UNIVERSIDAD NACIONAL DE CAAGUAZÚ

FACULTAD DE CIENCIAS Y TECNOLOGÍAS

CARRERA DE INGENIERÍA EN INFORMÁTICA

PROYECTO FINAL DE GRADO

Modelo Analítico para el Estudio de la Retención Estudiantil en la Facultad de Ciencias y Tecnologías de la Universidad Nacional del Caaguazú

AUTOR

Fabrizio Benjamin Villar Ferreira

TUTOR:

Prof. Ing. Víctor Manuel Melgarejo Riveros

CORONEL OVIEDO, DICIEMBRE DE 2025

DERECHO DE AUTOR

Quien suscribe, Fabrizio Benjamin Villar Ferreira, autor del trabajo de investigación titulado **"Modelo Analítico para el Estudio de la Retención Estudiantil en la Facultad de Ciencias y Tecnologías (FCyT–UNCA)"**, declara que voluntariamente cede a título gratuito en forma pura y simple ilimitada e irrevocablemente a favor de la Facultad de Ciencias y Tecnologías – UNCA, el derecho de autor de contenido patrimonial, que le corresponde sobre el trabajo de referencia. Conforme a lo anteriormente expresado, esta sesión le otorga a la FCyT la Facultad de comunicar la obra divulgarla, publicarla y reproducirla en soportes analógicos o digitales en la oportunidad que así lo estime conveniente. La FCyT deberá indicar qué autoría o creación del trabajo corresponde a mi persona y hará referencia al autor y a las personas que hayan colaborado en la realización del presente trabajo de investigación.

En la ciudad de Coronel Oviedo a los …. , del mes de ….. del 2025

…….

Fabrizio Benjamin Villar Ferreira

PÁGINA DE APROBACIÓN

Trabajo de fin de grado para la obtención del Título de Ingeniero Informático, aprobado en representación de la Facultad Ciencias y Tecnología de la Universidad Nacional de Caaguazú, por el Tribunal Examinador constituido por los siguientes profesores y con la siguiente nota final:

CALIFICACIÓN FINAL: _____

ACTA N°:_____

FECHA :_____

Prof. Ing.

Prof. Ing.

Prof. Ing.

DEDICATORIA

A Dios Todopoderoso, fuente de resiliencia y sabiduría, quien ha iluminado mi camino en los momentos de duda y dificultad, otorgándome serenidad, paciencia e ingenio para avanzar con fe y esperanza.

A mis padres, Sra. Norma que supo ser base de ejemplo de responsabilidad y dedicación para aplicar durante el curso de mi carrera, Sr. Fernando quién a su vez supo sentar en mí el valor por el esfuerzo y serenidad. Ambos dedicando sus días a ser sinónimo de paciencia y amor, sin ustedes este momento no sería posible.

A mi hermana, que con sus risas y cariño en las noches de estudio se convirtió en oxígeno necesario en momentos tensos, con su apoyo logré apuntar de forma más precisa al objetivo. También a mi abuela que durante incontables días con su cariño y apoyo formó parte del impulso para cumplir con este desafío.

A Luz, Carlos y Roberto; mis compañeros que fueron parte de los días, y de las noches, de los sacrificios y risas que afrontamos durante estos años. Unidos todos en lograr el mismo objetivo, gracias por sus enseñanzas y paciencia. Para Óscar Pizzurno (+), quién en el inicio fue parte de una promesa que hoy se está cumpliendo.

A mis docentes y tutores académicos, por su apoyo y guía. Me queda agradecer por haber compartido conmigo no sólo conocimiento académico, experiencias que fueron enriqueciendo mi aprendizaje y mi forma de observar el mundo. Todas estas experiencias compartidas propias y con ustedes serán de enorme ayuda para encarar las próximas etapas que vendrán.

Y para concluir, dedico este trabajo a todos quienes fueron parte de este proceso, quienes de forma directa o indirecta estuvieron presentes, con un consejo, una sonrisa o un mensaje de apoyo. A quienes fueron amigos y se convirtieron en familia. A ustedes, más que agradecimiento mi eterna gratitud.

AGRADECIMIENTOS

A Dios por haberme dado salud y sabiduría a lo largo de los años, fortaleciendo mi fe, siendo una constante en los momentos gratos y en los momentos de incertidumbre.

A mis familiares que desde el primer instante estuvieron firmes a mi lado, siendo un apoyo imprescindible para alcanzar esta meta.

A mis compañeros, por su ayuda, paciencia y amistad. Gracias por haber sido parte de las aventuras y desafíos que se presentaron durante estos años, convirtiéndose en una pieza clave de esta experiencia.

A mis profesores y amigos del sector docente de la Facultad de Ciencias y Tecnologías, por su profesionalismo, entrega y por asistir de forma integral mi formación académica y personal.

A mi tutor, el Ingeniero Víctor Melgarejo, por su invaluable guía, dedicación y paciencia durante el desarrollo de este trabajo. Sus enseñanzas y su apoyo constante fueron cruciales para alcanzar esta meta.

A la Facultad de Ciencias y Tecnologías de la Universidad Nacional de Caaguazú, mi institución, por brindarme la oportunidad de crecer profesionalmente y por ser el entorno propicio para mi desarrollo académico.

RESUMEN

El presente estudio aborda la retención estudiantil, enmarcándose en los Objetivos de Desarrollo Sostenible (ODS), particularmente el ODS 4 (Educación de Calidad), ODS 9 (Industria, Innovación e Infraestructura) y ODS 10 (Reducción de las Desigualdades), con el propósito de fortalecer la permanencia formativa en la Facultad de Ciencias y Tecnologías de la Universidad Nacional de Caaguazú, mediante el análisis de registros académicos históricos y la aplicación de modelos computacionales basados en aprendizaje automático orientados a modernizar la gestión académica institucional.

A partir de un dataset longitudinal de 75.937 registros, se construyeron indicadores académicos y temporales que permitieron caracterizar la trayectoria educativa de 1.422 estudiantes, promoviendo trayectorias más equitativas y sostenibles.

Se evaluaron distintos algoritmos de clasificación utilizando métricas estándar de desempeño, destacándose el modelo Random Forest por su mayor capacidad para identificar estudiantes en riesgo de pérdida de continuidad académica, facilitando la implementación de estrategias de intervención temprana.

El análisis de importancia de variables evidenció que los factores temporales, en particular la inactividad académica acumulada, constituyen los principales determinantes de la retención, superando al rendimiento académico como factor explicativo.

En conjunto, el estudio propone un modelo operativo de apoyo a la gestión académica que contribuye a fortalecer estrategias institucionales orientadas a sostener la continuidad formativa y el bienestar estudiantil.

Palabras clave: retención estudiantil, aprendizaje automático, Random Forest, minería de datos educativa, gestión académica.

ABSTRACT

The present study addresses student retention, framed within the Sustainable Development Goals (SDGs), particularly SDG 4 (Quality Education), SDG 9 (Industry, Innovation and Infrastructure), and SDG 10 (Reduced Inequalities), with the aim of strengthening academic persistence at the Faculty of Sciences and Technologies of the National University of Caaguazú, through the analysis of historical academic records and the application of computational models based on machine learning oriented toward the modernization of institutional academic management.

Based on a longitudinal dataset of 75,937 records, academic and temporal indicators were constructed to characterize the educational trajectories of 1,422 students, promoting more equitable and sustainable academic pathways.

Different classification algorithms were evaluated using standard performance metrics, with the Random Forest model standing out for its greater capacity to identify students at risk of losing academic continuity, thus facilitating the implementation of early intervention strategies.

The analysis of variable importance showed that temporal factors, particularly accumulated academic inactivity, constitute the main determinants of retention, surpassing academic performance as an explanatory factor.

Overall, the study proposes an operational model to support academic management, contributing to the strengthening of institutional strategies aimed at sustaining academic continuity and student well-being.

Keywords: student retention, machine learning, Random Forest, educational data mining, academic management.

