ISBN: 978-628-96613-3-0. ISSN: 2414-6390. Digital Object Identifier: https://dx.doi.org/10.18687/LACCEI2026.1.1.106

Comparativa de Modelos Boosting para Identificar Estudiantes en Modalidad Semipresencial en Educación Superior (2020–2024)

Moisés Evangelista Gamarra[1]; Jerremi Aron Chancan Labajos[1]; Seyda Sadith Rivas Hermitaño[1]; Carlos Adrian Versluys Magno Solsol[1]
[1]Servicio Nacional de Adiestramiento en Trabajo Industrial (SENATI), Lima, Perú, mevangelistag@senati.pe, 1520481@senati.pe, 1510258@senati.pe, 1509717@senati.pe

***Resumen*– El auge de los programas semipresenciales en las universidades de Perú ha traído consigo retos importantes al momento de planear como usar los recursos académicos en las instituciones. En este trabajo se hace una comparativa de cuatro modelos de tipo boosting: XGBoost, LightGBM, CatBoost y también el Gradient Boosting Machine; esto con el fin de clasificar si un alumno se matricula en modalidad presencial o semipresencial. Se uso un dataset de 87,008 registros entre los años 2020 y 2024. Para la parte técnica, se aplicó validación cruzada agrupando por estudiante (GroupKFold) y se separó el año 2024 como conjunto de evaluación externa. Como había un desbalance de clases muy fuerte (95/5), nos enfocamos en métricas como F1, AUC, MCC y PR-AUC, además de ajustar un umbral operativo para que el F1 sea máximo con un Recall de al menos 60%. Al final, el modelo de XGBoost fue el que mejor funcionó dando un F1 de 0.37 y un MCC de 0.41, logrando identificar 17 de los 26 casos reales semipresenciales. Los resultados muestran que estos modelos de boosting ganan por mucho a los baselines y que es clave estar recalibrando el umbral porque la distribución de los datos cambia con el tiempo.**

***Palabras clave*-- modalidad Académica, aprendizaje automático, desbalance de clases, educación superior peruana, fuga de datos.**

I. Introducción

La expansión de la educación semipresencial en las universidades peruanas a raíz de la pandemia del COVID-19 ha transformado de forma permanente la estructura de la oferta académica, lo que ha traído consigo retos para la planificación institucional en áreas como la gestión de docentes, la infraestructura tecnológica y el uso de espacios físicos [1]. En este panorama, el poder anticipar bajo qué modalidad se matriculará un estudiante se ha vuelto una necesidad operativa urgente para las áreas de gestión universitaria, que hoy en día suelen trabajar de forma reactiva por la falta de herramientas predictivas confiables. Desde el enfoque del aprendizaje automático, este problema se traduce en una clasificación binaria sobre datos tabulares con desafíos técnicos de gran calado; principalmente, existe un desbalance severo de clases donde la modalidad semipresencial apenas llega al 5% de los registros, lo cual tiende a sesgar a los modelos tradicionales que buscan optimizar la precisión global ignorando a la clase minoritaria. A esta dificultad se añade la dependencia temporal entre ciclos académicos y el riesgo crítico de fuga de información si no se maneja con rigor la selección de variables y el esquema de validación. Aunque los modelos de boosting han probado su superioridad en escenarios de datos tabulares desbalanceados [2] —siendo XGBoost, LightGBM, CatBoost y Gradient Boosting Machine los usados para este trabajo—, aún es escasa la literatura que compare estos algoritmos bajo validaciones temporales estrictas y con criterios de umbral pensados en el impacto institucional dentro del contexto latinoamericano. Por ello, este estudio aplica y contrasta estos cuatro modelos usando un dataset real de 87,008 matrículas de una universidad peruana (2020-2024), donde se implementó GroupKFold por estudiante para asegurar la independencia en la validación y se reservó el año 2024 como prueba externa. Además, el trabajo establece un umbral operativo basado en el costo de oportunidad entre la falta de identificación de alumnos y la sobreasignación de recursos, integrando finalmente un análisis de explicabilidad con valores SHAP y evaluaciones de equidad por subgrupos para garantizar que los resultados no solo sean precisos, sino también interpretables y éticamente aplicables en la toma de decisiones institucional.

II. Trabajos Relacionados

El análisis de la literatura muestra el uso del aprendizaje automático en la educación fue de pasar a ser algo experimental a ser una necesidad para la toma de decisiones.

En estos estudios actuales, se observa una tendencia clara hacia el uso de modelos estadísticos y de machine learning; por ejemplo, Adewale et al. [3] identificaron con su revisión de 64 investigaciones enfoques que tienen un valor del 29.69% en la educación a distancia y híbridos. Sin embargo, lo que más resalta es que, a pesar de este avance, todavía hay vacíos en cómo se aplican estos métodos en modelos de estudio híbridos.

En el ámbito regional, autores como Asencios-Trujillo et al. [4] han intentado mapear la adaptabilidad estudiantil en entornos online usando Random Forest para predecir niveles de adaptabilidad (‘Bajo’, ‘Moderado’, ‘Alto’). Sus resultados muestran que el modelo alcanzó una precisión del 91,29%, con valores de recall y f1-score sólidos, especialmente en las categorías ‘Bajo’ y ‘Moderado’, pero el hecho de su precisión elevada da a entender de un posible desbalance de clases. Este problema hace que los modelos no logren identificar correctamente a los grupos minoritarios lo cual es algo muy común en la minería de datos educativos (EDM) [5], lo que sugiere que la precisión por sí sola no es una métrica confiable en estos contextos educativos. De hecho, el desbalance de clases es el reto técnico que más se repite en los trabajos encontrados.

He y Garcia [6] ya advertían que con los algoritmos de machine learning tradicionales se vuelven "favorables" a la clase mayoritaria, arruinando el desempeño con los datos menos frecuentes viendo distintos algoritmos de machine learning entre ellos basados en boosting.

Para mitigar esto, investigadores como Chau y Phung [7] han propuesto mezclar técnicas de resampling con Random Forest, demostrando que es vital usar métricas adecuadas como Accuracy (Precisión) y Característica de funcionamiento del receptor (ROC) para verificar los resultados usando conjunto de datos por años.

Finalmente, la tendencia de esta investigaciones apunta por los modelos de boosting por su capacidad de captar patrones complejos en datos tabulares y manejo de desbalance de datos.

Lo que se extrae de esta revisión es que, aunque hay avances, falta una comparación que use validación temporal y análisis SHAP en el contexto peruano, que es justamente lo que se propone en este estudio.

III. Conjunto De Datos Y Preprocesamiento

Este estudio utilizó un conjunto de datos obtenido desde la plataforma nacional de datos abiertos del Perú, perteneciente a los registros de matrículas de la Universidad Nacional de Educación Enrique Guzmán y Valle (UNE) correspondientes a los años 2020–2024 [8], incluye un total de 87,008 registros y 19 variables académicas y demográficas [9].

A. Diccionario De Datos

A continuación en la Tabla I se observa únicamente las variables relevantes para el análisis con una breve descripción, esta información viene del diccionario de datos original [10].

