PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERIA

TEMA:

EVALUACIÓN Y COMPARACIÓN DE MODELOS PREDICTIVOS BASADOS EN MACHINE LEARNING PARA LA PREVENCIÓN DE LA DESERCIÓN ACADÉMICA EN UNA INSTITUCIÓN UNIVERSITARIA.

DIRECTOR

MSC. EDISON MORA LONDOÑO

AUTOR:

JENNY MARISOL OÑA TITUAÑA

TRABAJO PREVIO A LA OBTENCIÓN DEL TÍTULO DE MAGISTER EN SISTEMAS DE INFORMACIÓN CON MENCIÓN EN DATA SCIENE

QUITO– 2025

DEDICATORIA

Al regalo más grande que Dios me supo entregar mis hijos Mateo y Marcelo.
Las personas más importantes de mi vida, por ellos que me dan fuerzas y motivos para luchar y seguir adelante con todos mis sueños.

Por ellos y para ellos todo mi esfuerzo y dedicación.

AGRADECIMIENTO

Agradezco a Dios, por la salud y por permitirme terminar una etapa más en mi vida , por darme una gran familia la cual ha luchado de la mano conmigo para lograr esta meta.

A mi amada madre, cuyo amor incondicional ha sido mi refugio en los momentos más desafiantes. Gracias por ser la segunda madre de mis hijos y cuidar de ellos, en esos tiempos complejos.

A mi padre, mi principal motivador. Cada decisión que he tomado en mi vida ha sido impulsada por su confianza y su aliento. Gracias por creer siempre en mí.

A mi compañero de vida, mi esposo. Tu comprensión y paciencia fueron mi sostén en los días difíciles, cuando tuviste que asumir el rol de padre y madre en nuestro hogar. Gracias por tu amor y sacrificio.

A mi invaluable amigo Carlos, cuya guía y apoyo fueron fundamentales para alcanzar esta meta. Tu amistad y tu aliento fueron un regalo invaluable.

Finalmente, a mi dedicado tutor de tesis, Mi más sincero agradecimiento por su arduo trabajo, su tiempo generoso y su guía experta en cada paso de este camino. Su apoyo fue esencial para la culminación de este proyecto.

Este logro no habría sido posible sin el amor, el apoyo y la inspiración de cada uno de ustedes. ¡Gracias desde el fondo de mi corazón!

Contenido

CAPITULO I: INTRODUCCIÓN 6
1.1 Introducción. 6
1.2 Justificación 7
1.3 Planteamiento del problema 7
1.3.1 Objetivo general. 8
1.3.2 Objetivos específicos. 8
CAPITULO II. FUNDAMENTACIÓN TEORICA 9
2. Marco teórico 9
2.1 ¿Qué es la deserción estudiantil? 9
2.1.1 Factores asociados a la deserción 9
2.1.2 Perspectivas asociadas a la deserción 10
2.2 ¿Qué es el Machine Learning? 11
2.2.1 Aprendizaje Supervisado 12
2.3 ¿Qué es Ramdon Forest? 12
2.4 ¿Qué Adaptive Boosting? 13
2.5 ¿Qué es Gradient Boosting? 14
2.6 ¿Que es Visual Studio? 15
2.7 ¿Que Amazon Sagemaker? 15
2.7.1 ¿Qué es Canvas? 16
2.8 Fundamentos de programación en Python. 17
2.8.1 ¿Qué es Pyhton? 17
2.8.2 Python para la Ciencia de Datos. 17
2.8.3 Principales librerías y frameworks. 17
2.9 Metodología CRISP-DM 18
2.9.1 Fases de la Metodología CRISP-DM 18
CAPITULO III. DESARROLLO DE LA METODOLOGÍA 22
3.1 Comprensión del Negocio 22
3.1.1 Objetivo del Negocio 22
3.1.2 Objetivo de Minería 22
3.2 Comprensión de los Datos 23
3.2.1 Recolección de Datos 23
3.2.2 Descripción de los Datos 23
3.3 Preparación de Datos 28

3.3.1 Verificar la calidad de los datos 29
3.3.2 Limpieza de datos 29
3.3.3 Análisis exploratorio de datos 32
3.3.4 Construcción de nuevos datos 40
3.3.5 Integración de datos 45
3.4 Modelado 49
3.5 Evaluación 49
CAPITULO VI. CONCLUSIONES Y RECOMENDACIONES 58
6.1 CONCLUSIONES 58
6.2 RECOMENDACIONES 58
Bibliografía y referencias. 59

Indice de Figuras

Figura 1. Diagrama sobre el Machine Learning 11
Figura 2 Diagrama sobre el Aprendizaje Supervisado 12
Figura 3. Diagrama sobre el Aprendizaje 13
Figura 4. Dataset and Distribution with weak learner 14
Figura 5. Original dataset 15
Figura 6. Flujo de Trabajo en un Entorno de Preparación y Análisis de Datos 16
Figura 7. Logo de Python 17
Figura 9. Dataset 28
Figura 10. Data Columns 30
Figura 11. Distribución de Valores Únicos en Variables Categóricas de un Conjunto de Datos 46
Figura 11. Reporte de clasificación ..... **¡Error! Marcador no definido.**
Figura 12. Randon Forest 55
Figura 13. Adaptive Boosting 56

Indice de Graficos

Grafico 1. Distribución de la variable objetivo…… 32
Gráfico 2. Estado actual (No deserto vs Deserto)…… 33
Gráfico 3. Deserción por Etnia…… 34
Gráfico 4. Deserción por Edad…… 34
Gráfico 5. Deserción por Ciudad…… 35
Gráfico 6. Deserción por Estado Civil…… 36
Gráfico 7. Deserción por Colegio…… 36
Gráfico 8. Top 10 carreras…… 37
Gráfico 9. Estado actual y GPA por periodo…… 38
Gráfico 10. CredReg y el Estado Actual…… 39
Gráfico 11. Semestre y GPA por periodo…… 39
Gráfico 12. Colegio y GPA por periodo…… 40
Gráfico 13. Deserción por nota de colegio…… 43
Gráfico 14. Deserción por créditos registrados…… 43
Gráfico 15. Deserción por GPA periodo…… 44
Gráfico 16. Deserción por GPA Acumulado…… 44
Gráfico 17. Deserción por total de créditos aprobados…… 45
Gráfico 18. Matriz de Confusión…… 53
Gráfico 19. Curva Roc para la clase 0 (Deserción)…… **¡Error! Marcador no definido.**
Gráfico 20. Importancia de las características en AdaBoost…… 51

Indice de Tablas

Tabla 1. Nombre de variable, descripción, tipo y uso en el análisis.....24
Tabla 2. Datos académicos 28
Tabla 3. Discretización de los valores continuos.....41
Tabla 4. Variables categóricas en subset.....41
Tabla 5. Dataset final para el modelo 46
Tabla 6. Variables y su relación 47
Tabla 7. Relación nota del curso con variables 48
Tabla 8. Variables cuartil .....**¡Error! Marcador no definido.**
Tabla 9. Algoritmos de evaluación de modelos.....52

CAPITULO I: INTRODUCCIÓN

1.1 Introducción.

El presente trabajo de titulación tiene como objetivo desarrollar y comparar tres modelos de clasificación basados en algoritmos de Machine Learning, para predecir la deserción universitaria en una universidad privada del Ecuador. Estos modelos se plantean como una herramienta de alerta temprana, diseñada para identificar a los estudiantes en riesgo de abandonar sus estudios y, de esta manera, permitir a las instituciones de educación superior implementar medidas preventivas que fomenten la retención estudiantil.

Para cumplir este propósito se emplean los algoritmos Random Forest, Adaptive Boosting y Gradient Boosting debido a su robustez en problemas de clasificación. Random Forest combina múltiples árboles de decisión para manejar datos complejos y reducir el sobreajuste. Adaptive Boosting ajusta el peso de las observaciones mal clasificadas en cada iteración, mejorando la detección de patrones en datos desbalanceados. Gradient Boosting, al optimizar la función objetivo ajustando las predicciones residuales, es eficaz en la captura de relaciones complejas y no lineales.

El análisis se llevará a cabo utilizando un conjunto de datos privados que incluye variables representativas de factores académicos, personales, e institucionales asociados al riesgo de deserción. La metodología CRISP-DM (Cross-Industry Standard Process for Data Mining) guiará el proceso de desarrollo, asegurando un enfoque estructurado para la preparación, análisis y evaluación de los datos.

Para el desarrollo de los modelos Random Forest y XGBoost se emplearán diversas herramientas y frameworks en Amazon SageMaker el mismo que permitirá entrenar y desplegar los modelos de Machine Learning en un entorno escalable, Amazon Canvas se usará para análisis exploratorio y de predicción, este complementará el preprocesamiento de datos y validará resultados obtenidos mediante los modelos programados. En Visual Studio con Python se desarrollará el modelo de clasificación AdaBoosting, empleando sus bibliotecas especializadas como pandas, scikit-learn para el preprocesamiento, entrenamiento, validación y ajuste del algoritmo. Se utilizarán métricas de evaluación como Accuracy, Precision, Recall, F1-score y AUC-ROC para medir el desempeño de

cada modelo y determinar cuál ofrece el mejor balance entre detección de estudiantes en riesgo y minimización de falsos positivos.

Finalmente, El propósito de esta investigación es determinar el modelo predictivo óptimo para la detección temprana del riesgo de deserción estudiantil. La implementación de este modelo en la universidad facilitará la toma de decisiones informadas y la implementación de intervenciones personalizadas para apoyar a los estudiantes.

1.2 Justificación

La deserción estudiantil en la educación superior es un fenómeno complejo que afecta tanto a los estudiantes como a las instituciones de Educación Superior en el Ecuador. Para los estudiantes, abandonar sus estudios representa una pérdida de oportunidades educativas, laborales y personales; mientras que para las universidades implica costos económicos, deterioro de su reputación y desafíos en la gestión de recursos. Según estudios recientes, las tasas de deserción en muchas instituciones alcanzan niveles preocupantes, lo que evidencia la necesidad de desarrollar estrategias efectivas para abordar este problema.

En este contexto, la implementación de herramientas predictivas basadas en Machine Learning ofrece una oportunidad invaluable para identificar a los estudiantes en riesgo antes de que abandonen sus estudios. A diferencia de los métodos tradicionales, que suelen ser reactivos, estas técnicas permiten una intervención temprana y proactiva, optimizando el uso de recursos y aumentando las probabilidades de retención estudiantil.

1.3 Planteamiento del problema

Factores como el rendimiento académico, el entorno familiar y las características institucionales influyen directamente en la decisión de los estudiantes de abandonar sus estudios. Este fenómeno no solo representa un desafío para el desarrollo personal y profesional de los jóvenes, sino que también tiene repercusiones económicas y reputacionales para las universidades.

A pesar de los esfuerzos realizados por las instituciones para reducir las tasas de deserción, la identificación temprana de estudiantes en riesgo sigue siendo una tarea compleja debido a la diversidad y complejidad de los factores involucrados. El uso de tecnologías avanzadas, como el aprendizaje automático ofrece una oportunidad de solventar este problema mediante la construcción de modelos predictivos capaces de analizar grandes volúmenes de datos y detectar patrones ocultos que no serían evidentes mediante métodos tradicionales.

Dada su capacidad para abordar problemas de clasificación complejos, se ha optado por evaluar los algoritmos Random Forest, Adaptive Boosting y XGBoost. Si bien estos métodos utilizan técnicas de ensemble para construir predictores robustos, es necesario determinar cuál se adapta mejor a la predicción de la deserción estudiantil en el contexto específico de esta investigación.

1.3.1 Objetivo general.

Comparación de tres modelos de clasificación utilizando algoritmos de aprendizaje automático, para predecir la deserción estudiantil en una universidad, permitiendo identificar patrones de riesgo de deserción.

1.3.2 Objetivos específicos.

- ✓ Analizar y determinar los datos que más influyen en la deserción estudiantil.
- ✓ Construir y entrenar los modelos de clasificación utilizando técnicas de aprendizaje automático, para predecir la probabilidad de deserción de los estudiantes.
- ✓ Comparar la eficacia de los algoritmos para identificar cual es el más óptimo.

CAPITULO II. FUNDAMENTACIÓN TEORICA

2. Marco teórico

2.1 ¿Qué es la deserción estudiantil?

Analizando textualmente el significado de la palabra deserción, vamos a encontrar que esta deriva del vocablo desertar, del latín "Desertare", cuyo significado es abandonar (Lugo, 2024). Para hablar de deserción estudiantil universitaria, el estudiante debe haberse matriculado en el semestre y carrera respectiva; cuando este por motivos propios o ajenos a su voluntad abandona los estudios de forma temporal o permanente antes de obtener su título académico, se habla entonces de deserción académica. (Pacho, 2011). Desde cualquier perspectiva, la deserción estudiantil universitaria es un fenómeno complejo que se refiere a la interrupción o abandono de los estudios universitarios por parte de los estudiantes independientemente de sus causas. (Seminarqa, 2024). Para Arias et al (2018), la deserción estudiantil tiene un impacto negativo en el progreso del país, tanto a nivel social como científico, pues, agudiza los problemas de desigualdad y la exclusión, incrementa el riesgo de criminalidad y delincuencia, contribuye al desempleo y subempleo, y afecta negativamente la participación cívica y la salud pública. Al mismo tiempo, en el ámbito científico, la deserción escolar universitaria representa una pérdida de talento y potencial para la innovación y el desarrollo de la sociedad, ya que limita la capacidad de investigación y generación de conocimiento, y disminuye la competitividad global en la economía del conocimiento del siglo XXI. De acuerdo a las investigaciones consultadas, este fenómeno puede tener diversas causas y consecuencias, y su comprensión es determinante para desarrollar estrategias efectivas de prevención y mitigación.

2.1.1 Factores asociados a la deserción

Según (Effer Apaza, 2012) y (Ponce de León, 2018) describieron los principales factores asociados a la deserción de estudiantes de nivel superior, destacando los siguientes:

Factor psicológico: Los procesos psicológicos son individuales, aunque rasgos como la incapacidad de adaptarse a un nuevo ambiente es un indicador de que el estudiante no

podrá enfrentar el reto que supone estar en una institución de enseñanza superior, generando en muchas ocasiones la deserción estudiantil (Effer Apaza, 2012) como se citó en (Ponce de León, 2018).

Factor social: El factor social está vinculado también al aspecto económico, siendo una valla para muchos estudiantes de seguir estudiando, ya que al formar parte de una carrera no solo implica tiempo, sino recursos que deben ser afrontados por el estudiante o su entorno. Por otro lado, si el estudiante no cuenta con los conocimientos básicos se les dificulta estar en el mismo nivel de comprensión y aprendizaje que sus compañeros, siendo un determinante de deserción (Effer Apaza, 2012)

Factor institucional: Este factor sostiene que la deserción se vincula con las cualidades de la institución, evidenciado en la infraestructura, espacios, de estudio, y sobre todo la calidad de la educación. Cuando una institución no abastece al estudiante de elementos para que pueda desarrollarse y desenvolverse, se genera rechazo a la institución, habiendo caso de deserción. (Effer Apaza, 2012)

2.1.2 Perspectivas asociadas a la deserción

Enfoque Psicológico

La característica principal se relaciona con los rasgos de personalidad que diferencian a los estudiantes que completan sus estudios de aquellos que no.

Por lo tanto, la decisión de salir o continuar en un programa académico es influenciado por el comportamiento en el nivel secundario, actitudes sobre el abandono y persistencia, y reglas subjetivas sobre estas acciones, "generando una línea de comportamiento de ambigüedad, dejadez e indiferencia que se adopta al enfrentar situaciones de la vida" (Viale, 2014), describe que los comportamientos de las personas son directamente influenciados por su sistema de creencias, lo que marca la diferencia entre los desertores estudiantiles y los que completan sus estudios universitarios.

