Trabajo Práctico: Deserción Estudiantil

Modelo de Predicción de Bajas en Estudiantes Universitarios

Guadalupe Alesandro
guadalupealesandro@gmail.com

María Florencia Firpo
firpoflorencia@gmail.com

Sofía Forni
sof.forni@gmail.com

Abstract

La deserción estudiantil universitaria representa un desafío para las instituciones de educación superior. Este trabajo aborda su predicción mediante la construcción y comparación de modelos de aprendizaje automático. Utilizando un dataset con datos académicos y sociodemográficos de 57,510 estudiantes, se evaluó el desempeño de múltiples algoritmos bajo dos esquemas de clasificación: binario (desertor vs. no desertor) y multiclase (desertor, en curso, graduado). Se optimizaron modelos como Random Forest y XGBoost, utilizando el F1-score como métrica principal para manejar el desbalance de clases. Los resultados muestran que los modelos de ensamblado alcanzan un rendimiento superior, con un F1-score de hasta 0.89 en la clasificación binaria. Si bien el rendimiento disminuye en el escenario multiclase (F1-score de 0.79), este enfoque permite una caracterización más granular de las trayectorias. Se concluye que los modelos predictivos, especialmente los binarios, son herramientas robustas para la detección temprana de estudiantes en riesgo, y que el desempeño académico inicial es el factor más determinante, lo que permite a las instituciones diseñar intervenciones focalizadas.

Introducción

Los modelos de aprendizaje automático pueden utilizarse en el ámbito educativo para predecir la deserción estudiantil, lo que permite mejorar la toma de decisiones informadas y diseñar políticas de retención que incrementen la permanencia de los estudiantes. Se realizarán pruebas con distintos algoritmos de machine learning para evaluar su eficacia. Asimismo, se implementarán técnicas de ensamblado a fin de mejorar el rendimiento predictivo. Como resultado se espera obtener modelos de predicción robustos que puedan ser utilizados a futuro, así como identificar y caracterizar los factores que contribuyen en la continuidad o interrupción del trayecto educativo universitario.

La deserción estudiantil universitaria representa un desafío crítico en sistemas educativos de todo el mundo, con tasas que alcanzan hasta el 50 % en países latinoamericanos UNESCO (2020). Diversos estudios han aplicado técnicas de machine learning para abordar este problema, reportando resultados prometedores con algoritmos como árboles de decisión, Random Forest y redes neuronales Kemper et al. (2020). Sin embargo, existe una brecha significativa en la literatura: pocos trabajos han comparado sistemáticamente el desempeño de múltiples algoritmos en contextos con desbalance de clases, aspecto crucial dado que los datasets educativos típicamente. Este análisis busca aportar al entendimiento de la predicción de deserción mediante la comparación rigurosa de múltiples algoritmos bajo dos esquemas de clasificación: binaria y multiclase. Esta aproximación metodológica permitirá determinar qué estrategia de modelado maximiza la detección temprana de estudiantes en riesgo, evaluar el impacto del desbalance natural de clases en el desempeño predictivo, e identificar los factores más influyentes en cada trayectoria educativa.

El presente trabajo busca explorar de qué manera los algoritmos de machine learning se desempeñan en la predicción de categorías de desenlace académico y cuáles son los factores determinantes que caracterizan las distintas trayectorias educativas. En este marco, se exploraron la deserción y la graduación de los estudiantes, que originalmente se presentan como clases desbalanceadas. Para abordar la problemática, se implementará un diseño experimental que incluye la evaluación de múltiples algoritmos de clasificación en dos escenarios: binaria (desertor vs. no desertor) y multiclase (desertor vs. en curso vs. graduado). Se plantean las siguientes hipótesis: (H1) los modelos de ensamblado superarán en rendimiento predictivo a los algoritmos individuales para la predicción de clases binarizadas; (H2) el desbalance natural de clases afectará de manera diferencial el desempeño de los algoritmos, con mejor rendimiento en la clase mayoritaria; (H3) las variables de desempeño académico tendrán mayor peso predictivo que las variables sociodemográficas. El documento se organiza en las siguientes secciones: Materiales y Métodos describe el tratamiento de los datos, variables explicativas y el manejo de etiquetas; Manejo de Datos y Esquemas de Clasificación detalla los modelos binarios y multiclase implementados con sus respectivos hiperparámetros; Evaluación de Modelos especifica las métricas de desempeño; Resultados reporta el rendimiento comparativo, curvas de complejidad e importancia de atributos; Discusión vincula los hallazgos con la literatura y sus implicancias prácticas; y Conclusión sintetiza los aportes del trabajo.