Tabla I
Variables del conjunto de datos

Variable	Descripción breve
FECHA_CORTE	Fecha de generación del dataset
UUID	Código anonimizado del alumno
ANNIO	Año de matrícula
PERIODO	Periodo académico
NIVEL	Nivel de estudios
FACULTAD	Facultad del programa
PROGRAMA_ESTUDIOS	Programa de estudios
MODALIDAD*	Modalidad de estudio
TIPO_ESTUDIO	Tipo de estudio
DETALLE_TIPO_ESTUDIO	Detalle del tipo de estudio
PROMO	Promoción del alumno
SEDE	Sede de estudios
FECHA_MATRICULA	Fecha de matrícula
CANTCURMAT	Nº de cursos matriculados
CANTCREDMAT	Nº de créditos matriculados
CICLO	Ciclo académico
SECCION	Sección
SEXO	Sexo del alumno

* Modalidad es la variable objetivo a predecir

B. Desbalance De Clases

En la Fig. 1, presenta la distribución porcentual de la variable "Modalidad", categorizada como 0 para modalidad presencial (clase negativa) y 1 para modalidad semipresencial (clase positiva).

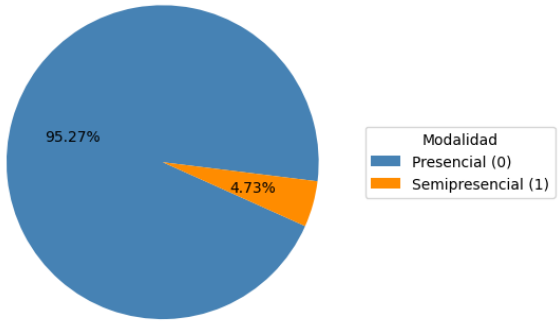

Fig 1. Distribución porcentual de la variable objetivo

Lo primero que se logra identificar en el análisis exploratorio de la variable objetivo “Modalidad”, es un desbalance de clases.

De los 87,008 registros, 82,889 (95.27%) corresponden a la modalidad presencial, mientras que con solo 4,119 (4.73%) pertenecen a la modalidad semipresencial.

Este desbalance representa un reto importante para los algoritmos de clasificación, ya que una distribución tan sesgada puede inducir a los modelos a favorecer la clase mayoritaria, obteniendo métricas aparentemente altas [6], sin verdaderamente captar los patrones de la clase minoritaria, que en este caso es la modalidad semipresencial.

C. Valores Atípicos

Para asegurarse de la calidad de los datos en el modelado, se realizó un tratamiento de valores atípicos (outliers) mediante el método del rango intercuartílico (IQR) lo cual identifico valores extremos, calculando los percentiles Q1 (25%) y Q3 (75%) [11]. Estos valores se consiguieron con los límites para detectar los datos atípicos [12], que se establecen con 2 fórmulas matemáticas.

Límite inferior que define el umbral mínimo por debajo del cual un dato es considerado atípico. Se calcula mediante la expresión:

$$Q1 - 1.5 \times IQR \quad (1)$$

Límite superior que representa el umbral máximo a partir del cual un dato se clasifica como atípico. Se obtiene con la ecuación:

$$Q3 + 1.5 \times IQR \quad (2)$$

Las variables de analizadas fueron **CANTCREDMAT**, **CICLO** y **CANTCURMAT**. La variable **CANTCURMAT** de la Fig. 2 no presentó valores fuera de lo común.

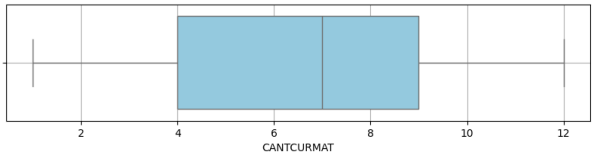

Fig. 2 Boxplot de la variable **CANTCURMAT**

La variable **CANTCREDMAT** identificó 7,007 casos con pocos créditos matriculados (entre 1 y 9 créditos), lo que podría estar asociado con estudiantes retirados, matriculados de manera parcial o enfrentan situaciones especiales (Fig. 3).

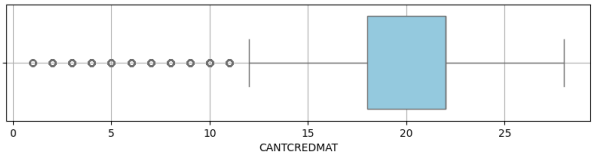

Fig. 3 Boxplot de la variable **CANTCREDMAT**

La variable **CICLO** presentó inconsistencias como valores nulos, no numéricos, igual a 0, y artificiales como 99 como se muestra en la Fig. 4, que no corresponden a ciclos válidos en la UNE, que son esencialmente de 10 ciclos.

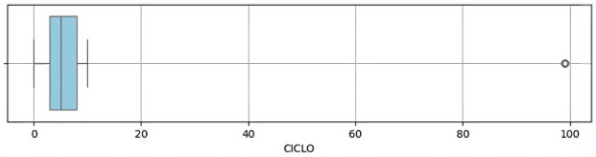

Fig. 4 Boxplot de la variable **CICLO**

Para tratar esto se hizo la conversión a tipo numérico, forzando errores como valores nulos, identificación de valores válidos entre 1 y 10 correspondientes al rango real de ciclos académicos universitarios, e imputación por mediana reemplazando valores nulos, iguales a 0 o mayores a 10 por la mediana obtenido de los datos válidos.

IV. METODOLOGÍA

En esta sección se explican cómo fueron los entornos de experimentación, cómo se seleccionaron las variables, qué técnicas se usaron para manejar el desbalance de clases, cómo se configuraron los hiperparámetros y qué modelos se aplicaron, destacando las decisiones que permitieron obtener mejores modelos de aprendizaje automático.

A. Configuración experimental

Con el fin de simular condiciones reales se emplea una estrategia de validación temporal del despliegue institucional. Para esto el conjunto de datos tiene 2 particiones.

Los registros de los años 2020-2023 son para entrenamiento y validación, luego los registros del 2024 se usan para un conjunto de prueba externo excluyéndolos de la fase de entrenamiento ni su selección de hiperparámetros.

Se utilizó GroupKFold con 5 fold en su validación cruzada para poder utilizar su variable UUID como variable de agrupación. Esto es para que cada estudiante permanezca dentro de un único fold y no combine datos de prueba con los datos de entrenamiento, esto para evitar la dependencia entre observaciones y asegurando una estimación más realista del rendimiento del modelo. Sus hiperparámetros de cada modelo utilizaron GridSearchCV en los datos de entrenamiento sin mirar lo datos del prueba externo en ningún momento.

Para la elección de un umbra (threshold) óptimo en cada modelo, se utilizó el conjunto de datos de entrenamiento y validación, buscando maximizar el F1-score con la restricción de mantener un Recall mínimo del 60% [13]. Este criterio se justifica debido a que el costo de no identificar un estudiante semipresencial (false negativo), involucraría una planificación insuficiente de recursos tecnológicos y docentes, considerando un mayor costo de una sobreasignación de recursos ante un falso positivo.