INDICE

DERECHO DE AUTOR 2

PÁGINA DE APROBACIÓN 3

DEDICATORIA 4

AGRADECIMIENTOS 5

RESUMEN 6

ABSTRACT 7

INDICE 8

ÍNDICE DE FIGURAS 10

ÍNDICE DE TABLAS 11

Modelo Analítico para el Estudio de la Retención Estudiantil en la Facultad de Ciencias y Tecnologías (FCyT–UNCA) 1

1. 14

1.1 Agrupamiento de datos y determinación de estructuras internas 4

1.1.1 K-Means 4

1.1.2 Método del codo 4

1.2 Modelos empleados para clasificación 5

1.2.1. Máquina de Soporte Vectorial (SVM) 5

1.3 19

1.3.1 Exactitud o *Accuracy* 8

1.3.2 Matriz de Confusión 9

1.3.3 Recall 9

1.3.4. Precisión 9

1.3.5. F1 Score 10

2 22

Objetivo General 11

Objetivos Específicos 11

3 23

3.1 23

3.2 Contenido del Archivo Institucional 12

Variables identificadoras 12

Variables personales 12

Variables académicas 13

3.3 Necesidad de transformación del Dataset 13

Agrupación por estudiante 13

Conversión de formatos 14

3.4 Transformación Final de los Datos 14

3.5 Determinación de la Importancia de las Variables 15

3.5.1 Fundamento del Método 15

3.5.2 Métrica de Importancia de Variables 16

3.5.3 Implementación computacional 16

3.5.4 Relevancia para el modelo de retención 17

4 28

4.1 Interpretación de los Resultados 17

4.2 Descripción del conjunto de datos 17

4.3 Rendimiento de los Modelos Predictivos 18

4.4 Interpretación General 19

4.5 Importancia de las Variables 19

4.6 Análisis interpretativo 20

5 32

6. GLOSARIO DE TÉRMINOS 21

7. BIBLIOGRAFÍA 24

8 ANEXOS 26

Script de Procesamiento 26

Script de Entrenamiento 34

Script de Evaluación 37

ÍNDICE DE FIGURAS E ILUSTRACIONES

Gráfico: 1 Diagrama de Funcionamiento de K-Means 4
Gráfico: 2 Método del Codo 5
Gráfico: 3 Regresión Logística 6
Gráfico: 4 KNN (K - Vecinos más Cercanos) 8

Ilustración I: Máquina de Soporte Vectorial (SVM) 6
Ilustración II Representación de Random Forest 7
Ilustración III: Árbol de Decisiones 7
Ilustración IV: Matriz de Confusión 9

ÍNDICE DE TABLAS

Tabla 1: Matriz Confusión 18
Tabla 2: Importancia de las Variables 19

Modelo Analítico para el Estudio de la Retención Estudiantil en la Facultad de Ciencias y Tecnologías (FCyT–UNCA)

La retención estudiantil constituye un componente decisivo para la calidad del sistema educativo, puesto que impacta directamente en la formación de capital humano y en la sostenibilidad del desarrollo académico y social del país [1]. Este proyecto surge con el propósito de comprender los factores que favorecen la continuidad educativa y de aprovechar herramientas modernas de análisis de datos y técnicas de Machine Learning para identificar tendencias que permitan fortalecer dicha permanencia estudiantil.

En Paraguay, la baja tasa de culminación de estudios universitarios pone en evidencia dificultades sistémicas asociadas a la trayectoria formativa de los estudiantes. Aproximadamente solo alrededor de un 10 % de los jóvenes que ingresan a instituciones de educación superior consiguen finalizar su carrera, lo que implica que la gran mayoría abandona sus estudios en algún punto del recorrido académico [1]. Este escenario no solo refleja desafíos individuales, sino también la necesidad de estrategias institucionales para apoyar la continuidad educativa.

En estudios previos realizados en instituciones nacionales como en la Universidad Católica Nuestra Señora de la Asunción [1], se han analizado patrones de permanencia y abandono entre estudiantes de distintas edades, modalidades laborales y tipos de universidades. Dichos resultados mostraron perfiles diferenciados de continuidad académica, donde una parte de los estudiantes mantiene un cursado regular, otra avanza con discontinuidad y un grupo significativo ya no continúa estudiando. Estos datos permiten dimensionar que la retención no puede explicarse por un único factor, sino por la interacción de variables académicas, personales y socioeconómicas.

En el ámbito de la Facultad de Ciencias y Tecnologías, la retención estudiantil constituye un aspecto sensible dentro de la gestión académica, evidenciado por trayectorias formativas discontinuas y dificultades en la permanencia a lo largo del recorrido universitario. Esta situación pone de manifiesto la necesidad de fortalecer los mecanismos de seguimiento y análisis de la continuidad académica, de manera que sea posible comprender con mayor profundidad los patrones de permanencia, transferencia y avance curricular, contribuyendo así a mejorar el rendimiento interno de las carreras y a optimizar el aprovechamiento del trayecto formativo de los estudiantes. Desde una

perspectiva técnica, la retención estudiantil se puede definir como la continuidad del estudiante dentro de su trayectoria académica desde el momento de ingreso hasta la finalización de sus estudios, sin interrupciones prolongadas que indiquen una desvinculación definitiva del sistema educativo [2]. En este trabajo se operacionaliza esta definición mediante una regla de negocio basada en la actividad académica registrada en los sistemas institucionales: un estudiante se considera retenido si registra actividad académica en un período no mayor a dos años consecutivos desde la última fecha de participación en el proceso académico (inscripción a materias, rendición de exámenes o registro de cursado). Por el contrario, si un estudiante no presenta actividad académica durante dos años consecutivos o más, se considera que ha abandonado sus estudios. Esta definición permite distinguir claramente entre estudiantes que se retrasan temporalmente, pero continúan activos y aquellos que realmente abandonan su formación formal.

La aplicación de modelos predictivos basados en Machine Learning se plantea como un recurso estratégico para anticipar situaciones que indiquen riesgo de abandono, permitiendo derivar alertas tempranas que orienten intervenciones de acompañamiento académico antes de que la deserción ocurra [2]. En lugar de centrarse únicamente en el abandono, la presente investigación enfoca el problema desde la perspectiva de retención: identificar qué factores ayudan a que un estudiante permanezca, progrese y complete sus estudios.

La literatura internacional también presenta aproximaciones similares. Por ejemplo, estudios realizados en universidades mexicanas han demostrado que algoritmos como árboles de decisión, k-vecinos (KNN) y técnicas de clasificación supervisada pueden utilizarse para analizar el comportamiento de datos estudiantiles y detectar patrones asociados a persistencia, continuidad académica o abandono [3]. Estos antecedentes resaltan el potencial de la minería de datos educativa para transformar registros académicos estáticos en conocimiento accionable que pueda reforzar la retención.

En suma, el modelo predictivo desarrollado en este proyecto busca otorgar a la institución una herramienta avanzada para comprender la trayectoria de sus estudiantes y contribuir a la construcción de un entorno académico que favorezca su permanencia y culminación exitosa de sus estudios.

1. MARCO TEÓRICO

Para emprender este proyecto fue fundamental identificar los componentes centrales de análisis y los enfoques metodológicos que lo sustentan. La propuesta se apoya en técnicas de ciencia de datos aplicadas a registros académicos con el propósito de caracterizar el comportamiento estudiantil y derivar patrones predictivos. La ciencia de datos, entendida como la disciplina que emplea métodos estadísticos [3], algoritmos y herramientas tecnológicas para extraer conocimiento útil a partir de grandes volúmenes de información, permite transformar datos brutos en insumos analíticos relevantes para la toma de decisiones institucionales.

En este sentido, uno de los primeros procedimientos consistió en la recolección estructurada de datos pertinentes para el fenómeno estudiado. Dichos datos abarcan variables de carácter académico, administrativo, como año de ingreso, trayectoria curricular, calificaciones obtenidas, programa académico cursado y género. La información será obtenida a partir de los repositorios institucionales en formatos como Excel o CSV y validada complementariamente mediante consulta con responsables académicos, a fin de asegurar consistencia y calidad.

Una vez reunida la información, se aplicaron procesos rigurosos de limpieza y normalización de datos, incluyendo la detección de inconsistencias, el tratamiento de valores faltantes y la estandarización de formatos [3]. Posteriormente, mediante análisis exploratorio, se buscó reconocer estructuras internas, comportamientos históricos y relaciones entre variables, lo que constituye un paso esencial antes de proceder al modelado predictivo.

Este abordaje sistemático permite desarrollar un modelo analítico robusto para analizar patrones de permanencia y detectar situaciones de riesgo, proporcionando información relevante que puede ser utilizada como insumo para la toma de decisiones y el diseño de estrategias institucionales de apoyo académico.