Materiales y métodos

Para el análisis se utilizaron los datos proporcionados por la cátedra en el campus de la materia. El dataset consta de 57.510 observaciones y 28 variables, que incluyen información sobre el trayecto educativo de cada estudiante, como carrera, turno de asistencia, cantidad de materias inscritas y aprobadas en el primer y segundo cuatrimestre, presencia de beca, presencia de deuda, puntaje en el examen de ingreso, número de evaluaciones y promedio de notas por cuatrimestre. También se incluyen datos sociodemográficos, tales como edad, estudios previos, sexo, estado civil, nivel educativo e ingresos de los padres. Cada observación está asociada a la etiqueta target: “desertor”, “en curso” o “graduado”. El dataset original presenta la siguiente distribución: 35.685 desertores (62 %), 14.087 estudiantes en curso (24 %) y 7.738 graduados (13 %), evidenciando un desbalance considerable entre clases. Según el tipo de análisis, las etiquetas se codificaron de manera distinta: para el análisis binario, los desertores se asignaron a 1 y las demás categorías a 0; para el análisis multiclase, los desertores se codificaron como 0, los estudiantes en curso como 1 y los graduados como 2. Como limitaciones del estudio, se identificaron posibles sesgos en los datos. El sesgo temporal implica que los datos corresponden a un período específico que podría no representar condiciones actuales o futuras, lo que hace que la explicación no sea generalizable a la comunidad estudiantil sino a este grupo en particular. El sesgo de censura surge porque los estudiantes clasificados como “en curso” representan trayectorias incompletas: aún no se conoce su desenlace final (graduación o deserción), lo que introduce incertidumbre en el modelado multiclase. Desde la perspectiva ética, los resultados deben interpretarse con cautela para evitar efectos de profecía autocumplida en intervenciones institucionales, donde etiquetar estudiantes como “en riesgo” podría afectar negativamente su motivación. Además, el dataset carece de información sobre el año en el cual se encuentra cada uno de los estudiantes, lo que presenta limitaciones a la hora de explicar la evolución temporal de la deserción en la comunidad estudiantil. Esta información ayudaría para pensar modelos generalizables que detecten características y comportamientos de los estudiantes en los diferentes estadios de sus estudios en la universidad que favorecen la deserción, lo que permitiría desarrollar estrategias para favorecer la permanencia y graduación.

Del dataset principal se tomó el 80 % de los datos para entrenamiento y el 20 % restante para prueba. La selección se realizó mediante el método `train_test_split` de `scikit-learn`, manteniendo la estratificación de clases para preservar las proporciones originales en ambos conjuntos. Previo al entrenamiento de los modelos, todas las variables numéricas fueron estandarizadas mediante StandardScaler, ajustando únicamente sobre el conjunto de entrenamiento para evitar filtración de información hacia el conjunto de prueba.

Manejo de datos y esquemas de clasificación

a) Clasificación binaria

Se realizó inicialmente un esquema de clasificación binaria, separando los datos en conjuntos de entrenamiento y prueba (train/test) con una proporción de 80/20. Para esto se utilizaron cuatro modelos de la librería sklearn: regresión logística, máquinas de soporte vectorial (SVM), K-Nearest Neighbors (KNN), Random Forest y XGBoost. Para el modelo de regresión logística se entrenaron dos versiones: una con el dataset original y otra aplicando SMOTE para balancear las clases.