Los experimentos fueron ejecutados en un entorno Google Colab con procesador Intel Xeon a 2.20 GHz, 12 GB de RAM y sin unidad de procesamiento gráfico (CPU only).

B. Selección De Variables

Se implementó un enfoque híbrido de selección de variables que combinó métodos estadísticos con enfoque de Machine Learning para optimizar el desempeño predictivo de los modelos [14].

El proceso utilizó dos técnicas complementarias como SelectKBest para evaluar la relación estadística entre cada variable independiente y la variable objetivo mediante el estadístico F de ANOVA, y Random Forest para calcular la relevancia de las variables basada en la disminución de la impureza de Gini.

Pero antes se incluyó la codificación de variables categóricas mediante LabelEncoder para poder usar en ambos modelos, para facilitar su comparación y cálculo se usó un puntaje combinado (**COMBINED_SCORE**) mediante promedio ponderado, esto se muestra en Fig. 5.

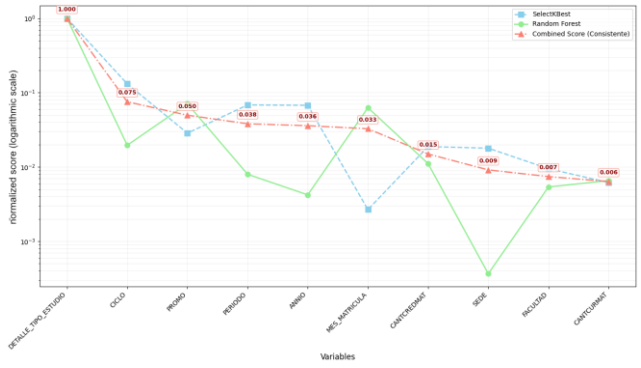

Fig. 5 Comparación de puntajes normalizados por variable

Lo más destacable que se puede ver es que la variable "**DETALLE_TIPO_ESTUDIO**" obtuvo el puntaje perfecto (1.00) en ambos enfoques. Otras variables como "**CICLO**", "**PROMO**", "**PERIODO**, "**ANNIO**" y "**MES_MATRICULA**" mostraron unas contribuciones moderadas, mientras que las demás presentaron valores bajos, indicando menor impacto en la predicción.

Se estableció un umbral >= 0.03 en el método "puntaje combinado", lo que permite filtrar aquellas características con mayor capacidad predictiva y descartar atributos con menos aporte marginal o ruido estadístico.

C. Fugas De Información

La identificación de la variable "**DETALLE_TIPO_ESTUDIO**" con puntación perfecta (Fig. 5) muestra una relación directa con la variable objetivo. En situaciones reales de clasificación, esta anomalía revela una fuga de información, es decir, la variable probablemente incluía una descripción contenida en la variable objetivo, lo que afecta la objetividad del análisis.

El análisis exploratorio corroboró que esta variable está estrechamente relacionada con la modalidad de matrícula, siendo prácticamente como una segunda variable objetivo.

Para evitar problemas, se decidió excluir del análisis a la variable "**DETALLE_TIPO_ESTUDIO**" de los datos que se usaron para el entrenamiento.

D. Modelos

Los modelos de *boosting* se han consolidado como la opción superior para clasificar datos tabulares cuando existe un desbalance marcado, especialmente si se integran con un ajuste fino de hiperparámetros y técnicas de peso de clases [13]. La gran diferencia con los métodos tradicionales es que los algoritmos de *boosting* trabajan de forma secuencial: cada nuevo modelo intenta corregir los fallos del anterior, lo que facilita enormemente la captura de esos patrones difíciles que definen a la clase minoritaria [15].

XGBoost destaca por su uso de *gradient boosting* regularizado. De los usados parámetros como gamma y colsample_bytree, permite un control muy preciso del sobreajuste, además de usar scale_pos_weight para darle más importancia a los errores en la clase menos frecuente [16].

LightGBM tiene a su favor en que hace crecer los árboles por hojas y no por niveles. Esto no solo acelera el entrenamiento, sino que maneja mucho mejor los conjuntos de datos grandes donde las clases no están equilibradas [17].

CatBoost es especialmente robusto porque gestiona de forma nativa las variables categóricas y aplica el *ordered boosting* para evitar sesgos en la predicción, funcionando muy bien ante el desbalance sin necesidad de pasos extra de preprocesamiento [18].

Gradient Boosting Machine (GBM) es la versión clásica de scikit-learn [19] como punto de referencia, para medir cuánto terreno han ganado realmente las versiones más modernas.

El Dummy Classifier se usó para marcar el rendimiento mínimo, ya que simplemente predice siempre la clase mayoritaria [20] y la Regresión Logística como baseline lineal; su propósito fue confirmar si la complejidad de los modelos de *boosting* realmente vale la pena frente a un método simple [21].

Lo esperable es que los modelos de *boosting* superen con creces a estos baselines en métricas como F1, MCC y PR-AUC, ya que están mejor equipados para entender la estructura no lineal de este problema educativo.

E. Optimización De Hiperparámetros

Para encontrar los mejores hiperparámetros, utilizamos **GridSearchCV** junto con un esquema de **GroupKFold** de 5 particiones. Como métrica principal de optimización elegimos el **PR-AUC** (average_precision), una decisión lógica considerando el fuerte desbalance que tienen los datos, en la siguiente Tabla II se mientras los hiperparámetros.

TABLE II
ESPACIO DE BÚSQUEDA GRIDSEARCHCV

Parámetro	XGBoost	LightGBM	CatBoost	Gradient Boosting
learning_rate (búsqueda)	[0.01, 0.05, 0.1]	[0.01, 0.05, 0.1, 0.2]	[0.01, 0.1, 0.2, 0.3]	[0.01, 0.1, 0.2, 0.3, 0.4]
learning_rate (final)	0.05	0.10	0.30	0.10
max_depth (búsqueda)	[3, 5, 7]	[3, 5, 7, 9, 12]	[3, 5, 7, 9]	[3, 5, 7, 9, 11, 13]
max_depth (final)	7	9	9	5
n_estimators (búsqueda)	[100, 200, 400]	[100, 200]	[100, 200, 300]	[100, 200, 250]
n_estimators (final)	200	100	100	100
subsample (búsqueda)	[0.8, 1.0]	[0.6, 0.8, 1.0]	[0.8, 1.0]	[0.8, 1.0]
subsample (final)	—	0.6	0.8	0.8
colsample (búsqueda)	[0.8, 1.0]	[0.8, 1.0]	[0.8, 1.0]	—
colsample (final)	0.8	0.8	0.8	—
gamma (búsqueda)	[0, 0.1, 0.5]	—	—	—
gamma (final)	0.1	—	—	—

*** En CatBoost el parámetro equivalente es colsample_bylevel**

También cabe mencionar la enorme diferencia en el esfuerzo computacional que requirió cada algoritmo, como se muestra a continuación en la Tabla III.