Enfoque Sociológico

Los primeros modelos en abordar la deserción estudiantil se dieron desde una perspectiva sociológica. Estas enfatizan la influencia de factores externos al individuo en abandonar el centro de estudio, los que se suman a lo psicológico.

(Viale, 2014) Destaca que el bajo apoyo en las relaciones sociales influye para el aumento de la deserción, en este sentido sostiene que "la incapacidad para integrarse en la sociedad, la baja conciencia moral y el apoyo escaso de las relaciones sociales son aspectos que promueven las deserciones estudiantiles" (p.61).

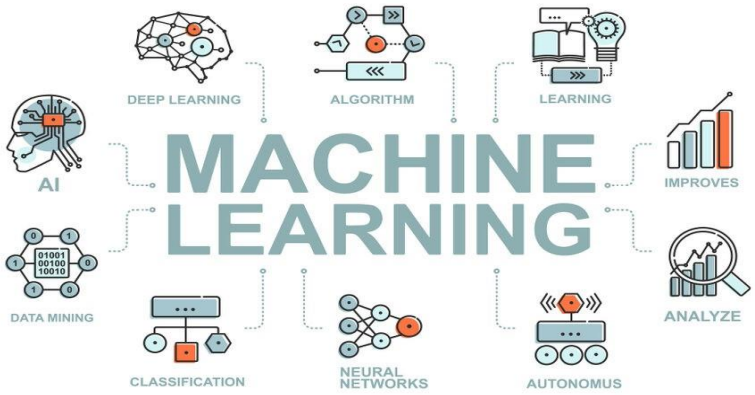
Enfoque Económico

El enfoque económico responde a la aplicación del enfoque de costo-beneficio se afirma que la racionalidad de los beneficios no siempre funciona como se esperaba, es decir, cuando los beneficios sociales y económicos generados por la universidad se percibe como inferior a los derivados de actividades alternativas, las personas eligen retirarse; es decir se declinan en seguir trabajando, ya que al percibir un sueldo y manejar dinero se dejan absorber por este sistema. (Viale, 2014)

2.2 ¿Qué es el Machine Learning?

El Machine Learning es un campo científico y, más particularmente, una subcategoría de inteligencia artificial, que consiste en dejar que los algoritmos descubran «patterns», es decir, patrones recurrentes, en conjuntos de datos. Esos datos pueden ser números, palabras, imágenes, estadísticas, etc. (DataScientest, 2023)

Figura 1. Diagrama Machine Learning



(Metaphorce, 2022) Descubre todo lo que necesitas saber sobre el Machine Learning [Imagen].

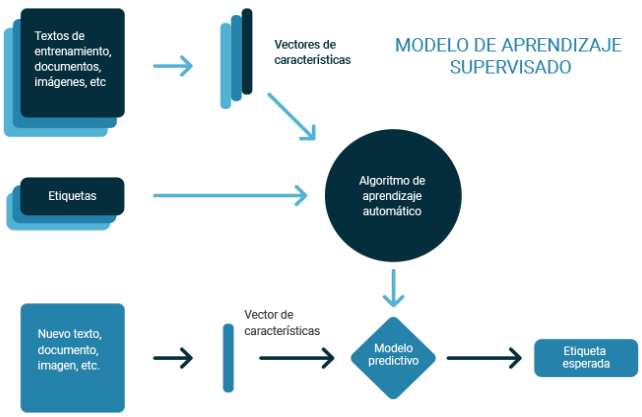
https://www.linkedin.com/pulse/descubre-todo-lo-que-necesitas-saber-sobre-el-machine-learning-/

2.2.1 Aprendizaje Supervisado

El aprendizaje supervisado es un campo del aprendizaje automático (Machine Learning) en el cual, partiendo de un conjunto de datos etiquetados, un algoritmo es capaz de entregar una predicción o una clasificación con precisión únicamente basado en los datos proporcionados.

Entre los principales algoritmos de aprendizaje supervisado se tiene la regresión lineal, la regresión logística, las máquinas de soporte vectorial, los árboles de decisión, entre otros.

Figura 2 Diagrama Aprendizaje Supervisado



(Gonzalez, 2018) Tipos de aprendizaje automático [Imagen], Meduim

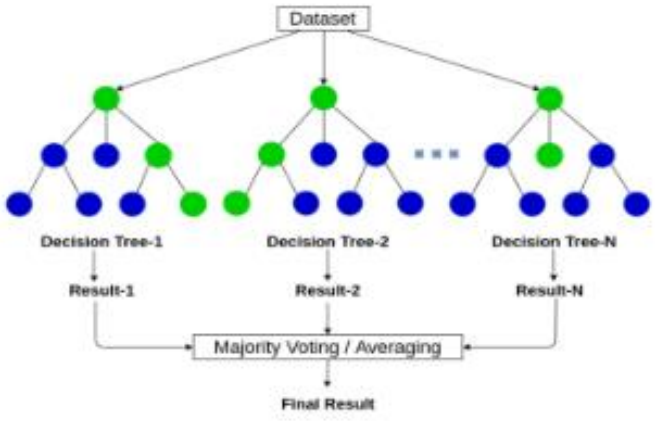
https://medium.com/soldai/tipos-de-aprendizaje-autom%C3%A1tico-6413e3c615e2

2.3 ¿Qué es Ramdon Forest?

Los algoritmos de bosque aleatorio tienen tres hiperparámetros principales que deben configurarse antes del entrenamiento. Estos incluyen el tamaño del nodo, la cantidad de árboles y la cantidad de características muestreadas. A partir de allí, el clasificador de bosque aleatorio se puede utilizar para resolver problemas de regresión o clasificación. (IBM, s.f.).

El algoritmo de bosque aleatorio se compone de una recopilación de árboles de decisión, y cada árbol del conjunto está compuesto por una muestra de datos extraída de un conjunto de entrenamiento con reemplazo, llamada muestra de arranque. (IBM, s.f.). De esa muestra de entrenamiento, un tercio se reserva como datos de prueba, conocida como muestra fuera de la bolsa, a la que volveremos más adelante. Luego se introduce otra instancia de aleatoriedad a través del empaquetamiento de características, lo que agrega más diversidad al conjunto de datos y reduce la correlación entre los árboles de decisión. Dependiendo del tipo de problema, la determinación de la predicción variará (IBM, s.f.)..Para una tarea de regresión, se promediarán los árboles de decisión individuales y, para una tarea de clasificación, un voto mayoritario (es decir, la variable categórica más frecuente) dará como resultado la clase predicha. Finalmente, la muestra fuera de la bolsa se utiliza para la validación cruzada, lo que finaliza esa predicción. (IBM, s.f.).

Figura 3. Diagrama Random Forest



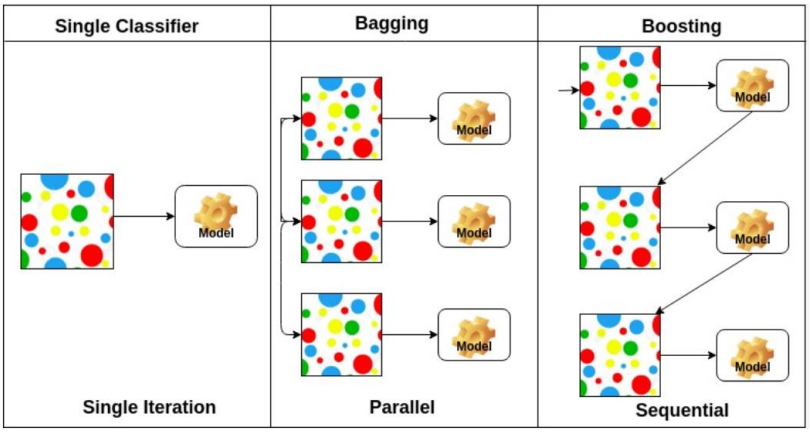
(Brital, 2021) https://anasbrital98.github.io/blog/2021/Random-Forest/

2.4 ¿Qué Adaptive Boosting?

Adaptive Boosting es un algoritmo de aprendizaje automático supervisado que se utiliza para mejorar la precisión de los modelos de clasificación débiles. (gamco, 2021) El algoritmo de AdaBoost entrena iterativamente una secuencia de clasificadores débiles en diferentes subconjuntos de datos, asignando mayores pesos a los datos que se clasificaron incorrectamente en iteraciones anteriores (gamco, 2021). Luego, combina los resultados de estos clasificadores débiles en un clasificador fuerte ponderado, en el que los

clasificadores débiles con un mejor rendimiento tienen un peso mayor en la clasificación final. (gamco, 2021)

Figura 4. Diagrama Adaptive Boosting



(datacamp, s.f.)

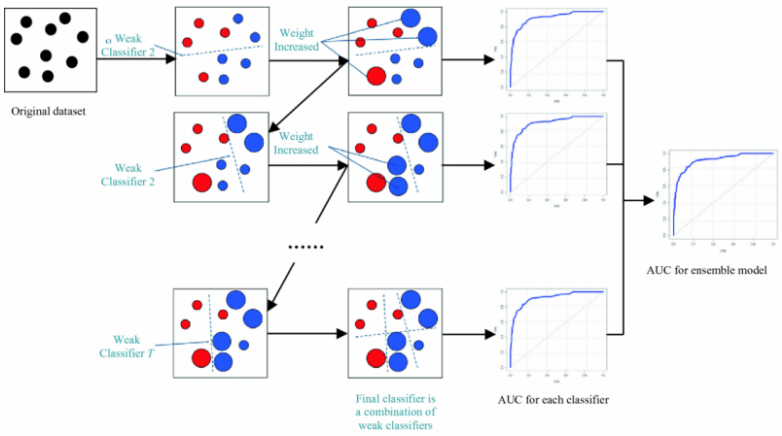
https://www.datacamp.com/tutorial/adaboost-classifier-python#rdl

2.5 ¿Qué es Gradient Boosting?

Gradient Boosting, es una familia de algoritmos usados tanto en clasificación como en regresión basados en la combinación de modelos predictivos débiles normalmente árboles de decisión para crear un modelo predictivo fuerte. La generación de los árboles de decisión débiles se realiza de forma secuencial, creándose cada árbol de forma que corrija los errores del árbol anterior. (Chaos, s.f.). Los aprendices suelen ser árboles "poco profundos" (*shallow trees*), de apenas uno, dos o tres niveles de profundidad, típicamente. (Chaos, s.f.)

Este tipo de algoritmos suelen ofrecer los mejores resultados en escenarios tabulares, y son especialmente destacables las implementaciones de LightBGM y XGBoost. (Chaos, s.f.) Uno de los parámetros de este tipo de argumentos es la tasa de aprendizaje, que controla el grado de mejora de un árbol respecto del anterior. Una tasa de aprendizaje pequeña supone una mejora más lenta pero adaptándose mejor a los datos, lo que se traduce generalmente en mejoras en el resultado a costa de un mayor consumo de recursos. (Chaos, s.f.).

Figura 5. Diagrama Gradient Boosting



(Datos, 2020)

https://datascience.eu/machine-learning/gradient-boosting-what-you-need-to-know/#google_vignette

2.6 ¿Que es Visual Studio?

Visual Studio es una herramienta de desarrollo eficaz que permite completar todo el ciclo de desarrollo en un solo lugar. Es un entorno de desarrollo integrado (IDE) completo que puede usar para escribir, editar, depurar y compilar el código. A continuación, implemente la aplicación. Visual Studio incluye compiladores, herramientas de finalización de código, control de código fuente, extensiones y muchas más características para mejorar cada fase del proceso de desarrollo de software. (Challenge, 2024).

2.7 ¿Que Amazon Sagemaker?

Amazon SageMaker es un servicio de aprendizaje automático (ML) totalmente gestionado (aws, 2024). Con SageMaker, los científicos de datos y los desarrolladores pueden crear, entrenar e implementar modelos de aprendizaje automático de forma rápida y segura en un entorno listo para la producción. Proporciona una experiencia de interfaz de usuario para ejecutar flujos de trabajo de aprendizaje automático que hace que las herramientas de SageMaker estén disponibles en varios entornos de desarrollo integrados. IDEs (aws, 2024)

SageMaker ofrece opciones de formación distribuidas y flexibles que se ajustan a sus flujos de trabajo específicos. En unos pocos pasos, puede implementar un modelo en un entorno seguro y escalable desde la SageMaker consola. (aws, 2024)

2.7.1 ¿Qué es Canvas?

Amazon SageMaker Canvas es una interfaz visual sin código que permite preparar datos, crear y desplegar modelos de ML de alta precisión, lo que agiliza el ciclo de vida integral del ML en un entorno unificado. Puede preparar y transformar datos a escala de petabytes con interacciones simples, en lenguaje natural y con la tecnología de SageMaker Data Wrangler. (IBM, s.f.). Se puede aprovechar la potencia de AutoML y crear modelos personalizados de ML de manera automática para la regresión, la clasificación, la previsión de series temporales, el procesamiento del lenguaje natural y la visión artificial, compatibles con SageMaker Autopilot. Con Canvas, acelera la innovación y aumenta la productividad al crear rápidamente modelos de ML personalizados o ajustar los modelos fundacionales para satisfacer las necesidades del usuario, sin importar su experiencia en codificación. (aws, 2024).

Figura 6. Flujo de Trabajo en un Entorno de Preparación y Análisis de Datos

(Services, s.f.) https://aws.amazon.com/es/free/

2.8 Fundamentos de programación en Python.

2.8.1 ¿Qué es Pyhton?

Python es un lenguaje de alto nivel de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. Se trata de un lenguaje de programación multiparadigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma. (Wikipedia, s.f).

Figura 7. Logo de Python

(Paz, 2023) Primeros pasos con Python: ¡Hola, mundo! [Imagen]

https://blog.vermiip.es/primeros-pasos-con-python-hola-mundo/

2.8.2 Python para la Ciencia de Datos.

Python ofrece varias características que lo vuelven atractivo para utilizarlo como lenguaje de programación en tareas de Machine Learning, su sintaxis fácil de leer y entender, el soporte de su comunidad y las librerías diseñadas específicamente para la gestión de datos, procesamiento analítico y la creación de modelos de aprendizaje automático lo convierten en una herramienta útil para de la Ciencia de Datos. (Noddar, 2022)

2.8.3 Principales librerías y frameworks.

Dentro de las principales librerías que tiene Python para el manejo de datos en ciencia de datos existen las librerías Pandas, Numpy, Matplotlib, Sckit-Learn y Keras/Tensorflow. (Verne Academy, 2022)

- **Pandas**

Facilita el manejo de datos, permite leer distintos tipos de archivos o bases de datos, extensiones tipo csv, htlm, xls, hdfs, entre otros. (IBM, s.f.).

- **Sckit-Learn**

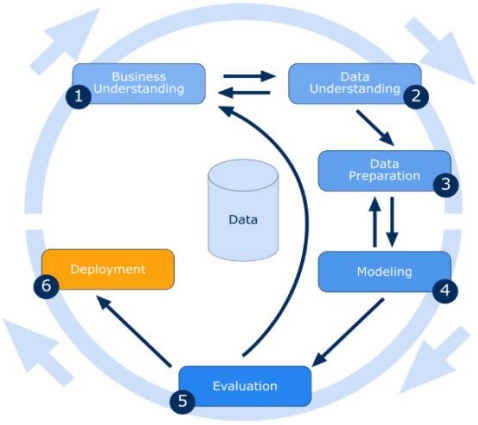
Contiene una gran variedad de algoritmos optimizados para tareas de machine Learning, dentro de su basto catalogo se destacan algoritmos de regresión, clasificación, reducción de dimensiones, clustering, entre otros. (IBM, s.f.).