Para cada modelo se definieron rangos de búsqueda de hiperparámetros que fueron explorados mediante validación cruzada. En ambos casos de **Regresión Logística**, se evaluaron los parámetros $solver \in \{\texttt{liblinear}, \texttt{lbfgs}, \texttt{saga}\}$, $C \in \{0{,}01, 0{,}1, 1, 10, 100\}$, $penalty \in \{\texttt{l2}\}$ y $class\_weight \in \{\texttt{None}, \texttt{balanced}\}$. Para **SVM**, se exploraron los parámetros $C \in \{0{,}1, 1, 5\}$, $kernel \in \{\texttt{linear}\}$ y $gamma \in \{\texttt{scale}, \texttt{auto}\}$. En el caso de **K-NN**, se evaluó el número de vecinos $n\_neighbors \in [3, 20]$, los pesos $weights \in \{\texttt{uniform}, \texttt{distance}\}$ y la métrica $metric \in \{\texttt{euclidean}, \texttt{manhattan}\}$. Para **Random Forest**, se utilizó `Optuna` para optimizar los hiperparámetros $n\_estimators \in [50, 300]$, $max\_depth \in [2, 20]$, $min\_samples\_split \in [2, 20]$, $min\_samples\_leaf \in [1, 10]$ y $max\_features \in \{\texttt{sqrt}, \texttt{log2}, \texttt{None}\}$, manteniendo fijo $class\_weight = \texttt{balanced}$. Finalmente, para **LightGBM** se empleó `RandomizedSearchCV`, explorando los parámetros $n\_estimators \in [100, 800]$, $max\_depth \in [3, 12]$, $learning\_rate \in [0{,}01, 0{,}2]$, $num\_leaves \in [20, 150]$, $min\_child\_samples \in [10, 100]$, $subsample \in [0{,}6, 1{,}0]$, $colsample\_bytree \in [0{,}6, 1{,}0]$, $reg\_alpha \in [0, 1]$ y $reg\_lambda \in [0, 1]$. La métrica empleada para la optimización de hiperparámetros fue el **F1-Score**, utilizada también como referencia principal para la evaluación final del rendimiento de los modelos.

Los mejores hiperparámetros encontrados para cada modelo se especifican en la Tabla 1:

Modelo	**Hiperparámetros**
Logistic Regression	{C: 1, class_weight: None, penalty: l2, solver: saga}
Logistic Regression con SMOTE	{C: 0.1, class_weight: None, penalty: l2, solver: saga}
SVM	{C: 1, gamma: scale, kernel: linear}
KNN	{metric: manhattan, n_neighbors: 20, weights: uniform}
Random Forest	{metric: manhattan, n_neighbors: 20, weights: uniform}
Boosting LightGBM	{colsample_bytree: 0.8428, learning_rate: 0.0652, max_depth: 3, min_child_samples: 98, n_estimators: 620, num_leaves: 107, reg_alpha: 0.3949, reg_lambda: 0.2935, subsample: 0.6056}

Tabla 1: Modelos e hiperparámetros óptimos obtenidos

Finalmente, se aplicó la técnica de `VotingClassifier` para integrar los modelos previamente entrenados, utilizando los mismos hiperparámetros seleccionados en la etapa de ajuste. De este modo, se combinaron sus predicciones con el objetivo de observar la robustez en la predicción del abandono, al promediar las salidas de los distintos algoritmos.

Clasificación multiclase

En segunda instancia se trabajó con un esquema de clasificación multiclase. Se utilizaron los mismos criterios de separación en *train* y *test*, y se entrenaron modelos de **K-NN**, **Random Forest Classifier**, **LightGBM** y **XGBoost**.

Para cada modelo se definieron rangos de búsqueda de hiperparámetros que fueron explorados mediante validación cruzada. En el caso de **K-NN**, se evaluó el número de vecinos $n\_neighbors \in [3, 21]$, los pesos $weights \in \{\texttt{uniform}, \texttt{distance}\}$, la métrica de distancia $metric \in \{\texttt{euclidean}, \texttt{manhattan}, \texttt{minkowski}\}$ y el algoritmo correspondiente. Para **Random Forest**, se optimizaron los hiperparámetros $n\_estimators \in [50, 300]$, $max\_depth \in [2, 20]$, $min\_samples\_split \in [2, 20]$, $min\_samples\_leaf \in [1, 10]$ y $max\_features \in \{\texttt{sqrt}, \texttt{log2}, \texttt{None}\}$. Finalmente, para **LightGBM** se empleó `RandomizedSearchCV`, explorando los parámetros $n\_estimators \in [100, 800]$, $max\_depth \in [3, 12]$, $learning\_rate \in$

$[0{,}01, 0{,}2]$, $num\_leaves \in [20, 150]$, $min\_child\_samples \in [10, 50]$, $subsample \in [0{,}4, 0{,}6]$, $colsample\_bytree \in [0{,}4, 0{,}6]$, $reg\_alpha \in [0, 1]$ y $reg\_lambda \in [0, 1]$. Se mantuvieron fijos los hiperparámetros `objective = multiclass`, `class_weight = balanced` y `num_class = 3`, para asegurar un entrenamiento adecuado considerando el desbalance de clases. La métrica empleada para la optimización de hiperparámetros fue el **F1-Score**, dado que también se utilizó como métrica principal para la evaluación final del rendimiento de los modelos.