TABLA III
TIEMPO DE ENTRENAMIENTO POR MODELO (GRIDSEARCHCV CON 5 FOLDS)

Modelo	**Tiempo GridSearch**	**Tiempo GridSearch (min)**	**Tiempo Test Externo**
XGBoost	1,369.11 seg	~22.8 min	0.11 seg
LightGBM	2,915.46 seg	~48.6 min	0.06 seg
CatBoost	3,480.60 seg	~58.0 min	0.02 seg
Gradient Boosting	10,141.88 seg	~169.0 min	—
Regresión Logística + Dummy	52.35 seg	~0.87 min	—

El Gradient Boosting fue, por mucho, el más lento, tomando cerca de 169 minutos en completar su entrenamiento.

En contraste, los modelos de boosting modernos fueron mucho más rápidos: CatBoost tardó 58 minutos, LightGBM unos 48 minutos y XGBoost se posicionó como el más eficiente, terminando en apenas 23 minutos. Por su parte, los modelos baseline fueron casi instantáneos, tomando menos de un minuto entre todos.

Es importante mencionar que todas las pruebas se corrieron usando solo CPU, sin depender de tarjetas gráficas (GPU). Esto significa que los tiempos reportados son un escenario conservador, lo que asegura que cualquier institución con hardware estándar podría replicar estos resultados sin problemas.

F. Selección De Umbral

Como la clasificación tiene un desbalance de clases tan marcado (Fig. 1), usar el umbral más común (que es 0.5 en muchos modelos son por defecto), no suele ser el adecuado, ya que puede favorecer la clasificación de la clase mayoritaria e ignorar a la clase minoritaria [13].

Entonces, para elegir un umbral de decisión adecuado se decidió usar dos tipos de umbrales. El primero, que llamamos umbral interno, se obtuvo durante la validación cruzada con los datos de entrenamiento de 2020 a 2023, como parte de optimización de los hiperparámetros. El segundo es el umbral operativo, obtenido mediante un análisis complementario sobre el conjunto de 2024, con el objetivo de identificar un punto de decisión adecuado para su aplicación operativa, buscando maximizar el F1-score pero cumpliendo con la regla de no bajar de un 60% de Recall [13].

TABLA IV
UMBRALES DE DECISION INTERNO Y OPERATIVO

Modelo	Umbral Interno	Umbral Operativo
XGBoost	0.32	0.13
LightGBM	0.33	0.16
CatBoost	0.29	0.23
Gradient Boosting	0.34	0.21

El umbral operativo se presenta como un análisis complementario ejecutado sobre el conjunto externo de 2024 para examinar la adaptación del criterio de decisión al comportamiento observado en dicho periodo. Este análisis no interviene en la selección, entrenamiento ni ajuste de los modelos.

La diferencia entre estos valores es muy reveladora. Mientras que los umbrales internos rondaban el 0.30–0.34, los operativos bajaron hasta el rango de 0.13–0.23.

Esto está pensado para la planeación de la universidad, no detectar a un alumno semipresencial es un falso negativo lo que trae problemas graves al asignar mal los docentes y la tecnología, mientras que si nos pasamos un poco con un falso positivo, el costo operativo es mucho más bajo y fácil de manejar.

V. RESULTADOS

La evaluación de cada modelo se hizo usando su propio umbral operativo realizado en la Sección III.F. Las métricas obtenidas sobre el conjunto de evaluación de 2024 se presentan en la tabla V, empleando indicadores adecuados para escenarios con desbalance de clases.

TABLA V
MÉTRICAS DE PRUEBA EXTERNO 2024 POR MODELO

Métrica	XGBoost	LightGBM	CatBoost	GBM	Reg. Log.	Dummy
Accuracy	0.993	0.993	0.989	0.986	0.834	0.003
Precision	0.258	0.258	0.179	0.167	0.019	0.003
Recall	0.654	0.615	0.731	0.846	1.000	1.000
Specificity	0.994	0.994	0.989	0.987	0.834	0.000
NPV	0.999	0.999	0.999	1.000	1.000	0.000
F1	0.370	0.364	0.288	0.279	0.037	0.006
AUC	0.970	0.987	0.987	0.991	0.987	0.500
PRAUC	0.158	0.196	0.132	0.261	0.147	0.003
BalAcc	0.824	0.805	0.860	0.916	0.917	0.500
MCC	0.408	0.396	0.358	0.372	0.125	0.000

Si nos fijamos en el F1-score, el XGBoost fue el que sacó el mejor puntaje con 0.370, y por debajo de quedaron LightGBM con 0.364 y el CatBoost con un 0.288. Sobre el MCC evalúa de manera equilibrada los aciertos y errores en las dos clases, XGBoost también obtuvo un resultado alcanzando con 0.4076. Asimismo, el AUC, mostró resultados adecuados en todos los modelos de Boosting con una capacidad de distinguir clases muy alta, moviéndose entre 0.969 y 0.991, siendo el Gradient Boosting el más alto en este punto específico.

En cuanto al Recall, el Gradient Boosting alcanzó el valor más alto con 0.846, detectando 22 de los 26 casos que de verdad eran semipresenciales, pero claro, eso le costó tener 110 falsos positivos. Por otro lado, el XGBoost encontró 17 de los 26 casos y generó 49 falsos positivos, lo que representa el mayor F1 y MCC del conjunto, lo que evidencia un mejor equilibrio entre detección de la clase minoritaria y control de falsas alarmas.

Asimismo, la matriz de confusión en la Tabla VI, muestra que el GBM alcanza el mejor balance al reducir falsos negativos (FN = 4) aunque esto implicó generar un mayor número de falsos positivos (FP=110), mientras la regresión logística y el clasificador dummy son deficientes al generar muchos falsos positivos, lo que confirma que los métodos de *boosting* ofrecen una ventaja clara en escenarios de desbalance severo.

TABLA VI
MATRIZ DE CONFUSIÓN RESUMIDA POR MODELO

Modelo	TP	FP	FN	TN
XGBoost	17	49	9	8182
LightGBM	16	46	10	8185
CatBoost	19	87	7	8144
GBM	22	110	4	8121
Reg. Logística	26	1371	0	6860
Dummy	26	8231	0	0

La comparación de curvas ROC en la Fig. 6 muestran la distinción entre las clases y entender su capacidad de generalización entre los modelos.