1.1 Agrupamiento de datos y determinación de estructuras internas

1.1.1 K-Means

K-Means es uno de los métodos de agrupamiento no supervisado más empleados para segmentar datos en un conjunto finito de grupos definidos por el parámetro K, como se puede apreciar en el gráfico 1, el procedimiento parte de seleccionar k puntos iniciales que actuarán como centroides. A partir de ellos, cada registro se asocia al centroide más próximo, originando una primera división del conjunto de datos [4]. Posteriormente, se recalculan los centroides considerando la media de los puntos asignados, volviendo a ejecutar el proceso de asignación y recalibración hasta alcanzar estabilidad (esto es, cuando los centroides dejan de variar significativamente).
El propósito del algoritmo es minimizar la dispersión interna de cada grupo mediante la reducción del error cuadrático promedio, promoviendo que cada cluster sea lo más homogéneo posible. [4]

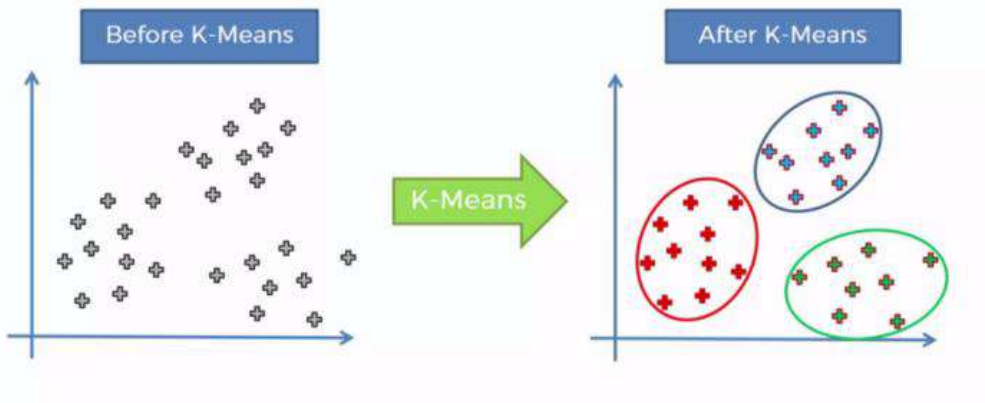

Gráfico: 1 Diagrama de Funcionamiento de K-Means

1.1.2 Método del codo

Para determinar cuántos grupos resultan adecuados para la partición, se emplea el conocido método del codo. La idea central consiste en entrenar repetidamente el algoritmo variando el número de clusters e inspeccionando cómo disminuye la suma de los errores cuadráticos internos.

Los resultados se representan graficando el número de grupos (eje X) contra la suma de distancias al cuadrado de cada elemento a su centroide (eje Y) de la misma forma que se representa en el gráfico 2. En general, la reducción del error es significativa solo hasta cierto punto; cuando esa disminución deja de ser pronunciada y forma una curvatura visible similar a un codo se interpreta dicho punto como el número más apropiado de clusters, ya que incrementos adicionales en k no generan mejoras relevantes en la compactación de los grupos. [5]

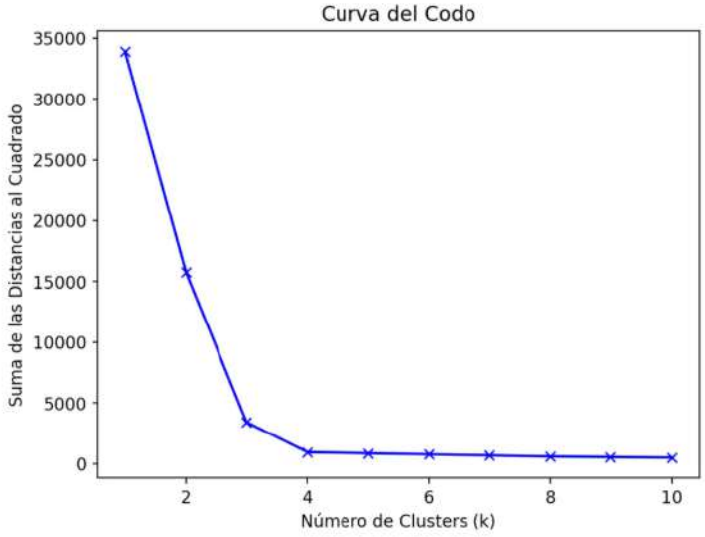

Gráfico: 2 Método del Codo

1.2 Modelos empleados para clasificación

Una vez identificadas las estructuras internas del conjunto de datos, se ajustaron diversos modelos de aprendizaje supervisado con el objetivo de estimar la categoría o clase de salida. Machine Learning se define como el campo que estudia algoritmos capaces de aprender patrones a partir de ejemplos, sin requerir instrucciones explícitas para cada posible caso.

1.2.1. Máquina de Soporte Vectorial (SVM)

Este enfoque construye un hiperplano que maximiza la separación entre clases. Cuando los datos no son separables de forma lineal en el espacio original, se aplican funciones kernel que permiten representarlos en un espacio dimensional ampliado donde la separación sea más factible. Así como representa la Ilustración I, la clasificación se

realiza mediante la ubicación del nuevo dato respecto a ese hiperplano decisorio. [6]

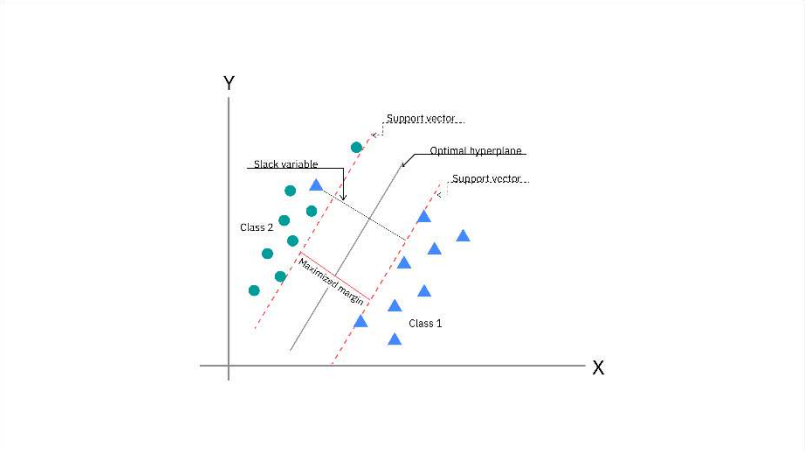

Ilustración I: Máquina de Soporte Vectorial (SVM)

1.2.2. Regresión Logística (RL)

La regresión logística estima la probabilidad de pertenecer a una categoría dada basándose en variables de entrada. Su fundamento consiste en ajustar una función sigmoide que transforma valores continuos en probabilidades entre 0 y 1 como se muestra en el gráfico 3. Es particularmente efectiva en problemas de clasificación binaria donde se requiere una interpretación probabilística. [6]

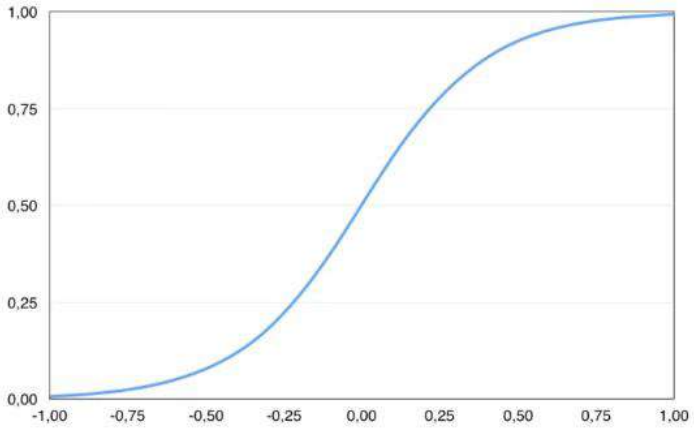

Gráfico: 3 Regresión Logística

1.2.3. Random Forest (RF)

Como se puede apreciar en la Ilustración II. Random Forest combina múltiples árboles de decisión generados con distintos subconjuntos de datos y/o variables, produciendo un modelo agregado que reduce la varianza y mejora la generalización. Como se referencia en la Ilustración II, El resultado final se obtiene mediante votación mayoritaria de los árboles en el caso de clasificación o promedio en regresión.

