8	1.6
Prism	**99.2**	44.2	94.7	66.2	55.6	1.7	93.8
Ridor	95.6	68.3	93.1	80.8	4.0	1.2	5.4
ADTree	**99.2**	78.3	**97.3**	88.1	21.0	3.0	63.0
J48	97.7	55.5	93.9	73.6	19.9	2.1	43.0
RandomTree	98.0	63.3	94.9	78.8	278.6	3.3	912.2
REPTree	97.9	60.0	94.5	76.6	30.0	1.9	68.4
SimpleCart	98.0	65.0	95.1	79.8	6.9	4.1	29.4
ICRM v1	92.0	**93.3**	92.1	**92.5**	2.0	2.4	4.9
ICRM v2	97.2	71.7	94.9	82.8	8.2	2.1	17.9
ICRM v3	75.9	85.0	76.7	79.0	4.0	0.9	3.8

Tabla 4.5-2: Resultado del entrenamiento de los modelos
Fuente: Márquez Vera (2015)

En cuanto a la matriz de confusión, los resultados de Verdaderos positivos son 170, falsos positivos 12, falsos negativos 19 y verdaderos positivos 7, en términos generales CatBoost tuvo un buen desempeño para identificar a los No reprobados como podemos ver en la figura 4.5-1.

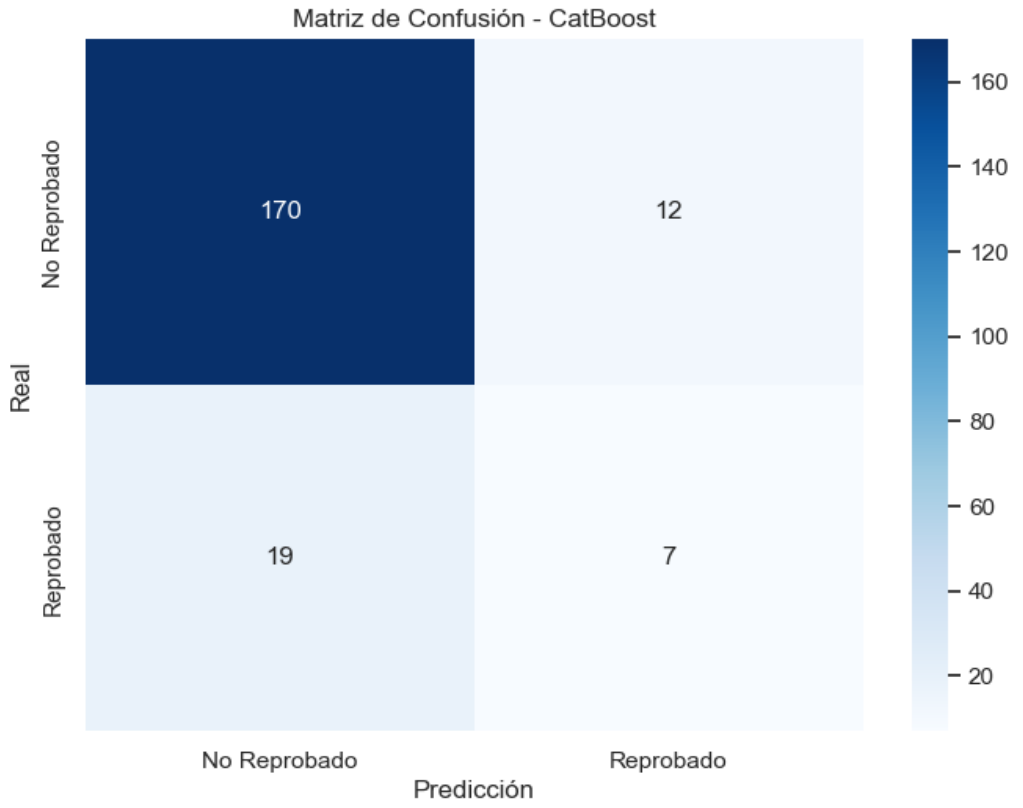


Figura 4.5-1: Matriz de confusión del mejor modelo
Fuente: Elaboración propia (2025)

En el caso de Márquez Vera identifica esta matriz como medidas de clasificación, para los verdaderos positivos tiene un total de 56, verdaderos negativos 363, falsos positivos y negativos 0 y 1 respectivamente como podemos ver en la tabla 4.5-3.

Resultados de las medidas de Clasificación:
Matriz de Confusión.

Actual vs Predicción	APROBÓ	REPROBÓ
APROBÓ	363	0
REPROBÓ	1	56

Tabla 4.5-3: Matriz de confusión de Márquez Vera
Fuente: Márquez Vera (2015)

A partir de estas predicciones, tenemos que para la gestión 2025 tendremos 23 reprobados, para el año 2026 17 y para 2027 12 estudiantes respectivamente como podemos ver en la figura 4.5-2.

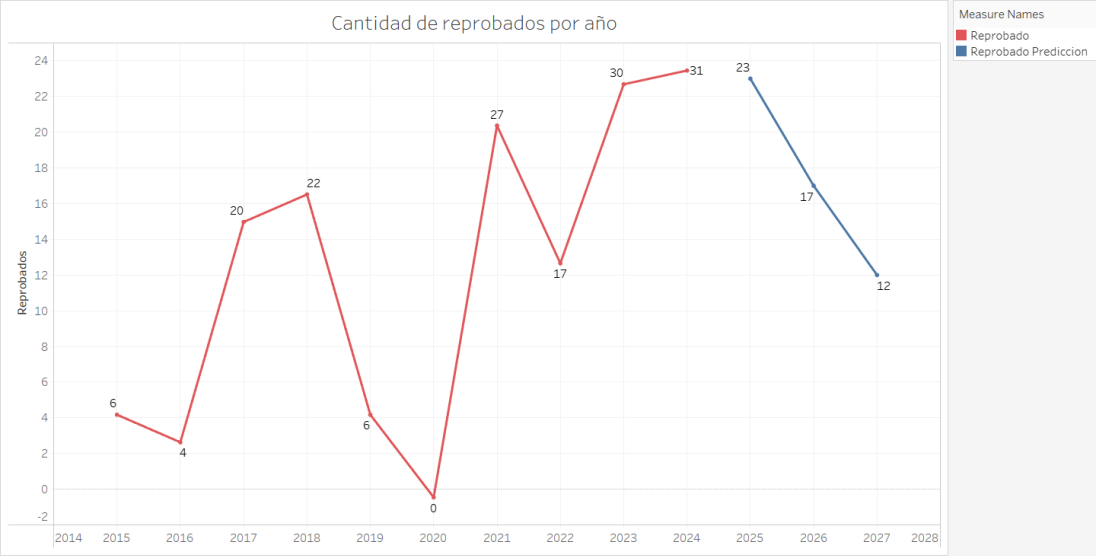


Figura 4.5-2: Gráfico de estudiantes reprobados + predicción
Fuente: Elaboración propia (2025)

Las predicciones no terminan en este punto, también se logró identificar que estudiantes tienen la mayor probabilidad de reprobar y dar una recomendación básica para intervenir como se ve en la figura 4.5-3.