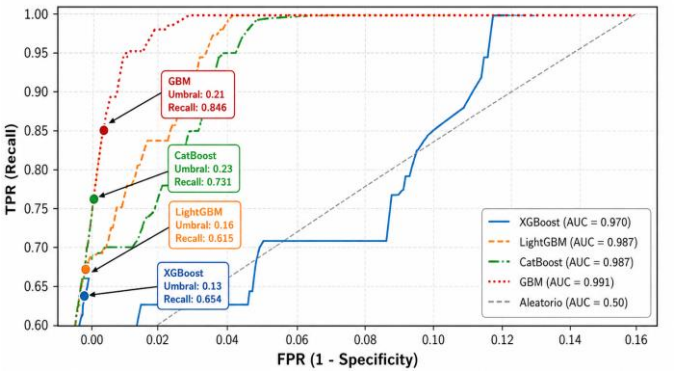

Fig. 6 Curvas ROC - Modelos de Boosting (Zoom en Curvatura)

El modelo GBM es el que llega a la mayor capacidad para distinguir las clases con un AUC de 0.9912, logrando un balance más bueno entre la sensibilidad y la especificidad, con un recall de 0.846 usando un umbral de 0.210. Por otro lado, tanto el LightGBM y CatBoost presentan valores de AUC muy similares de 0.9871, aunque sus niveles de Recall son diferentes alcanzando 0.615 y 0.731, respectivamente. Mientras tanto, el XGBoost se queda en un nivel que es competitivo pero algo más bajo con un AUC de 0.9695. Todo esto junto nos dice que, bajo estas condiciones, el modelo GBM muestra un mejor desempeño en términos de Recall, AUC, PR-AUC y Balanced Accuracy aunque XGBoost obtiene mejores resultados en F1 y MCC.

Si miramos la Fig. 7 con las curvas de precisión-recall se ven las diferencias para equilibrar la sensibilidad y la precisión entre cada modelo.

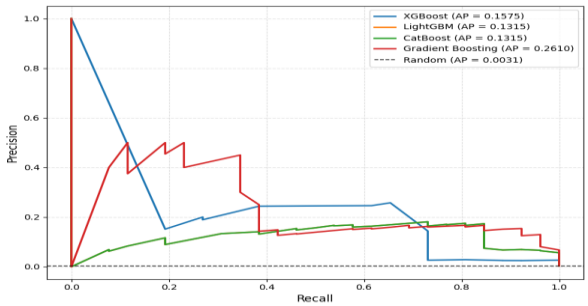

Fig. 7 Curvas de Precision-Recall de modelos boosting

El modelo GBM resalta con el valor más alto de PR-AUC, llegando a un AP de 0.261, lo que nos indica que funciona mejor para encontrar los positivos verdaderos sin arruinar tanto la precisión. Por otro lado, el XGBoost llega a un AP de 0.158, mientras que el LightGBM obtiene 0.196 y CatBoost registra 0.132, lo que muestra que son un poco menos eficaces con este set de datos.

La línea que marca el clasificador aleatorio tiene un AP de 0.003 y sirve como base para ver que todos los modelos de boosting son muchísimo mejores que el azar.

VI. EXPLICABILIDAD Y EQUIDAD

Para estar seguros de que los modelos que proponemos no solo funcionen bien, sino que también se puedan entender y sean justos, hicimos un análisis completo de explicabilidad y equidad. En esta parte mostramos los resultados de que tanto pesan las variables, el uso de SHAP para ver como decide el modelo, y como calibramos las probabilidades para que los resultados sean confiables.

También revisamos que el modelo sea equitativo entre los diferentes grupos académicos. Por último, explicamos cómo funciona el sistema de alertas tempranas que usamos para pasar todos estos datos a acciones reales en la universidad.

A. Importancia de la característica

Para revisar que tan parecidos son los modelos con sus variables, se compara la importancia normalizada usando el atributo "feature_importances_" de cada algoritmo.

La grafica de la Fig. 8 muestra claramente como cada algoritmo le da un peso diferente a las variables académicas y temporales.

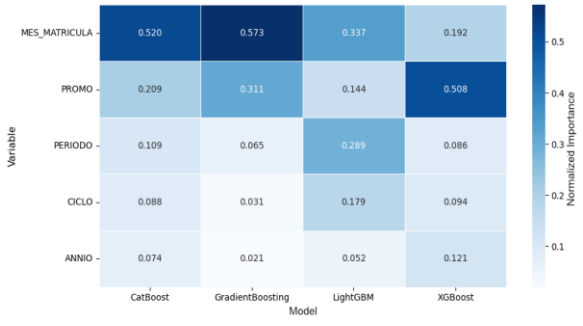

Fig. 8 Importancia de las funciones del Heatmap - Modelos Boosting

Sobre todo, **MES_MATRICULA** y **CICLO** salen como los factores que más pesan en casi todos los modelos, lo que confirma que el tiempo y como avanza el alumno en su carrera son claves para entender su comportamiento.

Por otra parte, **PROMO** y **PERIODO** varían mucho en su importancia, ya que unos modelos les hacen mucho caso mientras que otros las dejan casi de lado. Al contrario, **ANNO** suele tener una importancia más pequeña en todos los casos, lo que nos dice que solita no aporta tanto como las demás variables.

La Fig. 9, por otro lado, resume que tan fuerte es el efecto promedio de cada variable en el modelo. Aqui se vuelve a confirmar que **PROMO** es la característica que más influye en promedio, y le siguen **MES_MATRICULA** y **CICLO**, mientras que **PERIODO** y **ANNIO** tienen un peso mucho más bajo.

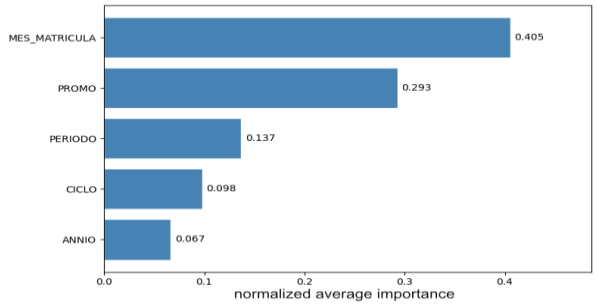

Fig. 9 Importancia Promedio entre Modelos Boosting

Los resultados muestran que **PROMO** tiene un efecto negativo en algunos casos específicos, pero su importancia en todos los modelos no es significativa; esto significa que no manda en las decisiones ni mete sesgos raros en la estructura. Aunque **PROMO** no es un peligro ni un sesgo por sí solo, si es una variable que el XGBoost usa todo el tiempo para poder distinguir bien entre las clases. Al final, funciona como un apoyo constante para que el algoritmo no se equivoque al clasificar, pero sin llegar a dominar el resultado final del análisis.

Este hallazgo es muy importante para el sector educativo, porque ayuda a elegir qué características usar y como revisar los resultados en distintas instituciones o contextos.

B. Análisis SHAP

Para entender mejor cómo funciona el modelo que estamos recomendando, se usó el análisis SHAP con TreeExplainer en una muestra de 1,000 registros del test externo de 2024.

Nos enfocamos solo en el XGBoost porque es el que mejor balance tiene entre F1 y MCC, lo que lo hace el candidato ideal para usarse en la universidad.

El grafico en la Fig. 10 para el XGBoost muestra cada variable afecta a lo que predice el modelo y también sirve para ver qué tan sólida es la estrategia.