Los mejores hiperparámetros encontrados para cada modelo se especifican en la Tabla 2:

Modelo	**Hiperparámetros**
KNN multiclase	{classifier_metric: manhattan, classifier_n_neighbors: 14, classifier_weights: distance}
Random Forest multiclase	{classifier_class_weight: balanced, classifier_max_depth: 20, classifier_min_samples_split: 5, classifier_n_estimators: 200}
Boosting LightGBM multiclase	{classifier_colsample_bytree: np.float64(0.7219125032632117), classifier_learning_rate: np.float64(0.04293117062858835), classifier_max_depth: 11, classifier_min_child_samples: 46, classifier_n_estimators: 324, classifier_num_leaves: 70, classifier_reg_alpha: np.float64(0.2694123337985215), classifier_reg_lambda: np.float64(0.24412552224777417), classifier_subsample: np.float64(0.6673164168691722)}
Boosting XGBoost multiclase	{classifier_colsample_bytree: np.float64(0.821105986734196), classifier_learning_rate: np.float64(0.12445849383416767), classifier_max_depth: 3, classifier_min_child_weight: 4, classifier_n_estimators: 290, classifier_reg_alpha: np.float64(0.30569701928718185), classifier_reg_lambda: np.float64(0.19091103115034602), classifier_subsample: np.float64(0.7073899427560627)}

Tabla 2: Modelos e hiperparámetros óptimos obtenidos

Evaluación de modelos

Para evaluar el rendimiento de los modelos se consideraron distintas métricas en función del tipo de análisis.

Para el análisis binario, se le dio importancia a la métrica `accuracy`, que representa la proporción de predicciones correctas sobre el total de observaciones.

Dado que para el caso multiclase el `accuracy` global puede resultar engañoso ante un fuerte desbalance, dado que no refleja el desempeño específico en cada categoría, se utilizó el `F1-score`, particularmente adecuado en contextos con clases desbalanceadas como el de este dataset (62 % desertores, 24 % en curso y 13 % graduados en la clasificación multiclase).

Finalmente, para ambos esquemas de clasificación se construyeron matrices de confusión, que facilitaron una inspección detallada de los aciertos y errores de clasificación en cada clase.

Resultados

En esta sección se presentan los hallazgos principales del estudio, comenzando por el rendimiento de los modelos en las tareas de clasificación binaria y multiclase, y finalizando con un análisis de los factores más influyentes en la predicción de la deserción.

Clasificación Binaria

Una vez identificados los mejores hiperparámetros para cada modelo, se evaluaron siete algoritmos de machine learning para la predicción binaria de deserción estudiantil (desertor vs. no desertor). La Tabla 3 resume el desempeño de cada modelo en el conjunto de prueba.

Los modelos de ensamblado, específicamente XGBoost y Random Forest, junto con el 'Voting-Classifier', demostraron el rendimiento más alto para la tarea objetivo (T) de clasificación binaria. Al analizar la totalidad de las métricas, XGBoost se posicionó como el modelo más performante, alcanzando un F1-score (P) de 0.8932, seguido de cerca por Random Forest (P = 0.8915) y el ensamble Voting (P = 0.8913). La paridad en el rendimiento de estos tres modelos sugiere que, si bien los ensambles son claramente superiores a los modelos individuales como KNN o Regresión Logística, la ganancia adicional de combinar múltiples ensambles en un 'VotingClassifier' es marginal en este contexto.

Tabla 3: Desempeño de modelos en clasificación binaria

Modelo	Accuracy	F1-Score	Precision	Recall	ROC-AUC	CV F1 Mean	CV F1 Std
Logistic Regression	0.8574	0.8885	0.8633	0.9151	0.9200	0.8585	0.0032
Logistic Regression con SMOTE	0.8570	0.8871	0.8697	0.9051	0.9189	0.8704	0.0368
SVM	0.8584	0.8895	0.8622	0.9186	0.9192	0.8575	0.0012
KNN	0.8462	0.8812	0.8461	0.9193	0.9042	0.8434	0.0025
Random Forest	0.8614	0.8915	0.8669	0.9175	0.9233	0.8585	0.0036
XGBoost	0.8638	0.8932	0.8693	0.9186	0.9273	0.8601	0.0041
Voting	0.8619	0.8913	0.8710	0.9126	0.8082		

Clasificación Multiclase

Posteriormente, se evaluaron cuatro algoritmos para la tarea de clasificación multiclase (T), que distingue entre las trayectorias "desertor", .^{en} cursoz "graduado". La Tabla 4 resume el desempeño de cada modelo en el conjunto de prueba.