Intervensión a estudiantes

Nombre_Completo	Nivel Rie..	Intervenciones Recomendadas	
ADRIANA BEJARANO CONDORI	Alto	Refuerzo académico específico, Monitoreo continuo del progreso	Secundaria
BEATRIZ CAMPOS CUELLAR	Muy Alto	Refuerzo académico específico, Monitoreo continuo del progreso	Secundaria
BRIGITTE AILEN CAMACHO URQ..	Alto	Refuerzo académico específico, Monitoreo continuo del progreso	Primario
DAYANA FIORELLA ARAUZ APAR..	Muy Alto	Refuerzo académico específico, Monitoreo continuo del progreso	Secundaria
DILAN PERES ROCA	Alto	Refuerzo académico específico, Monitoreo continuo del progreso	Primario
EZEQUIEL CLAURE CUELLAR	Muy Alto	Refuerzo académico específico, Monitoreo continuo del progreso	Primario
FAVIO ANDRES SURUBI VACA	Muy Alto	Refuerzo académico específico, Integración a grupos de estudio, Monitoreo continuo del progreso	Secundaria
GUILLERMO VERA TELMO	Muy Alto	Refuerzo académico específico, Monitoreo continuo del progreso	Secundaria
HIROSHI ISHIZAKI ORELLANO	Alto	Refuerzo académico específico, Monitoreo continuo del progreso	Secundaria
JAZMIN ALBA CORTEZ	Alto	Refuerzo académico específico, Monitoreo continuo del progreso	Primario
JOSE EDUARDO PEDRAZA PEDR..	Muy Alto	Refuerzo académico específico, Integración a grupos de estudio, Monitoreo continuo del progreso	Secundaria
MARCO ANTONIO SOLIZ SAMEJA	Alto	Refuerzo académico específico, Integración a grupos de estudio, Monitoreo continuo del progreso	Primario
MARIA GUADALUPE VACA ORRU..	Muy Alto	Refuerzo académico específico, Monitoreo continuo del progreso	Primario
MATIAS JOB BARRANCO JUSTIN..	Muy Alto	Refuerzo académico específico, Monitoreo continuo del progreso	Secundaria
RUBEN SOLIZ PEÑA	Alto	Refuerzo académico específico, Monitoreo continuo del progreso	Secundaria
SARA JIMENEZ PEREIRA VARGAS	Muy Alto	Refuerzo académico específico, Monitoreo continuo del progreso	Primario
VALENTINA SIANCAS CAMACHO	Alto	Refuerzo académico específico, Monitoreo continuo del progreso	Primario
YANETH YAMBAMINI SOLIZ	Alto	Refuerzo académico específico, Monitoreo continuo del progreso	Primario
YESSICA CHAMO ESTRADA	Muy Alto	Refuerzo académico específico, Monitoreo continuo del progreso	Secundaria
YOLVER CRESPO RAMOS	Alto	Refuerzo académico específico, Integración a grupos de estudio, Monitoreo continuo del progreso	Secundaria

Figura 4.5-3: Predicción de estudiantes con mayor riesgo de reprobación
Fuente: Elaboración propia (2025)

No solo se cuenta con los estudiantes con alto y muy alto porcentaje de reprobación, también podemos filtrar y ver a los estudiantes con riesgo bajo y medio como se ve en la figura 4.5-4.

Estudiantes en riesgo proximo año

Nombre_Completo	Nivel (predicciones completas proximo a..	Curso (predicciones ..	
YULIANA URABEBEY CORREA	Secundaria	6to A	0.03%
YULEISA YOPIE RIOS	Secundaria	2do A	0.06%
YOLVER CRESPO RAMOS	Secundaria	4to A	59.18%
YESSICA CHAMO ESTRADA	Secundaria	4to A	93.27%
YASMIN YAMBAMINI SOLIZ	Secundaria	5to A	0.76%
YANELA ORRURI TOMICHA	Secundaria	2do A	12.91%
YALIXA LINO YOPIYE	Secundaria	1ro A	0.17%
VALENTINA ARAUZ TOMICHA	Secundaria	2do A	0.15%
THALIA FERNANDEZ APIRANZAY	Secundaria	6to A	0.01%
STEVEN MANUEL EGUEZ SAUCEDO	Secundaria	5to A	0.05%
SIMON PAREDES CUELLAR	Secundaria	2do A	0.16%
SEVERO URAESAÑA RODRIGUEZ	Secundaria	3ro A	12.72%
SERGIO SANCHEZ MICHEL	Secundaria	4to A	0.01%
SEBASTIAN MASAY URAEZAÑA	Secundaria	6to A	1.65%
SAUL MENDOZA CRUZ	Secundaria	3ro A	7.24%
SANTIAGO URQUIZA LOPEZ	Secundaria	3ro A	18.33%
RUTH MARY TITO ARTEAGA	Secundaria	4to A	0.05%
RUBEN SOLIZ PEÑA	Secundaria	1ro A	65.29%
ROSY CATARI MARTINEZ	Secundaria	5to A	0.89%
ROSARIO LINO CUELLAR	Secundaria	2do A	4.97%
ROSARIO AÑEZ CESARI	Secundaria	1ro A	1.87%
ROLY GONZALES COCA	Secundaria	1ro A	4.51%
PARIS YHERALDINE MEJIA SALAZAR	Secundaria	2do A	1.91%
PABLO MONTERO AGUILERA	Secundaria	6to A	0.02%
NELSON PEÑARANDA GOMEZ	Secundaria	5to A	7.29%
NELSON CONDORI TICONA	Secundaria	4to A	0.62%
NATALY SOSA PATICU	Secundaria	2do A	3.21%
MOISES CHOQUE ROCA	Secundaria	1ro A	34.94%
MITSUE EMIKA NAGATANI DORADO	Secundaria	6to A	0.01%
MICHELLE ESTEFANY MEJIA SALAZAR	Secundaria	4to A	0.03%
MERLIN YESSICA FERREIRA CHORE	Secundaria	2do A	14.53%

Nivel Riesgo

- (All)
- Alto
- Bajo
- Medio
- Muy Alto

Figura 4.5-4: Estudiantes con riesgo de reprobación
Fuente: Elaboración propia (2025)

Los modelos predictivos pueden ayudar a identificar a los estudiantes en riesgo de reprobación con una buena precisión y no solo basta probar con uno o dos modelos, tener varios modelos y comparar las métricas de salida para luego seleccionar el mejor de todos ayuda a que nuestra predicción sea mucho más acertada, en este caso particular CatBoost fue el mejor, pero esto no siempre es así ya que la cantidad de datos, los datos que se tienen, pueden hacer que la precisión de los modelos cambien y la predicción sea totalmente diferente, tal es el caso de Marquez Vera que tenía datos diferentes y su enfoque fue también de ese modo , los modelos que uso también fueron diferentes lo que hace que no haya mucho que comparar con este proyecto en particular.

5. Conclusiones

El análisis exploratorio permitió descubrir que los estudiantes de género femenino tienen mayor promedio (73.73 puntos). La evolución de promedios generales permitió identificar que el año 2024 fue el año en el que se obtuvo el promedio más bajo (69.91 puntos) desde el año 2015 al presente.

El año 2021 esta unidad educativa tuvo mayor cantidad de estudiantes (342 inscritos) que otros años, en especial en el nivel secundario. La gestión con mayor cantidad de reprobados fue el año 2024 (31 reprobados), pero con relación a la cantidad de estudiantes inscritos por año, podemos concluir que el 2023 tuvo menos estudiantes inscritos y la cantidad de estudiantes reprobados fue proporcionalmente mayor (9.97%).

Las materias que representaron mayor complicación según el promedio para los estudiantes fueron Psicología (63.25 puntos de calificación), biogeografía/Ciencias naturales y matemáticas, para el nivel primario comunicación y lenguaje (42.97 puntos), para secundaria psicología y comunicación y lenguaje.

Basándonos en la cantidad de reprobados vemos que las materias de artes plásticas, biogeografía, comunicación y lenguaje, ciencias sociales, educación física, música, matemáticas, técnica general y especializada, valores espiritualidad y religión son las que tienen más cantidad de estudiantes reprobados (163 reprobados en total), el nivel secundario tiene mayor proporción de estudiantes que no aprueban el año escolar (6.77%), a nivel general se tiene 5.51% de estudiantes reprobados.

Para los años posteriores se prevé que reprueben 23 estudiantes el año 2025, 17 para el año 2026 y 12 para el 2027, la tendencia de estudiantes reprobados baja con el paso del tiempo, se identifican de forma temprana a estos estudiantes con mayor probabilidad para que los profesores tomen acción temprana sobre estos estudiantes.

A nivel general se puede concluir que el nivel secundario tiene más estudiantes reprobados y se prevé que este mismo tiene más estudiantes con alta probabilidad de reprobar, esta predicción específicamente salida a aparir del modelo CatBoost basado en F1-Score Weighted (0.8408), aunque esto puede cambiar según la cantidad y calidad de los datos.

El modelo cumple con lo que se quiere predecir que es identificar a los estudiantes específicamente y tener la cantidad por año. Con todo esto sumado a la analítica se cumplen con los objetivos de este proyecto.

6. Recomendaciones

Para tener un buen desempeño de la analítica y reprobación de los estudiantes de la unidad educativa, se recomienda tener los datos de las calificaciones anuales actualizadas. Tener datos muchos más limpios desde la fuente de datos original ayudaría bastante en una mayor precisión a la hora de predecir los estudiantes con riesgo de reprobación.

Incorporar variables adicionales puede extender el alcance de estudios en el futuro, también se sugiere explorar los motivos por los cuales los estudiantes obtienen estas calificaciones tanto las mejores, bajas y abandonos dentro de la unidad educativa. Extender este estudio a más unidades educativas es conveniente para ver de forma más clara y precisa la realidad en el ámbito educativo con el fin de corroborar si otros colegios tienen falencias en las mismas áreas y similar cantidad de reprobados, realizar comparativas entre diversas unidades educativas tanto de ciudades como de lugares alejados para identificar unidades educativas con mayores índices de estudiantes con bajo desempeño académico.

Se sugiere llevar a cabo este estudio luego de tener las calificaciones anuales disponibles para lograr trabajar con los estudiantes que presenten mayor probabilidad de reprobación. Debido a que este estudio solo tuvo un par de consultorías con una psicopedagoga sobre formas y concejos para prevenir bajas calificaciones y reprobaciones, se recomienda también consultar con más profesionales sobre estas medidas para posteriormente aplicarlas.

Si bien es cierto que este estudio predice que la cantidad de estudiantes tiende a una baja con el paso del tiempo, se puede incluir como parte importante el estudio de aptitudes, tipo de inteligencia, hábitos de estudio del o la estudiante para tener resultados más precisos sobre los motivos por los cuales los alumnos obtienen altas o bajas calificaciones.

Referencias bibliográficas

Bibliografía

Amonzabel, M. A. (5 de March de 2025). Obtenido de Youtube: https://www.youtube.com/live/gImhqTufEMQ

Asamblea legislativa plurinacional. (20 de December de 2010). *SEA*. Obtenido de sea.bog.bo: https://sea.gob.bo/digesto/CompendioII/D/28_L_70.pdf

Binns, M. (26 de Febrary de 2015). *borgen project*. Obtenido de borgenproject.org: https://borgenproject.org/top-4-reasons-education-in-boliva-lags/

Bravo, L. C., Bermudez, G. T., & Cardona, A. A. (2021). *Machine Learning aplicado al rendimiento académico en educación superior: factores, variables y herramientas.* Ciudad de Mexico: UD Editorial.