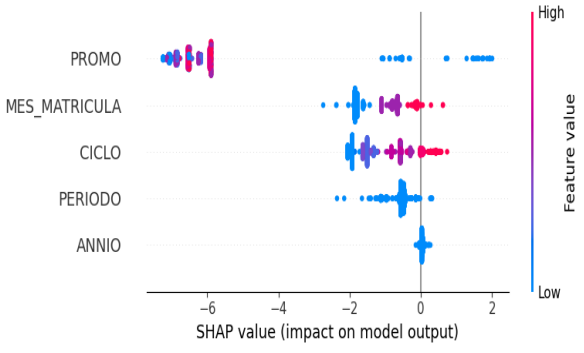

Fig. 10 Gráfico resumen SHAP - XGBoost

En este análisis, la variable **PROMO** aparece con una importancia relativamente baja frente a **MES_MATRICULA** y **CICLO** que resaltan por tener un impacto más constante, ya que sus valores pueden mover la predicción hacia lo positivo o lo negativo. Esto nos confirma que el tiempo y como va avanzando el alumno en sus estudios son los puntos centrales para clasificar bien. En cambio, **PERIODO** y **ANNIO** tienen efectos que casi siempre están cerca de cero, lo que nos dice que solitas no aportan tanto al resultado final.

El grafico de la Fig. 11, por otro lado, resume que tan fuerte es el efecto promedio de cada variable en el modelo.

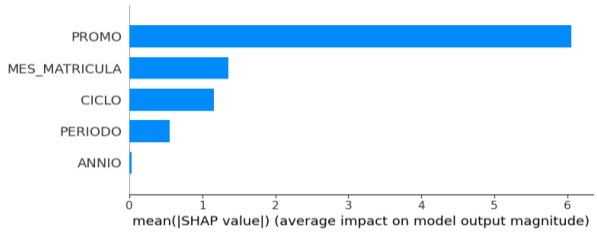

Fig. 11 Importancia de las características SHAP - XGBoost

Aqui se vuelve a confirmar que **PROMO** es la característica que más influye en promedio, seguido por detrás están **MES_MATRICULA** y **CICLO**, mientras que **PERIODO** y **ANNIO** tienen un peso mucho más bajo. Este orden ayuda a completar lo que vimos en el Fig. 10.

Los resultados muestran que **PROMO** tiene un efecto negativo en algunos casos específicos, pero su importancia en todo el modelo es baja (VI.A.); esto significa que no manda en las decisiones ni mete sesgos raros en la estructura.

Aunque **PROMO** no es un peligro ni un sesgo por sí solo, si es una variable que el XGBoost usa todo el tiempo para poder distinguir bien entre las clases. Al final, funciona como un apoyo constante para que el algoritmo no se equivoque al clasificar, pero sin llegar a dominar el resultado final del análisis.

C. Calibración de Probabilidad

Para ver que tan bien calibradas estaban las probabilidades, evaluamos tres configuraciones: las originales, Platt Scaling y la Regresión Isotónica, usando el Brier Score y el LogLoss como indicadores de calidad. Los resultados se presentan mediante la curvas de calibración y la comparación gráfica de las métricas, ver Figuras 12 y 13.

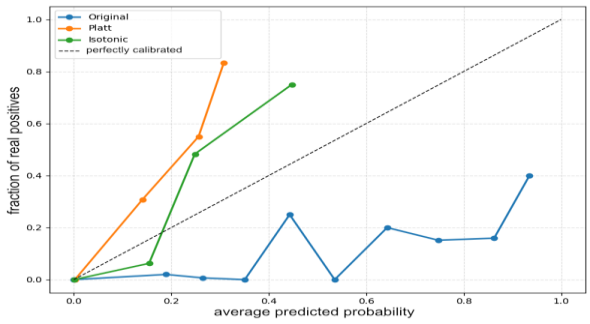

Fig. 12 Curva de Calibración – XGBoost (Original vs Platt vs Isotonic)

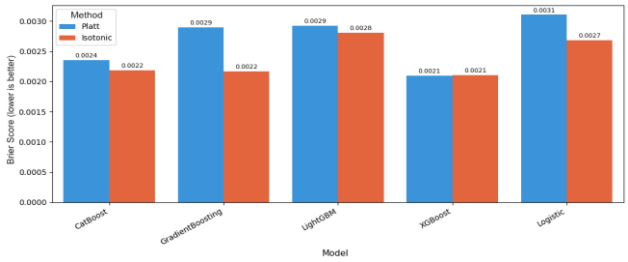

Fig. 13 Brier Score por Modelo y Método de Calibración

El XGBoost con Regresión Isotónica sacó el Brier Score más bajo (0.002) y también el menor LogLoss (0.007), lo que confirma que sus probabilidades son las más confiables para que la universidad las use como un indicador de riesgo real.

Un caso que llama mucho la atención es el del Gradient Boosting, donde el AUC se desploma a 0.424 si se usa Platt Scaling, pero sube hasta 0.998 con la Regresión Isotónica.

Esto revela que el Platt Scaling distorsiona incorrectamente a este modelo y no se debería usar para calibrarlo. Por su parte, los modelos baseline no mejoraron con ningún método, lo cual tiene sentido ya que solo son nuestra referencia mínima.

D. Equidad Por Subgrupos

Se hizo un análisis del AUC para el modelo XGBoost separándolo por subgrupos de **FACULTAD**, **SEDE** y **SEXO** usando los datos del test externo de 2024. Para que los resultados tengan sentido estadístico, se excluyeron a los grupos que tenían menos de 30 registros; toda esta información se puede revisar en las Figuras 14, 15 y 16.

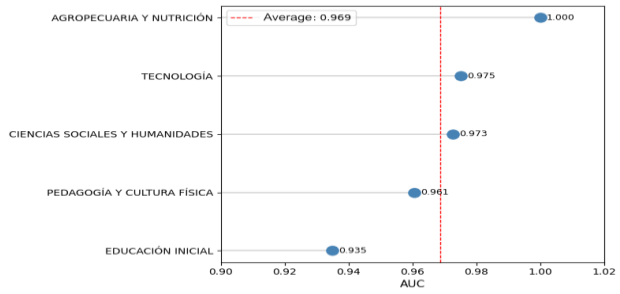

Fig. 14 AUC por Facultad - XGBoost

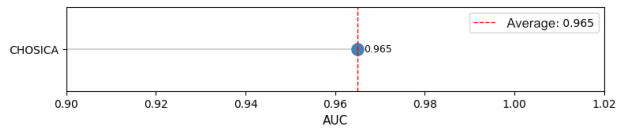

Fig. 15 AUC por Sede - XGBoost

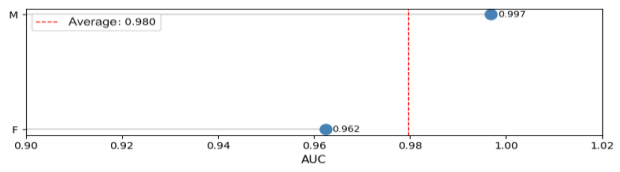

Fig. 16 AUC por Sexo - XGBoost

Los datos muestran que el modelo se porta de forma equitativa en todos los subgrupos que evaluamos. Por ejemplo, el AUC se mantuvo entre 0.934 y 1.000 en las cinco facultades, mientras que en la sede de Chosica llegó a 0.965. Si miramos el sexo, el puntaje fue de 0.996 para hombres y 0.962 para mujeres; esa diferencia de 0.034 puntos no se considera un sesgo importante.