En este escenario más complejo, el modelo XGBoost muestra ser el de mejor desempeño, obteniendo un F1-score ponderado (P) de 0.7876 y un área bajo la curva ROC ponderada (P) de 0.9070. Si bien se observa que las métricas de XGBoost son superiores, la diferencia de rendimiento entre los modelos de ensamblado (Random Forest, LightGBM y XGBoost) es menos pronunciada que en la clasificación binaria. Este resultado indica la dificultad inherente de la tarea multiclase, donde el desbalance entre las categorías .^{en} cursoz "graduadorepresenta un desafío para los algoritmos.

Tabla 4: Desempeño de modelos en clasificación multiclase

Modelo	Accuracy	F1-Score (weighted)	Precision (weighted)	Recall (weighted)	ROC-AUC (ponderada)	CV F1 Mean	CV F1 Std
KNN multiclase	0.7709	0.7522	0.7643	0.7709	0.8754	0.7457	0.0025
Random Forest multiclase	0.7843	0.7803	0.7837	0.7843	0.9004	0.7804	0.0028
LightGBM multiclase	0.7712	0.7743	0.7790	0.7712	0.9021	0.7740	0.0051
XGBoost multiclase	0.7970	0.7876	0.7907	0.7970	0.9070	0.7855	0.0037

El análisis de las curvas ROC del modelo XGBoost se realizó mediante la técnica *One vs. Rest (OvR)*, que evalúa cada categoría en contraposición al conjunto de las demás. Este enfoque transforma el problema multiclase en una serie de comparaciones binarias, permitiendo analizar la capacidad discriminante del modelo de forma individual para cada clase.

Los valores del área bajo la curva (AUC) obtenidos fueron 0.92 para la clase desertor, 0.87 para la clase graduado y 0.86 para la clase en curso. Tal como se observa en la figura 1, el modelo presenta su mejor desempeño al identificar desertores, lo que indica una alta capacidad para distinguirlos del resto de los estudiantes. En cambio, la capacidad de diferenciación es algo menor para las clases graduado y en curso, lo que sugiere una mayor dificultad debido al desbalance de las clases originales.

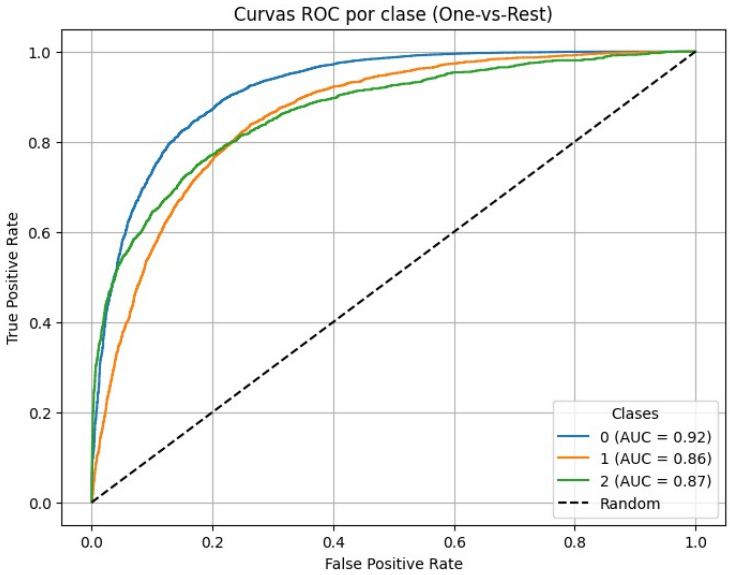

Figura 1: Curva roc-auc para cada clase en modelo XGBoost

Importancia de Atributos

Para identificar los factores más influyentes en la predicción de la deserción, se analizó la importancia de atributos de los modelos. A modo de ejemplo incluimos a La Figura 2 muestra los 50 atributos más relevantes para Random Forest clasificación binaria.