Breiman, L. (Octuber de 2001). Random forests. Machine Learning. *Springer*, págs. 5-32.

Brownlee, J. (2021). Random Forest for Machine Learning. *Machine Learning Mastery*. Obtenido de https://machinelearningmastery.com

Castillo Aráuz, D., & Martínez, J. J. (2023). Predicción del rendimiento académico en la UNADECA por medio de sistemas de clasificación. *Unaciencia Revista De Estudios E Investigaciones*, 17-35.

CEPAL. (2020). *Panorama social de América Latina 2020*. Comisión Económica para América Latina y el Caribe.

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Association for Computing Machinery*, 785-794.

Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, 5-7.

codificandobits. (18 de March de 2025). *codificandobits*. Obtenido de codificandobits.com: https://codificandobits.com/blog/maquinas-de-soporte-vectorial/

Cornejo, J. (2022). ¿Cuál es la nota minima para pasar el año en Bolivia? *todosloshechos.es*, https://todosloshechos.es/cual-es-la-nota-minima-para-pasar-de-ano-en-bolivia.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning. *Springer*, 273-297.

Duc, T. L., Leiva, R. G., Casari, P., & Östberg, P.-O. (13 de September de 2019). *Machine Learning Methods for Reliable Resource Provisioning in Edge-Cloud Computing: A Survey*. Obtenido de dl.acm.org: https://dl.acm.org/doi/10.1145/3341145

educabolivia. (31 de March de 2025). *educabolivia.com*. Obtenido de https://educabolivia.com: https://educabolivia.com/geografia/ubicacion-y-extension-territorial-de-bolivia/

Fix, E., & J. L. Hodges, J. (1951). Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties. *JSTOR*, 238-247.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 1189-1232.

Fromero, R. e. (2021). Factores socioeconómicos que influyen en el rendimiento académico en zonas rurales. *Universidad Nacional Abierta y a Distancia*.

geeksforgeeks. (16 de January de 2025). *geeksforgeeks*. Obtenido de geeksforgeeks.org: https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/

González, J. (2020). El rendimiento académico y sus factores asociados en educación básica y media. *Revista de Educación y Sociedad*, 45-60.

Gutierrez, B. (2022). Reflexiones e ideas para mejorar la Calidad Educativa en Bolivia, desde la Evaluación ERCE 2019. *Simbiosis*, 33-44. Obtenido de https://doi.org/10.59993/simbiosis.v2i4.19

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: Data mining, inference, and prediction (2nd ed.). *Springer*.

HQEDGAR. (26 de Enero de 2023). *La importancia de la educación en Bolivia*. Obtenido de librosdelministeriodeeducacion.com: https://librosdelministeriodeeducacion.com/blog/importancia-educacion-bolivia/

Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. En J. D. Hunter, *Matplotlib: A 2D Graphics Environment* (págs. 90-95). IEEE.

Ibm. (14 de March de 2025). *ibm*. Obtenido de ibm.com: https://www.ibm.com/es-es/topics/supervised-learning

Ibm. (15 de March de 2025). *ibm-topics*. Obtenido de ibm.com: https://www.ibm.com/es-es/topics/logistic-regression

Improvitz. (6 de September de 2024). *improvitz*. Obtenido de improvitz.com: https://improvitz.com/machine-learning-y-la-optimizacion-de-procesos-una-guia-completa-para-mejorar-la-eficiencia-operativa/

INE. (29 de August de 2024). *censo.ine*. Obtenido de https://censo.ine.gob.bo: https://censo.ine.gob.bo/somos-11-312-620-bolivianos-y-santa-cruz-es-el-departamento-que-mas-crecio-y-mas-poblado/

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., . . . Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. Advances in Neural Information Processing Systems. *NeurIPS*, 30-32.

Kichiuhua. (13 de Noviembre de 2024). *Kichiuhua*. Obtenido de https://www.kichihua.com/importancia-de-las-calificaciones/

Kowalczyk, A. (18 de March de 2025). *syncfusion*. Obtenido de syncfusion.com: https://www.syncfusion.com/succinctly-free-ebooks/support-vector-machines-succinctly/introduction

Laime, M. W. (2024). *ddigital*. Obtenido de ddigital.umss.edu: http://ddigital.umss.edu/bitstream/123456789/43657/1/MONOGRAFIA_LOPEZ%20LAIME%20MAYA%20WARA.pdf

López, D. E. (2022). Aplicación de modelos de aprendizaje automático en la predicción del rendimiento académico estudiantil. *Universidad Técnica de Ambato*.

Marquez Vera, C. (2015). *Predicción del fracaso y abandono escolar mediante técnica de minería*. Córdoba: Servicio de Publicaciones de la Universidad de Córdoba.

Mckinney, W. (2012). Data Structures for Statistical Computing in Python. En W. Mckinney, *Proceedings of the 9th Python in Science Conference* (págs. 51-56). IEEE.

MindMachineTV. (17 de Aug de 2020). *youtube*. Obtenido de youtube.com: https://youtu.be/tYPi6qcCQbg?si=m978xJUSz2fa8pGc

Ministerio de Educación del Estado plurinacional de Bolivia. (23 de Octubre de 2015). *minedu*. Obtenido de minedu.gob.bo: https://www.minedu.gob.bo/files/publicaciones/veaye/11.-R.M.-800-2015-Reglamento-de-Libretas-electronicas.pdf

Müller, A. C., & Guido, S. (2017). Introduction to Machine Learning with Python. *O'Reilly*.

Municipios de Bolivia. (31 de March de 2025). *municipio.com.bo*. Obtenido de https://www.municipio.com.bo: https://www.municipio.com.bo/

Navarro, R. E. (2003). El rendimiento académico: Concepto, investigación y desarrollo. *Revista Iberoamericana sobre*, 2-5.

Oliphant, T. E. (2006). *Guide to NumPy*. Trelgol.

Pedregosa, F. (2011). Scikit-learn: Machine Learning in Python. *ACM. DL. Digital Library*, 2825-2830.

Pedregosa, F., Varaquaux, G., Gramfort, A., Michel, V., & Thirion, B. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 2825-2830.

Perez, F., & Granger, B. E. (2008). IPython: A System for Interactive Scientific Computing. *IEEE*, 21-29.