Como ningún grupo evaluado presentó un AUC inferior a 0.930, los resultados muestran una capacidad discriminativa elevada y relativamente consistente entre los subgrupos académicos analizados, no observando diferencias marcadas en este indicador.

E. Sistema De Alerta Temprana

Como análisis complementario, armamos un sistema de alertas tempranas usando un ensemble con las probabilidades de los cuatro modelos de boosting. El riesgo de cada alumno se sacó promediando lo que decía cada modelo, y la incertidumbre la medimos con la desviación estándar; esto sirve para ver que tanto "discuten" los modelos entre ellos. Con estos datos pusimos cuatro niveles: **CRÍTICO** (prob ≥ 0.800), **ALTO** (prob ≥ 0.650), **MEDIO** (prob ≥ 0.450) y **BAJO** (prob < 0.450).

Los niveles de alerta se establecieron mediante rangos de probabilidad, mientras que el criterio de acción prioritaria utilizó adicionalmente un umbral de 0.70 y una incertidumbre inferior a 0.10, asegurando que todos los modelos estén de acuerdo en el resultado.

Esto se puede ver en la Figura 17, donde se nota que casi todos los estudiantes están en el nivel **BAJO** (más de 8,000 casos con una probabilidad promedio de 0.011), mientras que los grupos de riesgo son mucho más chicos: 20 en nivel **MEDIO** (0.557), 85 en **ALTO** (0.753) y 28 en **CRÍTICO** (0.828). Esta división deja claro que, aunque los críticos son poquitos, tienen un riesgo muy alto y necesitan atención rápida.

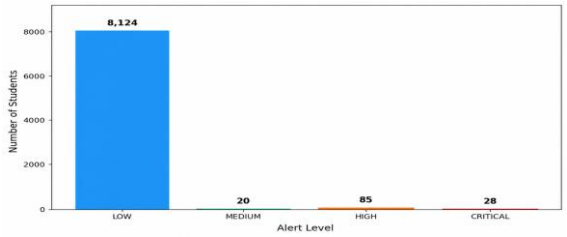

Fig. 17 Distribución de Alertas Tempranas - Test Externo 2024

Por otro lado, la Figura 18 muestra cómo se llevan el riesgo y la incertidumbre, verificando como se reparten los alumnos según la probabilidad del ensemble y que tanto varían los modelos.

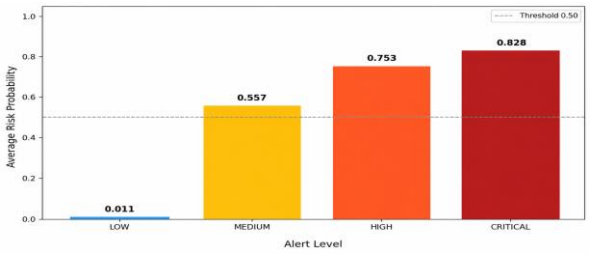

Fig. 18 Probabilidad Promedio por Nivel de Alerta

Los límites que pusimos ayudan a separar los casos de mucha probabilidad y poca incertidumbre —donde actuar es más confiable— de aquellos donde hay mucha duda y hace falta revisar más antes de gastar recursos.

En la Figura 19 se ve una simulación de impacto con tres tipos de intervención de la universidad. Los números dicen que se podrían evitar entre 1 y 5 cancelaciones, dependiendo de que tan fuerte sea la estrategia (conservadora, moderada o agresiva).

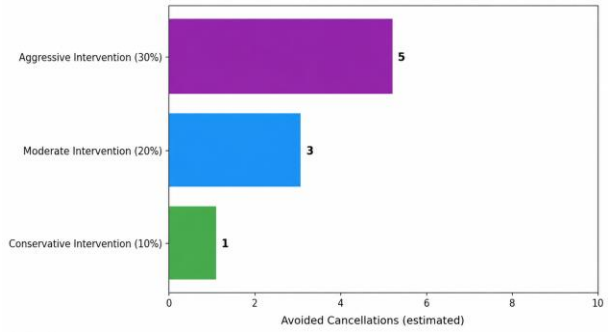

Fig. 19 Matriz de Impacto Contrafactual - Escenarios de Intervención

Por último, la Figura 20 muestra de forma combinada como interactúan el riesgo y la incertidumbre, resaltando los umbrales de acción prioritaria.

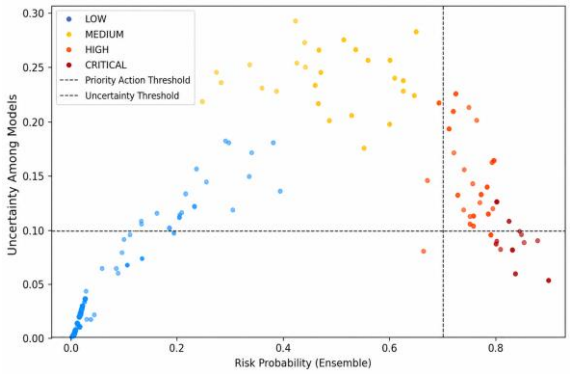

Fig. 20 Riesgo vs Incertidumbre por Nivel de Alerta

Este análisis deja identificar mejor los casos donde el riesgo es alto y la variación entre modelos es baja, lo que da más confianza para decidir cuándo intervenir.

VII. CONCLUSIONES

Se hizo una comparativa bien detallada de cuatro modelos de boosting —XGBoost, LightGBM, CatBoost y Gradient Boosting— para clasificar la modalidad de matrícula en una universidad del Perú. El estudio sirvió para ver cuáles son los puntos fuertes y las fallas de cada método en un escenario con un desbalance de clases muy pesado (95/5), donde también influyen mucho los ciclos académicos y el peligro de que se filtre información en el modelo.

Estos retos son puntos que nos dicen por dónde mejorar la metodología y como hacer las validaciones en otros lugares con características similares.

La evaluación realizada sobre los datos de 2024 muestra que XGBoost tuvo el mejor balance ante el desbalance, sacando un F1 de 0.370, un MCC de 0.408 y un AUC de 0.970; con esto detectó 17 de los 26 casos reales usando un umbral de 0.130. El Gradient Boosting tuvo el Recall más alto (0.846), atrapando 22 de 26 casos, aunque generó más falsos positivos. Por su parte, LightGBM y CatBoost compitieron bien con un AUC arriba de 0.980, siendo opciones validas según lo que la universidad prefiera priorizar. Los modelos boosting mostraron un desempeño superior a los baselines en las métricas orientadas a la clase minoritaria, particularmente F1 y MCC.

Un punto muy importante es que el umbral interno de la validación cruzada no detectó ningún positivo en las pruebas del 2024, lo que sugiere un cambio en la distribución de las observaciones entre el período de entrenamiento y el año 2024. Esto deja claro que el umbral operativo puede requerir calibración cuando empiecen nuevos ciclos académicos y justifica que usemos validaciones temporales estrictas en este tipo de predicciones educativas.