2.9 Metodología CRISP-DM

La metodología CRISP-DM (Cross-Industry Standard Process for Data Mining) integra todas las tareas necesarias en los proyectos de minería de datos, desde la fase de comprensión del problema hasta la puesta en producción de sistemas automatizados analíticos, predictivos y/o prospectivos. (Chapman, 2000)

2.9.1 Fases de la Metodología CRISP-DM

Figura 8. Etapas de desarrollo en Crisp-DM



(Gonçalves, 2020) CRISP-DM - Cross Industry Standard Process for Data Mining

https://www.ibm.com/docs/es/spss-modeler/saas?topic=dm-crisp-help-overview

CRISP-DM está compuesta por seis fases, las cuales dependen entre sí tanto en forma secuencial como cíclica, pudiendo encontrarse interacciones que permitan mejorar la aproximación obtenida en otras fases anteriores. (Chapman, 2000). Las fases son las siguientes:

1. Comprensión del negocio.
2. Comprensión de los datos.
3. Preparación de los datos.
4. Modelado.
5. Evaluación.
6. Despliegue.

- **Comprensión del Negocio.**

Esta es la primera fase por donde debe empezar todo proyecto de minería de datos, comprendiendo en profundidad el problema que se quiere resolver y estableciendo los objetivos del proyecto desde una perspectiva empresarial para luego trasladarlos a objetivos técnicos y a un plan de proyecto. (Chapman, 2000)
En primer lugar, se realiza una evaluación de la situación actual determinando los antecedentes y requisitos del problema, tanto en términos de negocio como de minería de datos. Y por último, se realiza un plan de proyecto donde se tiene en cuenta qué pasos se deben seguir y qué procedimientos se utilizarán en cada uno de ellos. (Chapman, 2000)

- **Comprensión de los Datos**

Es esta fase se lleva a cabo la recolección y exploración inicial de los datos, con el objetivo de establecer una primera relación con el problema. Esta fase suele ser crítica en el proyecto dado que un mal entendimiento de los datos tiene como consecuencia un aumento en el tiempo global del proyecto y también reduce las garantías de éxito. (Chapman, 2000)

Para llevar a cabo esta fase hay que desarrollar una serie de tareas. La primera es recolectar datos iniciales y adaptarlos a las necesidades del proyecto para su posterior procesamiento.
Luego se deben describir formalmente los datos obtenidos: número de instancias y atributos , el significado de los atributos y una descripción rigurosa del formato de los datos. (Chapman, 2000) Después se exploran los datos aplicando técnicas básicas de estadística descriptiva que revelan propiedades de estos. Por último, se lleva a cabo una verificación de los datos para determinar su consistencia, la cantidad y distribución de los valores nulos o valores fuera de rango que puedan provocar ruido en el modelado posterior. (Chapman, 2000)

- **Preparación de los Datos.**

La fase de preparación trata de seleccionar, limpiar y generar conjuntos de datos correctos, organizados y preparados para la fase de modelado. Esta es una fase sumamente crítica en un proyecto de minería de datos. (Chapman, 2000) Los errores en los datos que se pasan por alto y que no son resueltos en esta fase se trasladan hasta la fase de modelado, lo que genera una reducción en la exactitud de los modelos o incluso, es posible entregar al cliente resultados basados en datos que aún contienen errores no detectados.
Por esta razón, esta fase es crucial y generalmente demanda siempre el mayor esfuerzo y tiempo del proyecto, aproximadamente un 75% del tiempo total. (Chapman, 2000)

- **Modelado.**

En esta fase se lleva a cabo la creación de modelos de conocimiento a partir de los datos suministrados desde la fase anterior. (Chapman, 2000) Estos modelos de conocimiento pueden ser de distintos tipos, por ejemplo, se pueden crear modelos de clasificación o regresión con el objetivo de estimar o inferir el valor de una determinada variable.
Para afrontar esta fase hay que seguir una serie de pautas que nos ayudarán a obtener mejores resultados. Posteriormente se genera un plan de prueba, donde configuramos los valores de los parámetros que se usarán para los algoritmos de aprendizaje automático ya que muchos de estos pueden ser configurados para determinar las características del

modelo que se generará. También, se determinan las métricas de evaluación que se calcularán para evaluar los modelos.

- **Evaluación.**

Si los modelos obtenidos cumplen con las expectativas de negocio, se procede a la explotación del modelo. Si no, se evalúa en esta fase si se procede a iterar nuevamente sobre los pasos anteriores con el objetivo de encontrar nuevos resultados. (Chapman, 2000)

- **Despliegue.**

En esta última fase es donde se definen las estrategias para su implementación, monitorización y mantenimiento de los modelos. Esto nos ayudará a observar cualquier comportamiento irregular del sistema y corregirlo de forma que no provoque deficiencias en el servicio del cliente. (Chapman, 2000)

Finalmente, se lleva a cabo una revisión final de todo el proceso llevado a cabo en el proyecto, evaluando las acciones realizadas de forma correcta e incorrecta, con el fin de enumerar las lecciones aprendidas en el proyecto. (Chapman, 2000)

CAPITULO III. DESARROLLO DE LA METODOLOGÍA

3.1 Comprensión del Negocio

La universidad cuenta actualmente con aproximadamente 9 000 estudiantes regulares de pregrado, y se ha detectado que la deserción se concentra principalmente en el primer año, llegando a representar hasta un 10 % de la población estudiantil. Este problema se ha asociado con la insuficiente preparación académica de algunos estudiantes recién graduados del colegio, quienes ingresan con bases limitadas para afrontar la vida universitaria.

Ante esta situación, la universidad requiere hacer un análisis profundo de los datos disponibles e históricos de los estudiantes a partir del año 2023 hasta la actualidad para identificar que patrones, tendencias y causas subyacentes que inciden en la deserción, con el fin de diseñar estrategias efectivas que reduzcan el abandono y mejoren la retención de los estudiantes.

3.1.1 Objetivo del Negocio

Disminuir la tasa global de deserción estudiantil del 10% al 5%, con especial atención en los estudiantes de primer ingreso, a través del análisis de datos históricos y la aplicación de técnicas de machine learning para identificar factores de riesgo y desarrollar estrategias de intervención temprana.

3.1.2 Objetivo de Minería

Aplicar modelos de machine learning para clasificación con el objetivo de analizar patrones en los datos históricos de los estudiantes e identificar los factores que influyen en la deserción universitaria durante el primer año de estudios.

3.2 Comprensión de los Datos

En esta segunda fase se recolectará, analizará, interpretará y se hará una limpieza de los datos.

3.2.1 Recolección de Datos

Para el desarrollo de este trabajo se utilizará datos históricos privados que contiene información de estudiantes de una universidad privada del Ecuador desde el año 2023 hasta la actualidad.

Este data set contiene variables académicas y demográficas.

El dataset contiene 30 características y 119981 registros.

La estrategia para determinar la variable objetivo "Deserción" se base en las siguientes reglas:

- ✓ ***Desertó*:** Si el estatus académico del estudiante es: Inactivo, Retiro Temporal, Retiro Defnitivo, Separación Académica, Separación Disciplinaria, Desistimiento Administrativo, Suspensión Disciplinaria, Suspención Académica.

- ✓ ***No desertó:*** Si el estatus académico del estudiante es: Activo o Graduado-Inactivo.

3.2.2 Descripción de los Datos

Los datos para esta investigación fueron otorgados por el Departamento de Registro de la Universidad. Las características más importantes que fueron seleccionadas son por su pertinencia y alineación con los procesos y objetivos institucionales.

En la siguiente tabla se muestra el nombre de variable, la descripción, el tipo y uso en el análisis.

Tabla 1. Nombre de variable, descripción, tipo y uso en el análisis

Variable	Descripción	Tipo	Uso en el análisis
BANNER ID	Identificador único del estudiante.	Categórico	No se emplea como predictor.
CODIGO_GENERO	Código numérico que identifica el género del estudiante (1 = masculino, 2 = femenino).	Categórico	Permite agrupar y analizar diferencias de deserción por género.
GENERO	Género declarado del estudiante (masculino, femenino, etc.).	Categórico	Útil para segmentar el análisis (p. ej., comparar tasas de deserción entre géneros).
ESTADO CIVIL	Estado civil del estudiante (soltero, casado, etc.).	Categórico	Explora si la situación civil influye en la dedicación y probabilidad de deserción.
CODIGO DISCAPACIDAD	Código que indica la presencia y tipo de discapacidad.	Categórico	Evalúa el impacto de condiciones especiales en la continuidad de estudios.
ETNIA	Grupo étnico al que pertenece el estudiante (mestizo, afrodescendiente, indígena, etc.).	Categórico	Analiza si hay brechas de deserción asociadas a factores socio-culturales.

EDAD	Edad del estudiante (en años).	Numérico	Permite correlacionar la edad con la probabilidad de deserción (estudiantes más jóvenes vs. Mayores)
TIPO ESTUDIANTE	Clasificación del alumno regular.	Categórico	Identifica si ciertas condiciones de ingreso (becas, convenios) influyen en la retención.
NOMBRE COLEGIO	Nombre del colegio de procedencia del estudiante.	Categórico	Analiza posible relación entre el colegio de origen y la probabilidad de deserción.
NOTA_COLEGIO	Promedio o nota final obtenida en la educación secundaria.	Numérico	Indica el nivel académico previo; puede correlacionarse con el rendimiento universitario.
CIUDAD	Ciudad de origen o residencia del estudiante.	Categórico	Evalúa disparidades regionales en la deserción (p. ej., si el lugar de residencia influye en el abandono).
NIVEL	Nivel académico (por ejemplo, pregrado, posgrado) o curso actual.	Categórico	Distingue en qué fase o nivel de estudios se encuentra el alumno.
COD COLEGIO	Código interno del colegio de procedencia.	Categórico	Similar a “NOMBRE COLEGIO”; puede usarse para unificar y cruzar datos con otras fuentes.

COLEGIO	Descripción adicional o denominación oficial del colegio.	Categórico	Complementa “NOMBRE COLEGIO”; a veces se combina o depura para evitar duplicados.
CARRERA	Programa académico cursado (Ingeniería, Administración, etc.).	Categórico	Identifica si la deserción varía según la carrera elegida.
PRIMER_REGISTRO	Período o fecha de ingreso a la universidad.	Categórico /Fecha	Permite medir la duración de la trayectoria y detectar patrones de deserción temprana.
ULTIMO PERIODO REGISTRO	Período o fecha del último registro de matrícula del estudiante.	Categórico /Fecha	Útil para identificar hasta cuándo el estudiante estuvo activo y si existe retiro posterior.
SEMESTRE	Semestre o año académico en el que se encuentra el estudiante.	Categórico	Analiza la deserción por etapas académicas (1er semestre vs. semestres avanzados).
CRED_REG	Créditos inscritos en el período en curso.	Numérico	Evalúa carga académica asumida y su relación con la deserción.
CRED_APROB	Créditos aprobados hasta la fecha o en ese período.	Numérico	Refleja el rendimiento académico y permite calcular tasas de aprobación.
GPA_PERIODO	Promedio (GPA) en el período más reciente cursado.	Numérico	Indica desempeño reciente; posible predictor de riesgo de deserción.

GPA_ACUMULADO	Promedio (GPA) acumulado a lo largo de la carrera.	Numérico	Métrica global de rendimiento a largo plazo; factor importante en la permanencia.
CODIGO_ESTADO	Código que describe la situación académica del estudiante (activo, inactivo, egresado...).	Categórico	Permite identificar la condición actual dentro de la universidad.
ESTADO_ACAD	Estado académico en términos descriptivos (por ejemplo, “En curso”, “Retirado”, “Graduado”).	Categórico	Describe la continuidad o finalización de estudios; clave para etiquetar a desertores.
ESTADO_ACTUAL	Situación actual del estudiante.	Categórico	Útil para el proceso de etiquetar la deserción.
NOMBRE CURSO	Asignatura o materia específica que cursa o cursó el estudiante.	Categórico	Profundiza en el rendimiento por curso; posible exploración de materias con alta tasa de abandono.
NOTA CURSO	Calificación obtenida en dicha asignatura.	Numérico	Indica el desempeño en cada materia; podría asociarse a factores de riesgo de deserción si las notas son bajas.
REG_TITULACION	Indica si el estudiante está en proceso de titulación.	Categórico	Mide la fase final de la carrera; puede reflejar menor riesgo de deserción.

TOTAL_CREDITOS_APROBADOS	Créditos acumulados aprobados en toda la trayectoria académica.	Numérico	Muestra progreso global; puede relacionarse con la probabilidad de perseverar hasta graduarse.

3.3 Preparación de Datos

Se procede a la exploración de los datos se utilizando el lenguaje de programación Python con todas sus librerías necesarias para realizar un análisis detallado de las variables más relevantes que se usará en los modelos.

Se procede con la lectura del archivo Datos_academicos.xlsx

```
import pandas as pd
# Leer el nuevo archivo
df = pd.read_excel('Datos_academicos.xlsx')
```

Figura 9. Datos académicos

```
df.head(5)
```
✓ 0.2s Python

	CODIGO_GENERO	GENERO	ESTADO CIVIL	CODIGO DISCAPACIDAD	ETNIA	EDAD	TIPO ESTUDIANTE	NOMBRE COLEGIO	NOTA_COLEGIO	CIUDAD
0	F	0	S	NINGUNO	INDÍGENA	29.0	N	Republica Del Ecuador	8.95	Otavalo
1	F	0	S	NINGUNO	INDÍGENA	29.0	N	Republica Del Ecuador	8.95	Otavalo
2	F	0	S	NINGUNO	INDÍGENA	29.0	N	Republica Del Ecuador	8.95	Otavalo
3	F	0	S	NINGUNO	INDÍGENA	29.0	N	Republica Del Ecuador	8.95	Otavalo
4	F	0	S	NINGUNO	INDÍGENA	29.0	N	Republica Del Ecuador	8.95	Otavalo

```
df.shape
```
✓ 0.0s

```
(119981, 30)
```

En la Figura 9 se observa que el dataset tiene 119981 observaciones y registra 30 columnas, 29 variables predictoras y 1 variable respuesta.

3.3.1 Verificar la calidad de los datos

Se verifica si existen valores NaN o valores nulos y perdidos.

```
BANNER ID                      0
CODIGO_GENERO                  0
GENERO                         0
ESTADO CIVIL                   0
CODIGO DISCAPACIDAD            0
ETNIA                          0
EDAD                           0
TIPO ESTUDIANTE                0
NOMBRE COLEGIO                 0
NOTA_COLEGIO                 621
CIUDAD                       443
NIVEL                          0
COD COLEGIO                    0
COLEGIO                        0
CARRERA                        0
PRIMER_REGISTRO                0
ULTIMO PERIODO REGISTRO        0
SEMESTRE                       0
CRED_REG                       0
CRED_APROB                     0
GPA_PERIODO                    0
GPA_ACUMULADO                  0
CODIGO_ESTADO                  0
ESTADO_ACAD                    0
ESTADO_ACTUAL                  0
PERIODO                        0
NOMBRE CURSO                   0
NOTA CURSO                   118
REG_TITULACION                 0
TOTAL_CREDITOS_APROBADOS       0
dtype: int64
```

3.3.2 Limpieza de datos

Se procede a eliminar las filas NOTA_COLEGIO, CIUDAD, NOTA CURSO porque la cantidad de observaciones con valores nulos (621 en NOTA_COLEGIO, 443 en CIUDAD y 118 en NOTA CURSO) no representa un porcentaje alto como para dejar al conjunto de datos sin información relevante.

```
# Eliminar columnas con valores nulos
# NOTA_COLEGIO
# CIUDAD
# NOTA CURSO
```

```
# Eliminar filas con valores nulos
df.dropna(subset=['NOTA_COLEGIO', 'CIUDAD', 'NOTA CURSO'],
inplace=True)
```

Figura 10. Datos sin Valores Nulos

```
Data columns (total 30 columns):
 #   Column                    Non-Null Count   Dtype
---  ------                    --------------   -----
 0   BANNER ID                 119099 non-null  int64
 1   CODIGO_GENERO             119099 non-null  object
 2   GENERO                    119099 non-null  int64
 3   ESTADO CIVIL              119099 non-null  object
 4   CODIGO DISCAPACIDAD       119099 non-null  object
 5   ETNIA                     119099 non-null  object
 6   EDAD                      119099 non-null  int64
 7   TIPO ESTUDIANTE           119099 non-null  object
 8   NOMBRE COLEGIO            119099 non-null  object
 9   NOTA_COLEGIO              119099 non-null  float64
 10  CIUDAD                    119099 non-null  object
 11  NIVEL                     119099 non-null  object
 12  COD COLEGIO               119099 non-null  object
 13  COLEGIO                   119099 non-null  object
 14  CARRERA                   119099 non-null  object
 15  PRIMER_REGISTRO           119099 non-null  int64
 16  ULTIMO PERIODO REGISTRO   119099 non-null  int64
 17  SEMESTRE                  119099 non-null  object
 18  CRED_REG                  119099 non-null  int64
 19  CRED_APROB                119099 non-null  int64
...
 28  REG_TITULACION            119099 non-null  int64
 29  TOTAL_CREDITOS_APROBADOS  119099 non-null  int64
dtypes: float64(3), int64(10), object(17)
```

Luego de realizar una parte de la limpieza de los datos se puede observar en la Figura 10 que el dataset está completo, no mantiene valores perdidos o valores nulos.

```
# Remplazar NINGUNO POR METIZO en la columna ETNIA
df['ETNIA']. replace('NINGUNO', 'MESTIZO', inplace=True)
```

Se realiza este procesamiento de datos porque de acuerdo a la universidad ha indicado que asume que la mayoría de los estudiantes con valor NINGUNO son mestizos por tal razón se reemplaza por MESTIZO esto ayuda a minimizar la pérdida de información y permite mantener la integridad de los datos.

```
# Imputar los datos de la columna EDAD 5, 1, 2, 3, 4 por la
media de la columna

df['EDAD']. replace ([5, 1, 2, 3, 4], df['EDAD']. mean (),
inplace=True)
```

La imputación de la columna 'EDAD' se justifica por la necesidad de corregir errores en los datos y asegurar que los valores sean más representativos a la edad real de los

estudiantes. En este caso se elige la media debido a su robustez frente a valores atípicos y su capacidad para preservar la distribución original de los datos.

Los valores a reemplazar con la media son 5, 1, 2, 3 y 4, estos se reemplazan porque se han identificado como errores de tipeo.

```
# Cambiar nombres de las categóricas de la variable objetivo.
# Graduado - Inactivo -> No Deserto
# Activo -> No Deserto
# Inactivo -> Deserto
# Retiro Temporal -> Deserto
# Retiro Definitivo -> Deserto
# Separación Académica -> Deserto
# Separación Disciplinaria -> Deserto
# Desistimiento Admisión -> Deserto
# Suspensión Disciplinaria -> Deserto
# Suspension Académica -> Deserto
# Retiro Voluntario -> Deserto

df['ESTADO_ACTUAL'] = df['ESTADO_ACTUAL'].replace({'Graduado -
Inactivo': 'No Deserto', 'Activo': 'No Deserto', 'Inactivo':
'Deserto', 'Retiro Temporal': 'Deserto', 'Retiro Definitivo':
'Deserto', 'Separación Académica': 'Deserto', 'Separación
Disciplinaria': 'Deserto', 'Desistimiento Administrativo':
'Deserto', 'Suspension Disciplinaria': 'Deserto', 'Suspensión
Académica': 'Deserto'})
df['ESTADO_ACTUAL'].unique()
```

Para generar la variable objetivo ESTADO_ACTUAL, se aplicó una regla de recodificación basada en la normativa académica interna de la universidad. Esta regla define la deserción de la siguiente manera: los estudiantes con estados 'Inactivo', 'Retiro Temporal', 'Retiro Definitivo', 'Separación Académica', 'Separación Disciplinaria', 'Desistimiento Administrativo', 'Suspension Disciplinaria' y 'Suspensión Académica' se consideran desertores ('Deserto').

Los estudiantes con estados 'Graduado - Inactivo' y 'Activo' se consideran no desertores ('No Deserto').

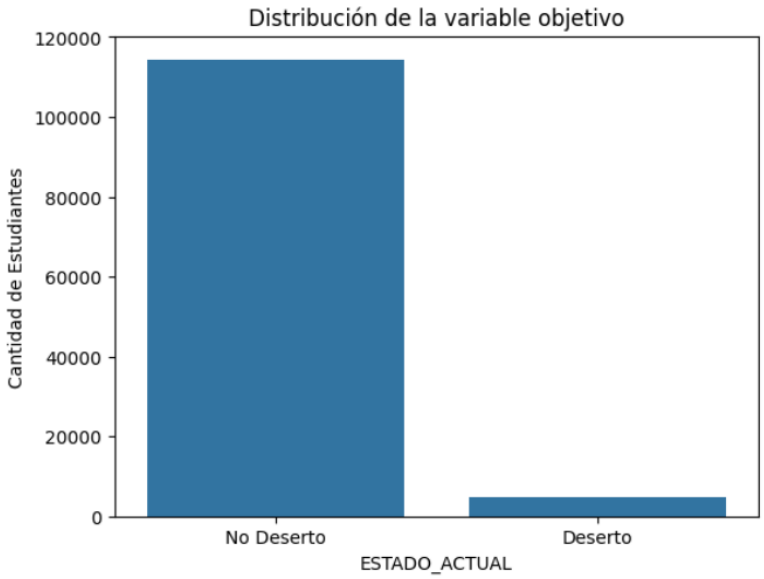
```
df['ESTADO_ACTUAL'].value counts ()

ESTADO_ACTUAL
No Deserto    114383
Deserto         4716
Name: count, dtype: int64
```

3.3.3 Análisis exploratorio de datos

Como parte inicial se realiza un análisis visual, se procede con la creación de un gráfico de barras para conocer la distribución de la variable objetivo.

```
# Graficar la variable objetivo.
import seaborn as sns
import matplotlib. pyplot as plt

sns.countplot(data=df, x='ESTADO_ACTUAL')
plt.title('Distribución de la variable objetivo')
plt.ylabel('Cantidad de Estudiantes')
plt.show()
```

Gráfico 1. Distribución de la variable objetivo

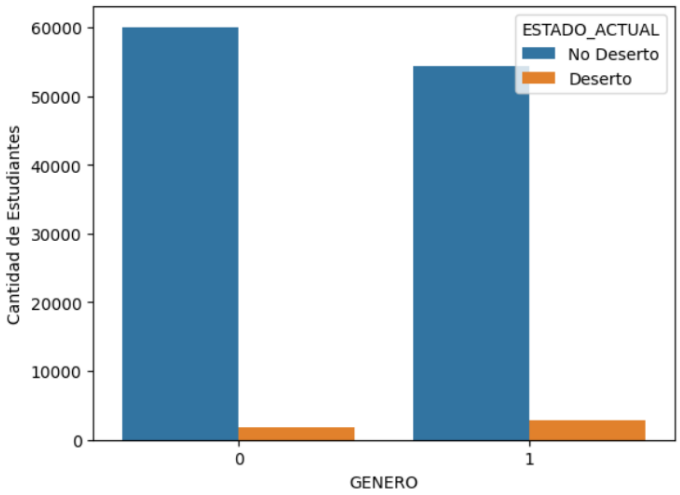


En el gráfico 1 se puede observar que existe un claro desbalance entre las clases. Las observaciones que pertenecen a la clase "No Deserto" son aproximadamente 110.000, mientras que las observaciones que pertenecen a la clase "Deserto" son cerca de 5.000, representando apenas alrededor del 4.3% del total de las observaciones.

```
# Desercion por Género
# 0 -> Femenino
# 1 -> Masculino
```

```
sns.countplot(data=df, x='GENERO', hue='ESTADO_ACTUAL')
plt.ylabel('Cantidad de Estudiantes')
plt.show()
```

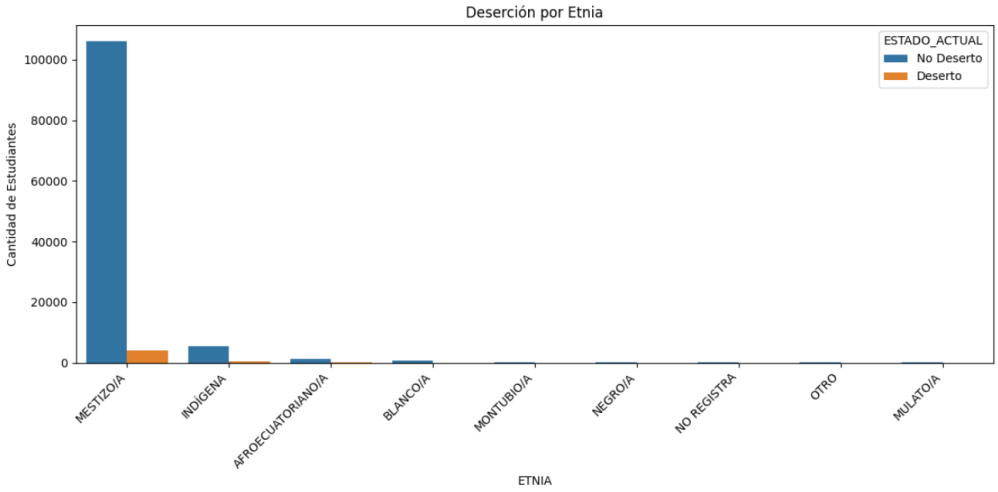
Gráfico 2. Estado actual (No deserto vs Deserto)



En el gráfico 2 se puede observar que, aunque ambos géneros tienen una baja tasa de deserción, los hombres muestran una tasa ligeramente mayor en comparación con las mujeres.

```
# Deserción por Etnia
plt.figure(figsize=(12, 6))  # Más ancho
order_etnias = df['ETNIA'].value_counts().index  # Ordena de
mayor a menor

sns.countplot(
    data=df,
    x='ETNIA',
    hue='ESTADO_ACTUAL',
    order=order_etnias
)

plt.xticks(rotation=45, ha='right')  # Rotación de etiquetas y
alineación
plt.title('Deserción por Etnia')
plt.tight_layout()
plt.show()
```

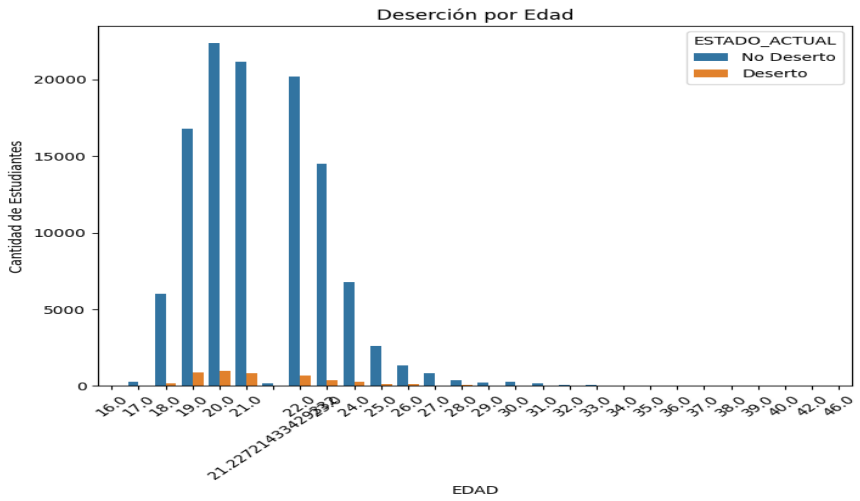
Gráfico 3. Deserción por Etnia



En el gráfico 3 se muestra la distribución por Etnia. La etnia mestiza es la que más estudiantes tiene en ambas categorías.

```
# Desercion por Edad
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='EDAD', hue='ESTADO_ACTUAL')
plt.title('Deserción por Edad')
plt.ylabel('Cantidad de Estudiantes')

plt.xticks(rotation=45)
plt.show()
```

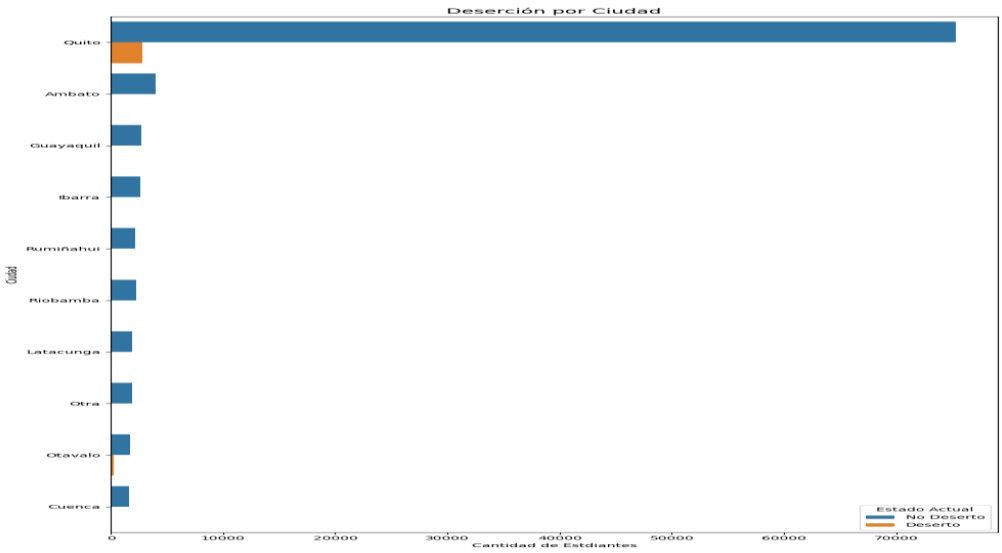
Gráfico 4. Deserción por Edad



En el gráfico 4 se observa un pico en la cantidad de estudiantes alrededor de los 19-21 años, lo que podría indicar un punto común para los estudiantes.

```
# Deserción por Ciudad.
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(10, 15))  # ajusta alto y ancho

# 1) Construir un orden de ciudades por frecuencia total EL TOP 10

order_cities = df['CIUDAD'].value_counts().head(10).index

sns.countplot(
    data=df,
    y='CIUDAD',
    hue='ESTADO_ACTUAL',
    order=order_cities
)

plt.title('Deserción por Ciudad', fontsize=14)
plt.xlabel('Cantidad de Estdiantes')
plt.ylabel('Ciudad')

plt.legend(title='Estado Actual', loc='lower right')  # Ajusta posición de la leyenda

plt.tight_layout()  # ajusta márgenes para evitar recortes
plt.show()
```

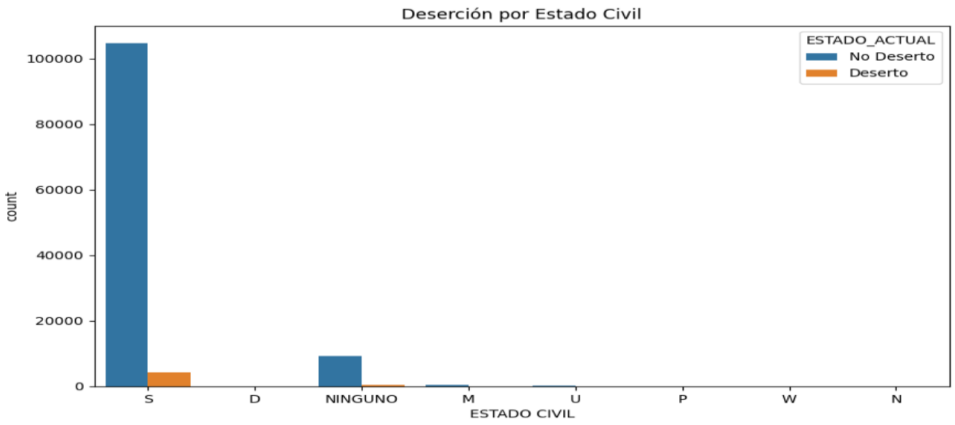
Gráfico 5. Deserción por Ciudad



En el gráfico 5 se observa que Quito tiene la mayor cantidad de estudiantes en comparación con las demás ciudades. También presenta un número significativo de estudiantes que han desertado.