Poulova, P. K., & Mikulecká, J. (2019). *Data Science—A Future Educational Potential*. Singapore: International Conference on Multimedia and .

Probabilidad y estadisitca. (17 de March de 2025). *probabilidadyestadistica*. Obtenido de probabilidadyestadistica.net: https://www.probabilidadyestadistica.net/arbol-de-decisiones/

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: Unbiased boosting with categorical features. Advances in Neural Information Processing Systems. *NeurIPS*, 31-32.

Ramírez, N. D., & Páez, A. R. (2024). Predicción de la aprobación a través de datos personales de estudiantes de medio superior. *Revista electrónica sobre tecnología, educación y sociedad*, 1-7.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 533-536.

Russo, C. (Mayo de 2019). *Sedici*. Obtenido de sedici.unlp.edu.ar: https://sedici.unlp.edu.ar/bitstream/handle/10915/79958/Documento_completo.pdf-PDFA1b.pdf?sequence=1&isAllowed=y

SAP Concur. (10 de October de 2021). *Machine Learning: ¿qué es y cómo funciona?* Obtenido de https://www.concur.com.mx/blog/article/machine-learning-que-es

Scikit-learn Developers. (2023). *scikit-learn*. Obtenido de scikit-learn.org: https://scikit-learn.org/stable/modules/neighbors.html

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 427-437.

Suguiura, F. O. (1 de April de 2022). *Árbol de Decisión en Aprendizaje Automático*. Obtenido de revistasbolivianas: http://www.revistasbolivianas.ciencia.bo/pdf/rv/n19/n19_a05.pdf

tiempos, L. (2021). Según ranking de la Unesco, Bolivia ocupa los últimos lugares de América Latina en calidad educativa. *Los tiempos*, 1.

UNESCO. (2021). Informe de seguimiento de la educación en el mundo 2021: Bolivia en contexto. Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura.

VanderPlas, J. (2016). *Python Data Science Handbook: Essential Tools for Working with Data* . Boston: O'Reilly Media.

Vygotsky, L. (1978). El desarrollo de los procesos psicológicos superiores. *Harvard University Press*.

Wikipedia. (16 de March de 2025). *Wikipedia*. Obtenido de wikipedia.org: https://en.wikipedia.org/wiki/Education_in_Bolivia

Anexos

Anexo 1. Fuente de datos original

Nro.	Apellidos y		Nombres	Código Rude	Gen	Fecha Nac	Lug Nac	Número C.I.	Estado Matrícula	Com.Lenguaje PA	L.Extranj. PA	Cs.Sociales PA	Edu.Musical PA	Art.Plástic.V. PA	E.Física.D. PA	Matemática PA	T.General PA	Física PA	Química PA	Bio y Geo. PA	Cos.Fil y Sic PA	Val.Esp.Rel. PA
	Paterno	Materno																				
1	BECERRA	RODRIGUEZ	LEONEL FRANCO	5195000120142605	M	28-12-2009	SC	13911865	PROMOVIDO	86	76	96	85	90	71	87	83	82	71	78	92	95
2	BURGOS	SUBUBI	BELEN	5195000120116805	F	11-06-2009	SC	15871159	PROMOVIDO	61	63	78	64	56	59	61	57	63	53	51	58	62
3	CABELLO	CUELLAR	MARIA YULIANA	7195000620132539	F	04-06-2008	SC	13913892	PROMOVIDO	58	60	66	77	55	67	63	56	58	52	53	56	57
4	CAMPOS	CUELLAR	BEATRIZ	7195000620114417	F	19-08-2009	SC	14597654	PROMOVIDO	60	59	81	93	68	78	69	72	72	59	54	60	60
5	CORTEZ	ARANCIBIA	CARLOS ALBERTO	7193000620121654	M	08-12-2006	SC	14278870	NO INCORPORAD													
6	CRUZ	TOLA	LEYDI	7195000620113993	F	10-12-2007	SC	13126047	PROMOVIDO	86	81	92	77	85	64	91	79	82	76	68	79	78
7	ESCOBAR	RODRIGUEZ	HERLIN MARK	5195000120142495	M	19-02-2010	SC	13095747	PROMOVIDO	93	90	98	93	95	96	99	93	87	93	93	91	97
8	GARECA	SORIA	IVER	7195004120135A	M	02-04-2008	SC	12014331	REPROBADO	42	39	51	53	51	70	50	42	35	39	35	36	40
9	GUTIERREZ	MATURANO	ADRIANA	5195000120152048	F	22-09-2009	SC	13185362	PROMOVIDO	66	61	77	83	69	81	74	59	72	62	56	76	70
10	GUTIERREZ	SALAZAR	ANABEL	7195000620141453	F	05-03-2009	SC	13913002	REPROBADO	51	51	59	70	51	51	50	39	49	43	46	53	53
11	JUSTINIANO	CASTEDO	ADRIAN	7195000620151374	M	11-10-2009	SC	11364918	PROMOVIDO	58	65	78	92	54	80	62	64	65	54	59	60	63
12	MENDEZ	APARICIO	DARWIN	5195000120152071	M	05-10-2009	SC	13095313	PROMOVIDO	64	52	83	78	51	87	76	56	55	58	60	65	58
13	MENDOZA	CRUZ	SAUL	5195000120152029	M	15-06-2010	SC	13915625	PROMOVIDO	53	60	62	93	57	69	61	54	58	52	54	51	53
14	ORTIZ	ALBAREZ	LADISLAO	7195000620141068	M	11-04-2010	SC	13095921	PROMOVIDO	81	81	90	92	83	97	95	76	85	69	82	85	94
15	PEDRAZA	PEDRAZA	JOSE EDUARDO	7195000620140980	M	23-03-2010	SC	13915595	REPROBADO	63	41	60	90	47	84	63	47	48	45	58	44	47
16	PEÑA	BECERRA	BRIANA ASHLEY	7195000620113577	F	10-04-2009	SC	14285271	REPROBADO	58	51	65	80	52	64	60	45	55	52	48	53	55
17	QUINTANA	MEJIAS	DAVID	7195000620113626	M	08-04-2009	SC	13913061	PROMOVIDO	63	51	64	93	55	77	64	52	54	51	59	56	55
18	RODRIGUEZ	ROCA	DAVID	7195004720141A	M	09-03-2008	SC	13913869	PROMOVIDO	60	68	86	90	70	70	66	73	75	68	57	76	78
19	ROSALES	PEREIRA	CARLOS	8198134820161016	M	10-09-2008	SC	15306435	PROMOVIDO	67	60	68	98	61	95	66	56	69	61	61	65	65
20	SAREZ	SALVADOR	NATALIA	519500012014272A	F	29-05-2009	SC	13941639	NO INCORPORAD													
21	TORRES	OVANDO	ORIANA	8198108720149119	F	11-01-2010	SC	15062432	RETIRADO TRASLADO													
22	TORREZ	CALUCHO	DIEGO CANDIDO	8198044020197903	M	12-01-2010	SC	13381566	PROMOVIDO	61	59	66	78	60	74	73	68	68	57	65	51	51
23	TORREZ	CORREA	JORGE MATIAS	5195000120142639	M	16-01-2010	SC	13095302	PROMOVIDO	64	68	80	93	72	87	83	74	76	64	65	74	60