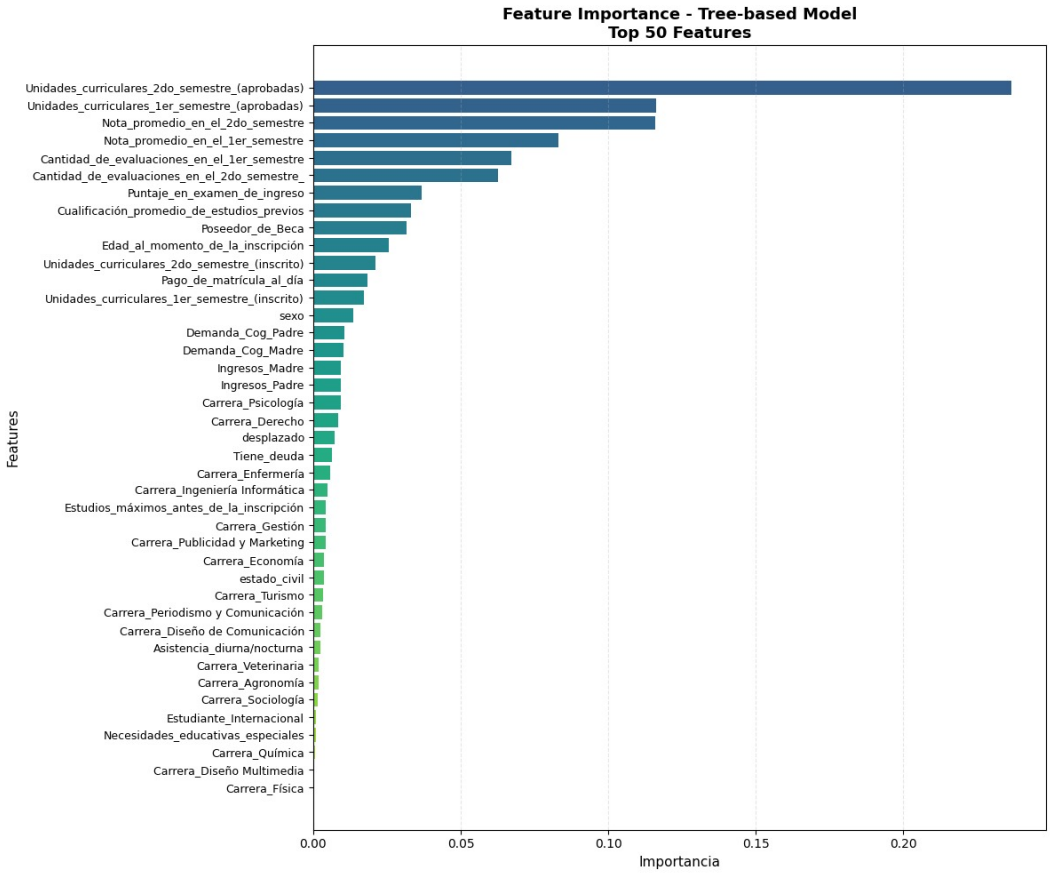

Figura 2: Feature Importance de Random Forest para clasificación binaria

La Figura 2 muestra que las variables de mayor peso predictivo están directamente relacionadas con el desempeño académico del estudiante. Los atributos $Unidades\_curriculares\_2do\_semestre\_(aprobadas)$, $Unidades\_curriculares\_1er\_semestre\_(aprobadas)$, $Nota\_promedio\_en\_el\_2do\_semestre$ y $Nota\_promedio\_en\_el\_1er\_semestre$ dominan la predicción. En un segundo nivel de importancia se encuentran factores de entrada como el $Puntaje\_en\_examen\_de\_ingreso$ y variables administrativas como $Poseedor\_de\_Beca$ y $Pago\_de\_matrícula\_al\_día$. Las variables sociodemográficas, como el sexo, los ingresos o el nivel educativo de los padres ($Demanda\_Cog\_Padre$, $Ingresos\_Madre$), muestran una influencia considerablemente menor en el desempeño predictivo.