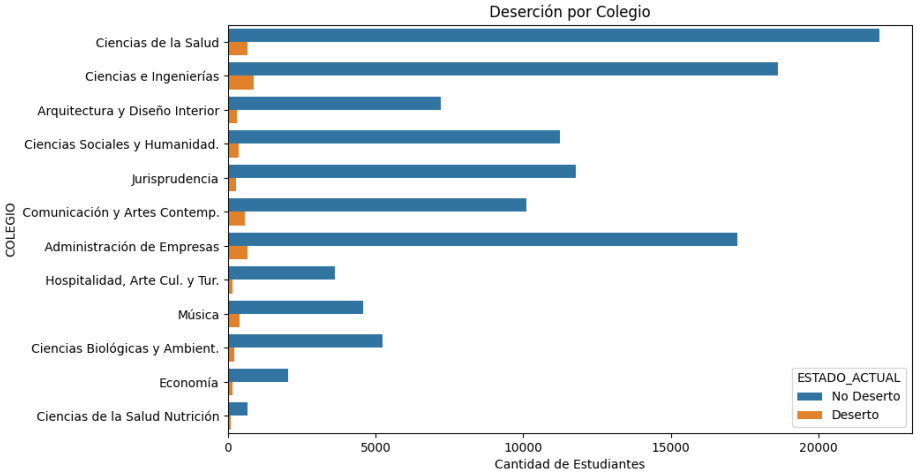
```
# Desercion por Estado Civil
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='ESTADO CIVIL', hue='ESTADO_ACTUAL')
plt.title('Deserción por Estado Civil')
plt.show()
```

Gráfico 6. Deserción por Estado Civil




```
# Deserción por Colegio
plt.figure(figsize=(10, 6))
sns.countplot(data=df, y='COLEGIO', hue='ESTADO_ACTUAL')
plt.title('Deserción por Colegio')
plt.xlabel('Cantidad de Estudiantes')
plt.show()
```

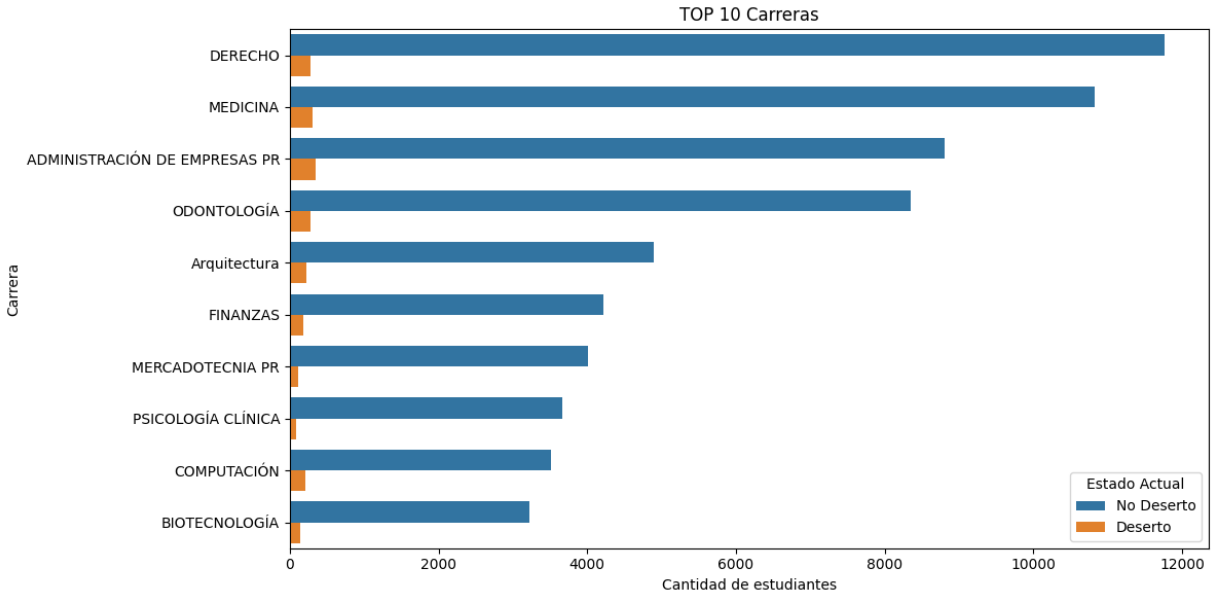
Gráfico 7. Deserción por Colegio



En gráfico 7 se puede observar que la deserción es más visible en los colegios de Ciencias de la Salud, Ciencias e Ingenierías y Administración de Empresas.

```
# Deserción por Carrera.
# 1) Obtener el listado de las 10 carreras más frecuentes
top_10_carreras = df['CARRERA'].value_counts().head(10).index

# 2) Filtrar el DataFrame para solo esas carreras
df_top10 = df[df['CARRERA'].isin(top_10_carreras)]

# 3) Graficar
plt.figure(figsize=(12, 6))
sns.countplot(
    data=df_top10,
    y='CARRERA',
    hue='ESTADO_ACTUAL',
    order=top_10_carreras
)

plt.title('TOP 10 Carreras')
plt.xlabel('Cantidad de estudiantes')
plt.ylabel('Carrera')
plt.legend(title='Estado Actual', loc='lower right')
plt.tight_layout()
plt.show()
```

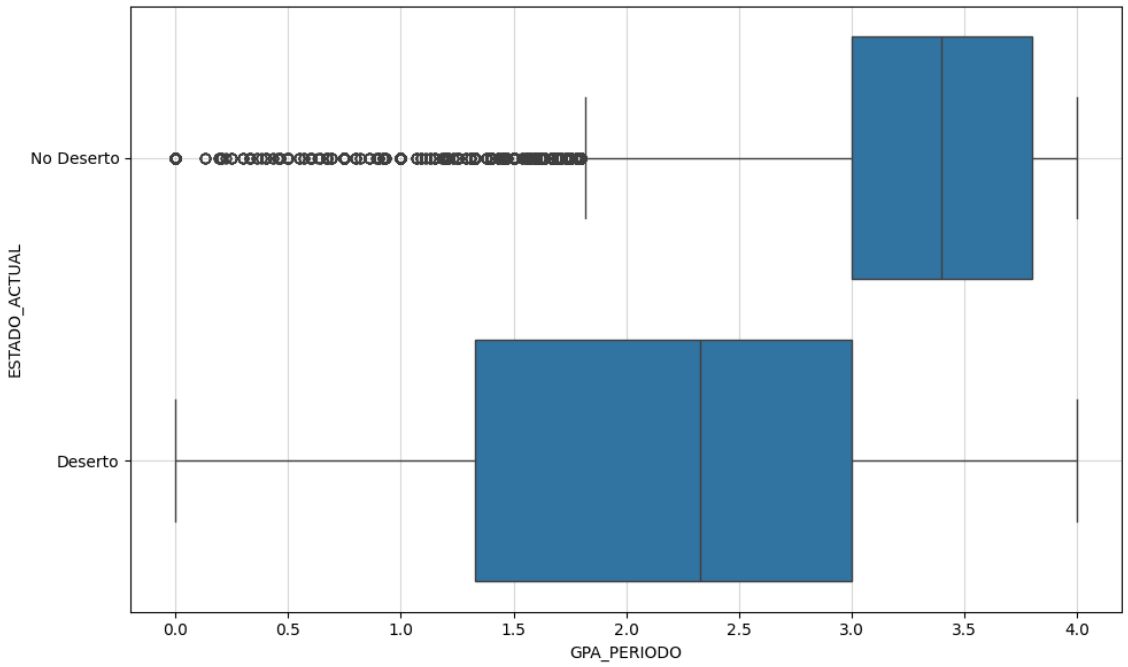
Gráfico 8. Top 10 carreras



En el gráfico 8 se muestra el Top 10 de las carreras y la cantidad de estudiantes que desertaron y que no desertaron.

```
# Deserción por GPA Periodo
plt.figure(figsize=(10, 6))
sns.boxplot(data=df, x='GPA_PERIODO', y='ESTADO_ACTUAL')
plt.grid(alpha=0.5)
plt.tight_layout()
plt.show()
```

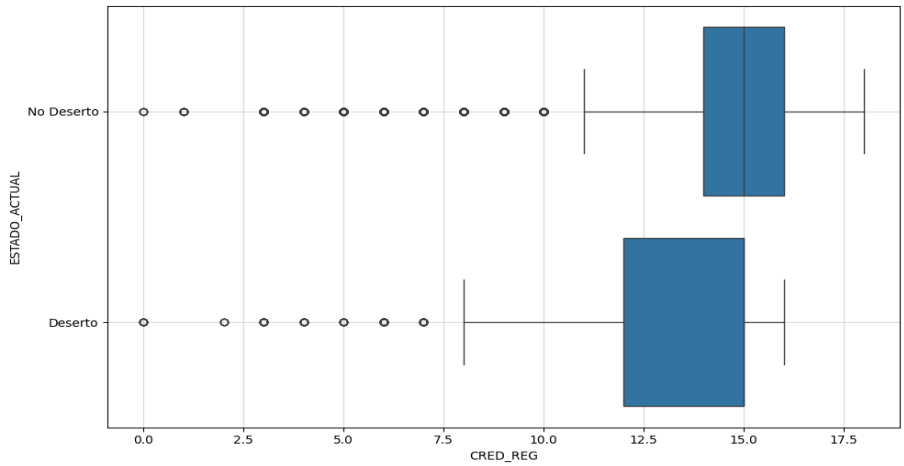
Gráfico 9. Estado actual y GPA por periodo



En el gráfico 9 podemos observar que la mediana está alrededor de 2.5, lo que sugiere que la mitad de los estudiantes que desertaron tienen un GPA de 2.5 o superior.

```
# Gráfico créditos registrados vs deserción
plt.figure(figsize=(10, 6))
sns.boxplot(data=df, x='CRED_REG', y='ESTADO_ACTUAL')
plt.grid(alpha=0.5)
plt.tight_layout()
```

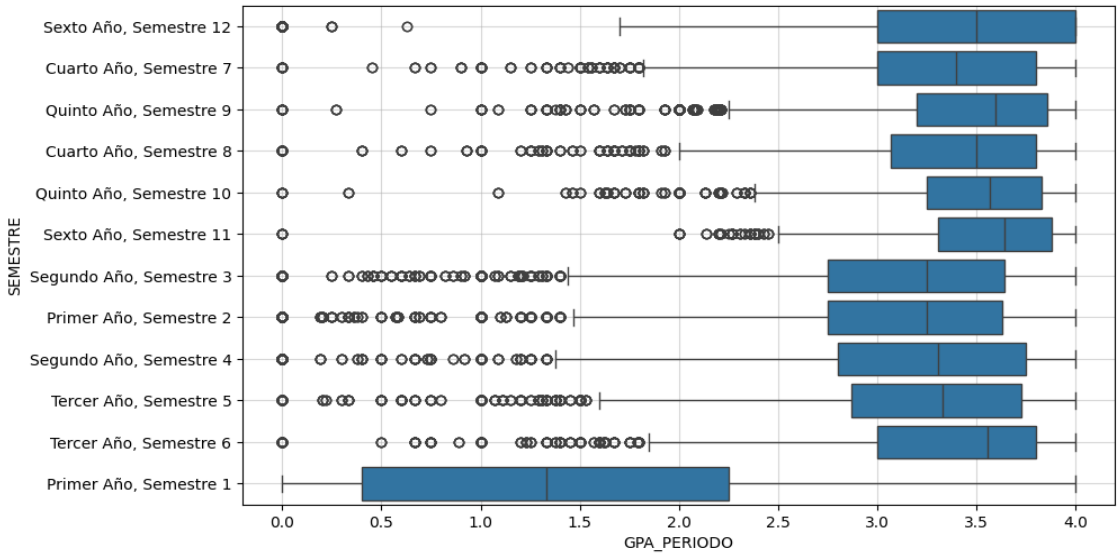
Gráfico 10. Créditos Registrados y el Estado Actual



El gráfico 10 muestra que la mediana de créditos registrados es más alta para los estudiantes que continúan en la universidad, con un rango más concentrado en torno a 12-15 créditos.

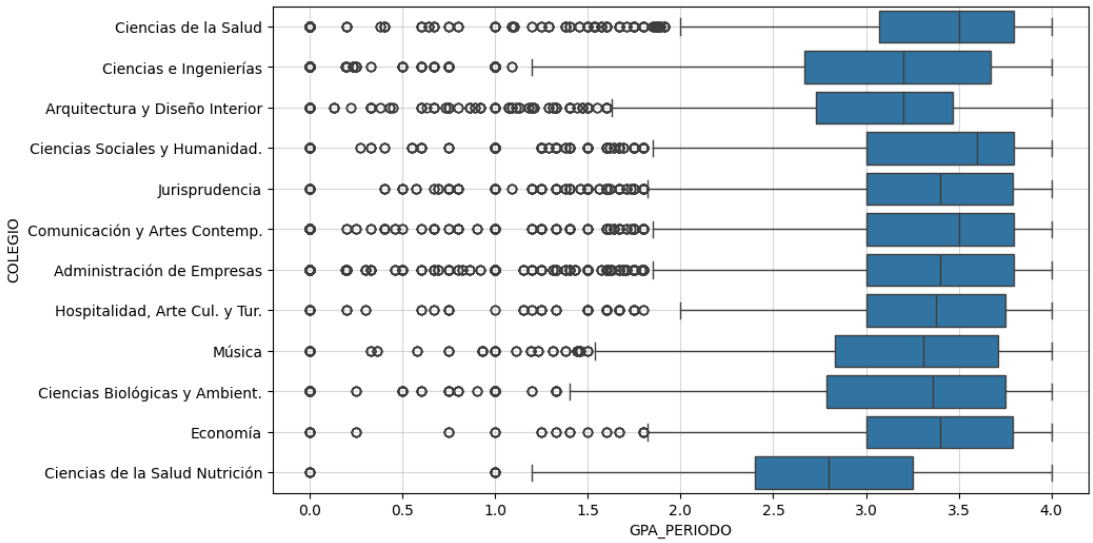
```
# Semestre y GPA_PERIODO
plt.figure(figsize=(10, 6))
sns.boxplot(data=df, x='GPA_PERIODO', y='SEMESTRE')
plt.grid(alpha=0.5)
```

Gráfico 11. Semestre y GPA por periodo



En el gráfico 11 se observa que existe una gran cantidad de valores atípicos en el primer semestre con GPA cercanos a 0, lo que indica que algunos estudiantes tienen dificultades desde el inicio.

Gráfico 12. Colegio y GPA por periodo



En el gráfico 12 se muestra que los colegios con mayor dispersión y presencia de valores atípicos son: Ciencias de la Salud, Ciencias e Ingenierías Arquitectura y Diseño Interior, Ciencias Sociales y Humanidades.

Estos colegios tienen una mayor cantidad de estudiantes con bajos GPAs, reflejado en los valores atípicos cercanos a 0.