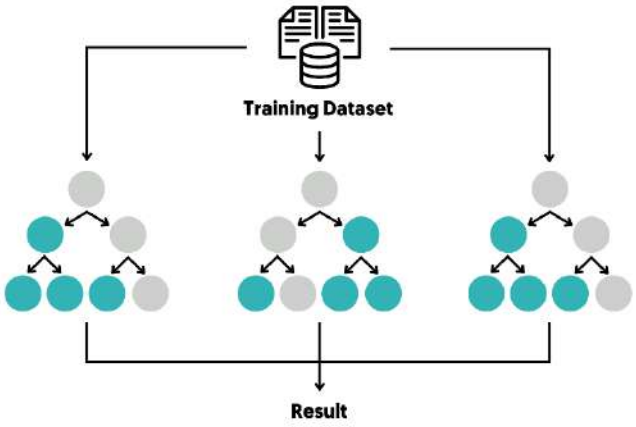

Ilustración II Representación de Random Forest

1.2.4. Árbol de Decisiones (DT)

Los árboles de decisión funcionan como una secuencia de condiciones jerárquicas que conducen a una clasificación final. Cada nodo interno representa una característica evaluada como se puede apreciar en la Ilustración III, las ramas corresponden a los resultados de la evaluación y las hojas contienen las decisiones finales. Este tipo de modelo es valorado por su interpretabilidad, ya que puede traducirse en reglas explícitas [7].

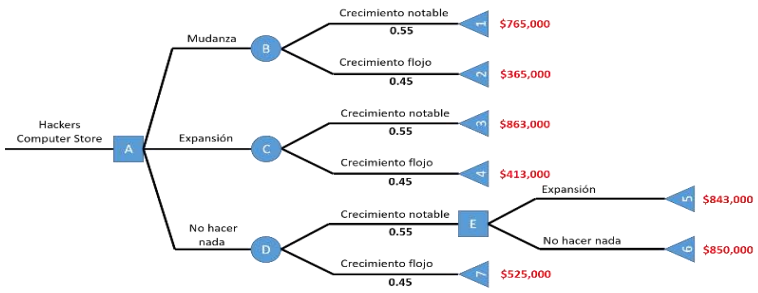

Ilustración III: Árbol de Decisiones

1.2.5. K Vecinos más Cercanos (KNN)

Este método clasifica un punto analizando las etiquetas de los **k** registros más próximos en el espacio de características. No presupone un modelo paramétrico ni una estructura de datos específica, por lo que resulta útil cuando la relación entre variables es desconocida o altamente compleja. La clase del nuevo dato depende de la mayoría de sus vecinos inmediatos. [6]

Gráfico: 4 KNN (K - Vecinos más Cercanos)

1.3 Métricas de Evaluación

Para valorar el rendimiento de los modelos de clasificación utilizados, se recurrió a varios indicadores que permiten analizar distintos aspectos del comportamiento predictivo.

1.3.1 Exactitud o *Accuracy*

La exactitud expresa el porcentaje de casos correctamente clasificados respecto al total de observaciones. Indica, de manera general, cuán frecuentemente el modelo acierta, aunque puede dar una impresión engañosa cuando existen desbalances en las clases.

$$Exactitud = \frac{VP + VN}{TOTAL\ DE\ MUESTRAS}$$

donde VP representa verdaderos positivos y VN verdaderos negativos.

1.3.2 Matriz de Confusión

La matriz de confusión proporciona una visión detallada de los aciertos y errores del modelo, mostrando cómo se distribuyen las predicciones entre categorías correctas e incorrectas. En el ideal de un clasificador fiable, los valores predominantes se encuentran en la diagonal principal de la matriz, lo que refleja un alto número de clasificaciones correctas. Las entradas fuera de la diagonal representan desaciertos o transiciones erróneas entre clases.

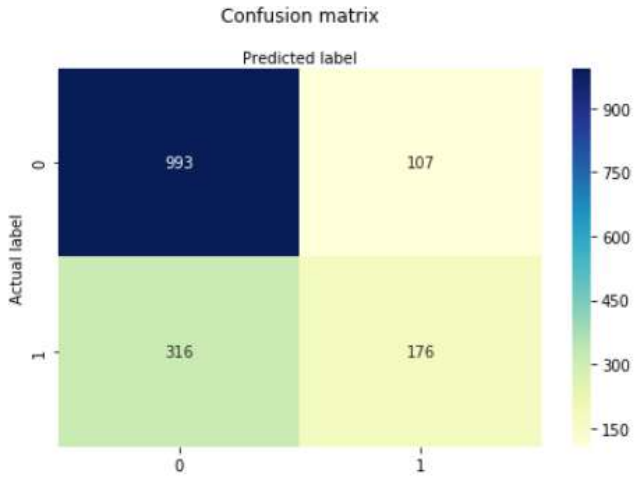

Ilustración IV: Matriz de Confusión

1.3.3 Recall

El *recall* (o sensibilidad) mide la proporción de elementos verdaderamente positivos que el modelo logra identificar correctamente. Es especialmente importante cuando el interés es no omitir casos críticos, como estudiantes en riesgo de abandono.

$$Recall = \frac{VP}{VP + FN}$$

donde FN hace referencia a falsos negativos.

1.3.4. Precisión

La precisión evalúa cuántas de las predicciones positivas hechas por el modelo corresponden realmente a casos positivos. En otras palabras, mide el grado de confianza que se puede tener en cada predicción positiva realizada.

$$Precisión = \frac{VP}{VP + FP}$$

1.3.5. F1 Score

El F1-Score combina precisión y *recall* en una sola métrica mediante su media armónica, favoreciendo modelos que mantienen un equilibrio entre ambos valores. Este indicador resulta útil cuando no se desea priorizar uno sobre otro, sino encontrar un punto óptimo entre sensibilidad y confiabilidad.

$$F1 = \frac{2.(Precisión \times Recall)}{Precisión + Recall}$$

2 OBJETIVOS

Objetivo General

- Desarrollar un modelo analítico que permita analizar y comprender los patrones de retención estudiantil en la Facultad de Ciencias y Tecnologías (FCyT), mediante técnicas de análisis de datos y aprendizaje automático aplicadas a registros académicos históricos

Objetivos Específicos

- Desarrollar un modelo de aprendizaje automático supervisado mediante métricas de desempeño, con el fin de identificar el más adecuado.
- Identificar los predictores con mayor influencia en la retención estudiantil, a partir de las medidas de importancia derivadas del modelo seleccionado.
- Analizar e interpretar los resultados obtenidos, generando información relevante que pueda servir de apoyo para la toma de decisiones institucionales.

3 METODOLOGÍA

3.1 Manejo y Recolección de Datos

Para la presente investigación, Para complementar la información obtenida mediante las encuestas aplicadas a los estudiantes y fortalecer el modelo predictivo desarrollado, se incorporó un segundo conjunto de datos institucionales proveniente de un archivo proveído por la Facultad de Ciencias y Tecnologías de la Universidad Nacional de Caaguazú (UNCA). Este archivo constituye el registro académico histórico de los alumnos y contiene 75.937 filas, representando un dataset longitudinal donde un mismo estudiante aparece múltiples veces, una por cada materia cursada, examen o instancia evaluativa registrada.

Este tipo de estructura permite un nivel de detalle significativo, aunque requiere procesos de transformación y agregación para poder ser utilizado adecuadamente en un modelo de predicción de retención estudiantil.

3.2 Contenido del Archivo Institucional

El archivo está compuesto por una serie de variables que describen tanto características del estudiante como su trayectoria académica. Entre las variables más relevantes se identificaron:

Variables identificadoras

- ID-Ingreso: Código asociado al ingreso del estudiante.
- IDAlumno: Identificador único para cada alumno.
- AñoIngreso: Año en que el estudiante inicia la carrera.

Variables personales

- Sexo: Clasificación binaria (Masculino/Femenino).

Variables académicas

- Carrera: Cuatro programas: Ingeniería Informática, Ingeniería Civil, Ingeniería Electrónica e Ingeniería en Electricidad.
- id-materia: Código único de cada materia (348 materias registradas).
- EstadoMateria: Estado de la materia (Aprobada, Reprobada, En curso, etc.).
- TipoExamen: Parcial, Final, Recuperatorio, entre otros.
- Curso: Nivel académico estimado (1°, 2°, 3°, 4° o 5°).
- PlanEstudios: Plan curricular vigente.
- TipoAlumno: Regular, Condicional, Reingresante.
- AñoCursado: Año efectivo en el que se cursa la materia.
- FechaExamen: Fecha exacta de evaluaciones, en formato tipo timestamp

3.3 Necesidad de transformación del Dataset

Dado que el archivo presenta información a nivel de materia y examen, no es apto directamente para el entrenamiento del modelo de retención. Para este propósito, fue necesario desarrollar procedimientos de:

Agrupación por estudiante

Cada alumno aparece desde unas pocas veces hasta cientos de veces dependiendo del avance en su carrera.
Fue necesario transformar los datos a nivel IDAlumno, generando métricas agregadas como:

- Cantidad total de materias aprobadas.
- Cantidad de materias reprobadas.
- Porcentaje de aprobación.
- Número de intentos por materia.
- Avance curricular estimado.
- Regularidad en evaluaciones.
- Cohorte y permanencia temporal.