El análisis de SHAP confirmo que **PROMO** y **MES_MATRICULA** son las variables que más pesan en el modelo, seguido de **CICLO, PERIODO** y **ANNIO** y la importancia de variables mostró que los cuatro modelos están de acuerdo en que lo temporal y lo académico es lo más relevante. También se observó una capacidad discriminativa elevada y relativamente consistente entre los subgrupos evaluados, ya que el AUC fue mayor a 0.930 en todas las facultades, sedes y sexos, sin mostrar señales de discriminación o sesgo.

Calibrar las probabilidades con Regresión Isotónica ayuda en la confianza de los modelos, siendo el XGBoost el más preciso con el Brier Score más bajo (0.002) y un LogLoss de 0.007. Estos resultados sugieren que el modelo puede ser útil para su utilización como indicador continuo de riesgo en la institución. Además, el sistema de alertas detectó 28 casos **CRÍTICOS** y 85 de nivel **ALTO** en 2024, son relevantes para planear los recursos académicos con tiempo.

Se hizo uso de cinco variables de tipo académico y temporal que sacamos de un dataset de la Plataforma Nacional de Datos Abiertos. Asimismo, no se agregaron factores socioeconómicos que podrían haber ayudado a predecir mejor, pero aun así los resultados dan una evidencia bastante sólida para lo que pasa en la Universidad Nacional de Educación.

Finalmente, para fortalecer estos hallazgos y tengan más peso en otros lados, va a ser necesario replicar y comparar el modelo en diferentes instituciones de educación superior y con datos que sean más variados y actualizados.

REFERENCIAS

[1] A. G. Floralba and S. V. María, *Experiencias docentes en tiempo de pandemia*. 2022. doi: 10.7476/9789978108222.

[2] Y. Singhal, A. Jain, S. Batra, Y. Varshney, and M. Rathi, "Review of bagging and Boosting Classification Performance on unbalanced binary Classification," *IEEE*, pp. 338–343, Dec. 2018, doi: 10.1109/iadcc.2018.8692138.

[3] M. D. Adewale, A. Azeta, A. Abayomi-Alli, and A. Sambo-Magaji, "Impact of artificial intelligence adoption on students' academic performance in open and distance learning: A systematic literature review," *Heliyon*, vol. 10, no. 22, p. e40025, Nov. 2024, doi: 10.1016/j.heliyon.2024.e40025.

[4] L. V. Asencios-Trujillo, L. Asencios-Trujillo, C. J. La-Rosa-Longobardi, D. Gallegos-Espinoza, and C. Piñas-Livia, "Determining the level of adaptability of students in online education using machine learning algorithms," *International Journal of Engineering Trends and Technology*, vol. 73, no. 1, pp. 305–312, Jan. 2025, doi: 10.14445/22315381/ijett-v73i1p125.

[5] L. Sánchez-Guerrero, J. Figueroa-González, B. González-Beltrán, and S. González-Brambila, "Evaluating Predictive Techniques in Educational Data Mining: An Unbalanced Data Set case of study," *Research in Computing Science*, vol. 148, no. 3, pp. 49–60, Dec. 2019, doi: 10.13053/rcs-148-3-4.

[6] H. He and E. A. Garcia, "Learning from Imbalanced Data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, Jul. 2009, doi: 10.1109/tkde.2008.239.

[7] V. T. N. Chau and N. H. Phung, "Imbalanced educational data Classification: an effective approach with resampling and random forest," *IEEE*, vol. 45, pp. 135–140, Nov. 2013, doi: 10.1109/rivf.2013.6719882.

[8] Plataforma Nacional de Datos Abiertos del Perú, "Lista de Alumnos Matriculados desde el 2020 al 2024 en la Universidad Nacional de Educación (UNE)," Gobierno del Perú, 2024. [En línea]. Available: https://www.datosabiertos.gob.pe/dataset/lista-de-alumnos-matriculados - desde-el-2020-al-2024-en-la-une-egyv. [Último acceso: 21 Abril 2026].

[9] Plataforma Nacional de Datos Abiertos del Perú, "Dataset: Alumnos Matriculados UNE 2020–2024," Gobierno del Perú, 2024. [En línea]. Available: https://www.datosabiertos.gob.pe/sites/default/files/MATRICULADOS-UNE-2020-2024_0.csv. [Último acceso: 21 Abril 2026].

[10] Plataforma Nacional de Datos Abiertos del Perú, "Diccionario de Datos: Alumnos Matriculados UNE 2020–2024," Gobierno del Perú, 2024. [En línea]. Available: https://www.datosabiertos.gob.pe/sites/default/files/DiccionarioDatos_matricula_alumnos_UNE.xlsx. [Último acceso: 21 Abril 2026].

[11] "Interquartile range," in *The Concise Encyclopedia of Statistics*, 2008, pp. 266–267. doi: 10.1007/978-0-387-32833-1_200.

[12] J. W. Tukey, *Exploratory data analysis*. Addison-Wesley Publishing Company, 1977.

[13] M. Abdelhamid and A. Desai, "Balancing the Scales: A comprehensive study on tackling class imbalance in binary classification," *arXiv (Cornell University)*, Sep. 2024, doi: 10.48550/arxiv.2409.19751.

[14] R. Iranzad and X. Liu, "A review of random forest-based feature selection methods for data science education and applications," *International Journal of Data Science and Analytics*, vol. 20, no. 2, pp. 197–211, Feb. 2024, doi: 10.1007/s41060-024-00509-w.

[15] C. Li, Z. Li, J. Xu, J. Li, and S. Li, "Collaborative optimization of multiclass imbalanced learning: Density-Aware and Region-Guided boosting," *arXiv (Cornell University)*, Dec. 2025, doi: 10.48550/arxiv.2512.22478.

[16] T. Chen and C. Guestrin, "XGBOOST: a scalable tree boosting System," *arXiv (Cornell University)*, Mar. 2016, doi: 10.48550/arxiv.1603.02754.

[17] X. Zhao, Y. Liu, and Q. Zhao, "Improved LightGBM for extremely imbalanced data and application to credit card fraud detection," *IEEE Access*, vol. 12, pp. 159316–159335, Jan. 2024, doi: 10.1109/access.2024.3487212.

[18] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: unbiased boosting with categorical features," *arXiv (Cornell University)*, vol. 31, pp. 6639–6649, Jun. 2017, doi: 10.48550/arxiv.1706.09516.

[19] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," *Frontiers in Neurorobotics*, vol. 7, p. 21, Jan. 2013, doi: 10.3389/fnbot.2013.00021.

[20] "DummyClassifier," *Scikit-learn*. https://scikit-learn.org/stable/modules/generated/sklearn.dummy.DummyClassifier.html

[21] H. Zhang, Z. Li, H. Shahriar, L. Tao, P. Bhattacharya, and Y. Qian, "Improving Prediction Accuracy for Logistic Regression on Imbalanced Datasets," *IEEE*, Jul. 2019, doi: 10.1109/compsac.2019.00140.