3.3.4 Construcción de nuevos datos

```
# Subset de variables numéricas
df_numeric = df.select_dtypes(include=['float64', 'int64'])

# subset de variables categóricas
df_categorical = df.select_dtypes(include=['object'])
```

Dividimos en subset numérico para hacer una discretización de los valores continuos y poderlos relacionar con la variable categórica objetivo ESTADO_ACTUAL.

Tabla 3. Discretización de los valores continuos

	GENERO	EDAD	NOTA_COLEGIO	PRIMER_REGISTRO	ULTIMO PERIODO REGISTRO	CRED_REG	CRED_APROB	GPA_PERIODO	G
0	0	29.0	8.95	201510	202310	12	12	3.25	
1	0	29.0	8.95	201510	202310	12	12	3.25	
2	0	29.0	8.95	201510	202310	12	12	3.25	
3	0	29.0	8.95	201510	202310	12	12	3.25	
4	0	29.0	8.95	201510	202310	12	12	3.25	
...	...	...	...	...	...	...	...	...	
119976	1	26.0	8.03	202410	202420	15	15	3.20	
119977	1	26.0	8.03	202410	202420	15	15	3.20	
119978	1	26.0	8.03	202410	202420	15	15	3.20	
119979	1	26.0	8.03	202410	202420	15	15	3.20	
119980	1	26.0	8.03	202410	202420	15	15	3.20	

119099 rows × 12 columns

Dividimos en subset de las variables categóricas para hacer analizar las interacciones entre variables categóricas y variables numéricas para comprender mejor las relaciones en los datos.

Tabla 4. Variables categóricas en subset

```
df_numeric.corr()
```

	EDAD	NOTA_COLEGIO	PRIMER_REGISTRO	ULTIMO PERIODO REGISTRO	CRED_REG	CRED_APROB	GPA_PERIODO
EDAD	1.000000	-0.342652	-0.759236	-0.285178	-0.147000	-0.130118	-0.001650
NOTA_COLEGIO	-0.342652	1.000000	0.293899	0.142457	0.176519	0.244918	0.333949
PRIMER_REGISTRO	-0.759236	0.293899	1.000000	0.366578	0.140796	0.107222	-0.046379
'IMO PERIODO REGISTRO	-0.285178	0.142457	0.366578	1.000000	0.158907	0.183858	0.068303
CRED_REG	-0.147000	0.176519	0.140796	0.158907	1.000000	0.738119	0.216524
CRED_APROB	-0.130118	0.244918	0.107222	0.183858	0.738119	1.000000	0.607206
GPA_PERIODO	-0.001650	0.333949	-0.046379	0.068303	0.216524	0.607206	1.000000
GPA_ACUMULADO	0.028219	0.433875	-0.091708	0.076290	0.229020	0.519259	0.813876
PERIODO	-0.203538	0.048420	0.264955	0.251215	-0.011327	-0.026465	-0.025102
REG_TITULACION	0.421626	-0.142089	-0.577843	-0.399894	-0.060567	-0.001555	0.124851
._CREDITOS_APROBADOS	0.656729	-0.127106	-0.896709	-0.259950	0.021563	0.092885	0.226867

```
df_numeric.columns
```

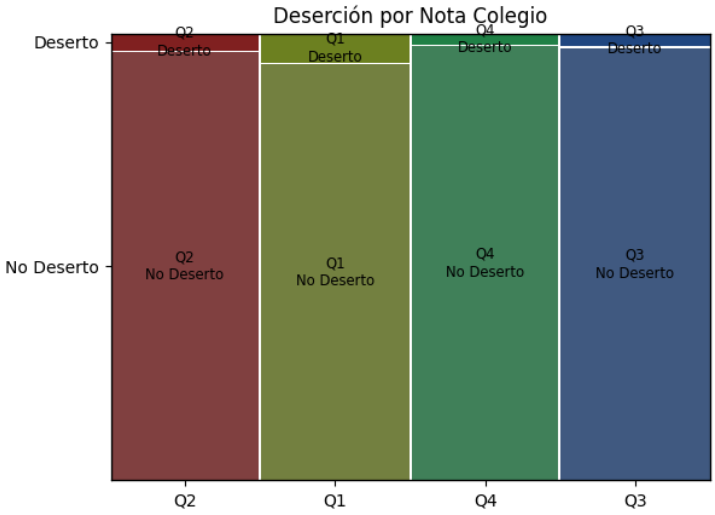
```
Index(['EDAD', 'NOTA_COLEGIO', 'PRIMER_REGISTRO', 'ULTIMO PERIODO REGISTRO',
       'CRED_REG', 'CRED_APROB', 'GPA_PERIODO', 'GPA_ACUMULADO', 'PERIODO',
       'REG_TITULACION', 'TOTAL_CREDITOS_APROBADOS'],
      dtype='object')
```

Se verifica las variables numéricas relevantes para formar cuartiles y observar cómo se encuentran distribuidos los datos.

```
# Cuartiles
#'EDAD', 'NOTA_COLEGIO', 'PRIMER_REGISTRO', 'ULTIMO PERIODO
REGISTRO','CRED_REG', 'CRED_APROB', 'GPA_PERIODO',
'GPA_ACUMULADO', 'PERIODO',REG_TITULACION',
'TOTAL_CREDITOS_APROBADOS'

df_numeric['EDAD_Cuartil'] = pd.qcut(df_numeric['EDAD'], 4,
labels=['Q1', 'Q2', 'Q3', 'Q4'])
df_numeric['NOTA_COLEGIO_Cuartil'] =
pd.qcut(df_numeric['NOTA_COLEGIO'], 4, labels=['Q1', 'Q2', 'Q3',
'Q4'])
df_numeric['PRIMER_REGISTRO_Cuartil'] =
pd.qcut(df_numeric['PRIMER_REGISTRO'], 4, labels=['Q1', 'Q2',
'Q3', 'Q4'])
#df_numeric['ULTIMO_PERIODO_REGISTRO_Cuartil'] =
pd.qcut(df_numeric['ULTIMO PERIODO REGISTRO'], 4, labels=['Q1',
'Q2', 'Q3', 'Q4'])
df_numeric['CRED_REG_Cuartil'] = pd.qcut(df_numeric['CRED_REG'],
4, labels=['Q1', 'Q2', 'Q3', 'Q4'])
#df_numeric['CRED_APROB_Cuartil'] =
pd.qcut(df_numeric['CRED_APROB'], 4, labels=['Q1', 'Q2', 'Q3',
'Q4'])
df_numeric['GPA_PERIODO_Cuartil'] =
pd.qcut(df_numeric['GPA_PERIODO'], 4, labels=['Q1', 'Q2', 'Q3',
'Q4'])
df_numeric['GPA_ACUMULADO_Cuartil'] =
pd.qcut(df_numeric['GPA_ACUMULADO'], 4, labels=['Q1', 'Q2',
'Q3', 'Q4'])
#df_numeric['PERIODO_Cuartil'] = pd.qcut(df_numeric['PERIODO'],
4, labels=['Q1', 'Q2', 'Q3', 'Q4'])
#df_numeric['REG_TITULACION_Cuartil'] =
pd.qcut(df_numeric['REG_TITULACION'], 4, labels=['Q1', 'Q2',
'Q3', 'Q4'])
df_numeric['TOTAL_CREDITOS_APROBADOS_Cuartil'] =
pd.qcut(df_numeric['TOTAL_CREDITOS_APROBADOS'], 4, labels=['Q1',
'Q2', 'Q3', 'Q4'])
```

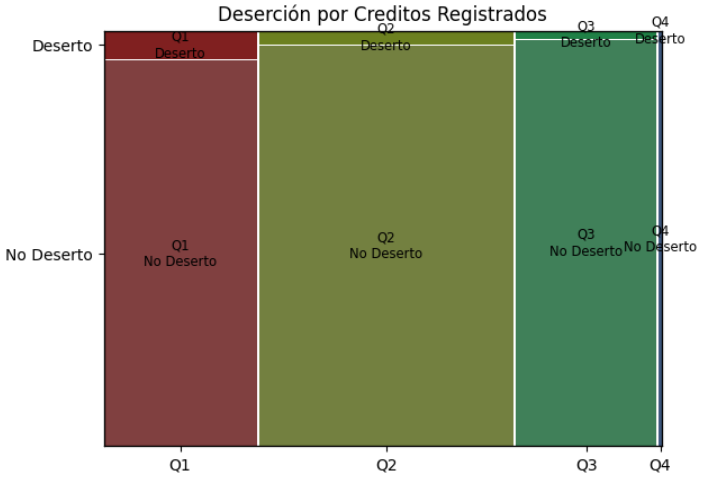
```
from statsmodels.graphics.mosaicplot import mosaic
plt.figure(figsize=(8,6))
mosaic(df_numeric, ['EDAD_Cuartil', 'ESTADO_ACTUAL'],
title='Deserción por Edad')
```

Gráfico 13. Deserción por nota de colegio



El gráfico 13 muestra que los estudiantes con notas más bajas en el colegio (cuartiles Q1 y Q2) tienen una mayor probabilidad de deserción universitaria. Específicamente, el cuartil más bajo (Q1) presenta la mayor proporción de deserción, lo que indica una fuerte correlación negativa entre las calificaciones del colegio de la secundaria y la retención universitaria.

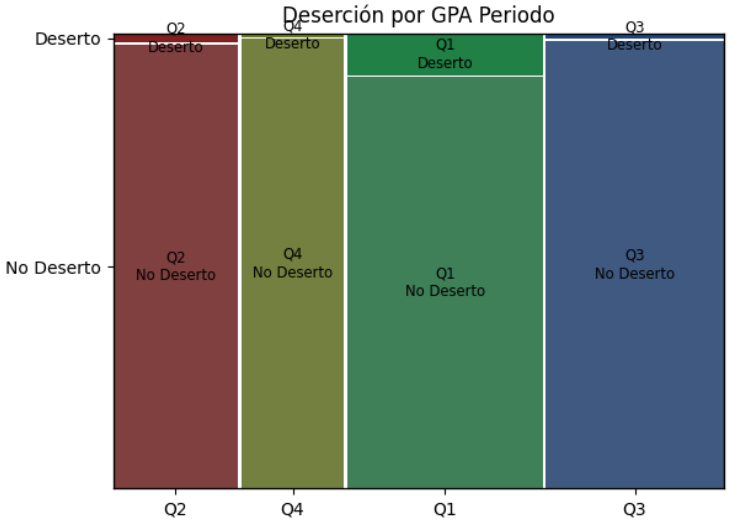
Gráfico 14. Deserción por créditos registrados



En el gráfico 14 los estudiantes que registran menos créditos (Q1) tienen la mayor tasa de deserción.

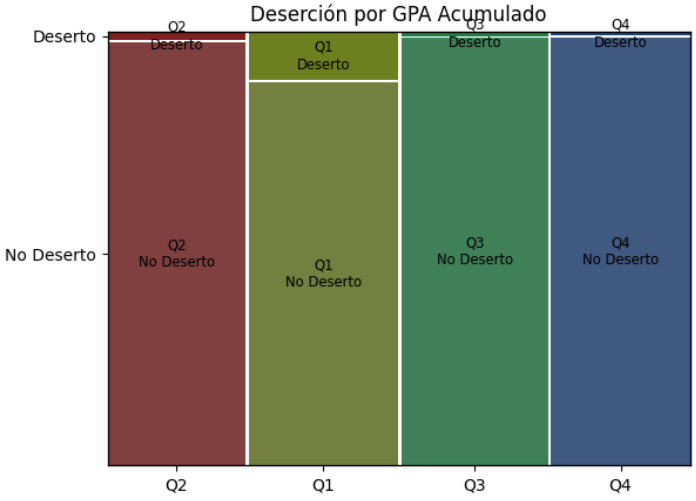
Se observa una franja significativa de deserción en Q1, lo que indica que los estudiantes que inscriben pocas materias tienen una mayor probabilidad de abandonar la universidad.

Gráfico 15. Deserción por GPA periodo



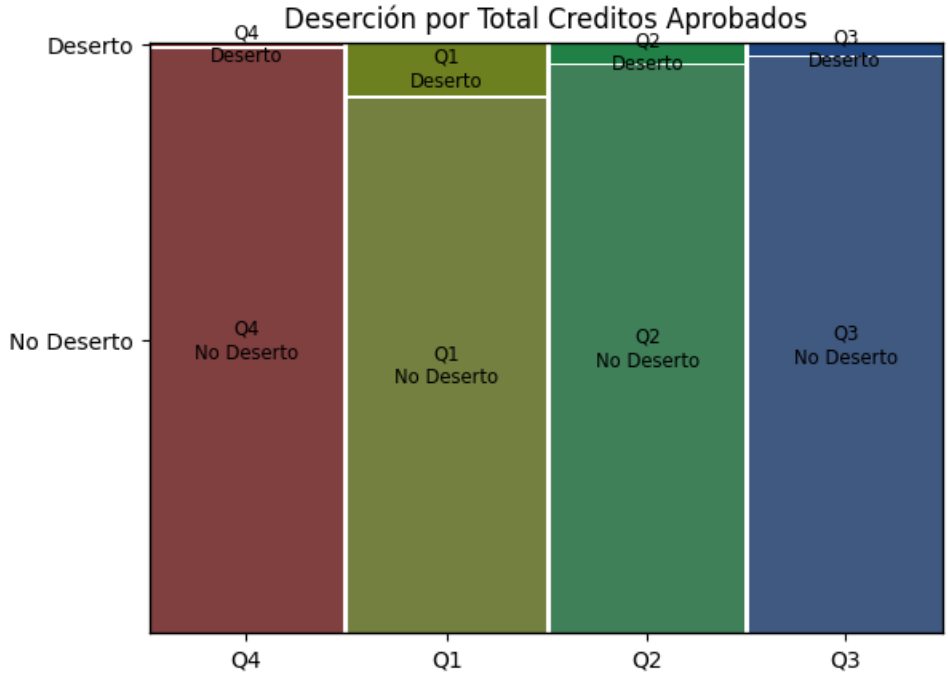
El gráfico 15 indica que existe una relación entre el GPA de los estudiantes en el periodo analizado y la probabilidad de deserción. Los estudiantes con un GPA más bajo (cuartiles Q1 y Q2) tienen una mayor probabilidad de desertar.

Gráfico 16. Deserción por GPA Acumulado



El gráfico 16 indica que existe una relación entre el GPA acumulado de los estudiantes y la probabilidad de deserción. Los estudiantes con un GPA acumulado más bajo (cuartiles Q1 y Q2) tienen una mayor probabilidad de desertar.