Figura 1-1: Contenido de los archivos originales

Fuente: Fuente de datos de la unidad educativa San José Obrero (2025)

Por petición de la directora encargada, estos archivos en formato pdf no se subirán, sin embargo, se tienen subidos los archivos convertidos a un repositorio de GitHub.

Enlace de ubicación en GitHub:

https://github.com/LimbergVillcaCoraite/Proyecto-Dip.-Ciencia-de-datos/tree/c471b96613103599499df300cdf06bed7a7a28ba/data-source

Ubicación en CD: Proyecto-Dip.-Ciencia-de-datos/data-source/{Notas Primaria} {Notas Secundaria}

Anexo 2. Código fuente

Proyecto de Ciencia de datos

Proyecto: Analisis y prediccion de estudiantes reprobados en la Unidad Educativa San Jose Obrero

Estudiante: Limberg Villca Coraite

Carrera: Lic.Ingenieria de Sistemas

Fase 1: Comprension del Negocio (problema)

Analizar las calificaciones de los estudiantes, determinar las materias que tienen mayor falencia y predecir que estudiantes tienen la mayor probabilidad de reprobar

Fase 2: Comprension de los Datos

Adquisicion de datos

[1]:
```
import os

ruta_local = "L:/Materiales cursos y diplomados/datascience/Notas San Jose Obrero/Proyecto-Dip.-Ciencia-de-datos/data-source"
ruta_repo = "https://github.com/LimbergVillcaCoraite/Proyecto-Dip.-Ciencia-de-datos.git"
nombre_carpeta_repo = "Proyecto-Dip.-Ciencia-de-datos"

if os.path.exists(ruta_local):
    print("Usando ruta local:", ruta_local)
    os.chdir("L:/Materiales cursos y diplomados/datascience/Notas San Jose Obrero/Proyecto-Dip.-Ciencia-de-datos/data-source")
    print("Directorio de trabajo: " + os.getcwd())
elif os.path.exists(nombre_carpeta_repo):
    print(f"Se encontró la carpeta del repositorio clonado previamente: {nombre_carpeta_repo}")
    os.chdir("/content/Proyecto-Dip.-Ciencia-de-datos/data-source")
    print("Directorio de trabajo: " + os.getcwd())
else:
    try:
        print("La ruta local no existe. Clonando desde el repositorio remoto...")
        !git clone {ruta_repo}
        print("Repositorio clonado exitosamente.")
        os.chdir("/content/Proyecto-Dip.-Ciencia-de-datos/data-source")
        print("Directorio de trabajo: " + os.getcwd())
```

Figura 2-1: Código del proyecto
Fuente: Elaboración propia (2025)

Enlace de ubicación en GitHub: https://github.com/LimbergVillcaCoraite/Proyecto-Dip.-Ciencia-de-datos/blob/c471b96613103599499df300cdf06bed7a7a28ba/Codigo%20fuente/proyecto_ciencia_de_datos.ipynb

Ubicación en CD: Proyecto-Dip.-Ciencia-de-datos/Codigo fuente/proyecto_ciencia_de_datos.ipynb

Anexo 3. Proyección de cantidad de estudiantes para los siguientes años

Para tener una mejor perspectiva, también se realizó la predicción para tener una aproximación de la cantidad de estudiantes inscritos teniendo como salida 316 estudiantes para el año 2025, 320 para 2026 y 325 estudiantes para el año 2027.

Las columnas usadas para realizar la predicción fueron la gestión y el código rude de cada estudiante, con estas dos variables podemos realizar un conteo de cuantos estudiantes tenemos por año como vemos en la figura 3-1.

```python
def cargar_y_preparar_datos(ruta_archivo='./Salidas del programa/calificaciones.xlsx'):
    """Carga y prepara los datos para el análisis predictivo."""
    try:
        # Cargar datos
        df_calificaciones = pd.read_excel(ruta_archivo)

        # Detectar columnas clave
        cols_gestion = ['gestion', 'año', 'anio', 'periodo', 'gestión']
        cols_rude = ['rude', 'Codigo Rude', 'estudiante', 'alumno', 'codigo']

        columna_gestion = next((col for col in df_calificaciones.columns
                                if any(key in col.lower() for key in cols_gestion)), None)

        columna_rude = next((col for col in df_calificaciones.columns
                             if any(key in col.lower() for key in cols_rude)), None)

        if not columna_gestion or not columna_rude:
            raise ValueError("No se encontraron las columnas necesarias")

        # Agrupar por gestión y contar estudiantes únicos
        df = df_calificaciones.groupby(columna_gestion)[columna_rude].nunique().reset_index()
        df.columns = ['Gestion', 'Cantidad_Estudiantes']

        # Asegurar tipo de datos correcto
        df['Gestion'] = pd.to_numeric(df['Gestion'], errors='coerce')
        df = df.dropna().reset_index(drop=True)

        # Ordenar por gestión
        df = df.sort_values('Gestion').reset_index(drop=True)
```

Figura 3-1: Selección de las comunas gestión y Codigo Rude
Fuente: Elaboración propia (2025)