Conversión de formatos

Varias columnas presentaban formatos mixtos:

- **FechaExamen**: formato datetime irregular.
- **AñoCursado**: presente en unos casos, faltante en muchos otros.
- **EstadoMateria**: valores inconsistentes que debían normalizarse.

Se aplicaron correcciones tipológicas para garantizar coherencia interna.

3.4 Transformación Final de los Datos

La integración de la información contenida en el archivo institucional resumen_estadodealumnocorregido.xlsx permitió consolidar un conjunto de datos adecuado para el entrenamiento del modelo de predicción de retención estudiantil. Este archivo, de naturaleza longitudinal y compuesto por 75.937 registros, incluye múltiples apariciones de cada estudiante según su historial de materias, evaluaciones y estados académicos, por lo que fue necesario aplicar procesos de depuración, estandarización tipológica y reconstrucción temporal antes de su utilización. Uno de los desafíos principales fue la ausencia sistemática de la variable “Año Cursado” en numerosos registros, la cual fue recuperada mediante inferencias basadas en el año de ingreso del estudiante, los años de cursado presentes en parte del archivo, la secuencia cronológica de materias inscritas y las fechas exactas de evaluaciones disponibles. Posteriormente, los datos fueron agrupados a nivel de estudiante, generando indicadores académicos consolidados tales como el promedio general acumulado, el número total de materias reprobadas, el porcentaje de aprobación, el avance curricular estimado y otros atributos derivados que reflejan la trayectoria académica individual. Para asegurar la compatibilidad con modelos de aprendizaje automático, se efectuó la codificación de variables categóricas mediante técnicas de tipo one-hot y se creó una etiqueta binaria que distingue entre estudiantes retenidos y estudiantes en riesgo de abandono. Finalmente, el proceso se completó con la exportación de tres archivos esenciales para las siguientes etapas del proyecto: dataset_limpio.csv, que contiene la consolidación completa de los datos; X_processed.csv, correspondiente a las variables predictoras ya transformadas; y y_processed.csv, que registra la variable objetivo necesaria para el entrenamiento y evaluación de los algoritmos empleados en los scripts posteriores.

3.5 Determinación de la Importancia de las Variables

En el marco de la presente investigación, la identificación de las variables más relevantes para la modelización del estado de retención estudiantil se llevó a cabo mediante el uso del algoritmo Random Forest, seleccionado tanto por su capacidad predictiva como por su potencial explicativo en problemas de clasificación supervisada [8].

A diferencia de otros modelos de tipo “caja negra”, Random Forest permite cuantificar la contribución individual de cada variable al proceso de clasificación, lo cual resulta especialmente pertinente en estudios de carácter académico e institucional, donde no solo se busca predecir un fenómeno, sino también comprender los factores que lo determinan.

3.5.1 Fundamento del Método

Random Forest es un algoritmo de aprendizaje ensamblado (ensemble learning) basado en la construcción de múltiples árboles de decisión entrenados de manera independiente sobre subconjuntos aleatorios del conjunto de datos. Cada árbol realiza divisiones sucesivas del espacio de características con el objetivo de maximizar la separación entre clases [9].

En problemas de clasificación binaria, como el abordado en esta investigación (estudiante retenido vs. estudiante en riesgo), el criterio de separación utilizado en cada nodo corresponde al Índice de Gini, el cual mide el grado de impureza de un conjunto de observaciones.

El Índice de Gini se define como:

$$Gini = 1 - \sum_{i=1}^{C} \rho_i^2$$

Donde ρ_i representa la proporción de observaciones pertenecientes a la clase i y C el número total de clases. Una disminución del valor del índice de Gini indica una mejora en la homogeneidad del nodo, es decir, una mejor separación entre las clases objetivo.

3.5.2 Métrica de Importancia de Variables

La importancia de cada variable fue estimada utilizando la métrica conocida como reducción promedio de impureza, también denominada Mean Decrease in Impurity (MDI) [10]. Esta métrica cuantifica cuánto contribuye una variable a reducir la impureza del modelo a lo largo de todos los árboles que conforman el bosque.

Formalmente, la importancia de una variable j se expresa como:

$$Importancia_j = \sum_{t \in T} \sum_{n \in N_t} \Delta I(n) \cdot 1(v(n) = j)$$

Donde:

- T representa el conjunto de árboles del bosque,
- N_t es el conjunto de nodos del árbol t,
- $\Delta I(n)$ corresponde a la reducción de impurezas generada en el nodo n,
- $v(n)$ indica la variable utilizada en dicho nodo,
- $1(\cdot)$ es la función indicadora.

Las importancias obtenidas se normalizan posteriormente, de modo que la suma total de todas las variables sea igual a 1, permitiendo una interpretación relativa y comparativa.

3.5.3 Implementación computacional

La estimación de la importancia de las variables se realizó a partir del modelo Random Forest previamente entrenado, utilizando el atributo:

*feature_importances_*

Este atributo devuelve un vector numérico donde cada valor representa la contribución relativa de una variable al desempeño global del modelo. Dichos valores fueron asociados a los nombres de las variables de entrada y exportados en un archivo independiente (*importancias.csv*), facilitando su análisis e interpretación posterior.

Cabe destacar que el cálculo de estas importancias se realiza sobre las variables previamente escaladas y seleccionadas, garantizando la consistencia del análisis y evitando sesgos derivados de diferencias en la magnitud de las variables.

3.5.4 Relevancia para el modelo de retención

El análisis de la importancia de variables constituye un elemento central del modelo propuesto, ya que permite:

Validar empíricamente la selección de las variables utilizadas.

Identificar factores críticos para el diseño de políticas institucionales de acompañamiento.

Justificar la elección del algoritmo Random Forest como herramienta principal para el análisis de la retención estudiantil.

Asimismo, este enfoque fortalece la interpretabilidad del modelo y aporta evidencia cuantitativa que respalda las conclusiones del estudio, alineando los resultados obtenidos con el objetivo general de comprender y analizar los factores asociados a la permanencia académica.

4 RESULTADOS

4.1 Interpretación de los Resultados

Las variables con mayor importancia corresponden a aquellas que:

- Generan mayores reducciones de impureza en los nodos de decisión.
- Aparecen con mayor frecuencia en niveles superiores de los árboles.

Contribuyen de forma significativa a la correcta clasificación del estado de retención.

Los resultados evidencian que las variables asociadas a la inactividad académica prolongada, la trayectoria temporal del estudiante y los indicadores acumulativos de riesgo concentran la mayor proporción de importancia dentro del modelo.

Este hallazgo sugiere que la retención estudiantil no depende exclusivamente del rendimiento académico puntual, sino que está fuertemente condicionada por la continuidad sostenida de la actividad académica a lo largo del tiempo.

4.2 Descripción del conjunto de datos

El proceso de depuración y estandarización permitió obtener un conjunto final de 1.422 estudiantes, cada uno con 14 variables, incluyendo características académicas,

indicadores de actividad y la variable objetivo de retención. La distribución final de la etiqueta fue:

- Retención (0): 879 estudiantes
- A/E (1): 543 estudiantes

Como puede observarse, la distribución presenta un leve desbalance natural (62% vs. 38%), habitual en estudios de permanencia universitaria.

4.3 Rendimiento de los Modelos Predictivos

Se entrenaron cuatro algoritmos de clasificación supervisada:
Random Forest, Regresión Logística, Máquinas de Vectores de Soporte (SVM) y K-Nearest Neighbors (KNN). A continuación en la Tabla 1, se presentan los resultados obtenidos en cada aspecto. La matriz de confusión desglosa las predicciones en cuatro categorías:

- TN (True Negative):
  Estudiantes correctamente identificados como retenidos.
- FP (False Positive):
  Estudiantes clasificados como en riesgo que en realidad están retenidos.
- FN (False Negative):
  Estudiantes en riesgo que el modelo no detectó.
- TP (True Positive):
  Estudiantes en riesgo correctamente identificados.

Tabla 1: Matriz Confusión

Modelo	Accuracy	Precision	Recall	F1-score	TN	FP	FN	TP
Random Forest	0.828	0.837	0.764	0.798	204	16	32	105
Regresión Logística	0.813	0.824	0.729	0.773	201	19	37	100
SVM	0.815	0.829	0.734	0.778	202	18	36	101
KNN	0.819	0.818	0.748	0.781	200	20	34	103

4.4 Interpretación General

Los resultados muestran que, si bien todos los modelos presentan un rendimiento adecuado, el Random Forest obtiene el mejor desempeño global, destacándose especialmente en Accuracy (82.8%), Precision (83.7%) y Recall (76.4%).

Este último indicador resulta de particular relevancia, ya que permite identificar una mayor proporción de estudiantes en riesgo, aspecto crítico para modelos de retención que buscan captar tempranamente posibles casos de abandono.

Los modelos lineales (como la Regresión Logística) y basados en distancias (como KNN) demostraron desempeños competitivos, aunque ligeramente inferiores en la detección de casos positivos (abandono), presentando más falsos negativos.

4.5 Importancia de las Variables

La estimación de importancia generada a partir del modelo Random Forest permitió identificar los factores con mayor incidencia en la predicción de la retención académica. La tabla 2 resume la magnitud relativa de las variables más influyentes.

Tabla 2: Importancia de las Variables

Variable	Importancia
Riesgo suave	0.3317
Inactividad normalizada	0.2156
Años de inactividad	0.1960
calif_norm	0.0690
Calificación media	0.0647
Calificación	0.0639
Materias cursadas	0.0591
Resto de variables	≈ 0.00

4.6 Análisis interpretativo

Los resultados confirman que el componente temporal asociado a la continuidad académica constituye el principal predictor de retención, concentrando más del 74 % de la importancia total del modelo. Esto incluye:

- tiempo sin actividad académica,
- métricas de inactividad suavizadas,
- años acumulados sin rendir materias.

Este hallazgo coincide con la literatura internacional, donde diversos autores sostienen que la continuidad y la participación activa del estudiante son factores decisivos para su retención en la educación superior. Tinto destaca que la permanencia depende fundamentalmente del nivel de integración académica y social del estudiante dentro de la institución [1], mientras que que la persistencia se ve fuertemente influida por la interacción continua del alumno con los procesos académicos y administrativos universitarios [2]. Por tanto, la inactividad prolongada constituye un indicador crítico para predecir la pérdida de continuidad en la trayectoria formativa.

Por otro lado, las variables de rendimiento académico presentan una influencia moderada (≈ 6–7%), lo cual sugiere que:

La retención no se explica únicamente por el desempeño académico, sino por la presencia o ausencia de actividad sostenida en el tiempo.

Finalmente, variables como ProyectoFinal_SinNota y tasa_aprobacion mostraron importancia nula o residual, posiblemente debido a su baja presencia o a la dispersión existente en los registros institucionales.

5 CONCLUSIONES

El análisis realizado a partir del registro académico longitudinal de la Facultad de Ciencias y Tecnologías de la Universidad Nacional de Caaguazú permitió caracterizar con precisión los factores que inciden en la continuidad educativa de los estudiantes, consolidando un marco metodológico robusto para el estudio de la retención estudiantil. Los resultados obtenidos muestran que las dinámicas temporales de participación académica particularmente los periodos de inactividad y su acumulación constituyen los elementos de mayor influencia sobre la permanencia dentro de la institución, superando ampliamente a los indicadores tradicionales de rendimiento. Este hallazgo refuerza lo señalado en la literatura especializada, que subraya la importancia de la integración académica sostenida como condición necesaria para garantizar la continuidad formativa. Si bien las calificaciones y el desempeño académico presentan una contribución significativa, su efecto resulta secundario frente al peso explicativo del compromiso temporal del estudiante. Asimismo, variables de bajo registro o dispersión institucional mostraron una influencia marginal. En conjunto, estos resultados ofrecen una visión clara de la trayectoria estudiantil y habilitan a la institución a diseñar estrategias de intervención temprana fundamentadas en evidencia, permitiendo orientar acciones de acompañamiento dirigidas a fortalecer la retención y mejorar el rendimiento interno de las carreras. El presente trabajo sienta así las bases para la implementación de sistemas de monitoreo continuo de la actividad académica y para el desarrollo de políticas institucionales que aborden de manera integral los factores asociados a la permanencia estudiantil.

6. GLOSARIO DE TÉRMINOS

El presente glosario reúne los principales conceptos técnicos, metodológicos y analíticos empleados a lo largo del desarrollo del proyecto *"Modelo Analítico para el Estudio de la Retención Estudiantil en la Facultad de Ciencias y Tecnologías (FCyT–UNCA)"*. Su objetivo es facilitar la comprensión del contenido, estandarizar el significado de los términos utilizados y servir como referencia conceptual para el lector.

- **Aprendizaje Automático (Machine Learning)**

Rama de la inteligencia artificial que estudia el desarrollo de algoritmos capaces de aprender patrones a partir de datos históricos y realizar predicciones o clasificaciones sin ser programados explícitamente para cada situación.

- **Aprendizaje Supervisado**

Tipo de aprendizaje automático en el que el modelo se entrena utilizando datos etiquetados, es decir, registros donde la variable objetivo es conocida previamente. En este estudio se utiliza para predecir el estado de retención estudiantil.

- **Aprendizaje No Supervisado**

Modalidad de aprendizaje automático orientada a identificar patrones o estructuras internas en los datos sin contar con una variable objetivo definida, como ocurre en los procesos de agrupamiento.

- **Accuracy (Exactitud)**

Métrica de evaluación que indica la proporción de predicciones correctas realizadas por un modelo respecto al total de observaciones analizadas.

- **Árbol de Decisiones (Decision Tree)**

Modelo predictivo basado en una estructura jerárquica de decisiones, donde cada nodo evalúa una condición sobre una variable y conduce a una clasificación final. Se

- **Clasificación Binaria**

Problema de aprendizaje automático en el que la variable objetivo solo puede tomar dos valores posibles. En este trabajo corresponde a estudiantes retenidos y estudiantes en riesgo de abandono.

- **Clúster (Cluster)**

Grupo de observaciones que presentan alta similitud entre sí y diferencias significativas respecto a otros grupos del conjunto de datos.

- **Clustering (Agrupamiento)**

Técnica de aprendizaje no supervisado que consiste en dividir un conjunto de datos en grupos homogéneos según criterios de similitud.

- **Cohorte**

Conjunto de estudiantes que ingresan a una carrera o institución académica en un mismo período lectivo.

- **Dataset Longitudinal**

Conjunto de datos que registra información de los mismos individuos a lo largo del tiempo, permitiendo analizar trayectorias académicas y patrones de continuidad o abandono.

- **Desbalance de Clases**

Situación en la que una categoría de la variable objetivo tiene mayor representación que otra, lo cual puede afectar la interpretación de métricas como la exactitud.

- **Feature Importance (Importancia de Variables)**

Medida que cuantifica la contribución relativa de cada variable al desempeño predictivo del modelo, especialmente relevante en algoritmos como Random Forest.

- **Hiperplano**

Superficie de decisión utilizada por modelos como las Máquinas de Soporte Vectorial para separar clases en espacios multidimensionales.

- **Índice de Gini**

Medida de impureza utilizada en árboles de decisión y Random Forest que evalúa el grado de heterogeneidad de un nodo respecto a las clases.

7 BIBLIOGRAFÍA

[1] Á. L. L. Angela Cynthia Elizabeth Ibarra de Leguizamon, "Causas de la deserción estudiantil según estudiantes desertores de dos unidades académicas en la Universidad Nacional de Concepción," *Ciencia Latina Revista Científica Multidisciplinar ,* pp. 4273-4290, 2023.

[2] V. Tinto, "Leaving College: Rethinking the Causes and Cures of Student Attrition," *University of Chicago Press,* 1993.

[3] AWS, "¿Qué es la ciencia de datos?," 2024. [Online]. Available: https://aws.amazon.com/es/what-is/data-science/.

[4] D. P. R. M. Trupti M. Kodinariya, "Review on determining number of Cluster in K-Means Clustering," *International Journal of Advance Research in Computer Science and Management Studies,* vol. 1, no. 6, pp. 90-95, 2013.

[5] D. Rodríguez, "Método del codo (Elbow method) para seleccionar el número óptimo de clústeres en K-means," analyticslane, 09 06 2023. [Online]. Available: https://www.analyticslane.com/2023/06/09/metodo-del-codo-elbow-method-para-seleccionar-el-numero-optimo-de-clusteres-en-k-means/.

[6] P. C. L. L. P. S. P. P. K. Mohammadmehdi Saberioon, "omparative Performance Analysis of Support Vector Machine, Random Forest, Logistic Regression and k-Nearest Neighbours in Rainbow Trout (Oncorhynchus Mykiss) Classification Using Image-Based Features," *Sensors,* vol. 18, no. 4, p. 1027, 2018.

[7] N. C. R. G. A. M. d. C. G. T. Rocío Erandi Barrientos Martínez, "Árboles de decisión como herramienta en el diagnóstico médico," *Revista Médica de UV,* pp. 19-24, 2009.

[8] L. Breiman, "Random Forests," *Machine Learning,* vol. 45, no. 1, pp. 5-32, 2001.

[9] T. T. R. F. J. Hastie, "The Elements of Statistical Learning," 2009.

[10] F. Podegrosa, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research,* 2011.

[11] N. R. G. Duarte, Modelo de alerta temprana para la deserción estudiantil en la Facultad de Ciencias y Tecnologías basado en la estimación de factores académicos, Coronel Oviedo, Paraguay: Facultad de Ciencias y Tecnologías, 2024.

8 ANEXOS

Script de Procesamiento

```
# scripts/01_preprocesar.py
import os
import pandas as pd
import numpy as np
from datetime import datetime
from sklearn.preprocessing import MinMaxScaler

BASE_DIR = r"C:\Users\ultrabook\proyecto_retencion"
INPUT_PATH = os.path.join(BASE_DIR, "data",
"resumen_estadodealumnocorregido.xlsx")
PROCESSED_DIR = os.path.join(BASE_DIR, "data", "processed")
os.makedirs(PROCESSED_DIR, exist_ok=True)

OUTPUT_DATASET = os.path.join(PROCESSED_DIR, "dataset_limpio.csv")
OUTPUT_X = os.path.join(PROCESSED_DIR, "X_processed.csv")
OUTPUT_Y = os.path.join(PROCESSED_DIR, "y_processed.csv")
DIAG_OUT = os.path.join(PROCESSED_DIR, "preproc_diagnostics.csv")

RANDOM_SEED = 42
NOISE_FLIP_RATE = 0.08  # 8% ruido estructurado

def find_id_column(df):
    candidates = ["id_alumno", "id-alumno", "id", "id-ingreso", "id_ingreso", "idusuario",
"idusuario"]
    cols = [c.lower() for c in df.columns]
    for cand in candidates:
        for col in df.columns:
            if cand == col.lower():
```

```
            return col
    # try fuzzy: contains 'id'
    for col in df.columns:
        if "id" in col.lower() and len(col) < 25:
            return col
    raise KeyError("No se encontró columna identificadora (ID). Asegurate que exista 'IDAlumno' o similar.")

def safe_to_datetime(s):
    try:
        return pd.to_datetime(s, errors="coerce")
    except Exception:
        return pd.to_datetime(s, errors="coerce")

def aggregate_per_student(df, id_col):
    """Devuelve DataFrame agregado por alumno con variables numéricas y proxies."""
    # Ensure FechaExamen is datetime if present
    if "FechaExamen" in df.columns:
        df["FechaExamen"] = safe_to_datetime(df["FechaExamen"])
    else:
        # try to find a fecha column
        fecha_cols = [c for c in df.columns if "fecha" in c.lower()]
        if fecha_cols:
            df["FechaExamen"] = safe_to_datetime(df[fecha_cols[0]])
        else:
            df["FechaExamen"] = pd.NaT

    # Create ProyectoFinal_SinNota flag at row level
    if "Materia" in df.columns and "Calificación" in df.columns:
        pfg_flag = (df["Materia"].astype(str).str.contains("Proyecto Final", case=False, na=False)) & df["Calificación"].isna()
        df["ProyectoFinal_SinNota"] = pfg_flag.astype(int)
```

```
    else:
        df["ProyectoFinal_SinNota"] = 0

    # numeric col candidates for aggregation
    numeric_candidates = ["Calificación", "EP", "EF", "ET", "IntentosFinal",
"RegularidadesMateria"]
    numeric_present = [c for c in numeric_candidates if c in df.columns]

    # Ensure numeric conversion
    for c in numeric_present:
        df[c] = pd.to_numeric(df[c], errors="coerce")

    # Aggregations per student
    agg_dict = {}
    for c in numeric_present:
        # use mean for scores, sum for counts if name suggests count
        if c.lower() in ["regularidadesmateria", "intentosfinal"]:
            agg_dict[c] = "sum"
        else:
            agg_dict[c] = "mean"

    # counts and approvals
    agg_dict.update({
        # count of rows = materias cursadas
        "Materia": "count",
        # count of approved (if EstadoMateria exists)
    })

    grouped = df.groupby(id_col).agg(agg_dict)

    # rename
    grouped = grouped.rename(columns={"Materia": "materias_cursadas"})
```

```
    # materias_aprobadas
    if "EstadoMateria" in df.columns:
        aprob = df.groupby(id_col)["EstadoMateria"].apply(lambda s:
s.astype(str).str.lower().isin(["aprobado", "aprobada", "aprob"]).sum())
        grouped["materias_aprobadas"] = aprob
    else:
        grouped["materias_aprobadas"] = 0

    # tasa aprobacion
    grouped["tasa_aprobacion"] = grouped["materias_aprobadas"] /
grouped["materias_cursadas"].replace(0, np.nan)
    grouped["tasa_aprobacion"] = grouped["tasa_aprobacion"].fillna(0)

    # Ultima actividad e InactividadAnios
    ult = df.groupby(id_col)["FechaExamen"].max().rename("UltimaActividad")
    grouped = grouped.join(ult)

    today = pd.Timestamp.today()
    grouped["InactividadAnios"] = (today - grouped["UltimaActividad"]).dt.days / 365.25
    grouped["InactividadAnios"] =
grouped["InactividadAnios"].fillna(grouped["InactividadAnios"].median())

    # ProyectoFinal_SinNota aggregated (max -> if any row has PFG sin nota)
    pfg = df.groupby(id_col)["ProyectoFinal_SinNota"].max().fillna(0)
    grouped["ProyectoFinal_SinNota"] = pfg

    # Calificacion mean (if exists)
    if "Calificación" in df.columns:
        grouped["Calificacion_media"] = grouped.get("Calificación", np.nan)
        # if original 'Calificación' not in aggregated columns, compute
        if "Calificacion_media" not in grouped.columns:
```

```
        grouped["Calificacion_media"] =
df.groupby(id_col)["Calificación"].mean().fillna(df["Calificación"].median())

    # Fill NaNs for numeric cols
    numeric_cols = grouped.select_dtypes(include=[np.number]).columns.tolist()
    grouped[numeric_cols] =
grouped[numeric_cols].fillna(grouped[numeric_cols].median())

    # reset index to have ID as column
    grouped = grouped.reset_index().rename(columns={id_col: "IDAlumno"})

    return grouped

def build_riesgo_and_label(df_student):
    """Construye riesgo_suave y retencion_model tal como la Opción A."""
    print("== Construyendo riesgo suave y retencion_model ==")

    # Ensure required cols exist
    if "InactividadAnios" not in df_student.columns:
        df_student["InactividadAnios"] = 0.0
    if "ProyectoFinal_SinNota" not in df_student.columns:
        df_student["ProyectoFinal_SinNota"] = 0
    if "Calificacion_media" not in df_student.columns:
        # try alternative numeric columns
        numcols = [c for c in df_student.columns if c.lower().startswith("calif") or "promedio"
in c.lower()]
        if numcols:
            df_student["Calificacion_media"] = df_student[numcols[0]]
        else:
            df_student["Calificacion_media"] =
df_student.select_dtypes(include=[np.number]).mean(axis=1).fillna(0)
```

```
    scaler = MinMaxScaler()
    # reshape requires 2D
    df_student["inact_norm"] =
scaler.fit_transform(df_student[["InactividadAnios"]].fillna(0))
    df_student["pfg_norm"] = df_student["ProyectoFinal_SinNota"].astype(float)
    df_student["calif_norm"] =
scaler.fit_transform(df_student[["Calificacion_media"]].fillna(0))

    # riesgo_suave = 0.55*inact + 0.30*pfg + 0.15*(1 - calif)
    df_student["riesgo_suave"] = (
        0.55 * df_student["inact_norm"]
        + 0.30 * df_student["pfg_norm"]
        + 0.15 * (1.0 - df_student["calif_norm"])
    )

    # ruido normal pequeño
    np.random.seed(RANDOM_SEED)
    ruido = np.random.normal(0, 0.05, size=len(df_student))
    df_student["riesgo_suave"] = (df_student["riesgo_suave"] + ruido).clip(0, 1)

    # umbral percentil 65
    p65 = df_student["riesgo_suave"].quantile(0.65)
    df_student["retencion_model"] = (df_student["riesgo_suave"] > p65).astype(int)

    # flip noise (8%)
    flip_mask = np.random.RandomState(RANDOM_SEED).rand(len(df_student)) <
NOISE_FLIP_RATE
    df_student.loc[flip_mask, "retencion_model"] = 1 - df_student.loc[flip_mask,
"retencion_model"]

    return df_student
```

```
def keep_numeric_only_and_save(df_student):
    # Save full dataset per student for records
    df_student.to_csv(OUTPUT_DATASET, index=False)

    # Build X (numeric only) and y
    X = df_student.drop(columns=["retencion_model", "IDAlumno", "UltimaActividad"],
errors="ignore")
    X_numeric = X.select_dtypes(include=[np.number]).copy()

    # If no numeric columns remain, create simple proxies
    if X_numeric.shape[1] == 0:
        X_numeric["dummy"] = 0

    y = df_student["retencion_model"].astype(int)

    # Save
    X_numeric.to_csv(OUTPUT_X, index=False)
    y.to_csv(OUTPUT_Y, index=False)

    # Diagnostics
    diag = {
        "n_students": len(df_student),
        "n_features": X_numeric.shape[1],
        "retention_counts": y.value_counts().to_dict()
    }
    pd.DataFrame([diag]).to_csv(DIAG_OUT, index=False)
    print("Guardados: ")
    print(" - full student dataset:", OUTPUT_DATASET)
    print(" - X (numeric):", OUTPUT_X)
    print(" - y:", OUTPUT_Y)
    print(" - diagnostics:", DIAG_OUT)
```

```
def main():
    print("=== Script 1: Preprocesamiento robusto ===")
    print("Cargando:", INPUT_PATH)
    if not os.path.exists(INPUT_PATH):
        raise FileNotFoundError(f"No se encontró el archivo de entrada: {INPUT_PATH}")

    df_raw = pd.read_excel(INPUT_PATH)
    print("Archivo cargado. Filas:", df_raw.shape[0], "Columnas:", df_raw.shape[1])

    # Normalize column names
    df_raw.columns = [str(c).strip().replace(" ", "_") for c in df_raw.columns]

    try:
        id_col = find_id_column(df_raw)
        print("Columna ID detectada:", id_col)
    except KeyError as e:
        print("ERROR:", e)
        raise

    # Aggregate per student
    df_students = aggregate_per_student(df_raw, id_col)

    # Build riesgo and label
    df_students = build_riesgo_and_label(df_students)

    # Final numeric-only X and y saved
    keep_numeric_only_and_save(df_students)

    print("=== Preprocesamiento finalizado correctamente ===")

if __name__ == "__main__":
    main()
```

Script de Entrenamiento

```
import os
import pandas as pd
import joblib

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.utils import shuffle

from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier

from sklearn.metrics import (
    accuracy_score,
    precision_score,
    recall_score,
    f1_score,
    confusion_matrix
)

BASE_DIR = r"C:\Users\ultrabook\proyecto_retencion"
PROCESSED_DIR = os.path.join(BASE_DIR, "data", "processed")
MODELS_DIR = os.path.join(BASE_DIR, "models")

os.makedirs(MODELS_DIR, exist_ok=True)

X_PATH = os.path.join(PROCESSED_DIR, "X_processed.csv")
Y_PATH = os.path.join(PROCESSED_DIR, "y_processed.csv")
METRICS_OUT = os.path.join(PROCESSED_DIR, "tabla_metricas_modelos.csv")
```

```
def cargar_datos():
    if not os.path.exists(X_PATH) or not os.path.exists(Y_PATH):
        raise FileNotFoundError("Archivos procesados no encontrados. Asegúrese de ejecutar Script 1.")

    X = pd.read_csv(X_PATH)
    y = pd.read_csv(Y_PATH).iloc[:, 0]

    # Sanitizar NaN
    X = X.fillna(X.median(numeric_only=True))

    # Confirmar solo numéricos
    X = X.select_dtypes(include=["number"])

    print("\n=== CARGA DE DATOS ===")
    print("X shape:", X.shape)
    print("Distribución y:")
    print(y.value_counts(), "\n")

    # Mezclar dataset
    X, y = shuffle(X, y, random_state=42)
    return X, y

def entrenar_modelos(X_train, y_train, X_test, y_test):
    modelos = {
        "RandomForest": RandomForestClassifier(
            n_estimators=200, class_weight="balanced", random_state=42
        ),
        "LogisticRegression": LogisticRegression(max_iter=500),
        "SVM": SVC(probability=True),
        "KNN": KNeighborsClassifier(n_neighbors=7)
```

```
}

registros = []

for nombre, modelo in modelos.items():
    print(f"\nEntrenando modelo: {nombre}")

    modelo.fit(X_train, y_train)
    pred = modelo.predict(X_test)

    cm = confusion_matrix(y_test, pred)

    registro = {
        "Modelo": nombre,
        "Accuracy": accuracy_score(y_test, pred),
        "Precision": precision_score(y_test, pred, zero_division=0),
        "Recall": recall_score(y_test, pred, zero_division=0),
        "F1": f1_score(y_test, pred, zero_division=0),
        "CM_TN": cm[0, 0],
        "CM_FP": cm[0, 1],
        "CM_FN": cm[1, 0],
        "CM_TP": cm[1, 1],
    }

    registros.append(registro)

    # Guardar modelo
    joblib.dump(modelo, os.path.join(MODELS_DIR, f"{nombre}.pkl"))

return pd.DataFrame(registros)
```

```
def main():
    X, y = cargar_datos()

    # ESCALADO
    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(X)

    joblib.dump(scaler, os.path.join(MODELS_DIR, "scaler.pkl"))

    # SPLIT
    X_train, X_test, y_train, y_test = train_test_split(
        X_scaled, y,
        test_size=0.25,
        random_state=42,
        stratify=y
    )

    # ENTRENAMIENTO
    tabla = entrenar_modelos(X_train, y_train, X_test, y_test)
    tabla.to_csv(METRICS_OUT, index=False)

    print("\n=== TABLA DE MÉTRICAS GENERADA ===")
    print(METRICS_OUT)

if __name__ == "__main__":
    main()
```

Script de Evaluación

```
import os
import pandas as pd
import joblib
import matplotlib.pyplot as plt
```

```
from sklearn.metrics import roc_curve, auc

BASE_DIR = r"C:\Users\ultrabook\proyecto_retencion"
PROCESSED_DIR = os.path.join(BASE_DIR, "data", "processed")
MODELS_DIR = os.path.join(BASE_DIR, "models")
OUT_DIR = os.path.join(BASE_DIR, "outputs")

os.makedirs(OUT_DIR, exist_ok=True)

DATASET = os.path.join(PROCESSED_DIR, "dataset_limpio.csv")
MODEL_PATH = os.path.join(MODELS_DIR, "RandomForest.pkl")
SCALER_PATH = os.path.join(MODELS_DIR, "scaler.pkl")

def main():

    df = pd.read_csv(DATASET)
    print("Dataset limpio cargado:", df.shape)

    # Variables
    X = df.drop(columns=["retencion_model", "IDAlumno", "UltimaActividad"],
errors="ignore")
    X = X.select_dtypes(include=["number"])
    y = df["retencion_model"]

    scaler = joblib.load(SCALER_PATH)
    modelo = joblib.load(MODEL_PATH)

    # Escalar
    X_scaled = scaler.transform(X)

    # ROC
    probs = modelo.predict_proba(X_scaled)[:, 1]
```

```
    fpr, tpr, _ = roc_curve(y, probs)
    roc_auc = auc(fpr, tpr)

    # Graficar
    plt.plot(fpr, tpr, label=f"AUC={roc_auc:.3f}")
    plt.plot([0, 1], [0, 1], "--")
    plt.xlabel("False Positive Rate")
    plt.ylabel("True Positive Rate")
    plt.title("Curva ROC – Proyección de Retención")
    plt.legend()
    plt.savefig(os.path.join(OUT_DIR, "roc_curve.png"))
    plt.close()

    # Importancia de variables
    imp = pd.DataFrame({
        "variable": X.columns,
        "importancia": modelo.feature_importances_
    }).sort_values("importancia", ascending=False)

    imp.to_csv(os.path.join(OUT_DIR, "importancias.csv"), index=False)

    print("Evaluación completada. Resultados guardados.")

if __name__ == "__main__":
    main()
```