Gráfico 17. Deserción por total de créditos aprobados



3.3.5 Integración de datos

Una vez trabajadas todas las variables a través del preprocesamiento, se seleccionan los características con las que se va generar el set de datos para entrenar los modelos.

```
# DF FINAL PARA EL MODELO

df_final = pd.concat([df_numeric, df_categorical], axis=1)
df_final
```

Tabla 5. Dataset final para el modelo

	EDAD	NOTA_COLEGIO	PRIMER_REGISTRO	ULTIMO PERIODO REGISTRO	CRED_REG	CRED_APROB	GPA_PERIODO	GPA_ACUMULADO	I
0	29.0	8.95	201510	202310	12	12	3.25	3.2685	
1	29.0	8.95	201510	202310	12	12	3.25	3.2685	
2	29.0	8.95	201510	202310	12	12	3.25	3.2685	
3	29.0	8.95	201510	202310	12	12	3.25	3.2685	
4	29.0	8.95	201510	202310	12	12	3.25	3.2685	

Se verifica los valores únicos para las variables categóricas.

```
# Lista de columnas categóricas
categorical_columns = ['CODIGO_GENERO', 'ESTADO CIVIL', 'CODIGO
DISCAPACIDAD', 'ETNIA','TIPO ESTUDIANTE', 'NOMBRE COLEGIO',
'CIUDAD', 'NIVEL', 'COD COLEGIO','COLEGIO', 'CARRERA',
'SEMESTRE', 'CODIGO_ESTADO', 'ESTADO_ACAD','ESTADO_ACTUAL',
'NOMBRE CURSO', 'NOTA CURSO']

# Imprimir los valores únicos de cada columna categórica
for col in categorical_columns:
    print(f"Valores únicos en
'{col}':\n{df_categorical[col].unique()}\n"
```

Figura 11. Distribución de Valores Únicos en Variables Categóricas de un Conjunto de Datos

```
['F' 'M']

Valores únicos en 'ESTADO CIVIL':
['S' 'D' 'NINGUNO' 'M' 'U' 'P' 'W' 'N']

Valores únicos en 'CODIGO DISCAPACIDAD':
['NINGUNO' 'PA' 'LC' 'D2' 'TS' 'TI' 'DO' 'MS' 'EP' 'DZ' 'CA' 'AO' 'DE'
 'EG' 'ER' 'EF' 'AG' 'D5' 'DF' 'TO' 'BI' 'EZ' 'MG' 'SH' 'DT' 'DI' 'TL'
 'HF' 'TB' 'AX' 'AL' 'GM' 'EI' 'DP' 'EA' 'ED' 'LU' 'D6' 'NC' 'MR' 'TT'
 'CN' 'TX' 'DB' 'CI' 'QA' 'AP' 'D3' 'D1' 'ES' 'SP' 'HA' 'GA' 'LE' 'EC'
 'CB' 'DD' 'AC' 'EN' 'DC' 'VH' 'DS' 'SC' 'SU' 'DG' 'IL' 'EM' 'PS' 'LS'
 'SD' 'TZ' 'DY' 'TF' 'SA' 'SO' 'TW' 'AB' 'EQ' 'D4' 'AA' 'DR']

Valores únicos en 'ETNIA':
['INDÍGENA' 'MESTIZO/A' 'AFROECUATORIANO/A' 'MONTUBIO/A' 'BLANCO/A'
 'NEGRO/A' 'MULATO/A' 'OTRO' 'NO REGISTRA']

Valores únicos en 'TIPO ESTUDIANTE':
['N']

Valores únicos en 'NOMBRE COLEGIO':
['Republica Del Ecuador' 'Saint Patrick' 'Nacional Cumbaya'
 'U.E. Int.Pensionado Atahualpa' 'U.E. Sagrados Corazones De Rum'
 'Computer World' 'John F Kennedy' 'Letort' 'Juan De Velasco'
...

Valores únicos en 'NOTA CURSO':
['B' 'A' 'D' 'C' 'F' 'P' 'N' 'W' 'I' 'H']
```

```
# Se elimina las columnas que no se van a utilizar NOMBRE CURSO, NOMBRE COLEGIO.
df_categorical.drop(columns=['NOMBRE CURSO', 'NOMBRE COLEGIO'], inplace=True)
```

```
df_categorical.drop(columns=['NIVEL'], inplace=True)

df_categorical.drop(columns=['TIPO ESTUDIANTE'], inplace=True)
```

```
df_categorical
```

Tabla 6. Variables y su relación

	CODIGO_GENERO	ESTADO CIVIL	CODIGO DISCAPACIDAD	ETNIA	CIUDAD	COD COLEGIO	COLEGIO	CARRERA	SEMEST
0	F	S	NINGUNO	INDÍGENA	Otavalo	CS	Ciencias de la Salud	ODONTOLOGÍA	Sexto A Semes
1	F	S	NINGUNO	INDÍGENA	Otavalo	CS	Ciencias de la Salud	ODONTOLOGÍA	Sexto A Semes
2	F	S	NINGUNO	INDÍGENA	Otavalo	CS	Ciencias de la Salud	ODONTOLOGÍA	Sexto A Semes
3	F	S	NINGUNO	INDÍGENA	Otavalo	CS	Ciencias de la Salud	ODONTOLOGÍA	Sexto A Semes
4	F	S	NINGUNO	INDÍGENA	Otavalo	CS	Ciencias de la Salud	ODONTOLOGÍA	Sexto A Semes
...	...	...	...	...	...	...	...	...	
119976	M	NINGUNO	NINGUNO	MESTIZO/A	Guayaquil	CC	Comunicación y Artes Contemp.	PUBLICIDAD	Ter A Semestr
119977	M	NINGUNO	NINGUNO	MESTIZO/A	Guayaquil	CC	Comunicación y Artes Contemp.	PUBLICIDAD	Ter A Semestr

```
df_numeric.drop(columns=['EDAD',
'NOTA_COLEGIO','PRIMER_REGISTRO', 'ULTIMO PERIODO REGISTRO',
'CRED_REG', 'CRED_APROB', 'GPA_PERIODO', 'GPA_ACUMULADO',
'PERIODO','REG_TITULACION', 'TOTAL_CREDITOS_APROBADOS'],
inplace=True)
```

```
df_final = pd.concat([df_numeric, df_categorical], axis=1)
df_final
```

Tabla 7. Data Set Final con Todas las Variables

CIUDAD	COD COLEGIO	COLEGIO	CARRERA	SEMESTRE	CODIGO_ESTADO	ESTADO_ACAD	ESTADO_ACTUAL	NOTA CURSO
Otavalo	CS	Ciencias de la Salud	ODONTOLOGÍA	Sexto Año, Semestre 12	00	Estado Acad. Regular	No Deserto	B
Otavalo	CS	Ciencias de la Salud	ODONTOLOGÍA	Sexto Año, Semestre 12	00	Estado Acad. Regular	No Deserto	A
Otavalo	CS	Ciencias de la Salud	ODONTOLOGÍA	Sexto Año, Semestre 12	00	Estado Acad. Regular	No Deserto	B
Otavalo	CS	Ciencias de la Salud	ODONTOLOGÍA	Sexto Año, Semestre 12	00	Estado Acad. Regular	No Deserto	B
Otavalo	CS	Ciencias de la Salud	ODONTOLOGÍA	Sexto Año, Semestre 12	00	Estado Acad. Regular	No Deserto	B
...	...	...	...	...	...	...	...	...
Guayaquil	CC	Comunicación y Artes Contemp.	PUBLICIDAD	Tercer Año, Semestre 5	00	Estado Acad. Regular	No Deserto	A

```
df_final["ESTADO_ACTUAL"].value_counts()
```

```
ESTADO_ACTUAL
1    114383
0      4716
Name: count, dtype: int64
```

```
# Separar en características (X) y variable objetivo (y)
X = df_final.drop(columns=["ESTADO_ACTUAL"]) # TODAS LAS COLUMNAS MENOS LA VARIABLE DEPENDIENTE (ESTADO ACTUAL)
y = df_final["ESTADO_ACTUAL"] # SOLO LA VARIABLE A PREDECIR LA VARIABLE DEPENDIENTE
```

Se separa el data set original en dos partes: una que contiene las características que se utilizarán para entrenar el modelo (X) y otra que contiene la variable que se quiere predecir (y). Esta separación es un paso esencial en el proceso de aprendizaje automático, ya que permite al modelo aprender la relación entre las características y la variable objetivo.

```
# Imprimir los nombres de las características utilizadas en el modelo.
```

```
feature_names = X.columns.tolist()
feature_names
```

```
['EDAD_Cuartil',
 'NOTA_COLEGIO_Cuartil',
 'PRIMER_REGISTRO_Cuartil',
 'CRED_REG_Cuartil',
 'GPA_PERIODO_Cuartil',
 'GPA_ACUMULADO_Cuartil',
 'TOTAL_CREDITOS_APROBADOS_Cuartil',
 'CODIGO_GENERO',
 'ESTADO CIVIL',
 'CODIGO DISCAPACIDAD',
 'ETNIA',
 'CIUDAD',
 'COD COLEGIO',
 'COLEGIO',
 'CARRERA',
 'SEMESTRE',
 'CODIGO_ESTADO',
 'ESTADO_ACAD',
 'NOTA CURSO']
```

3.4 Modelado

3.4.1 Selección de técnicas de modelado

En esta tercera fase se realiza la implementación de los algoritmos.

3.4.2 Generación de los Modelos

Se implementa 3 algoritmos:

AdaBoost implementado con Phyton, XGBoost y Random Forest en Canvas.

3.4.3 División del conjunto de datos

```
from sklearn.model_selection import train_test_split
# Dividir en conjunto de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=42, stratify=y) # estratify es
porque el data set está desbalanceado.
```

Se estableció que el 30% de los datos se utilizarán para la prueba del modelo y el 70% restante para el entrenamiento. Se eligió este porcentaje porque permite evaluar de manera

efectiva la capacidad predictiva sin comprometer la cantidad de datos disponibles para el entrenamiento.

En cuanto al random_state=42 se utilizó un valor fijo para asegurar que la división de los datos sea reproducible en cada ejecución del código.

Se aplicó estratificación en la variable objetivo, debido a que el dataset está desbalanceado esto permite garantizar que la proporción de clases se mantenga tanto en el conjunto de entrenamiento como en el de prueba, evitando sesgos en el modelo.

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.model_selection import cross_val_score
from sklearn.metrics import make_scorer, roc_auc_score


# Inicializar y entrenar el modelo AdaBoost
base_estimator = DecisionTreeClassifier(max_depth=7,
class_weight='balanced')
model = AdaBoostClassifier(estimator=base_estimator,
n_estimators=200, random_state=42)
model.fit(X_train, y_train)
```

Se crea un modelo AdaBoost para clasificación. Este especifica que el árbol de decisión creado previamente será el estimador base utilizado por AdaBoost (estimator=base_estimator).

Se indica que AdaBoost entrenará 200 árboles de decisión en secuencia (n_estimators=200), además se establece una semilla para la generación de números aleatorios (random_state=42) para asegurar que los resultados sean reproducibles.

Por otro lado como el dataset mantiene problemas de desbalanceo es decir, hay más ejemplos de una clase que de otra, se ha utilizado class_weight='balanced' en el árbol de decisión base para ajustar los pesos de las clases automáticamente, dándole mayor importancia a la clase minoritaria.

3.4.4 Ajuste de Pesos para la Clase 0 (Balanceo de Datos)

```
from imblearn.over_sampling import SMOTE
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import AdaBoostClassifier
import numpy as np

# Se aplica SMOTE para balancear las clases.
smote = SMOTE(sampling_strategy='auto', random_state=42)
X_train_bal, y_train_bal = smote.fit_resample(X_train, y_train)

# Definir pesos manualmente: darle más peso a la clase 0
(desertores)
class_weights = {0: 5, 1: 1}  # Se da 5 veces más peso a la
clase 0
sample_weights = np.array([class_weights[y] for y in
y_train_bal])  # Asigno pesos a cada muestra

# Inicializar y entrenar el modelo AdaBoost con datos
balanceados
base_estimator = DecisionTreeClassifier(max_depth=7,
class_weight='balanced')
model = AdaBoostClassifier(estimator=base_estimator,
n_estimators=200, random_state=42)

# Entrenar el modelo con los pesos ajustados
model.fit(X_train_bal, y_train_bal,
sample_weight=sample_weights)
```

3.4.5 Aplicación de SMOTE

Se genera muestras sintéticas de la clase minoritaria para dar más peso a la clase 0 (desertores) en el entrenamiento de AdaBoost, Esto hace que el modelo preste más atención a la clase minoritaria durante el entrenamiento.

Por tal razón se crea una instancia de SMOTE con sampling_strategy='auto' (que ajusta automáticamente la estrategia de sobremuestreo) y random_state=42 (para reproducibilidad).

Se aplicó fit_resample a los datos de entrenamiento X_train e y_train para generar un conjunto de datos balanceado X_train_bal e y_train_bal.

3. 4.6 Definición de Pesos Manuales

En este caso, se da 5 veces más peso a la clase 0 (desertores) que a la clase 1,

la razón es porque como se está usando SMOTE para generar muestras sintéticas de la clase minoritaria. Esto ya ayuda a equilibrar las clases, por lo que no es necesario un peso extremadamente alto.

3.5 Evaluación

En esta quinta fase se evaluá los modelos de clasificación.

3.5.1 Selección de Métricas

Dado que el objetivo de este trabajo es desarrollar y comparar tres modelos de clasificación que clasifique correctamente a los estudiantes en riesgo de deserción, es fundamental que el modelo logre detectar con precisión los casos positivos.

Por esta razón, las métricas clave a utilizar en los tres modelos es , Recall (Sensibilidad) y Especificidad por el alto desbalance de los datos.

3.5.2 Análisis de Resultados

Tabla 9. Algoritmos de evaluación de modelos de clasificación

Modelo	Accuracy	Recall (Clase 0)	F1-Score	ROC AUC
Adaptive Boosting	97.00	97.00	71.00	99.00
XGBoost	99.59	97.50	96.00	100.00
Random Forest	97.83	96.45	77.65	99.40

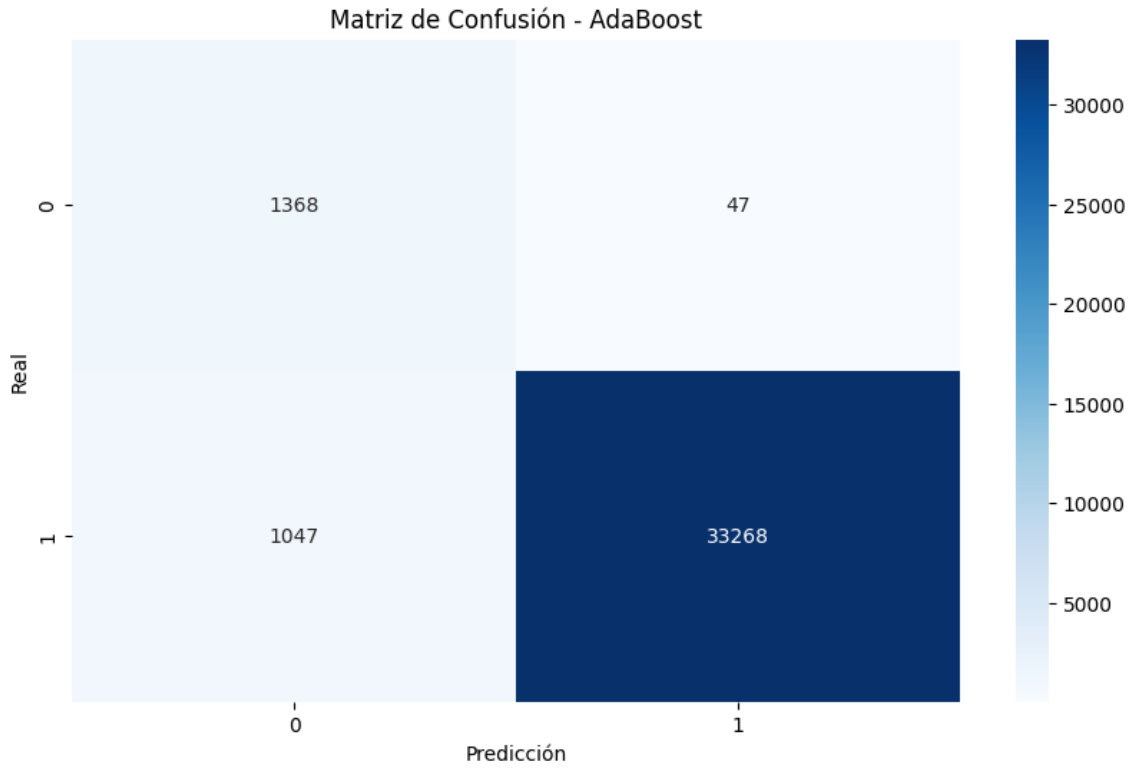
La tabla 9 presenta los resultados de evaluación de los tres modelos de clasificación.

3.5.3 Resultados de Adaptive Boosting

```
Classification Report:
              precision    recall  f1-score   support

           0       0.57      0.97      0.71      1415
           1       1.00      0.97      0.98     34315

    accuracy                           0.97     35730
   macro avg       0.78      0.97      0.85     35730
weighted avg       0.98      0.97      0.97     35730
```

3.5.4 Matriz de Confusión

Gráfico 18. Matriz de Confusión



Se realiza un análisis sobre la matriz de confusión y se interpreta lo siguiente:

Verdaderos Negativos 1368**,** esto representa a los estudiantes que desertaron en la realidad, y que el modelo los identificó correctamente como desertores.

Falsos Positivos 47, son estudiantes que en realidad no desertaron, pero que el modelo erróneamente clasificó como desertores. Este número es relativamente bajo, indicando que el modelo tiene pocas alarmas falsas.

Falsos Negativos 1047, son estudiantes que desertaron en la realidad, pero que el modelo clasificó erróneamente como no desertores.

Verdaderos Positivos 33268, corresponden a estudiantes que no desertaron y que el modelo identificó correctamente como no desertores.

3.5.5 Curva ROC

```
ROC-AUC Score: 0.9976532015534728
```

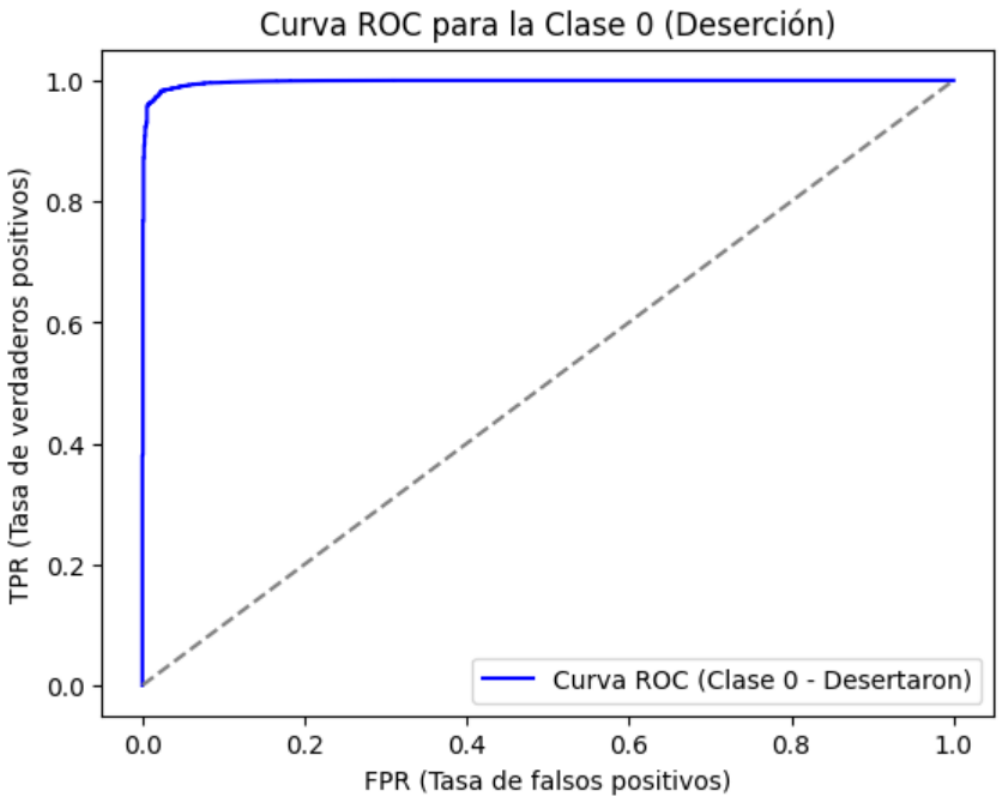
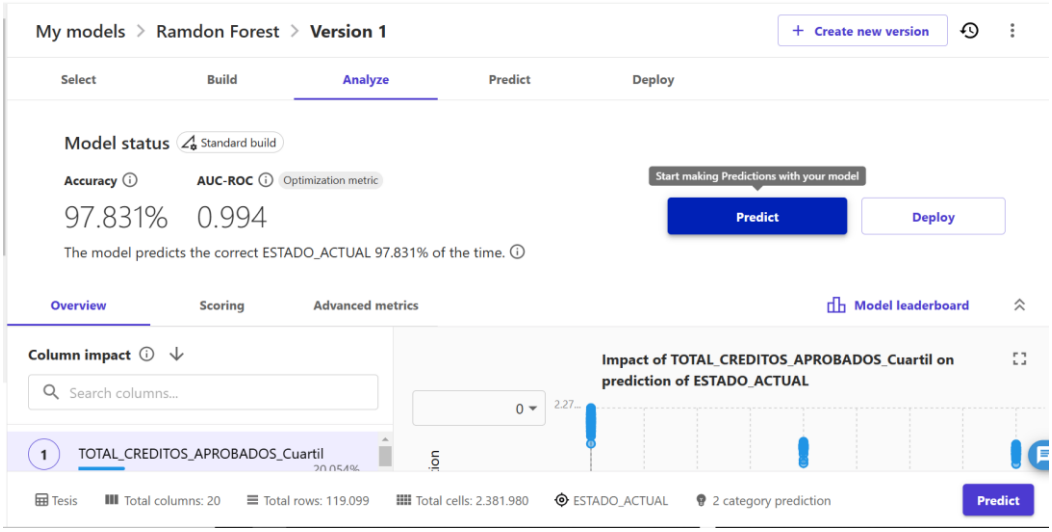


Figura 13. Curva Roc

3.5.6 Resultados de Random Forest

Figura 12. Algoritmo Random Forest

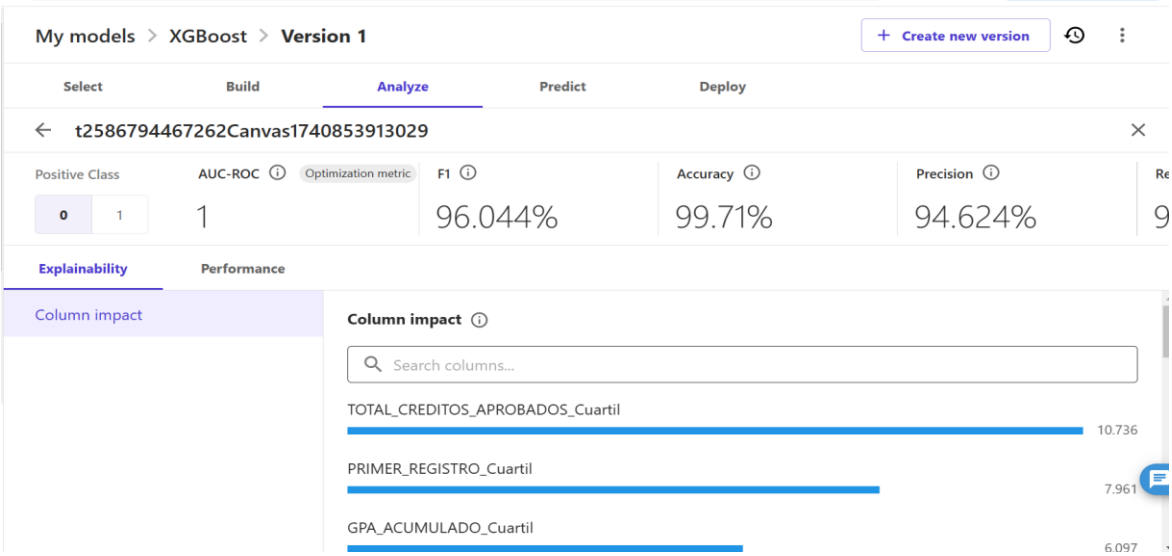


Para el segundo modelo se desarrolló un modelo de clasificación utilizando el algoritmo Random Forest en AWS Canvas para predecir la variable ESTADO_ACTUAL.

El modelo alcanzó una precisión del 97.831% y un AUC-ROC de 0.994, lo que indica un excelente rendimiento. La variable más influyente en la predicción fue TOTAL_CREDITOS_APROBADOS_Cuartil, con un impacto del 20.054%. Estos resultados indican que el modelo es capaz de predecir el ESTADO_ACTUAL con alta precisión, lo que podría tener implicaciones importantes para la Universidad.

3.5.7 Resultados de XGBoost

Figura 13. Algoritmo XGBoost



Del mismo modo, se desarrolló un modelo de clasificación utilizando el algoritmo XGBoost en AWS Canvas para predecir la variable ESTADO_ACTUAL. El modelo alcanzó una precisión del 97.831% y un AUC-ROC de 0.994, lo que indica un excelente rendimiento. La variable más influyente en la predicción fue TOTAL_CREDITOS_APROBADOS_Cuartil, con un impacto del 20.054%. Estos resultados indican que el modelo es capaz de predecir el ESTADO_ACTUAL con alta precisión, lo que podría tener implicaciones importantes para la Universidad. Es importante destacar que para el análisis de los datos, se utilizó el parámetro stratify=y durante la división del conjunto de datos en entrenamiento y prueba. Esto aseguró que la proporción de clases en ambos conjuntos fuera representativa del conjunto de datos original, lo cual es crucial para obtener resultados confiables en problemas de clasificación con datos desbalanceados.

Con estos resultados, se observa que el modelo XGBoost supera al modelo Random Forest en todas las métricas evaluadas, lo que lo convierte en el modelo con mejor rendimiento para este conjunto de datos.

CAPITULO VI. CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

Se utilizó la metodología CRISP-DM, aplicando técnicas de minería de datos para analizar la proporción de estudiantes según su estado académico. Se contrastó empíricamente la hipótesis planteada por la universidad, confirmando que la mayor tasa de deserción ocurre durante el primer año.

Se implementaron métodos de aprendizaje automático, desarrollando modelos capaces de predecir la deserción estudiantil con el objetivo de mejorar la retención y optimizar la toma de decisiones institucionales.

Se evaluaron diferentes modelos de aprendizaje automático, incluyendo AdaBoost, Random Forest y XGBoost. Tras analizar su desempeño, se evidenció que XGBoost fue el modelo más efectivo para esta investigación debido a su capacidad para manejar datos desbalanceados, su optimización en el uso de memoria y su capacidad para evitar el sobreajuste mediante técnicas de regularización. Además, XGBoost presentó una mayor precisión y recall en la clasificación, lo que lo convirtió en la mejor opción para este problema.

6.2 RECOMENDACIONES

Se evidenció los problemas académicos de los estudiantes dentro de la institución, por consiguiente, la universidad debe tomar medidas de seguimiento de los educandos con la

finalidad de proponerles acciones de mejora a su situación, esto puede ser tutorías desde el primer semestre.

Se debe seguir generando y probando modelos que ayuden a predecir la deserción estudiantil pero con una mayor información de datos, para que estos puedan servir en proyectos futuros, facilitando a las autoridades una mejor toma de decisiones.

Con base en los resultados obtenidos, se recomienda a la universidad proceder con la fase de despliegue del modelo XGBoost, el cual demostró ser el más efectivo en la predicción de la deserción estudiantil. Se propone implementar este modelo en la plataforma AWS, debido a su escalabilidad, flexibilidad y las ventajas que ofrece para el despliegue y gestión de modelos de machine learning.

Es importante destacar que AWS es una plataforma de pago, por lo que la universidad deberá realizar una inversión económica. No obstante, esta inversión se justifica por el potencial del modelo para proporcionar información valiosa y en tiempo real, lo que permitirá a la universidad tomar decisiones estratégicas. Por ejemplo, el modelo podría identificar qué carreras y semestres presentan mayores tasas de deserción, permitiendo a la universidad enfocar sus recursos y estrategias de retención de manera más efectiva. Al tener acceso a esta información, la institución podrá optimizar sus recursos y mejorar la retención estudiantil.

Bibliografía y referencias.

1. *Lugo, B. (01 de enero de 2013). La deserción estudiantil: ¿Realmente es un problema social? Rev. Postgrado FACE-UC, 7(12). Retrieved 09 de febrero de 2024, from* http://www.arje.bc.uc.edu.ve/arj12/art17.pdf

2. *Pacho, F., & Chiqui, D. (2011). Estudio de las causas de la deserción escolar. Tesis Licenciatura, Universidad de Cuenca, Departamento de Filosofía, letras y ciencias de la Educación, Ecuador.*

3. *Seminarqa, M., & Aparicio, M. (15 de junio de 2018). La deserción universitaria ¿un concepto equívoco? Revisión de estudios latinoamericanos sobre conceptos alternativos. Rev. Orientación Educacional, 32(61), 44-72. Retrieved 09 de febrero de 2024.*

4. Baesens, B., Vlasselaer, V. V., & Verbeke, W. (2015). *Fraud analytics using descriptive, predictive, and social network techniques: A guide to data science for fraud detection*. John Wiley & *Sons.*

5. *Awoyemi, J. O., Adetunmbi, A. O., & Oluwadare, S. A. (2017). Credit card fraud detection using machine learning techniques: A comparative analysis. 2017 International Conference on Computing Networking and Informatics (ICCNI).* https://doi.org/10.1109/iccni.2017.8123782

6. *Alcalá Castro, A. (2019). Validación de un instrumento para la identificación de las determinantes causales de la deserción estudiantil . Universidad del bosque, 1-64.*

7. *Effer Apaz (Effer Apaza, Factores determinantes que inciden en la deserción de los estudiantes universitarios, 2012)a* https://dialnet.unirioja.es/servlet/articulo?codigo=4031617

8. *Asale, R.-., & Rae. (s. f.). fraude | Diccionario de la lengua española. «Diccionario de la Lengua Española» - Edición del Tricentenario.* https://dle.rae.es/fraude

9. *Viale, H. (2014). Una aproximación teórica de la deserción estudiantil universiratia. Revista Digital Ridu, 1-18.*

10. *Daniel Álvarez Gil 14 enero, 2021 Adictos al Trabajo* https://adictosaltrabajo.com/2021/01/14/metodologia-crisp-dm/

11. *¿Qué es el machine learning (ML)? | IBM. (s. f.).* https://www.ibm.com/es-es/topics/machine-learning

12. *DataScientest.com. (2023, 30 octubre). Machine Learning: definición, funcionamiento, usos. Formación En Ciencia de Datos | DataScientest.com.* https://datascientest.com/es/machine-learningdefinicionfuncionamientousos#:~:text=El%20Machine%20Learning%20o%20aprendizaje,%2C%20im%C3%A1genes%2C%20estad%C3%ADsticas%2C%20etc

13. *Metaphorce. (2022, 18 agosto). Descubre todo lo que necesitas saber sobre el Machine Learning.* https://www.linkedin.com/pulse/descubre-todo-lo-que-necesitas-saber-sobre-el-machine-learning-/

14. *Gonzalez, F. (s. f.). Machine Learning: La base que tenes que tener.* https://somospnt.com/blog/53-introduccion-a-machine-learning

15. *Gonzalez, J. L. (2020, 13 julio). Tipos de aprendizaje automático - SoldAI - Medium. Medium.* https://medium.com/soldai/tipos-de-aprendizaje-autom%C3%A1tico-6413e3c615e2

16. *Servicios web de Amazon. (sf).* AWS Canvas: Interfaz de preparación de datos y construcción de modelos *[Imagen]. Recuperado el 19 de marzo de 2025, de* https://aws.amazon.com/es/free/

17. *Nodd3r. (s.f.). ¿Por qué se utiliza Python en la ciencia de datos? Nodd3r. Recuperado de* https://nodd3r.com/blog/por-que-se-utiliza-python-en-la-ciencia-de-datos

18. *Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). CRISP-DM 1.0 step-by-step data mining guide.*