Para evaluar se usará el 0.7 o 70% para entrenamiento y el restante para pruebas como podemos ver en la figura 3-2.

```python
def dividir_datos(df, test_size=0.3):
    """Divide los datos respetando la secuencia temporal."""
    n = len(df)
    train_size = max(int(n * (1-test_size)), n-2)
    df_train = df.iloc[:train_size].reset_index(drop=True)
    df_test = df.iloc[train_size:].reset_index(drop=True)
    return df_train, df_test
```

Figura 3-2: Dividiendo los datos en entrenamiento y prueba
Fuente: Elaboración propia (2025)

Para realizar esta predicción se probaron modelos como ser: SES, Holt, Sarima, Crecimiento y MediaMovil, ya que para la cantidad de datos y años era más indicado, en un principio se probó con modelos como Prophet y Arima, pero estos causaban sobre ajuste y necesitan mayor cantidad de datos, requieren más parámetros, demasiado complejos para crecimiento estudiantil, en síntesis, era como: "Matar una mosca a cañonazos", los modelos mencionados para la predicción se entrenaron como podemos apreciar en la figura 3-3.

```python
def generar_predicciones_futuras(df, modelos, mejor_modelo, años_a_predecir=3):
    """Genera predicciones para los próximos años con todos los modelos e incluye datos históricos."""
    ultimo_año = df['Gestion'].max()
    años_futuros = np.array([ultimo_año + i for i in range(1, años_a_predecir+1)])

    # Crear un DataFrame consolidado que incluirá datos históricos y predicciones
    consolidado_df = pd.DataFrame({'Año': list(df['Gestion']) + list(años_futuros)})

    # Añadir datos históricos
    consolidado_df['Histórico'] = list(df['Cantidad_Estudiantes']) + [None] * años_a_predecir

    # DataFrame solo para predicciones futuras (para mantener la funcionalidad original)
    predicciones_df = pd.DataFrame({'Año': años_futuros})

    # Series completas para reentrenamiento
    y_completo = df['Cantidad_Estudiantes'].values

    for nombre, info in modelos.items():
        try:
            if nombre == 'SES':
                # Reentrenar con todos los datos
                modelo = SimpleExpSmoothing(y_completo).fit(
                    smoothing_level=info['params']['alpha'], optimized=False)
                pred = modelo.forecast(años_a_predecir)

            elif nombre == 'Holt':
                # Reentrenar con todos los datos
                modelo = ExponentialSmoothing(
                    y_completo, trend=info['params']['trend'], seasonal=None
                ).fit(
                    smoothing_level=info['params']['alpha'],
                    smoothing_trend=info['params']['beta']
                )
                pred = modelo.forecast(años_a_predecir)

            elif nombre == 'SARIMA':
                # Reentrenar con todos los datos
                order = info['params']['order']
                modelo = SARIMAX(y_completo, order=order, simple_differencing=True)
                res = modelo.fit(disp=False)
                pred = res.forecast(años_a_predecir)
```

Figura 3-3: Entrenamiento de los modelos seleccionados
Fuente: Elaboración propia (2025)

Dentro de los modelos que, seleccionados, el mejor modelo según sus parámetros de salida fue Holt como podemos apreciar en la tabla 3-1.

Modelo	MSE	RMSE	MAE	MEJOR MODELO
Holt	13.053520	3.612966	3.545193	0.568479
MediaMovil	30.610000	5.532630	5.500000	-0.011901
SES	32.267219	5.680424	5.500000	-0.066685
Sarima	52.740191	7.262244	5.595218	-0.743477
Crecimiento	695.991304	26.381647	24.513430	-22.007977

Tabla 3-1: Métricas de los modelos para predecir la cantidad de inscritos en los próximos años
Fuente: Elaboración propia (2025)

Como resultado de los modelos tenemos la taba 3-2.

Año	SES	Holt	Sarima	Crecimiento	MediaMovil
2025	308	316	305	303	307
2026	308	320	305	294	307
2027	308	325	305	285	307

Tabla 3-2: Predicciones de los modelos usados
Fuente: Elaboración propia

También podemos ver este resultado en forma gráfica en la figura 3-4.

Figura 3-4: Gráfico del histórico y las predicciones de los modelos
Fuente: Elaboración propia

Teniendo este resultado, podemos apreciar una gráfica de la cantidad de estudiantes desde el año 2015 hasta el año 2024 (histórico), dentro de la misma grafica la selección del mejor modelo para esta predicción, desde el año 2025 al 2027 (Mejor Modelo) como podemos ver en la figura 3-5.

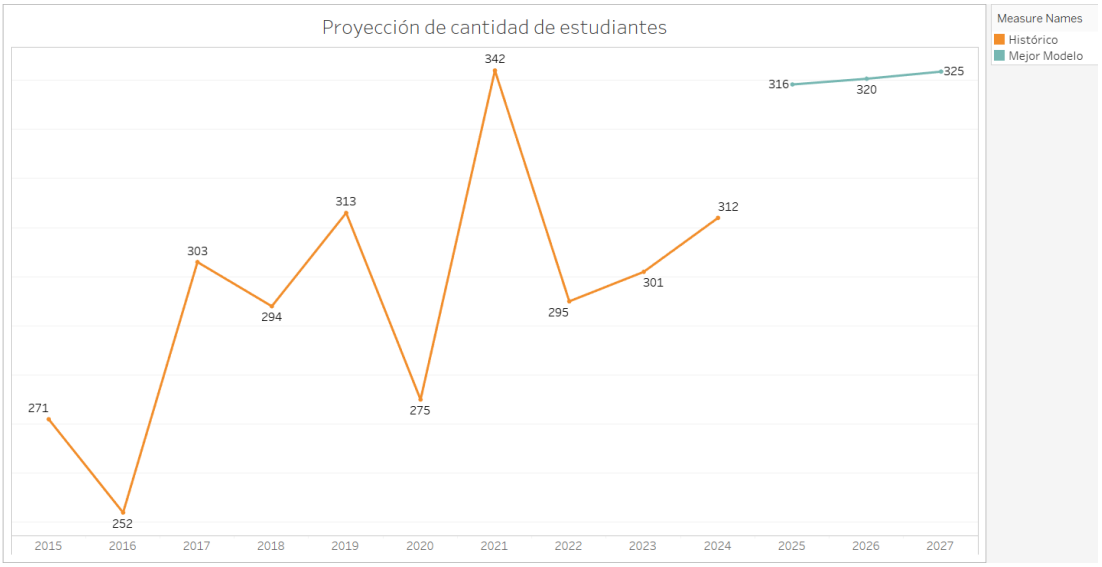


Figura 3-5: Grafico de cantidad de estudiantes hasta el año 2027
Fuente: Elaboración propia (2025)

Anexo 4. Estudio de Hábitos de estudio

A finales del año 2024, se aplicaron encuestas a los estudiantes de 3ro de Secundaria de la Unidad Educativa, esto se realizó en conjunto con la dirección y el profesor encargado, en dicha encuesta se tuvo resultados positivos como se muestra en la figura 4.1, donde el 57.14% de estudiantes tiene Tendencia positiva en hábitos de estudio, esto puede ser un indicador clave para la parte predictiva ya que ayuda a confirmar que los estudiantes empiezan a tomar conciencia sobre su educación.

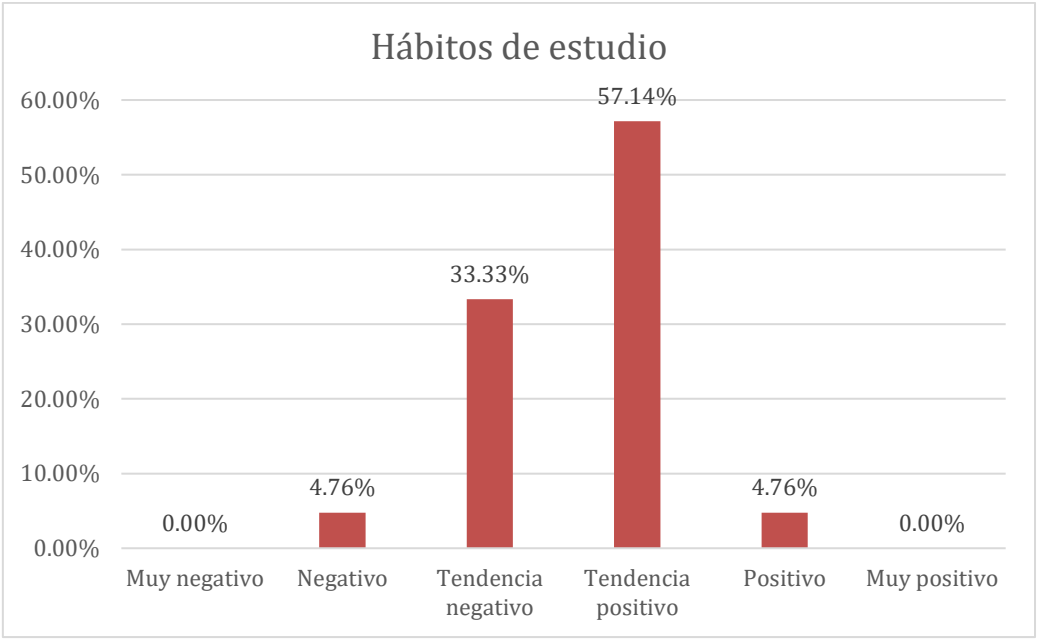


Figura 4-1: Gráfico de hábitos de estudio

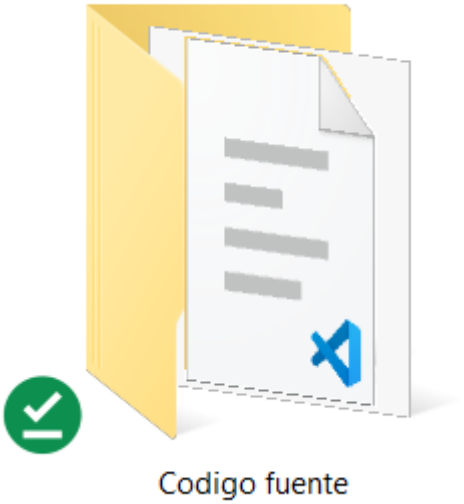
Fuente: elaboración propia (2024)

Si bien es cierto que la tendencia negativa parece estar no tan alejada de la primera esto se refleja en las predicciones con el descenso de la cantidad de estudiantes gradualmente para los próximos años.

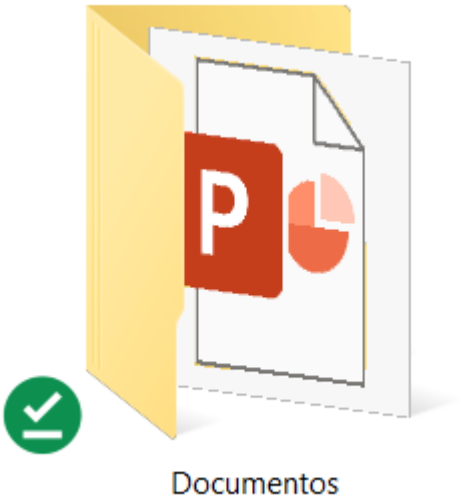
Anexo 5. Contenido del CD

En este apartado se describe el contenido del CD, donde estarán disponibles las carpetas y los archivos necesarios del proyecto.
En esta carpeta se tiene lo siguiente:

- Carpetas: Primaria y Secundaria, donde están contenidas las calificaciones de los estudiantes desde la gestión 2015 a 2024.
- Carpeta Salidas del programa, en esta ubicación se encuentran los archivos en formato csv y xlsx que contienen todos los datos necesarios para ejecutar este proyecto.
- Carpeta catboost_info, archivos y correspondientes al entrenamiento, errores y aprendizaje de este modelo en particular.
- Archivo student_analysis.log, contiene el log de la ejecución del código desde la parte del modelamiento de machine learning.



- Contiene el código fuente del proyecto realizado en python, ordenado según la metodología crisp-dm.



- Contiene el documento de la monografía en formato pdf con el título "MonografiaLimbergVillcaCoraite.pdf".
- Presentación en formato pptx (Power point), que contiene información representativa e importante de este proyecto.

Contiene dos carpetas:

- Tableau, contiene un archivo con el nombre "Proyecto Final Tableau.twbx", dentro de este archivo se encuentra el dashboard realizado para este proyecto.
- Salida del código, contiene algunos gráficos extraídos al momento de analizar y predecir en este proyecto, dichos gráficos son un complemento a lo realizado en Tableau.