Discusión

Los resultados de este estudio demuestran la capacidad predictiva de los modelos de aprendizaje automático para identificar la deserción estudiantil, respondiendo así a la pregunta inicial sobre cómo estos algoritmos se desempeñan en la predicción de desenlaces académicos. Se encontró que los modelos de ensamblado, particularmente XGBoost y Random Forest, alcanzan un rendimiento levemente superior en la clasificación binaria (desertor vs. no desertor), con un F1-score cercano a 0.89. Si bien el desempeño disminuye en el escenario multiclase (desertor vs. en curso vs. graduado), el modelo XGBoost mantiene una capacidad predictiva robusta (F1-score ponderado de 0.79), evidenciando la complejidad adicional que implica distinguir entre trayectorias académicas diversas.

Estos hallazgos permiten abordar las hipótesis y los vacíos de conocimiento planteados en la introducción. En primer lugar, se refuta parcialmente la hipótesis H1, que postulaba que los modelos de ensamblado como ‘VotingClassifier‘ superarían a los algoritmos individuales. Los resultados muestran que el ‘VotingClassifier‘ no mejoró significativamente el rendimiento de los mejores modelos

base que lo componían (XGBoost y Random Forest). Esto sugiere que, cuando se combinan algoritmos de ensamblado ya de por sí potentes, el beneficio marginal de una capa adicional de votación es mínimo, lo que lleva a preferir los modelos individuales por su menor complejidad computacional y mayor interpretabilidad. En próximos estudios quedaría pendiente indagar acerca de la significatividad de esta diferencia para evaluar cuales son mejores.

En segundo lugar, los resultados confirman la hipótesis H2: el desbalance de clases afecta diferencialmente el desempeño de los algoritmos. La brecha de rendimiento entre la clasificación binaria y la multiclase es una manifestación de este fenómeno. Mientras que el modelo binario predice con eficacia la clase mayoritaria ("desertor"), el modelo multiclase encuentra mayores dificultades para diferenciar las clases minoritarias ("en curso" y "graduado"). Este hallazgo es consistente con la literatura que advierte sobre los desafíos del desbalance en datasets educativos Kemper et al. (2020), pero nuestro aporte radica en la cuantificación directa de este impacto mediante la comparación de esquemas de clasificación. Si el objetivo institucional es una intervención temprana y general, un modelo binario es suficiente y más preciso. Sin embargo, si se busca diseñar estrategias diferenciadas para retener estudiantes en "curso" y confirmar la graduación, un modelo multiclase, aunque menos preciso globalmente, ofrece una granularidad necesaria.

Finalmente, los modelos de árboles como Random Forest y XGBoost consistentemente señalan que las variables de desempeño académico del primer y segundo semestre (materias aprobadas, promedio) son los predictores más influyentes, validando la hipótesis H3, como se observa en la figura 1. El éxito temprano es más determinante que las condiciones sociodemográficas de entrada. En la práctica, esto justifica la implementación de sistemas de alerta temprana y programas de apoyo académico focalizados en los primeros semestres, ya que es el período de mayor impacto para prevenir la deserción.

El aporte más significativo de este trabajo es la comparación sistemática del rendimiento de múltiples algoritmos bajo esquemas de clasificación binaria y multiclase, demostrando el trade-off entre la precisión predictiva y la riqueza informativa. La elección entre un modelo u otro dependerá del objetivo estratégico. No obstante, el estudio presenta limitaciones. El sesgo inherente a la clase "en curso", introduce incertidumbre sobre el desenlace final de estos estudiantes, afectando la validez a largo plazo del modelo multiclase. Asimismo, la ausencia de variables temporales, como el año de cursada, impide modelar la evolución del riesgo de deserción a lo largo de la carrera, una dimensión clave para intervenciones más dinámicas y personalizadas.

Conclusión

En base a nuestros resultados, podemos concluir que los modelos de ensamblado de árboles como XGBoost y Random Forest permiten resolver la tarea de predicción de la deserción estudiantil a partir de datos de desempeño académicos y sociodemográficos. Estos modelos tienen mejores resultados predictivos en el esquema de clasificación binaria, y demuestran que las variables de desempeño académico son los predictores más influyentes. El estudio establece que, si bien los modelos multiclase ofrecen una visión más granular de las trayectorias, los modelos binarios constituyen una herramienta más robusta y precisa para la identificación temprana de estudiantes en riesgo de deserción, facilitando así el diseño de intervenciones institucionales para mejorar la retención.

Bibliografía

Kemper, L., G. Vorhoff, y B. U. Wigger
2020. Predicting student dropout: A machine learning approach. *European Journal of Higher Education*, 10(1):28–47.

UNESCO
2020. *Inclusión y educación: Todos y todas sin excepción*. Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura.