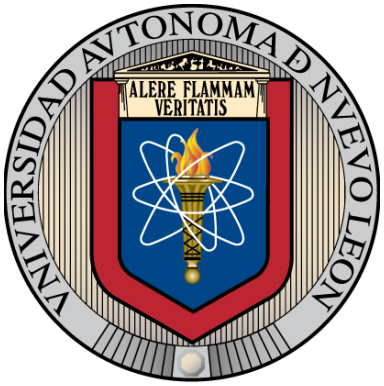
UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS



ALGORITMOS DE *MACHINE LEARNING* PARA LA IDENTIFICACIÓN DE FACTORES ACADÉMICOS QUE PONEN EN RIESGO DE DESERCIÓN A LOS ESTUDIANTES UNIVERSITARIOS

Por

ORESTES BOFFILL BELTRÁN

Como requisito parcial para obtener el Grado de

MAESTRÍA EN CIENCIA DE DATOS

Noviembre, 2023

Universidad Autónoma de Nuevo León

Facultad de Ciencias Físico Matemáticas

Los miembros del Comité de Tesis recomendamos que la Tesina "Algoritmos de *Machine Learning* para la identificación de factores académicos que ponen en riesgo de deserción a los estudiantes universitarios", realizada por el alumno Orestes Boffill Beltrán, con número de matrícula 2085377, sea aceptada para su defensa como opción al grado de Maestría en Ciencia de Datos.

El Comité de Tesis

_____

Dr. Álvaro Eduardo Cordero Franco
Director

_____

Dr. José Apolinar Loyola Rodríguez
Revisor

_____

Dra. Azucena Yoloxóchitl Ríos Mercado
Revisor

Vo.Bo.

_____

Dra. Azucena Yoloxóchitl Ríos Mercado
Coordinadora de la Maestría en Ciencia de Datos

San Nicolás de los Garza, N.L.

DEDICATORIA

A mis padres por haberme educado y guiado hacia la consecución de metas mediante el esfuerzo y el trabajo.

A Dani y a Deyi, quienes brindaron un apoyo y una motivación invaluable para concluir este proyecto que se convirtió en parte de su compromiso.

A mi familia, quienes han sido un apoyo constante en mi vida.

A todas las personas de las cuales he aprendido a lo largo de mi trayectoria.

ÍNDICE GENERAL

.

ÍNDICE DE FIGURAS

Página

ÍNDICE DE TABLAS

AGRADECIMIENTOS

Al Comité de Tesis por su valiosa contribución a nuestro trabajo. Sus sugerencias y correcciones fueron fundamentales durante la elaboración del documento final de la Tesina.

A mis Maestros en la parte curricular, quienes brindaron aportes conceptuales y prácticos que sentaron las bases fundamentales para la realización de este trabajo en el posgrado.

Al Dr. Álvaro Eduardo Cordero Franco y a la Dra. Jessica Margarita Rubiano Moreno por el apoyo incondicional, guía, supervisión y acertada dirección durante todo el desarrollo del proyecto.

Al Dr. Atilano Martínez Huerta, Director de la Facultad de Ciencias Físico Matemáticas de la Universidad Autónoma de Nuevo León, por su inestimable ayuda facilitando el acceso a los datos utilizados y por otorgarme el apoyo necesario para la culminación de mis estudios.

CAPÍTULO 1
INTRODUCCIÓN

La deserción estudiantil es uno de los problemas que aborda la mayoría de las Instituciones de Educación Superior (IES) de todo el mundo. Definir el concepto de deserción estudiantil es una tarea compleja, sin embargo, existe consenso en precisarla como un abandono que puede ser explicado por diferentes categorías de variables: socioeconómicas, individuales, institucionales y académicas (Ruiz et al., 2009).

Las altas tasas de deserción y la finalización tardía de la Educación Superior están asociadas con costos personales y sociales considerables. La deserción de la Educación Superior representa un costo para el gobierno y la sociedad, un gasto innecesario para la familia y una experiencia de fracaso para el estudiante universitario. Dentro de los tipos de deserción escolar se incluye la deserción temprana o primera deserción (*first drop-out*), que se manifiesta cuando el estudiante abandona sus estudios en los primeros semestres del programa académico.

En este sentido, algunos países han comenzado a diseñar profundos procesos de mejoramiento para aumentar la retención en los primeros años de estudios universitarios (Peralta, 2008). Enmarcados en estos cambios surgió una nueva disciplina científica conocida como minería de datos educativos (EDM, *siglas en inglés*), sustentada por el rápido desarrollo de la inteligencia artificial, *Machine Learning* (ML) y los métodos estadísticos, además por los volúmenes de datos acumulados por las universidades (Kiss et al., 2019).

En la última década, se ha incrementado la investigación educativa analítica predictiva y se han desarrollado sistemas de apoyo a la toma de

decisiones basados en inteligencia artificial para ayudar a las partes interesadas en la Educación Superior. La aplicación de ML en universidades puede proporcionar información procesable y de calidad para implementar intervenciones educativas, como el apoyo oportuno para estudiantes en riesgo de abandonar la escuela (Dutt et al., 2017).

La identificación de los grupos, desertores y no desertores, y el cálculo de la probabilidad de pertenecer a uno u otro conjunto, dadas ciertas características, permiten diseñar políticas permanentes, maximizando así los recursos disponibles en las universidades.

La presente investigación es resultado de la aplicación de varias técnicas de ML con el objetivo de identificar los factores académicos que causan deserción estudiantil en la Facultad de Ciencias Físico Matemáticas (FCFM) de la Universidad Autónoma de Nuevo León (UANL). La identificación de estudiantes en riesgo permitirá a la Facultad adoptar medidas anticipadas de mitigación que impacten positivamente en los índices de retención y eficiencia terminal; así se refleja en el Plan de Desarrollo Institucional UANL 2022-2030.

CAPÍTULO 2
DELIMITACIÓN Y PLANTEAMIENTO DEL PROBLEMA DE INVESTIGACIÓN

La deserción o abandono estudiantil constituye una problemática actual, que afecta negativamente a los sistemas educativos alrededor del mundo, incidiendo en su efectividad, eficiencia y prestigio. Generando, además, consecuencias económicas y/o psicosociales negativas en los estudiantes y sus familias. En el ámbito de la Educación Superior, implica el retiro del estudiante de su carrera antes de alcanzar la titulación(Matos, 2021).

Según informes de la UNESCO y el Banco Mundial, en América Latina y el Caribe, menos de la mitad de todos los jóvenes que comenzaron los cursos de Educación Superior se gradúan (tasa de graduación del 46%), exceptuando a EUA con un 67% (Ferreyra et al., 2017).

La identificación de las posibles causas de la deserción se ha convertido en una tarea compleja para las universidades. Entre los factores causales podemos citar los académicos, psicosociales, familiares, económicos, factores psicológicos, así como los relacionados con las propias IES: infraestructura, vida estudiantil, entre otros. Ante esta complejidad, resulta vital la identificación temprana de aquellos estudiantes en riesgo de abandono, pues esto le permitirá a las IES adoptar diferentes medidas para mitigar el fracaso académico. Entre ellas se puede incluir, la asistencia individualizada de estudiantes, cursos de recuperación y sesiones de tutoría (Alvarado-Uribe et. al., 2022).

La UANL a través de su Plan de Desarrollo Institucional 2022-2030 convoca a incorporar diversas actividades y a fortalecer políticas institucionales que impacten en los índices de eficiencia terminal en escuelas y facultades.

La presente investigación está orientada a identificar los principales factores que desde el punto de vista académico inciden en la deserción estudiantil en la FCFM de la UANL, con el empleo de técnicas de ML y utilizando registros de calificaciones de los estudiantes en el período comprendido de enero – junio de 2015 hasta agosto – diciembre de 2022.

Por lo antes expuesto, nuestro proyecto pretende responder a las siguientes preguntas de investigación:

- ¿Los algoritmos de ML permiten clasificar, con métricas de desempeño aceptables, a los estudiantes en riesgo de abandono tomando como información el primer semestre, y el primer año?
- ¿Cuáles son los factores académicos que inciden en la deserción, tomando los resultados de los algoritmos de clasificación?

CAPÍTULO 3
JUSTIFICACIÓN

En México el abandono estudiantil se manifiesta en todos los niveles educativos, con especial énfasis en las IES. Por otro lado, la identificación temprana de estudiantes en riesgo y el análisis de los principales factores de deserción han generado una nueva línea de investigación educativa que permite a las universidades adoptar diferentes medidas para mitigar el fracaso académico. Esta problemática representa una oportunidad para la aplicación de los métodos estadísticos y algoritmos de ML a la información académica de los estudiantes (Alvarado-Uribe et. al., 2022).

El empleo de algoritmos de ML, unido a la disponibilidad de datos acumulados en los sistemas administrativos, posibilita el análisis y la detección de los factores académicos causantes de la deserción estudiantil en la FCFM, lo que permitirá implementar intervenciones oportunas, de apoyo a los estudiantes en riesgo.

Hasta donde sabemos, no existen estudios previos similares sobre la deserción estudiantil en la FCFM.

CAPÍTULO 4
FORMULACIÓN DE OBJETIVOS

4.1 Objetivo General

Aplicar algoritmos de *Machine Learning* para identificar factores académicos que ponen en riesgo de deserción a los estudiantes de la Facultad de Ciencias Físico Matemáticas de la Universidad Autónoma de Nuevo León.

4.2 Objetivos Específicos

- Utilizar algoritmos de *Machine Learning* para clasificar a los estudiantes en riesgo de deserción.
- Identificar los factores académicos que inciden en la deserción, tomando los resultados de los algoritmos de clasificación.
- Interpretar los resultados obtenidos y llegar a conclusiones sobre el abandono estudiantil.

CAPÍTULO 5
MARCO TEÓRICO

La deserción escolar en las IES ha sido objeto de estudio y preocupación a nivel mundial. Dicho fenómeno no solo afecta a los estudiantes que abandonan sus estudios, sino que también tiene implicaciones económicas, sociales y culturales para los individuos y las sociedades en su conjunto. En este sentido, la comprensión de los factores que inciden en la deserción escolar y la identificación de estrategias para prevenirla son fundamentales para garantizar una educación de calidad y para el desarrollo sostenible de las comunidades. En este capítulo se presenta un marco teórico que aborda el fenómeno de la deserción escolar considerando sus causas y sus implicaciones en los sistemas educativos tanto de América Latina como de México. Se parte de una visión general y se particulariza para el caso de las universidades y su incidencia en el fracaso académico. A continuación, se muestra la aplicación de la Ciencia de Datos para el análisis de información, en particular se describen algunas técnicas de ML para el procesamiento de datos educativos. Por último, se incluyen algunas investigaciones que han abordado el tema con el empleo de ML para la predicción de la deserción de los estudiantes. Este marco teórico constituye un referente fundamental para el diseño y desarrollo de la investigación que se presenta en esta Tesina.

5.1 Deserción Escolar

La definición exacta de este concepto resulta compleja debido a que no existen parámetros teóricos claros que lo delimiten, más allá del indicador con

el que se refiere al ausentismo o abandono de un estudiante de la institución donde se matriculó para cursar el año escolar (Rochin Berumen, 2021). A pesar de la complejidad del tema, autores como Spady (1970) asocian “la deserción escolar universitaria con cualquier persona que se retira de una institución educativa antes de recibir su diploma”.

Tinto (1982) define deserción como “una situación a la que se enfrenta un estudiante cuando aspira y no logra concluir su proyecto educativo”, considerando como desertor al estudiante de una institución de Educación Superior que no presenta actividad académica durante tres semestres académicos consecutivos. Otros autores se apoyan en la heterogeneidad del concepto y basan su análisis desde dos puntos de vista: el temporal y el espacial. La deserción con respecto al tiempo se clasifica, según Ruiz et. al. (2009) en:

- Precoz: individuo que habiendo sido admitido por la institución de Educación Superior no se matricula.
- Temprana: individuo que abandona sus estudios en los primeros semestres del programa.
- Tardía: individuo que abandona los estudios en los últimos semestres.

Mientras que la deserción espacial, se divide en:

- Institucional: se presenta cuando el alumno decide abandonar la institución.
- Interna o del programa académico: se presenta cuando el estudiante decide cambiarse a otro programa dentro de la misma universidad.

5.1.1 Causas de la Deserción Escolar Universitaria

Diversos modelos y teorías se han desarrollado para abordar el problema de la deserción escolar. Estos enfoques se centran en la evaluación de factores académicos, socioeconómicos, psicológicos y familiares de los estudiantes al inicio de sus estudios en la Educación Superior. Autores como, Santamaría y Bustos (2013) identifican las deficiencias en la preparación académica previa

como una de las principales causas de la deserción, lo que impide la integración social en el nuevo nivel educativo. Mientras que Torres Castillo (2012) apunta a la responsabilidad que recae en los programas académicos y su falta de actualización.

Donoso y Schiefelbein (2007) indican que la percepción que tienen los estudiantes sobre su capacidad para costear los gastos asociados con los estudios universitarios puede ser un factor económico relevante que influye en la decisión de abandonar. Para abordar esta problemática, las instituciones ofrecen opciones como créditos a largo plazo, tasas de interés relativamente bajas, subsidios parciales o totales, becas de matrícula y de alimentación, entre otras. Moreno Bernal (2013) sostiene que, paradójicamente, la misma institución educativa puede fomentar la deserción debido a los altos costos de matrícula, lo cual impide a los estudiantes de bajos recursos continuar con sus estudios.

En América Latina, el embarazo en la adolescencia es otra causa común de deserción escolar, porque los futuros padres deben asumir responsabilidades familiares que pueden afectar su desempeño académico. En este contexto, la investigación de Moreno Torres et. al. (2016) resalta las desventajas que enfrentan las mujeres en cuanto a las oportunidades educativas.

Otro elemento importante para tener en cuenta es la escolaridad de los padres, así como la convivencia o falta de ella con sus hijos. Ciertos factores como la desmotivación, el desinterés por la escuela, las adicciones o la violencia también pueden contribuir al abandono escolar (RUIZ, 2018).

La desmotivación y actitud del estudiante son igualmente contribuyentes significativos a la deserción escolar. De modo que, tanto las instituciones educativas con sus programas, como los docentes con sus prácticas pedagógicas deben jugar un papel clave para abordar este problema. Aunque la actitud de los estudiantes puede estar influenciada por su desarrollo personal, la trayectoria escolar también tiene un impacto en su desarrollo social. Según Donoso y Schiefelbein (2007), es importante que todos los miembros de la

comunidad educativa se interesen por los problemas que enfrentan los estudiantes y ofrezcan actividades que promuevan un ambiente de integración.

Además de lo anterior, sería beneficioso crear programas de orientación y motivación vocacional para aquellos que estudian una carrera universitaria. Esto no solo beneficiaría a los individuos, sino también a la sociedad en general. El propósito es mejorar la actitud de las personas hacia el logro de sus objetivos, lo que puede conducir al éxito académico y personal. Si se puede lograr un cambio positivo en la mentalidad de los estudiantes, y si se tiene en cuenta el desarrollo individual de cada uno, podrían tener las herramientas necesarias para enfrentar además de los desafíos académicos, los personales. El reto consiste en prevenir el abandono escolar, pero para lograrlo es necesario analizar cuidadosamente los diversos factores que pueden causarlo.

5.1.2 Deserción Escolar en América Latina y México

Años atrás, las investigaciones que abordaban las diferentes causas por las cuales un estudiante decidía abandonar sus estudios universitarios en América Latina eran escasas, al igual que las posibles soluciones educativas encaminadas a aumentar la permanencia de los estudiantes dentro del sistema de Educación Superior. Afortunadamente, la deserción estudiantil constituye actualmente una de las problemáticas que aborda la mayoría de las universidades de todo el continente (Díaz Peralta, 2008).

Las diversas investigaciones publicadas dan cuenta de un considerable número de estudiantes que no logran culminar sus estudios universitarios, con el consecuente costo social asociado. Como consecuencia, algunos países han comenzado a diseñar procesos de mejora para aumentar la retención en los primeros años de estudios universitarios (Palacios-Pacheco et. al., 2019).

Estudios sobre deserción realizados en universidades de Chile, Colombia, Argentina, Uruguay, Puerto Rico, entre otros, muestran desde diferentes perspectivas la problemática de la deserción escolar. A través de las distintas publicaciones se muestran los esfuerzos realizados por todos los entes

participantes en los respectivos sistemas de educación para el seguimiento y evaluación de la deserción estudiantil, así como de las acciones diseñadas para disminuir la deserción. Según cálculos del Instituto Internacional de la UNESCO para la Educación Superior en América Latina y el Caribe (IESALC), en el 2005 el costo de la deserción fue estimado en US$11.1 billones de dólares al año para 15 países de América Latina y el Caribe (Ruiz et al., 2009).

En el caso particular de México, la Secretaría de Educación Pública (SEP) define el abandono escolar de la siguiente forma: número de alumnos que dejan la escuela de un ciclo escolar a otro, por cada cien alumnos que se matricularon al inicio de cursos de un mismo nivel educativo (SEP, 2019).

El abandono escolar forma parte de la triada de indicadores de eficiencia más representativa del sistema educativo en México (reprobación, abandono y eficiencia terminal). La información obtenida a partir de su análisis permite determinar, entre otros, el tiempo exacto en años que permanecen dentro del sistema educativo los estudiantes que abandonan definitivamente la escuela. El abandono se puede dar en dos momentos: el que ocurre durante el ciclo escolar (intracurricular) y el que se da al finalizar el ciclo escolar (intercurricular). La suma de ambos es el abandono que se denomina abandono escolar según los lineamientos para la formulación de indicadores educativos de la Secretaría de Educación Pública (2019).

Según el extinto Instituto Nacional para la Evaluación de la Educación en México [INEE] (2017), (ahora Comisión Nacional para la Mejora Continua de la Educación), los problemas relacionados con la deserción escolar son causados por factores tanto intersistémicos como intrasistémicos. Los intersistémicos se refieren a la oferta educativa, la desigualdad en la calidad de los servicios educativos y los mecanismos de acceso (designación de escuela o facultad, modalidad y turno). Por otro lado, los elementos intrasistémicos están relacionados con prácticas pedagógicas inconvenientes, un personal docente poco preparado, condiciones laborales limitadas, equipamiento escaso y un currículo poco pertinente, adicionalmente una gestión escolar deficiente y una

participación limitada de padres y estudiantes en la escuela (Rochin Berumen, 2021).

5.1.3 Deserción / Abandono Estudiantil en las Universidades

Las IES se encuentran sujetas constantemente a procesos de evaluación por parte de entidades gubernamentales, así como de organismos internacionales que se encargan del aseguramiento de la calidad académica. Entre los principales factores que se referencian en los procesos de evaluación están: el abandono de estudios (deserción), la repitencia y la eficacia académica.

Medir la calidad de la educación es difícil, pero las tasas de graduación son una indicación preocupante de que algunas cosas no están funcionando. Para las universidades se ha convertido en una tarea muy compleja detectar las posibles causas de la deserción. Estas causas hasta hace unos años se basaban en factores como la falta de información previa sobre la carrera y la dificultad del estudiante para adaptarse a un entorno universitario. Actualmente, los estudios realizados sobre el tema incluyen nuevas variables que pueden ser analizadas y buscan dar respuesta a los porcentajes de influencia que tienen para que el estudiante abandone sus estudios (Palacios-Pacheco et. al., 2019).

Un informe publicado por la UNESCO muestra la evolución de la matrícula de Educación Superior de 2000 a 2018, con un aumento del 19% al 38%. A pesar de las disparidades regionales, América Latina y el Caribe muestran avances significativos. Sin embargo, también se señala la brecha que existe entre las tasas de matriculación y graduación en la Educación Superior. El informe concluye que los países deben prestar atención a las tasas de deserción, así como a las tasas de progresión, lograr no solo altas tasas de matriculación, que midan el proceso, sino también altas tasas de graduación, que midan el resultado de sus esfuerzos. Sugiere, además, prestar especial atención a la inclusión y permanencia de los estudiantes vulnerables (UNESCO, 2020).

Desde el punto de vista institucional, todos los estudiantes que abandonan su Educación Superior pueden ser clasificados como desertores. Es así como varios autores asocian la deserción con los fenómenos de “mortalidad” académica y retiro forzoso. En este sentido, cada estudiante que abandona la institución crea un lugar vacante que pudo ser ocupado por otro alumno que permaneciera en sus estudios, por lo cual la pérdida de estudiantes causa serios problemas financieros a las instituciones al producir inestabilidad en la fuente de sus ingresos (Tinto 1982). Sin embargo, no es claro que todos los tipos de abandono requieran la misma atención o exijan similares formas de intervención por parte de la institución, siendo ésta la gran dificultad que enfrentan las instituciones educativas. El conocimiento de estas diferencias constituye la base para elaborar políticas universitarias eficaces con el fin de aumentar la retención estudiantil.

Es importante analizar los factores que influyen en la deserción escolar a nivel universitario. La atención a los estudiantes universitarios, especialmente a los nuevos ingresos, es crucial para comprender los factores que contribuyen a la deserción escolar. El primer año universitario es un período crítico en el que se toma la decisión de continuar o abandonar los estudios (Silva Laya, 2011). Es esencial acompañar a estos estudiantes para enfrentar tanto las dificultades externas, como las variables personales, en particular la autoestima. Debemos tener en cuenta que los estudiantes de nuevo ingreso se enfrentan a un nuevo ámbito académico con responsabilidades con las que no están familiarizados. La familia también debe involucrarse como un apoyo invaluable para sus hijos, ya que impactaría positivamente en su motivación, conducta y rendimiento académico (Rochin Berumen, 2021).

5.1.3.1 Panorámica en la UANL

En el documento que recoge el Plan de Desarrollo Institucional UANL 2022-2030, se analiza la tasa de reprobación promedio de los programas educativos que ofrece cada dependencia, donde se aprecia una disminución

respecto a etapas anteriores. Sin embargo, en algunos programas y facultades este indicador sigue siendo muy alto, lo que requiere de especial atención a corto plazo (UANL, 2022).

La tasa de eficiencia terminal para el 2021 para las facultades de la UANL cerró en un rango muy amplio, entre el 9% y el 67%, lo que da cuenta de amplias brechas de eficiencia en la operación de los programas de licenciatura que es necesario superar para incrementar el índice de competitividad académica y hacer realidad la Visión 2030. Para ello se plantea que deben incorporarse y fortalecerse diversas actividades y políticas institucionales, entre las que destacan la revisión de las prácticas docentes centradas en el aprendizaje (componente importante del Modelo Educativo de la UANL), la sistematización del programa institucional de tutorías, la incorporación del seguimiento de egresados y su utilización en la actualización de los programas educativos, entre otras (UANL, 2022).

El referido Plan de Desarrollo Institucional 2022-2030 señala como imprescindible dar un impulso renovado a la mejora continua de la calidad de los procesos de gestión académico-administrativa que permitan contar con elementos para dar un seguimiento más preciso de los índices de retención y eficiencia terminal en cada una de las escuelas.

5.1.3.2 Antecedentes de Estudios de Deserción Escolar en la UANL

Tal y como se ha expuesto anteriormente, las universidades desempeñan un papel clave en la prevención de la deserción escolar. La UANL a través de sus escuelas y facultades se ha insertado en el estudio de esta problemática. Muestra fehaciente son las numerosas investigaciones respecto al tema incluidas en el repositorio académico digital de esta institución.

La tesis de Rodríguez Schaeffer (2000), se inscribió como uno de los primeros intentos por descubrir las causas que influían significativamente en que los estudiantes de licenciatura abandonaran tempranamente sus estudios. Se logró esbozar las causas de deserción en la Facultad de Ingeniería

Mecánica y Eléctrica (FIME), a través de encuestas a los desertores. A partir de entonces y hasta hoy, disímiles investigaciones se han desarrollado entorno a la deserción estudiantil, analizando desde diferentes perspectivas los factores incidentes en el abandono escolar: académicos (Rodríguez Pérez, 2013), sociodemográficos y sociales (Guerra Turrubiates, 2020), económicos (Gómez Triana, 2013). Otros trabajos de tesis y artículos contienen propuestas encaminadas a la prevención de este fenómeno dentro de la universidad: como programas de tutorías y otras estrategias educativas (Gadea Cavazos et al., 2011; Fernández, 2014).

5.2 Ciencia de Datos

El concepto de la Ciencia de Datos es definido por Tukey (1962) como un conjunto de procedimientos para analizar datos, técnicas para interpretar los resultados de dichos procedimientos, formas de planificar la recopilación de datos para hacer su análisis más fácil, más preciso o acertado, y toda la maquinaria y los resultados de las estadísticas matemáticas que se aplican al análisis de datos, con el objetivo de aprender cada vez más acerca de los datos y sus modelos, para lograr una mejor precisión en cuanto al conocimiento del negocio.

De acuerdo con lo expresado por Moreno Salinas (2017), hoy como nunca estamos más conectados con personas y dispositivos, con más acceso a redes y servicios, consumiendo mayor cantidad de datos e información, por lo cual se requiere contar con las habilidades, conocimientos, experiencias y técnicas de los científicos de datos para procesar, analizar y visualizar de formas más inteligentes los datos en información, promoviendo así más y mejores conocimientos de nuestra realidad en sus contextos, en especial con los datos de la organización.

5.2.1 Inteligencia Artificial

El objetivo de la inteligencia artificial (IA) es dotar a las computadoras con habilidades similares a las que puede realizar la mente. Algunas de estas habilidades, como el razonamiento, suelen ser consideradas "inteligentes", mientras que otras, como la visión, no lo son. Sin embargo, todas ellas implican competencias psicológicas, como la percepción, la asociación, la predicción, la planificación y el control motor, que permiten a los seres humanos y otros animales alcanzar sus objetivos. La inteligencia no se limita a una única dimensión, sino que abarca un amplio espacio de capacidades diversas para procesar la información. De manera similar, la IA emplea una variedad de técnicas para resolver una amplia gama de tareas (Boden, 2017).

Podríamos decir que "pensar en IA es pensar en ordenadores", pero eso sería parcialmente cierto. Los ordenadores en sí no son el foco central; lo realmente relevante es lo que pueden hacer. En otras palabras, aunque la IA depende de máquinas físicas (ordenadores), es más preciso considerar que utiliza lo que los especialistas en sistemas llaman "máquinas virtuales". Estas máquinas virtuales no son réplicas de máquinas en un entorno de realidad virtual, ni se asemejan a un motor de automóvil simulado para estudiar mecánica. En realidad, son sistemas de procesamiento de información que los programadores conciben al escribir un programa y que los usuarios tienen en mente al utilizarlo (Boden, 2017).

5.2.2 Aprendizaje Automático (*Machine Learning*)

Según Camargo García (2020), el enfoque de aprendizaje automático es ampliamente utilizado en procesos de clasificación y predicción de datos. El ML es una forma de IA que permite a un sistema aprender de los datos en lugar de usar programación explícita. Se genera un modelo predictivo entrenando un algoritmo de ML con datos, lo que resulta en una salida de información para generar un pronóstico basado en los datos que se utilizaron para entrenar el

modelo. Algunos métodos de ML incluyen clasificadores bayesianos, clasificadores basados en instancias, máquinas de soporte vectorial, árboles de decisión, redes neuronales artificiales, lógica difusa, bosques aleatorios y modelos de Markov, entre otros. A continuación, se refieren algunos algoritmos predictores individuales, el método de aprendizaje en conjunto (*ensemble)* y algoritmos que son tipo *ensemble*.

5.2.2.1 Máquinas de Soporte Vectorial (SVM)

En los años 90, Boser et al. (1992) desarrollaron las SVM como una técnica de aprendizaje estadístico para la clasificación binaria. Desde entonces, se han utilizado para resolver problemas de optimización de agrupamiento, clasificación múltiple, regresión y en diversas áreas como identificación de imágenes, clasificación de texto e hipertexto, procesamiento de lenguaje natural y estudio de series temporales.

Las SVM mapean los vectores de entrada no linealmente a un espacio de atributos de alta dimensión y construyen un hiperplano de separación lineal en ese espacio. Son capaces de realizar tanto predicciones numéricas como clasificaciones y su objetivo es minimizar el error cuadrático en la clasificación, creando un hiperplano que separe con precisión un conjunto de datos de alta dimensionalidad. La estrategia es seleccionar un hiperplano óptimo que se sitúe equidistante de las clases, logrando un margen máximo a cada lado del hiperplano. Para definir los vectores de soporte, sólo se tienen en cuenta aquellos de cada clase que caen justo en la frontera de los márgenes. Las propiedades especiales de este hiperplano garantizan una alta capacidad de generalización del modelo de aprendizaje.

Las SVM de clasificación binaria discriminan puntos de datos pertenecientes a dos posibles categorías. Estos puntos de datos pertenecen exclusivamente a una de las dos clases y un clasificador lineal los separa mediante un hiperplano, tal y como se puede apreciar en la Figura 1, donde

observamos múltiples clasificadores lineales que logran separar de manera acertada estas dos clases en un ejemplo bidimensional.

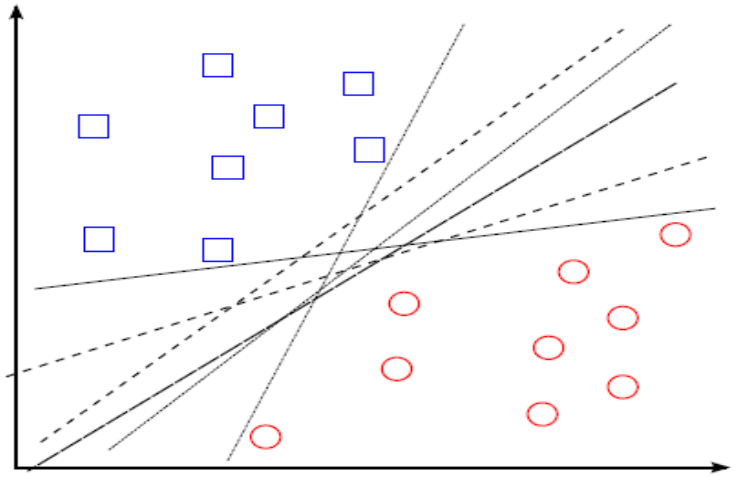


Figura 1. Hiperplanos de separación en un espacio bidimensional de un conjunto separable en dos clases. Fuente: Carmona Suárez, 2016.

Las SVM son sensibles a las escalas de características, como se puede ver en la Figura 2: a la izquierda, la escala vertical es mucho más grande que la escala horizontal, por lo que la calle más ancha posible está cerca de la horizontal. Después de escalar características (por ejemplo, usando *StandardScaler* de *scikit-learn*), el límite de decisión en la gráfica derecha se ve mucho mejor.

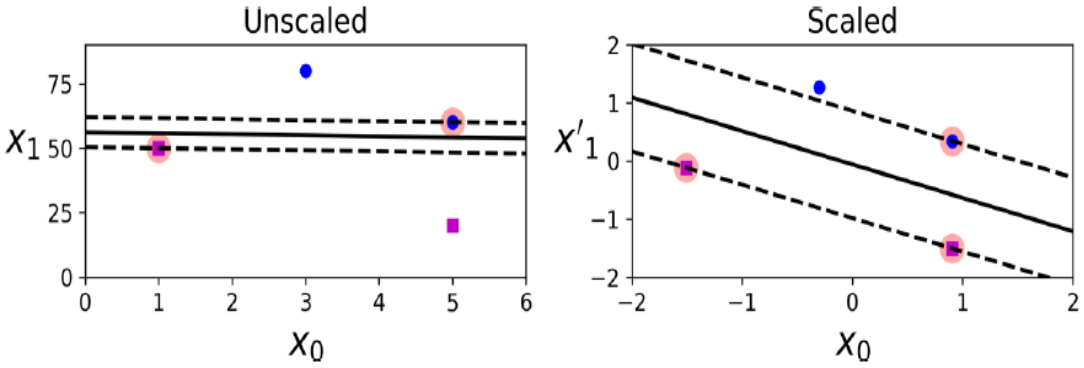


Figura 2. Sensibilidad al escalado de características. Fuente: Géron, 2019.

Según Betancourt (2005), las fortalezas de las SVM incluyen su entrenamiento fácil, la ausencia de óptimos locales, el buen escalado para datos de alta dimensión, la posibilidad de contrastar claramente la relación entre el error y la complejidad del clasificador, y la capacidad de utilizar cadenas de

caracteres y árboles como entrada en lugar de vectores de características. Sin embargo, las SVM también tienen debilidades, como la necesidad de contar con una buena función *kernel* y la requerida eficiencia en la recepción de los parámetros de inicialización.

5.2.2.2 Regresión Logística

La regresión logística (LR) es una técnica de modelado estadístico que se utiliza para analizar la relación entre una variable dependiente categórica y una o más variables independientes continuas o categóricas, es decir, puede utilizarse en análisis multivariados y puede emplearse en análisis univariados para examinar la relación entre una variable dependiente categórica y una única variable independiente (Hosmer et al., 2013). Por tanto, la LR es una herramienta estadística versátil que puede utilizarse en diversos contextos de análisis, de uso tanto explicativo como predictivo. Resulta útil su empleo cuando se tiene una variable dependiente dicotómica (un atributo cuya ausencia o presencia hemos puntuado con los valores cero y uno, respectivamente) y un conjunto de variables predictoras o independientes, que pueden ser cuantitativas (que se denominan covariables o covariadas) o categóricas. En este último caso, se requiere que sean transformadas en variables "*dummy*", es decir: variables simuladas (Chitarroni, 2002).

La LR se emplea para correlacionar la probabilidad de ocurrencia de una variable cualitativa binaria con un conjunto de variables escalares, predice la probabilidad de ocurrencia de un evento binario utilizando una función *logit*, cuyos datos siguen una distribución binomial como muestra la siguiente expresión:

$$Y_i \sim B(p_i, n_i), para\ i = 1,2, \dots, m \tag{5.1}$$

Donde:

- n_i representa el número de ensayos Bernoulli, que son conocidos.
- p_i representa las probabilidades de éxito, que son desconocidas.
- m representa el número de observaciones o casos diferentes.

Los valores de las probabilidades binomiales desconocidas conforman el modelo de regresión logística, dado por:

$$logit(p) = \ln\left(\frac{p}{1-p}\right) = x_i^T\beta = \beta_0 + \sum_{j=1}^{p}\beta_j x_j = \eta \qquad (5.2)$$

La probabilidad de pertenencia a cualesquiera de las dos categorías en el suceso se aproxima a través de la función logística que resulta de realizar transformaciones algebraicas a (5.2):

$$p_i = logit^{-1}\,\eta = \frac{1}{1+e^{-\eta}} \qquad (5.3)$$

La identificación del mejor modelo de regresión logística se realiza mediante la comparación de modelos utilizando el cociente de verosimilitud, que indica, a partir de los datos de la muestra, cuánto más probable es un modelo frente al otro. La diferencia de los cocientes de verosimilitud entre dos modelos se distribuye según la ley de la Chi-cuadrada con los grados de libertad correspondientes a la diferencia en el número de variables entre ambos modelos. Si a partir de este coeficiente no se puede demostrar que un modelo resulta mejor que el otro, se considerará como el más adecuado, el más sencillo (Silva Fuente-Alba & Molina Villagra, 2017).

A continuación, se describen las fortalezas y debilidades del algoritmo de modo general:

- Fortalezas: fácil de entender e interpretar, útil para problemas lineales bien definidos.
- Debilidades: puede tener un desempeño bajo cuando hay interacciones no lineales entre las características.

5.2.2.3 K Vecinos más Cercanos (KNN)

KNN, siglas en inglés para "*K-Nearest Neighbors*", es un algoritmo de aprendizaje automático supervisado utilizado para la clasificación y la regresión. Su principio básico es que los objetos similares se agrupan en regiones similares. De acuerdo con (Bruce et al., 2020), el KNN para cada registro clasificado o predicho realiza los siguientes pasos:

1. Encontrar los "K" registros que tengan características similares (es decir, valores predictores similares).
2. Para la clasificación, averiguar cuál es la clase mayoritaria entre esos registros similares.
3. Para la predicción (también llamada regresión KNN), encontrar el promedio entre aquellos similares y predice ese promedio para el nuevo registro.

Se basa en la identificación de los ejemplares de entrenamiento más cercanos en el espacio de las características para clasificar objetos, utiliza el voto mayoritario de los vecinos más cercanos para asignar un objeto a una clase determinada. El número de vecinos utilizados se determina por un valor "K", que es un entero positivo.

Es importante tener en cuenta que el valor de "K" puede afectar significativamente la precisión del algoritmo. Si "K" es demasiado pequeño, el modelo puede ser demasiado sensible a las fluctuaciones en los datos y no generalizar bien a nuevos datos. Si "K" es demasiado grande, el modelo puede perder detalles importantes en los datos y clasificar o regresar valores incorrectos.

Por otro lado, la similitud (cercanía) se determina utilizando una métrica de distancia, que es una función que mide qué tan lejos están dos registros entre sí. La métrica de distancia más popular entre dos vectores es la distancia euclidiana.

Sean $\vec{u} = (u_1, u_2, \dots, u_p)$ y $\vec{v} = (v_1, v_2, \dots, v_p)$ dos vectores del espacio de características, la distancia euclidiana $D_e(\vec{u}, \vec{v})$ se expresa como:

$$D_e(\vec{u}, \vec{v}) = \sqrt{(u_1 - v_1)^2 + (u_2 - v_2)^2 + \cdots + (u_p - v_p)^2} \qquad (5.4)$$

Otra métrica de distancia común para los datos numéricos es la distancia de Manhattan $D_M(\vec{u}, \vec{v})$, dada por:

$$D_M(\vec{u}, \vec{v}) = |u_1 - v_1| + |u_2 - v_2| + \cdots + |u_p - v_p| \qquad (5.5)$$

Existen muchas otras métricas para medir la distancia entre vectores. Para los datos numéricos, la distancia de Mahalanobis $D_{mh}(\vec{u},\vec{v})$, expresada en (5.6) con $\sum^{-1}$ siendo la inversa de la matriz de covarianza, es atractiva porque explica la correlación entre dos variables. Esto es útil ya que, si dos variables están altamente correlacionadas, Mahalanobis esencialmente las tratará como una sola variable en términos de distancia; a diferencia de la distancia euclidiana y la de Manhattan que no explican la correlación, colocando efectivamente un mayor peso en el atributo que subyace a esas características (Bruce et al., 2020).

$$D_{mh}(\vec{u},\vec{v}) = \sqrt{(\vec{u}-\vec{v})^T \sum^{-1} (\vec{u}-\vec{v})} \tag{5.6}$$

Al medir la distancia entre dos vectores, las variables (características) que se miden con una escala comparativamente grande dominarán la medida. Por lo tanto, las variables de relación no contarían prácticamente nada en comparación. Este problema se resuelve estandarizando los datos. En la medición, a menudo no estamos tan interesados en "cuánto" sino en "qué tan diferente del promedio". La estandarización, también llamada normalización, pone todas las variables en escalas similares restando la media y dividiendo por la desviación estándar, como muestra la ecuación (5.7); el resultado de esta transformación se conoce comúnmente como una puntuación z. Las mediciones se representan en forma de "desviaciones estándar de la media" para evitar que una variable ejerza una influencia excesiva en un modelo debido únicamente a la escala de su medición original.

$$z = \frac{x-\mu}{\sigma} \tag{5.7}$$

La normalización en este contexto estadístico no debe confundirse con la normalización de la base de datos, que es la eliminación de datos redundantes y la verificación de las dependencias de datos (Bruce et al., 2020).

En resumen, KNN es un algoritmo simple pero potente para la clasificación y regresión de datos, especialmente en casos donde los datos son

no-lineales y el número de características es pequeño. A continuación, se describen las fortalezas y las debilidades de modo general:

- Fortalezas: simple y fácil de implementar, útil en conjuntos de datos pequeños.
- Debilidades: puede ser muy sensible al ruido y no funciona bien con datos dispersos.

5.2.2.4 *Ensemble*

Supongamos que planteamos una pregunta compleja a miles de personas al azar y luego agregamos sus respuestas. En muchos casos, encontraremos que esta respuesta agregada es mejor que la respuesta de un experto. A esto se denomina la sabiduría de la multitud. Del mismo modo, si agregamos las predicciones de un grupo de predictores (como clasificadores o regresores), a menudo obtendremos mejores predicciones que con el mejor predictor individual. Un grupo de predictores se llama conjunto; por lo tanto, esta técnica se conoce como *Ensemble Learning*, y un algoritmo de *Ensemble Learning* se llama método *Ensemble* (Géron, 2019). La Figura 3 muestra un esquema del método.

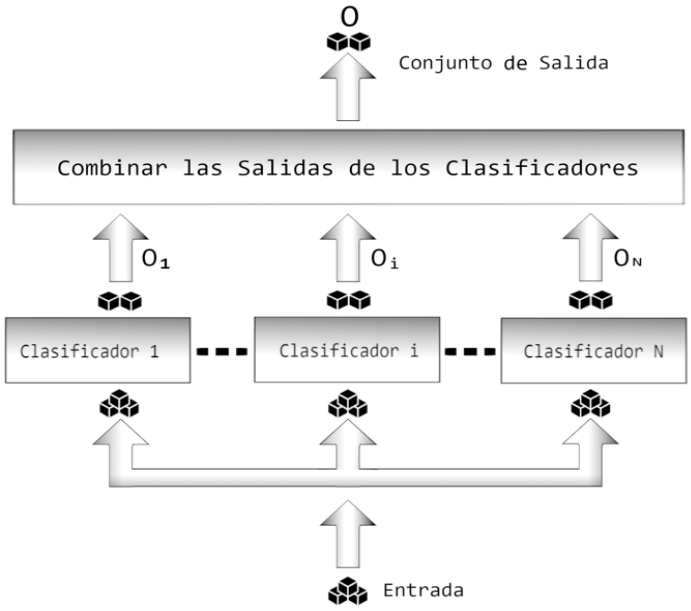


Figura 3. Esquema del método *Ensemble*. Fuente: Shi et al., 2010.

Los métodos de conjunto son algoritmos de aprendizaje que construyen un conjunto de clasificadores y luego clasifican nuevos puntos de datos tomando un voto (ponderado) de sus predicciones. El método de conjunto original es el promedio bayesiano, pero los algoritmos más recientes incluyen la codificación de salida para corrección de errores, la agregación de arranque y el impulso. Mediante el uso de estos conjuntos, normalmente se obtiene un mejor rendimiento que cualquier clasificador individual, al existir metodologías bien establecidas para obtener clasificadores altamente precisos mediante la combinación de clasificadores menos precisos (Dietterich, 2000).

El aprendizaje por conjuntos (*Ensemble Learning*) es un método de aprendizaje automático que consiste en construir un conjunto de clasificadores básicos y combinar sus predicciones para clasificar nuevos ejemplos. En la comunidad de aprendizaje automático, este enfoque ha ganado considerable popularidad y se posiciona como una de las principales direcciones en el campo en la actualidad. El conjunto de clasificadores básicos se entrena primero con los ejemplos de entrenamiento, y luego se combinan las predicciones de cada uno de ellos para producir la salida del conjunto. Para mejorar la precisión de las predicciones, el aprendizaje por conjuntos emplea algoritmos como el *bagging* (*bootstrap aggregation)* y el *boosting*, que son ampliamente reconocidos y populares en este enfoque.

El *bagging* es como el algoritmo básico para conjuntos, excepto que, en lugar de ajustar los diversos modelos a los mismos datos, cada nuevo modelo se ajusta a un remuestreo de *bootstrap*. A continuación, se muestra el algoritmo presentado más formalmente por Bruce et al. (2020):

1. Inicializar M, el número de modelos que se ajustarán, y n, el número de registros a elegir ($n < N$). Establezca la iteración $m = 1$.
2. Tomar una remuestra de arranque (es decir, con reemplazo) de n registros de los datos de entrenamiento para formar una submuestra Y_m y X_m (la bolsa).

3. Entrenar un modelo usando Y_m y X_m para crear un conjunto de reglas de decisión $\hat{f}_m(X)$.
4. Incremente el contador del modelo $m = m + 1$. Si $m <= M$, vaya al paso 2.

En el caso en que $\hat{f}_m$ predice la probabilidad $Y = 1$, la estimación en bolsas viene dada por:

$$\hat{f} = \frac{1}{M}\left(\hat{f}_1(X) + \hat{f}_2(X) + \cdots + \hat{f}_M(X)\right) \tag{5.8}$$

Hay varios algoritmos de impulso (*boosting*), y la idea básica detrás de todos ellos es esencialmente la misma. El más fácil de entender es *Adaboost*, que procede de la siguiente manera (Bruce et al., 2020):

1. Inicialice M, el número máximo de modelos que se ajustarán, y establezca el contador de iteración $m = 1$. Inicialice los pesos de observación $w_i = 1/N$ para $i = 1, 2, \ldots, N$. Inicialice el modelo de conjunto $\hat{F}_0 = 0$.
2. Usando los pesos de las observaciones $w_1, w_2, \ldots, w_N$, entrene un modelo $\hat{f}_m$ que minimice el error ponderado e_m definido por la suma de los pesos de las observaciones clasificadas erróneamente.
3. Agregar el modelo al conjunto: $\hat{F}_m = \hat{F}_{m-1} + \alpha_m \hat{f}_m$, para valores de $\alpha_m = \frac{\ln(1-e_m)}{e_m}$.
4. Actualizar los pesos $w_1, w_2, \ldots, w_N$, de modo que se aumenten los pesos para las observaciones que fueron mal clasificadas. El tamaño del aumento depende de α_m, donde valores más grandes de α_m conducen a pesos más grandes.
5. Incremente el contador del modelo $m = m + 1$. Si $m <= M$, vaya al paso 2.

La estimación impulsada viene dada por:

$$\hat{F} = \alpha_1 \hat{f}_1 + \alpha_2 \hat{f}_2 + \cdots + \alpha_M \hat{f}_M \tag{5.9}$$

5.2.2.4.1 Bosque Aleatorio (*Random Forest*)

Al igual que los SVM, los árboles de decisión son algoritmos versátiles de aprendizaje automático que pueden realizar tareas de clasificación y regresión, e incluso tareas de salidas múltiples. Son algoritmos poderosos, capaces de adaptarse a conjuntos de datos complejos. Una de las muchas cualidades de los árboles de decisión es que requieren muy poca preparación de datos. De hecho, no requieren escalado o centrado de características en absoluto. Los árboles de decisión son también los componentes fundamentales de los bosques aleatorios, que se encuentran entre los algoritmos de aprendizaje automático más potentes disponibles en la actualidad (Géron, 2019).

El *Random Forest* (RF) es un enfoque de *Ensemble Learning*, desarrollado por L. Breiman (2001), para resolver problemas de clasificación y regresión. Como hemos mencionado, el aprendizaje en conjunto es un esquema de aprendizaje automático para aumentar la precisión mediante la integración de múltiples modelos para resolver el mismo problema. En particular, varios clasificadores participan en la clasificación de conjuntos para obtener resultados más precisos en comparación con un solo clasificador. En otras palabras, la integración de múltiples clasificadores disminuye la varianza, especialmente en el caso de clasificadores inestables, y puede producir resultados más confiables. Seguidamente, se diseña un escenario de votación para asignar una etiqueta a muestras no etiquetadas.

Según Hastie et al. (2017), el *bagging* es una técnica de aprendizaje de conjunto que reduce la varianza de una función de predicción estimada. El *bagging* parece funcionar especialmente bien para procedimientos de alta varianza y bajo sesgo, como ocurre con los árboles de decisión. Para la clasificación, un conjunto de árboles predice la clase de un punto de prueba, o voto, y la predicción del bosque aleatorio será la clase que tenga mayoría de votos. Los bosques aleatorios (Breiman, 2001) son una modificación sustancial del *bagging* que construye una gran colección de árboles no correlacionados, y luego los promedia. En muchos problemas, el rendimiento de los bosques

aleatorios es muy similar al *boosting*, y son fáciles de entrenar y afinar. Como consecuencia, al azar los bosques son populares y se implementan en una variedad de paquetes. A continuación, mostramos el pseudocódigo de RF para regresión y para clasificación (Hastie et al.,2017):

1. $Para\ b\ =\ 1\ hasta\ B$:
    a) Obtener una muestra de inicio Z^* de tamaño N a partir de los datos de entrenamiento.
    b) Crear un árbol de bosque aleatorio T_b para los datos de inicio, repitiendo recursivamente los siguientes pasos para cada nodo terminal del árbol, hasta que se alcance el tamaño mínimo de nodo n_{min}.
        i. Seleccionar m variables al azar de entre las p variables.
        ii. Elegir la mejor variable o punto de división entre los m.
        iii. Dividir el nodo en dos nodos hijos.
2. Salida del conjunto de árboles $\{T_b\}_1^B$

Para hacer una predicción en un nuevo punto x:

- *Regresión:* $\hat{f}_{rf}^B(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$
- *Clasificación:* Sea $\hat{C}_b(x)$ la predicción de clase del $b-ésimo$ árbol del bosque aleatorio. Entonces, $\hat{C}_{rf}^B(x) = voto\ mayoritario\ \{\hat{C}_b(x)\}_1^B$

Donde B es el número de árboles, $\hat{C}_b(x)$ define el voto del $b-ésimo$ árbol y $C_1^B = \{\hat{C}_1(x), \hat{C}_2(x), \dots, \hat{C}_B(x)\}$.

Para nuestro trabajo usamos el RF para la clasificación. Con el modelo entrenado, para clasificar una nueva muestra x se deben analizar los votos de cada árbol presente en el modelo creado en el entrenamiento. Matemáticamente, el bosque aleatorio se gobierna de acuerdo con la ecuación:

$$\hat{C}_b(x) = \arg\max_c \sum_{i=1}^{B} \delta_{c,\hat{C}_i(x)} \qquad (5.10)$$

$$sujeto\ a\ c \in \{1,2,\dots,m\}$$

Donde $\delta_{i,j}$ (delta de Kronecker) está dada por:

$$\delta_{i,j} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \tag{5.11}$$

Según se ha señalado, los árboles de decisión tienen una alta varianza. Siempre se intenta reducir la varianza del modelo ajustando los parámetros. Una forma de mejorar el resultado en este sentido es ajustando el número de entidades seleccionadas para cada árbol (parámetro p en el pseudocódigo). Al aumentar este parámetro, el modelo estará menos sujeto a la varianza del conjunto de datos. Sin embargo, el aumento de este valor conduce a un aumento en el sesgo del sistema. En problemas de clasificación, generalmente se utiliza como valor de p la raíz cuadrada del número total de características (Marins, 2016).

En relación con la cantidad de árboles empleados en RF, esta cantidad puede variar según la disponibilidad de capacidad de procesamiento computacional. A medida que se incrementa el número de árboles, el modelo se vuelve más preciso y menos propenso a la variabilidad. Sin embargo, esta mejora en los resultados eventualmente se estabiliza, alcanzando un punto en el cual agregar más árboles no brinda beneficios significativos en términos de eficiencia. A partir de ese punto, el aumento continuo de árboles puede tener un efecto similar al sobreajuste, reduciendo la capacidad de generalización del modelo. Por lo tanto, al ajustar estos parámetros, es importante considerar la posibilidad de que ocurra un sobreajuste (Marins, 2016).

Según Marins (2016), en los problemas donde los datos tienen dimensiones muy grandes, los criterios para elegir los puntos de separación no son aplicables a primera vista. Para los problemas de clasificación los más comunes son: mediante la obtención de información y mediante el índice de Gini.

5.2.2.4.2 *Gradient Boosting*

Gradient Boosting (GB) es una técnica de aprendizaje automático supervisado utilizada para la regresión y la clasificación, que también sigue un

enfoque de aprendizaje en conjunto. Su objetivo es construir un modelo predictivo fuerte combinando varios modelos predictivos débiles (también conocidos como "árboles de decisión débiles") en una serie. Cada árbol se construye para corregir los errores del modelo anterior, de modo que el modelo final tenga en cuenta todos los errores previos y genere mejores predicciones.

El proceso de entrenamiento del algoritmo de GB se puede dividir en tres partes:

1. Construcción del primer árbol: se entrena un modelo predictivo débil, generalmente un árbol de decisión simple, utilizando los datos de entrenamiento.
2. Construcción de árboles adicionales: se construyen varios árboles adicionales de manera secuencial. Cada árbol se ajusta a los residuos (diferencia entre las predicciones del modelo actual y los valores reales) generados por el modelo anterior. El objetivo es minimizar los residuos del modelo actual.
3. Predicción final: se calcula la predicción final sumando las predicciones de todos los árboles. El resultado final es un modelo de GB que puede usarse para hacer predicciones en datos nuevos.

GB utiliza un enfoque de optimización de gradiente para minimizar la función de pérdida del modelo. La función de pérdida puede variar dependiendo del problema que se esté abordando, como la regresión o la clasificación. El pseudocódigo asociado a este algoritmo en regresión queda del siguiente modo (Hastie et al.,2017):

1. Inicializar $f_0(x) = \arg\min_{\gamma} \sum_{i=1}^{N} L(y_i, \gamma)$
2. $For\ m = 1\ to\ M$:
   a) $For\ i = 1\ to\ N$:

$$r_{im} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f=f_{m-1}}$$

   b) Ajustar un árbol de regresión al borde de los objetivos r_{im} dando regiones terminales $R_{jm}, j = 1, 2, \dots, J_m$

c) $For\ m = 1\ to\ J_m$:

$$\gamma_{jm} = \arg\min_{\gamma} \sum_{x_i \in R_{jm}} L(y_i, f_{m-1}(x_i) + \gamma)$$

d) Actualizar $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$

3. Devuelve $\hat{f}(x) = f_{kM}(x), para\ k = 1, 2, \dots, K$

El algoritmo para clasificación es análogo, las líneas del paso 2 se repiten "K" veces en las "m" iteraciones, una vez para cada clase usando:

$$-g_{ikm} = \left[\frac{\partial L(y_i, f_1(x_i), f_2(x_i), \dots, f_K(x_i))}{\partial f_k(x_i)}\right]_{f(x_i)=f_{m-1}(x_i)} \quad (5.12)$$

O equivalentemente:

$$-g_{ikm} = I(y_i = G_k) - p_k(x_i) \quad (5.13)$$

Cada árbol T_{km} se ajusta a su respectivo vector de gradiente negativo g_{km}. Para la clasificación, la función de pérdida es la desviación multinomial dada en (5.12), y "K" árboles de mínimos cuadrados se construyen en cada iteración. Como resultado en la línea 3 se obtienen "K" diferentes expansiones de árboles (acoplados). Estos producen probabilidades a través de (5.13) o hacen clasificación como en (5.14).

$$L\big(y, p(x)\big) = -\sum_{k=1}^{K} I(y = G_k)\, f_k(x) + \ln\Big(\sum_{l}^{K} e^{f_l(x)}\Big) \quad (5.14)$$

$$p_k(x) = \frac{e^{f_k(x)}}{\sum_{l=1}^{K} e^{f_l(x)}} \quad (5.15)$$

$$G(x) = G_k, donde\ k = arg\max_{l} p_l(x) \quad (5.16)$$

En resumen, GB es un método de aprendizaje automático que combina varios modelos débiles en una serie para construir un modelo predictivo fuerte. Cada modelo en la serie se ajusta a los residuos generados por el modelo anterior, lo que mejora la precisión del modelo final. A continuación, se describen las fortalezas y debilidades de modo general:

- Fortalezas: gran capacidad para manejar problemas complejos, muy preciso y útil en grandes conjuntos de datos.
- Debilidades: más difícil de ajustar que otros modelos y más propenso a sobreajuste.

5.2.3 Técnicas y Medidas para Evaluar Desempeño

Los modelos de ML necesitan evaluarse con el objetivo de calibrar sus parámetros y evaluar su mejor rendimiento, y así, comparar entre ellos para detectar aquel con mejor desempeño. En esta sección hablaremos sobre la matriz de confusión y las métricas que se derivan de esta, así como de las curvas ROC y AUC. Se incluyen conceptos de muestreo como el *K-Fold Cross Validation* y métodos para medir el impacto de las variables del modelo.

5.2.3.1 Matriz de Confusión

Los resultados de los modelos pueden ser medidos a través de tablas de contingencias, llamadas matrices de confusión. La Figura 4 muestra la estructura de una matriz de confusión para dos clases.

A partir de los resultados en la experimentación, plasmados en la matriz de confusión podemos calcular varias métricas que nos permitirán evaluar los modelos correspondientes. Las fórmulas propuestas por Bruce et al. (2020) para realizar el cálculo con dos clases, se muestran desde (5.17) a (5.21).

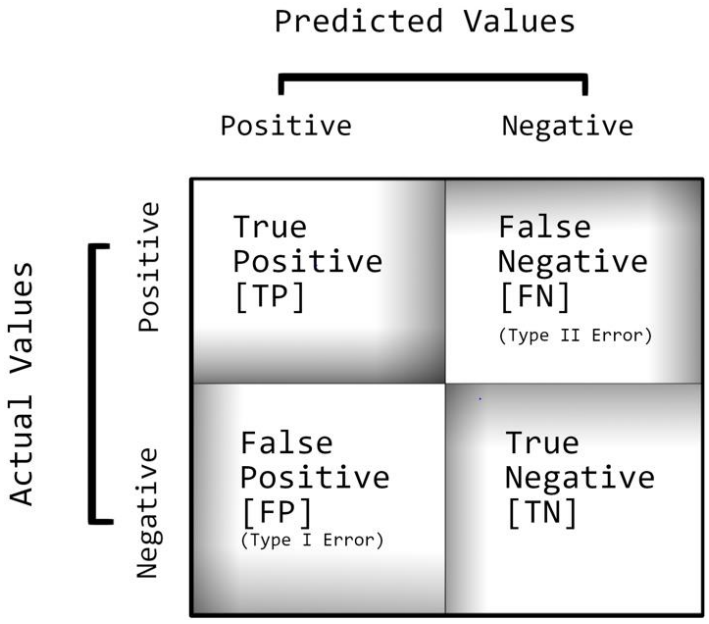


Figura 4. Matriz de confusión para dos clases. Fuente: elaboración propia.

$$precision_P = \frac{\sum TP}{\sum TP + \sum FP} \quad (5.17)$$

$$recall_P = \frac{\sum TP}{\sum TP + \sum FN} \quad (5.18)$$

$$specificity = recall_N = \frac{\sum TN}{\sum TN + \sum FP} \quad (5.19)$$

$$accuracy = \frac{\sum TP + \sum TN}{\sum TP + \sum FP + \sum TN + \sum FN} \quad (5.20)$$

$$f_1 - score_P = 2 * \frac{precision_P * recall_P}{precision_P + recall_P} \quad (5.21)$$

Donde:

- TP: número de puntos clasificados correctamente como positivos.
- FP (error de Tipo I): número de puntos clasificados como positivos cuando en realidad son negativos.
- TN: número de puntos clasificados correctamente como negativos.
- FN (error de Tipo II): número de puntos clasificados como negativos cuando en realidad son positivos.

La explicación de las métricas anteriores se describe, a continuación:

- La precisión (*precision*): es utilizada para medir los patrones positivos que se predicen correctamente del total de patrones predichos en una clase positiva.
- La sensibilidad – exhaustividad (*recall*): es utilizada para medir las etiquetas positivas reales (TP) que se predicen correctamente.
- La especificidad (*specificity*): mide la proporción de verdaderos negativos (TN) identificados correctamente en relación con el número total de negativos reales.
- La exactitud (*accuracy*): es utilizada para evaluar la capacidad de generalización de los clasificadores, sobre la base del total de instancias que el clasificador entrenado predice correctamente usando datos no vistos.
- La media armónica entre la precisión y la exhaustividad es representada por el *"f1-score",* y se utiliza para evaluar modelos de clasificación en situaciones desequilibradas.

Las tasas de falsos positivos/negativos a menudo se confunden o se combinan con la especificidad o la sensibilidad (¡incluso en publicaciones y software!). A veces, la tasa de falsos positivos se define como la proporción de verdaderos negativos que dan positivo. En muchos casos (como la detección de intrusiones en la red), el término se utiliza para referirse a la proporción de señales positivas que son verdaderos negativos (Bruce et al., 2020).

5.2.3.2 Índice de Kappa de Cohen

El índice de Kappa de Cohen (K) se utiliza para evaluar si la concordancia observada entre los evaluadores es mayor que la concordancia que podría esperarse simplemente por azar. En otras palabras, mide el acuerdo corrigiendo el acuerdo observado por el acuerdo esperado debido al azar y se calcula utilizando la siguiente expresión:

$$\mathrm{K} = \frac{p_0 - p_e}{1 - p_e} \tag{5.22}$$

Donde:

- p_0: acuerdo relativo observado entre los observadores.
- p_e: probabilidad hipotética de acuerdo por azar.

El valor máximo de concordancia alcanzable es $\mathrm{K} = 1$. Asimismo, un valor de $\mathrm{K} = 0$ se logra cuando la concordancia observada es igual a la que se esperaría puramente por azar. Si la concordancia es mayor que la esperada por azar, entonces $\mathrm{K} > 0$, mientras que, si es menor, $\mathrm{K} < 0$. El valor mínimo posible de K está determinado por las distribuciones marginales. Cuando se trata de interpretar el valor de K, resulta beneficioso contar con una escala, ilustrada en la Tabla 1, a pesar de que su establecimiento sea subjetivo (de Ullibarri Galparsoro, 1999).

Tabla 1. Escala para el índice de Kappa de Cohen. Fuente: de Ullibarri Galparsoro, 1999.

Valoración del índice de Kappa de Cohen	
Valor de K	Fuerza de la Concordancia
< 0.20	Pobre
0.21 - 0.40	Débil
0.41 - 0.60	Moderada
0.61 - 0.80	Buena
0.81 - 1.00	Muy buena

Según Fleiss et al. (1969), la Kappa se utiliza para evaluar el grado de concordancia entre dos evaluadores en una escala nominal. Ellos refieren que las fórmulas para calcular el error estándar de esta estadística, presenta errores que tienden a sobreestimar los valores, lo que lleva a realizar pruebas de significancia y a establecer intervalos de confianza de manera más cautelosa al emplearla. Del mismo modo, proporcionan fórmulas válidas para calcular la varianza aproximada en muestras grandes e ilustran su cálculo a través de un ejemplo numérico en este trabajo investigativo.

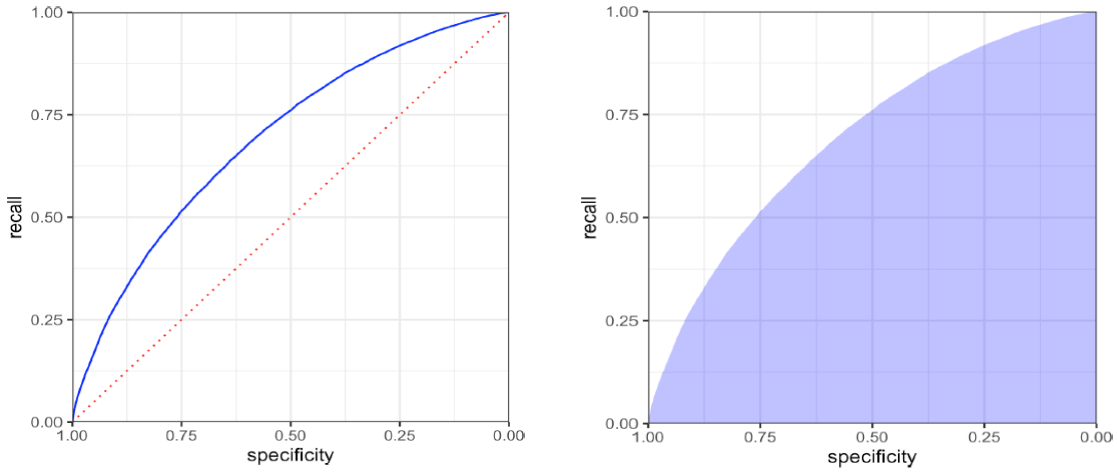
5.2.3.3 Curva ROC

Entre las técnicas más apropiadas para evaluar clasificadores en problemas con clases desequilibradas se encuentra la curva ROC (*Receiver Operating Characteristic*), que es una herramienta para visualizar, organizar y seleccionar clasificadores en función de su equilibrio entre beneficios (TP) y costos (FP) (V. García et al., 2012). Esta curva resume todas las matrices de confusión producidas por los diferentes umbrales. De acuerdo con Bruce et al. (2020) el proceso para calcular la curva ROC, cuya representación gráfica se observa en la Figura 5a), es:

1. Ordenar los registros por la probabilidad prevista de ser un “1”, comenzando con el más probable y terminando con los menos probables.
2. Calcular la especificidad acumulada y el *recall* en función de los registros ordenados.

5.2.3.4 Curva AUC

La curva ROC es una herramienta gráfica valiosa, sin embargo, no proporciona una medida única del rendimiento de un clasificador. La curva ROC se puede utilizar para producir la métrica de área debajo de la curva (AUC: *Area Under the Curve*), como se muestra en la Figura 5b). AUC es simplemente el área total bajo la curva ROC. Cuanto mayor sea el valor del AUC, más efectivo será el clasificador. Un AUC de "1" indica un clasificador perfecto: obtiene todos los "1" correctamente clasificados, y no clasifica erróneamente ningún "0" como "1".



a) ROC b) AUC

Figura 5. Curvas ROC y AUC. Fuente: Bruce et al., 2020.

El uso del AUC como métrica para evaluar un modelo es una mejora sobre la precisión simple, porque puede evaluar qué tan bien un clasificador maneja la compensación entre la precisión general y la necesidad de identificar los "1" más importantes. Pero no aborda completamente el problema de casos raros, donde debe reducir el límite de probabilidad del modelo por debajo de 0.5 para evitar que todos los registros se clasifiquen como "0". En tales casos, para que un registro se clasifique como "1", podría ser suficiente tener una probabilidad de 0.4, 0.3 o inferior. En efecto, terminamos identificando más "1", reflejando su mayor importancia (Bruce et al., 2020).

Cambiar este límite mejorará sus posibilidades de atrapar los “1” (a costa de clasificar erróneamente más “0” como “1”). Esto nos lleva a preguntarnos: ¿cuál es el límite óptimo? El concepto de elevación nos permite diferir la respuesta a esa pregunta. En su lugar, considera los registros en orden de su probabilidad prevista de ser “1”. Digamos, del 10% superior clasificado como “1”, ¿cuánto mejor lo hizo el algoritmo, en comparación con el punto de referencia de simplemente elegir a ciegas? Un gráfico de elevación (gráfico de ganancias) cuantifica esto sobre el rango de los datos. Se puede producir decil por decil, o continuamente sobre el rango de los datos. La curva de elevación es la relación entre las ganancias acumuladas y la línea diagonal correspondiente a la selección aleatoria. Los gráficos de ganancias de deciles son una de las técnicas más antiguas en el modelado predictivo, que data de los días anteriores al comercio por Internet. Una curva de elevación le permite ver las consecuencias de establecer diferentes límites de probabilidad para clasificar registros como “1”. Puede ser un paso intermedio para establecer un nivel de corte apropiado (Bruce et al., 2020).

Bruce (2020) relaciona algunas ideas que se deben tener presente:

- El *accuracy* (el porcentaje de clasificaciones predichas que son correctas) no es más que un primer paso en la evaluación de un modelo.
- Otras métricas (*recall, specificity, precision*) se centran en características de rendimiento más específicas (por ejemplo, el *recall* mide qué tan bueno es un modelo para identificar correctamente “1”).
- AUC (área bajo la curva ROC) es una métrica común para la capacidad de un modelo para distinguir “1” de “0”.
- La elevación mide qué tan efectivo es un modelo para identificar los “1”, y a menudo se calcula decil por decil, comenzando con los “1” más probables.

5.2.3.5 *K-Fold Cross Validation*

Una de las técnicas más simples y ampliamente utilizadas para estimar el error de predicción es la validación cruzada, que generalmente proporciona una buena estimación del error esperado. Si tuviéramos suficientes datos, podríamos reservar un conjunto de validación para evaluar el rendimiento de nuestro modelo de predicción. Sin embargo, dado que los datos pueden ser escasos, esto suele no ser posible. Para abordar este problema, la validación cruzada de *K-Fold* utiliza una parte de los datos disponibles para ajustar el modelo y otra parte para probarlo, permitiendo así refinar la evaluación del modelo (Hastie et al., 2017). Dividimos los datos en K partes aproximadamente del mismo tamaño. En la Figura 6, se muestra el escenario cuando $K = 5$.

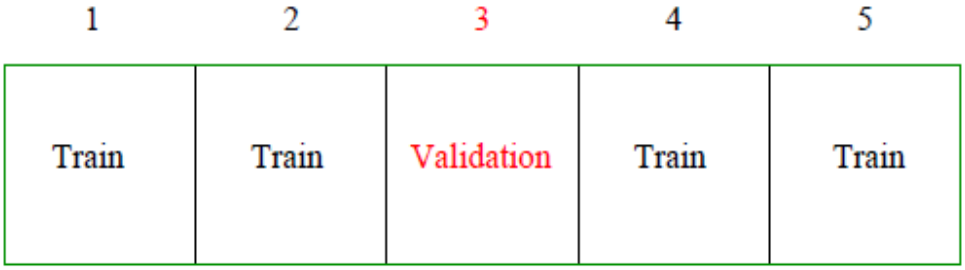


Figura 6. *K-Fold Cross Validation* para 5 pliegues. Fuente: Hastie et al., 2017.

Para la $k - ésima$ parte (tercera en la imagen) tratamos de calcular el error de predicción, entrenando el modelo con los otros $K - 1$ pliegues, y esto se realiza para $k = 1, 2, \ldots, K$ (5 veces para la imagen) combinando las K estimaciones del error de predicción. Con un mayor número de observaciones en los pliegues de entrenamiento, se puede evitar una sobreestimación del error de predicción.

5.2.3.6 *One Hot Encoder*

La técnica *"one hot encoder"* es un tipo común de codificación utilizado en la comunidad de aprendizaje automático en la que se conservan todos los niveles de factor; útil para ciertos algoritmos de aprendizaje automático, aunque

este enfoque no es apropiado para la regresión lineal múltiple (Bruce et al., 2020). Es una técnica que se usa para convertir datos categóricos en datos numéricos. Las variables factoriales no ordenadas generalmente se convierten en un conjunto de variables binarias $(0/1)$ utilizando esta técnica. En la regresión lineal y logística, esta codificación causa problemas con multicolinealidad; en tales casos, se omite y su valor se puede inferir de otros valores. Esto no es un problema con KNN.

La librería *scikit-learn* de *Python* proporciona la clase *"OneHotEncoder"* para convertir valores categóricos en vectores de $(0/1)$.

5.2.3.7 Análisis de SHAP

El análisis de SHAP (*Shapley Additive exPlanation*) se utiliza para interpretar la salida de un modelo de aprendizaje automático. Se basa en la idea de la Teoría de Juegos de Shapley, que es una técnica utilizada para asignar el valor a los jugadores en un juego cooperativo.

Este método requiere volver a entrenar el modelo en todos los subconjuntos de características $S \subseteq F$, donde F es el conjunto de todas las características. Asigna un valor de importancia a cada entidad que representa el efecto en la predicción del modelo de incluir esa característica. Los valores SHAP son una medida unificada de la importancia de las características.

El cálculo exacto de los valores SHAP es un desafío. Sin embargo, al combinar los conocimientos de los métodos actuales de atribución de características aditivas, podemos aproximarlos. Cuando se utilizan estos métodos, la independencia de características y la linealidad del modelo son dos supuestos opcionales que simplifican el cálculo de los valores esperados (Lundberg & Lee, 2017).

La librería *"shap"* de *Python* está provista de varias funciones que permiten la realización de los cálculos de valores SHAP para cada muestra de prueba y generar gráficos de resumen de los valores SHAP que nos permiten entender la importancia relativa de cada característica en la predicción del

modelo. Es importante tener en cuenta que el análisis de SHAP puede ser computacionalmente costoso para conjuntos de datos grandes y modelos complejos, y es posible que se necesiten técnicas de reducción de la complejidad, como análisis de componentes principales (PCA, por sus siglas en inglés) o análisis discriminante lineal (LDA), para manejar este problema. En nuestro contexto, el análisis SHAP se utilizó para determinar la importancia de cada característica en la salida del modelo correspondiente. El análisis de SHAP proporciona una explicación individual para cada predicción del modelo y muestra cómo cada característica contribuye a la salida final.

5.2.3.8 Análisis de PDP

Una vez identificadas las variables más relevantes, se puede aplicar un análisis de PDP (*Partial Dependence Plots*) e intentar comprender la naturaleza de la dependencia de la aproximación $f(X)$ de sus valores conjuntos. Las representaciones gráficas de $f(X)$ en función de sus argumentos proporcionan un resumen completo de su dependencia de los valores conjuntos de las variables de entrada (Hastie et al., 2017).

La visualización se encuentra limitada en dimensiones bajas. Es sencillo mostrar las funciones con uno o dos argumentos, ya sean continuos, discretos o una combinación de ambos, de diversas formas. Sin embargo, resulta más complicado visualizar las funciones de argumentos en dimensiones superiores, especialmente cuando se trata de más de dos o tres variables.

Una alternativa útil en ocasiones es observar una colección de gráficos, donde cada uno muestra la dependencia parcial de la aproximación $f(X)$ en un pequeño subconjunto seleccionado de las variables de entrada. Esto nos permite obtener una idea de cómo la función de aproximación se relaciona con cada subconjunto de variables, a pesar de las dificultades que surgen al visualizar dependencias en dimensiones superiores.

A partir de un vector X_S de dimensión $l < p$, contenido en el vector de las variables predictoras de entrada $X^T = (X_1, X_2, \dots, X_p)$ e indexado por

$S \subset \{1,2,\ldots,p\}$, sea C el complemento de S, de modo que $S \cup C = \{1,2,\ldots,p\}$. Una función general $f(X)$ dependerá en principio de todas las variables de entrada: $f(X) = f(X_S, X_C)$. Una forma de definir la dependencia media o parcial de $f(X)$ de X_S es:

$$f_S(X_S) = E_{X_C} f(X_S, X_C) \tag{5.23}$$

La ecuación (5.23) muestra un promedio marginal de f, y puede servir como una descripción útil del efecto del subconjunto elegido sobre $f(X)$ cuando, por ejemplo, las variables en X_S no tienen interacciones fuertes con las de X_C. Las funciones de dependencia parcial se pueden utilizar para interpretar los resultados de cualquier método de aprendizaje de "caja negra" (*Support Vector Machine, Random Forest, Gradient Boosting*, entre otros). Se pueden estimar usando la siguiente expresión:

$$\bar{f}_S(X_S) = \frac{1}{N} \sum_{i=1}^{N} f(X_S, x_{iC}) \tag{5.24}$$

Donde $\{x_{1C}, x_{2C}, \ldots, x_{NC}\}$ son los valores de X_C que aparecen en los datos de entrenamiento. Las funciones de dependencia parcial definidas en (5.24) representan el efecto de X_S sobre $f(X)$ después de tener en cuenta los efectos (promedio) de las otras variables X_C sobre $f(X)$. Estas funciones no representan el efecto de X_S en $f(X)$ ignorando los efectos de X_C. Este último viene dado por la expectativa condicional:

$$\tilde{f}_S(X_S) = E(f(X_S, X_C) | X_S) \tag{5.25}$$

Esta es la mejor aproximación de mínimos cuadrados dependiendo sólo de X_S. Las cantidades $\tilde{f}_S(X_S)$ y $\bar{f}_S(X_S)$ serán las mismas sólo en el improbable caso de que X_S y X_C sean independientes (Hastie et al., 2017).

El análisis PDP, en nuestro trabajo, mostró cómo el modelo correspondiente cambia su predicción a medida que se varía una variable predictora, manteniendo constantes las demás variables. Para ello, generamos y observamos la gráfica de algunos de los factores más influyentes según otros análisis, haciendo énfasis en aquellos factores que estaban relacionados con

las experiencias planteadas por los Coordinadores Académicos de cada uno de los programas educativos de licenciatura que se imparten en la FCFM.

5.2.3.9 Optimización Multiobjetivo

En nuestro trabajo se utilizó una metodología fundamentada en los conceptos de frontera de Pareto de optimización multiobjetivo para llevar a cabo comparaciones entre los factores que ejercen la mayor influencia en la clasificación de estudiantes en riesgo. En esta sección, proporcionaremos una descripción general de los conceptos que aplicamos en nuestra investigación.

El problema general de optimización multiobjetivo se plantea de la siguiente manera:

$$\begin{aligned} &\underset{x\in E^n}{\text{Minimizar}}\, F(x) = [F_1(x), F_2(x), \dots, F_k(x)]^T \\ &sujeto\ a\ g_j(x) \leq 0,\ j = 1,2,..,m \\ &h_l(x) = 0,\ l = 1,2,..,e \end{aligned} \tag{5.26}$$

donde "k" es el número de funciones objetivo, "m" es el número de restricciones de desigualdad y "e" es el número de restricciones de igualdad. La $x \in E^n$ es un vector de variables de diseño (también llamadas variables de decisión), donde "n" es el número de variables independientes x_i. $F(x) \in E^k$ es un vector de funciones objetivo $F_i(x)$: $E^n \rightarrow E^1$. También se les llama objetivos, criterios, funciones de pago, funciones de coste o funciones de valor.

Todo método de solución de (5.26) debe usar la comparación entre vectores(" $\leq$ "," $\geq$ ", $etc.$), misma que se implementa mediante una relación de orden definida sobre el espacio de los objetivos, por ejemplo, la comparación componente a componente es una relación de orden que cumple que $a \leq b$ (donde a y b son vectores) sí, y sólo si $a_i \leq b_i$ para todo $i \in 1, \cdots, n$ (Marler & Arora, 2004).

Cuando trabajamos con un problema de optimización multiobjetivo, en el espacio de salida (donde están las soluciones) hablamos de eficiencia y el espacio de llegada (imágenes de esas soluciones) es el de objetivos, en el que

hablamos de no dominancia. Es decir, eficiencia y no dominancia difieren. En el trabajo se emplearon las definiciones que exponemos a continuación (Steuer, 1989):

- Definición de eficiencia (Pareto): un punto, $x^* \in X$, es eficiente sí, y sólo si no existe otro punto, $x \in X$, tal que $F(x) \leq F(x^*)$ donde se cumpla al menos un $F_i(x) < F_i(x^*)$. De lo contrario, x^* es ineficiente.
- Definición de frontera eficiente: el conjunto de todos los puntos eficientes se llama frontera eficiente.
- Puntos no dominados y dominados: un vector de funciones objetivo, $F(x^*) \in Z$, es no dominado sí, y sólo si no existe cualquier otro vector $F(x) \in Z$, tal que $F(x) \leq F(x^*)$ con al menos un $F_i(x) \leq F_i(x^*)$. De lo contrario, $F(x^*)$ está dominado.

La solución al problema de optimización multiobjetivo en el contexto de Pareto no será única, sino que estará conformada por el conjunto de todos los vectores que no son dominados por otros, también conocido como frente de Pareto. La metodología que se empleó en nuestro estudio reconoce la importancia de aplicar los conceptos de no dominancia al considerar distintos ordenamientos de los factores que influyen en la deserción estudiantil, obtenidos una vez que se aplicaron técnicas de interpretación a los algoritmos utilizados.

5.2.4 Datos

Previo a referirnos a los datos y habiendo mencionado tantos conceptos de ML, hagamos un paréntesis para puntualizar el panorama general, como propone Géron (2019):

- El aprendizaje automático consiste en hacer que las máquinas mejoren en alguna tarea aprendiendo de los datos, en lugar de tener que codificar explícitamente las reglas.
- Hay diferentes tipos de sistemas de ML: supervisados o no, por lotes o en línea, basados en instancias o basados en modelos.

- En un proyecto de aprendizaje automático, se recolectan datos en un conjunto de entrenamiento y se introducen en un algoritmo de aprendizaje. Si el algoritmo se basa en modelos, ajusta ciertos parámetros para adaptar el modelo al conjunto de entrenamiento (es decir, lograr buenas predicciones en relación con el conjunto de entrenamiento en sí mismo). Posteriormente, con suerte, también será capaz de hacer buenas predicciones sobre nuevos casos. Por otro lado, si el algoritmo está basado en instancias, simplemente aprende los ejemplos almacenados en memoria y generaliza a nuevas instancias utilizando una medida de similitud para compararlas con las instancias aprendidas.
- El sistema no funcionará bien si el conjunto de entrenamiento es demasiado pequeño, o si los datos no son representativos, son ruidosos o están contaminados con características irrelevantes (basura dentro, basura fuera). Por último, el modelo no debe ser ni demasiado simple (en cuyo caso se ajustará poco) ni demasiado complejo (en cuyo caso habría sobreajuste).
- Una vez que haya entrenado un modelo, no se desea simplemente "esperar" que se generalice a nuevos casos. Se debe evaluar y ajustar si es necesario.

5.2.4.1 Datos de Entrenamiento No Representativos

Para generalizar bien, es crucial que los datos de entrenamiento sean representativos de los nuevos casos a los que se quiere generalizar. Esto es cierto tanto para el aprendizaje basado en instancias como el aprendizaje basado en modelos. Es importante utilizar un conjunto de entrenamiento que sea representativo de los casos a los que desea generalizar. Esto es a menudo más difícil de lo que parece: si la muestra es demasiado pequeña, aparece ruido de muestreo (es decir, datos no representativos como resultado del azar), pero incluso las muestras muy grandes pueden no ser representativas si el

método de muestreo es defectuoso. Esto último se denomina sesgo de muestreo (Géron, 2019).

5.2.4.2 Datos de Mala Calidad

Si los datos de entrenamiento contienen errores, ya sean valores atípicos o ruido (por ejemplo, debido a mediciones de mala calidad), será más difícil para el sistema detectar los patrones subyacentes, por lo que es menos probable que el sistema funcione bien. Es recomendable dedicar tiempo a limpiar los datos de entrenamiento. La mayoría de los científicos de datos pasan una parte significativa de su tiempo haciendo precisamente eso. Si algunas instancias son claramente valores atípicos, puede ser útil simplemente descartarlas o intentar corregir los errores manualmente. Si a varias instancias les faltan algunas características, debe decidir si desea ignorar este atributo por completo, ignorar estas instancias, completar los valores que faltan o entrenar un modelo con la característica y un modelo sin ella (Géron, 2019).

5.2.4.3 Características Irrelevantes

El sistema sólo será capaz de aprender si los datos de entrenamiento contienen suficientes características relevantes y no demasiadas irrelevantes. Una parte crítica del éxito de un proyecto de aprendizaje automático es crear un buen conjunto de características para entrenar. Este proceso, denominado ingeniería de características, implica los siguientes pasos (Géron, 2019):

- Selección de características (selección de las características más útiles para entrenar, entre las características existentes).
- Extracción de características (combinación de características existentes para producir una más útil; así construimos algunas nuevas medidas en nuestro trabajo).
- Creación de nuevas características mediante la recopilación de nuevos datos (la incorporación de datos del Departamento de Asesorías).

5.2.4.4 Sobreajuste de los Datos de Entrenamiento

En ML el sobreajuste significa que el modelo funciona bien en los datos de entrenamiento, pero no generaliza bien. Los modelos complejos, como las redes neuronales profundas, pueden detectar patrones sutiles en los datos, pero si el conjunto de entrenamiento es ruidoso o si es demasiado pequeño (lo que introduce ruido de muestreo), es probable que el modelo detecte patrones en el ruido en sí. El sobreajuste ocurre cuando el modelo es demasiado complejo en relación con la cantidad y el ruido de los datos de entrenamiento. A continuación, relacionamos posibles soluciones (Géron, 2019):

- Simplificar el modelo seleccionando uno con menos parámetros (por ejemplo, un modelo lineal en lugar de un modelo polinómico de alto grado), reduciendo el número de atributos en los datos de entrenamiento o restringiendo el modelo.
- Recopilar más datos de entrenamiento.
- Reducir el ruido en los datos de entrenamiento (por ejemplo, corrija los errores de datos y elimine valores atípicos).

La técnica de restringir un modelo con el fin de simplificarlo y reducir el riesgo de sobreajuste se conoce como regularización. El objetivo es encontrar un equilibrio adecuado entre ajustar los datos de entrenamiento de manera óptima y mantener el modelo lo suficientemente simple para garantizar una buena generalización (Géron, 2019).

La cantidad de regularización a aplicar durante el aprendizaje puede ser controlada por un hiperparámetro. Un hiperparámetro es un parámetro de un algoritmo de aprendizaje (no del modelo). Como tal, no se ve afectado por el algoritmo de aprendizaje en sí; debe establecerse antes del entrenamiento y permanece constante durante el entrenamiento. Si establece el hiperparámetro de regularización en un valor muy grande, se obtendrá un modelo casi plano (una pendiente cercana a cero); es casi seguro que el algoritmo de aprendizaje no mostrará sobreajuste con los datos de entrenamiento, pero será menos

probable que encuentre una buena solución. El ajuste de hiperparámetros es una parte importante de la creación de un sistema de aprendizaje automático.

5.2.4.5 Ajuste Insuficiente de los Datos de Entrenamiento

El ajuste insuficiente es lo opuesto al sobreajuste: ocurre cuando el modelo es demasiado simple para aprender la estructura subyacente de los datos. Ocurre cuando la realidad es más compleja que el modelo, por lo que sus predicciones están destinadas a ser inexactas, incluso en los ejemplos de entrenamiento. Las principales opciones para solucionar este problema, según propone Géron (2019), son:

- Seleccionar un modelo más potente, con más parámetros.
- Alimentar mejores características al algoritmo de aprendizaje (ingeniería de características).
- Reducir las restricciones del modelo (por ejemplo, reducir el hiperparámetro de regularización).

5.2.4.6 Pruebas y Validación

La única manera de saber qué tan bien un modelo se generalizará a nuevos casos es probarlo realmente en nuevos casos. Una forma de hacerlo es poner el modelo en producción y monitorear su funcionamiento. Si el modelo no funciona bien, una mejor opción es dividir los datos en dos conjuntos: (1) el conjunto de entrenamiento y (2) el conjunto de prueba. (1) para entrenar el modelo, y (2) para probarlo. La tasa de error en casos nuevos se denomina error de generalización (o error fuera de muestra) y, al evaluar el modelo en el conjunto de pruebas, se obtiene una estimación de este error. Este valor indica si el modelo funciona bien en instancias nunca vistas. Si el error de entrenamiento es bajo (es decir, el modelo comete pocos errores en el conjunto de entrenamiento) pero el error de generalización es alto, significa que el modelo presenta sobreajuste con los datos de entrenamiento (Géron, 2019).

5.2.4.7 Ajuste de Hiperparámetros

"GridSearchCV" es una herramienta de la biblioteca *scikit-learn* en *Python,* que permite ajustar los hiperparámetros de un algoritmo de aprendizaje automático a través de una búsqueda exhaustiva en un espacio predefinido de hiperparámetros. Esta búsqueda se realiza mediante validación cruzada para evaluar el rendimiento del modelo con diversas combinaciones de hiperparámetros. En lugar de realizar manualmente el tedioso trabajo de ajustar los hiperparámetros hasta encontrar una combinación óptima, podemos utilizar *"GridSearchCV"*. Sólo necesitamos especificar los hiperparámetros en los que deseamos experimentar y los valores que queremos probar, siguiendo la documentación del algoritmo de clasificación que estamos ajustando. *"GridSearchCV"* utiliza la validación cruzada para evaluar todas las combinaciones posibles de valores de hiperparámetros. Si se inicializa *"GridSearchCV"* con *refit=True* (valor predeterminado), una vez que encuentra el mejor estimador mediante validación cruzada, lo reentrena con todo el conjunto de entrenamiento. Esto suele ser una buena estrategia, ya que alimentarlo con más datos probablemente mejorará su rendimiento (Géron, 2019).

De acuerdo con Géron (2019), el enfoque de búsqueda en cuadrícula está bien cuando se exploran relativamente pocas combinaciones, pero cuando el espacio de búsqueda del hiperparámetro es grande, a menudo es preferible usar *"RandomizedSearchCV",* también de la biblioteca *scikit-learn*, en su lugar. Esta clase se puede utilizar de la misma manera que la clase *"GridSearchCV"*, pero en lugar de probar todas las combinaciones posibles, evalúa un número dado de combinaciones aleatorias seleccionando un valor aleatorio para cada hiperparámetro en cada iteración. Este enfoque tiene dos beneficios principales:

- Si la búsqueda aleatoria se ejecuta durante 1.000 iteraciones, este enfoque explorará 1.000 valores diferentes para cada hiperparámetro (en lugar de sólo unos pocos valores por hiperparámetro con el enfoque de búsqueda en cuadrícula).

- Estableciendo el número de iteraciones, tiene más control sobre el presupuesto informático que desea asignar a la búsqueda de hiperparámetros.

"BayesSearchCV" es una herramienta de la biblioteca *skopt* que utiliza la optimización bayesiana para encontrar los valores de hiperparámetros que maximizan la métrica de evaluación en cada iteración. Esto significa que se puede explorar de manera más efectiva el espacio de hiperparámetros, lo que permite una búsqueda más eficiente y una mejor interpretación de los resultados. Para los ajustes de hiperparámetros, la que más utilizamos fue "*BayesSearchCV*".

Cuando se realizan múltiples ajustes de hiperparámetros, es común que el rendimiento general sea ligeramente inferior al que se obtuvo mediante validación cruzada. Esto ocurre porque el sistema se ajusta para funcionar bien con los datos de validación, pero es probable que no funcione tan bien con conjuntos de datos desconocidos. En este escenario, es importante resistir la tentación de ajustar los hiperparámetros sólo para que los números se vean bien en el conjunto de prueba. Las mejoras obtenidas probablemente no se generalicen a nuevos datos (Géron, 2019).

Una alternativa para ajustar su sistema es intentar combinar los modelos que presenten un mejor rendimiento. En general, un grupo o conjunto de modelos suele superar al mejor modelo individual, al igual que los bosques aleatorios superan a los árboles de decisión individuales en los que se basan. Esto es especialmente cierto cuando los modelos individuales cometen diferentes tipos de errores (Géron, 2019). En nuestro trabajo incorporamos esta idea.

La evaluación de un modelo es relativamente sencilla: simplemente se utiliza un conjunto de prueba. A la pregunta de cómo decidir entre diferentes modelos, una opción es entrenar los modelos y comparar su capacidad de generalización utilizando el conjunto de prueba. En algunos casos, se realiza una evaluación del error de generalización múltiples veces utilizando el conjunto de prueba. Esto permite ajustar el modelo y adaptar los hiperparámetros con el

objetivo de obtener el mejor rendimiento en ese conjunto específico. Sin embargo, esto implica que es poco probable que el modelo funcione igual de bien con datos nuevos.

Una solución común a este problema se llama validación de *holdout*: simplemente mantenga parte del conjunto de entrenamiento para evaluar varios modelos candidatos y seleccionar el mejor. El nuevo conjunto retenido se denomina conjunto de validación. En efecto, entrenamos varios modelos con varios hiperparámetros en el conjunto de entrenamiento reducido (es decir, el conjunto de entrenamiento completo menos el conjunto de validación) y seleccionamos el modelo que mejor funciona en el conjunto de validación. Después de este proceso de validación de resistencia, entrenamos el mejor modelo en el conjunto de entrenamiento completo (incluido el conjunto de validación), y esto da el modelo final. Por último, evaluamos este modelo final en el conjunto de pruebas para obtener una estimación del error de generalización.

Esta solución generalmente funciona bastante bien. Sin embargo, si el conjunto de validación es demasiado pequeño, las evaluaciones del modelo serán imprecisas: puede terminar seleccionando un modelo subóptimo por error. Por el contrario, si el conjunto de validación es demasiado grande, el conjunto de entrenamiento restante será mucho más pequeño que el conjunto de entrenamiento completo. Esto es malo dado que el modelo final se entrenará en el conjunto de entrenamiento completo y no es ideal comparar modelos candidatos entrenados en un conjunto de entrenamiento mucho más pequeño. Una forma de resolver este problema es realizar una validación cruzada repetida, utilizando muchos conjuntos de validación pequeños. Cada modelo se evalúa una vez por conjunto de validación después de que se entrena en el resto de los datos. Al promediar todas las evaluaciones de un modelo, se obtiene una medida mucho más precisa de su rendimiento. Hay un inconveniente: el tiempo de entrenamiento se multiplica por el número de conjuntos de validación (Géron, 2019).

5.2.4.8 Discrepancia de Datos

En ocasiones, es posible obtener una gran cantidad de datos para el entrenamiento, pero es probable que estos datos no sean perfectamente representativos de los datos que se utilizarán en producción. En tal caso, es fundamental recordar la regla principal: tanto el conjunto de validación como el conjunto de prueba deben ser lo más representativos posible de los datos que se esperan utilizar en producción. Por lo tanto, estos conjuntos deben estar compuestos exclusivamente por observaciones representativas, y es recomendable mezclar los datos en ambos conjuntos evitando que se duplique información. Luego de entrenar el modelo, si observa que el rendimiento del modelo en el conjunto de validación es decepcionante, no sabremos si esto se debe a que el modelo muestra sobreajuste con el conjunto de entrenamiento, o si esto se debe sólo a la falta de representatividad. Una solución es mantener algunas de las observaciones de entrenamiento en un conjunto *train-dev*. Después de entrenar el modelo (en el conjunto de entrenamiento, no en el conjunto *train-dev*), puede evaluarlo en el conjunto *train-dev*. Si funciona bien, entonces el modelo no muestra sobreajuste con el conjunto de entrenamiento. Si funciona mal en el conjunto de validación, el problema debe provenir de la falta de coincidencia de datos (Géron, 2019).

Por otro lado, un modelo es una versión simplificada de las observaciones. Las simplificaciones están destinadas a descartar los detalles superfluos que es poco probable que se generalicen a nuevas instancias. Para decidir qué datos descartar y qué datos para conservar, debe hacer suposiciones. En un famoso artículo de 1996, David Wolpert demostró que, si no se hace absolutamente ninguna suposición sobre los datos, entonces no hay razón para preferir un modelo sobre cualquier otro. Para algunos conjuntos de datos, el mejor modelo es un modelo lineal, mientras que para otros conjuntos de datos es una red neuronal. No hay ningún modelo que esté garantizado a priori para funcionar mejor. La única manera de saber con certeza qué modelo es mejor es evaluarlos todos. Dado que esto no es posible, en la práctica se

hacen algunas suposiciones razonables sobre los datos y se evalúan sólo unos pocos modelos razonables (Géron, 2019).

5.3 Aplicación de ML en Estudios de Deserción Escolar

La riqueza de datos acumulados en los sistemas administrativos educativos junto con el desarrollo de métodos eficientes de Estadística y ML han abierto enfoques novedosos para abordar el problema de la deserción estudiantil, generando una nueva línea de investigación. Como se ha referido anteriormente, se aprecia un aumento significativo en el número de investigaciones en el campo de la educación analítica predictiva. Además, se han desarrollado sistemas basados en IA para brindar apoyo en la toma de decisiones a las partes involucradas en la Educación Superior.

Algunos estudios se centran en la aplicación del análisis de aprendizaje (*Learning Analytics*) como estrategia prometedora para abordar los desafíos educativos persistentes en América Latina, como las altas tasas de deserción (Hilliger et. al., 2020; Namoun & Alshanqiti, 2020).

Por otro lado, diferentes estudios de predicción de la deserción se basan en medidas de rendimiento previas a la inscripción (calificaciones de la escuela secundaria, pruebas de evaluación) y datos personales; algunos también consideran indicadores de rendimiento universitario del primer semestre, como las calificaciones de los cursos. También se han encontrado otros factores con poder predictivo incremental sobre el rendimiento académico y la retención, como el alojamiento dentro o fuera del campus, el nivel socioeconómico, factores psicológicos como el afrontamiento y la inteligencia emocional, y antecedentes escolares de los padres. En este sentido, se deben proponer y aplicar principios éticos sobre la recopilación y el uso de datos educativos con el objetivo de proteger la privacidad de los estudiantes, como el principio ético de considerar el desempeño de los estudiantes como una variable dinámica (Alvarado-Uribe et. al., 2022).

Trabajos previos en el tema analizan la deserción considerando modelos y herramientas estadísticas, midiendo factores económicos y académicos, su cálculo principal está segmentado a si el estudiante se matricula o no en el siguiente período. La Tabla 2 muestra ejemplos de investigaciones centradas en el fenómeno de la deserción estudiantil aplicando algoritmos de ML.

Tabla 2. Selección de estudios sobre deserción escolar utilizando algoritmos de *Machine Learning.* Fuente: elaboración propia.

Autor/ Año	Objetivo	Resultados	Algoritmos
Jia & Mareboyana (2014)	Medir la retención de estudiantes de pregrado a través de la clasificación de datos estudiantiles	Los factores más influyentes en la retención de los estudiantes estaban asociados al promedio de calificaciones y la cantidad de créditos tomados	Árboles de decisión, SVM y redes neuronales compatibles con el kit de software WEKA
Rowtho (2017)	Detección temprana de la deserción en el Instituto Charles Telfair, Mauricio	Proponen una nueva técnica para determinar predictores significativos del rendimiento académico e identificar a los estudiantes en riesgo en las IES.	Regresión lineal y análisis discriminante lineal
Dicovskiy Riobóo & Pedroza Pacheco (2018)	La predicción de la deserción temprana en estudiantes universitarios	Se pudo clasificar los estudiantes que llegan a quinto año de forma exitosa o no, con un alto nivel de acierto (79%)	Estadística univariada y multivariada, con análisis discriminante
Solis et al. (2018)	Abandono estudiantil	Lograron una predicción correcta del 91% de los abandonos con una sensibilidad del 87%.	*Random Forest*, *Neural Networks*, *Support Vector Machines* and *Logistic Regression*
Manrique et al. (2019)	Detección de la deserción temprana.	Se concluye que la deserción se predice con precisión utilizando las calificaciones de algunos cursos básicos	Modelos predictivos y series de tiempo junto con algoritmos de aprendizaje apropiados para cada una de ellas.
Kemper et al. (2020)	Predecir el abandono de estudiantes utilizando los datos de exámenes realizados	Ambos métodos producen altas precisiones de predicción de hasta un 95% después de tres semestres, aunque los árboles de decisión dan mejores resultados.	Regresiones logísticas y árboles de decisión
Vivek Raj (2020)	Gestión del desempeño estudiantil para mejorar el rendimiento.	El algoritmo de árbol de representación superó al resto al clasificar a los estudiantes que tienen más probabilidades de reprobar los exámenes.	*Rep Tree*, *Jrip, Random Forest, Decision Tree* y *Naives Bayes*

CAPÍTULO 6
METODOLOGÍA

Con el objetivo de aplicar algoritmos para identificar factores que ponen en riesgo de abandono estudiantil, se emplearon modelos de clasificación de ML para tratar de predecir la deserción de estudiantes de la FCFM usando dos análisis:

1. Calificaciones por X semestres cursados, $X = 1,2$. Utilizando los datos de Kardex con las unidades de aprendizaje de los X semestres cursados, se analizó si las calificaciones eran factor para pronosticar la deserción.
2. Tiempo en avanzar, medido como el tiempo en terminar completamente Y semestres, con $Y = 1,2$. Utilizando los datos de Kardex hasta terminar todas las unidades de aprendizaje que completan el semestre Y; se analizó si el tiempo en avanzar es factor para pronosticar la deserción.

El objetivo fue encontrar factores que incidieran en la deserción para que las autoridades académicas busquen estrategias remediales y mitigarla. El factor carrera se utilizó cuando así se ameritó para mejorar las predicciones y se utilizaron variables artificiales creadas a partir del Kardex.

6.1 Población y Muestra de Análisis

El estudio tomó como población a todos los estudiantes pertenecientes a la FCFM en el período enero-junio 2015 hasta agosto-diciembre 2022, los cuáles se inscribieron a una de las seis carreras que relacionamos, a continuación:

- Licenciatura en Actuaría (LA)
- Licenciatura en Ciencias Computacionales (LCC)
- Licenciatura en Física (LF)
- Licenciatura en Matemáticas (LM)
- Licenciatura en Multimedia y Animación Digital (LMAD)
- Licenciatura en Seguridad en Tecnologías de Información (LSTI)

Se seleccionaron todas sus calificaciones correspondientes a 8 años, que representan 16 semestres de cada una de las carreras. Por el criterio de selección de la muestra, ésta fue no-probabilística, pues se eligieron los estudiantes pertenecientes al Modelo Educativo 420, siendo en ese período de tiempo uno de los modelos educativos con mayor cantidad de observaciones. Por Modelo Educativo 420, nos referimos al plan de estudios vigente para cada una de las carreras que se imparten en la FCFM, al momento de la realización de la presente investigación. El modelo tiene la peculiaridad de que todos los estudiantes inscritos cursan las mismas materias en el primer semestre, independientemente de la carrera a la que pertenezcan (tronco común, Anexo 1).

6.2 Área y Tipo de Estudio del Trabajo de Investigación

El área de estudio de nuestro trabajo de investigación fue la educación. Se realizó la recopilación de datos sobre las diferentes carreras de la FCFM durante un período de 8 años para estudiar la deserción estudiantil. El estudio fue longitudinal y de cohortes, pues al tiempo que se recopilaron datos de estudiantes durante el período mencionado, se establecieron estrategias de avances en períodos iniciales de sus carreras para identificar factores incidentes en la deserción o abandono. Además, nuestro estudio fue de predicción y de modelado, porque se utilizaron técnicas estadísticas y de modelado.

6.3 Métodos y Técnicas de Recolección de Datos

Se realizó una solicitud de la información contenida en los Kardex de los estudiantes durante un período de tiempo a las autoridades de la FCFM, la cual fue proporcionada para el uso de los datos respetando las políticas de privacidad y confidencialidad establecidas por la institución. Por lo tanto, para nuestra investigación, se utilizaron los datos que ya han sido recolectados y registrados por la institución, lo que se considera una recolección de datos secundarios.

Por otro lado, se realizaron entrevistas ocasionales a los Coordinadores Académicos, para identificar desde sus experiencias las materias más críticas en el modelo educativo seleccionado.

6.3.1 Recopilación de Datos

Los datos utilizados corresponden a los Kardex de estudiantes, como hemos mencionado, de todas las carreras en el período descrito. Dicha información fue proporcionada por el Departamento Escolar de la FCFM bajo la autorización con fines académicos por las autoridades competentes. Además, se empleó la base de datos del Departamento de Asesorías, que incluye registros de las asesorías realizadas desde el semestre enero – junio de 2018 hasta agosto – diciembre de 2022, con el visto bueno de las autoridades facultadas.

Los Kardex se recibieron en formato texto y se convirtieron en documentos de *Excel*, utilizando *scripts* en *Python*. En la Tabla 3 se muestran los nombres de las columnas de las bases de datos proporcionadas.

Tabla 3. Columnas de las bases de datos originales. Fuente: elaboración propia.

Base de datos Kardex	**Base de datos Asesorías**
Matrícula	Nombre
Nombre	Sexo
Carrera	Matrícula

Modelo educativo	Carrera
Código de la materia	Fecha
Materia	Hora inicio
Frecuencia por materia	Hora término
Créditos por materia	Unidad de aprendizaje
Oportunidades presentadas	Solucionó la duda
Fecha de registro de la calificación	Tema
Calificaciones	Profesor/dependencia
	Nombre completo del asesor

A partir de allí, los datos se reorganizaron hasta consolidar 7 bases de datos para los análisis posteriores, como se describe seguidamente:

- Kardex de estudiantes de todas las carreras hasta el primer semestre en el Modelo Educativo 420.
- Kardex por carrera en el Modelo Educativo 420 (6 bases de datos).

6.4 Preprocesamiento de las Bases Datos

Después de haber convertido todos los documentos de formato "$.txt$" a formato "$.xlsx$", se realizó una limpieza de estos. Durante este proceso, se eliminaron los espacios innecesarios de los valores en formato texto, se restauraron los caracteres especiales que se vieron afectados durante la conversión, se eliminaron la información de matrícula y nombre de los estudiantes, y se generaron identificadores aleatorios, entre otros ajustes.

Para la ejecución de los dos análisis objeto de estudio, se realizó un análisis descriptivo donde se observó el comportamiento de la deserción estudiantil para el Modelo Educativo 420; rutina que permitió identificar los porcientos de abandono por carreras para decidir en cuáles bases de datos enfocarnos, evitando así realizar un estudio exhaustivo. Para este procesamiento nos apoyamos en *Microsoft Excel* 365, usando tablas dinámicas sobre la base de datos que contiene estudiantes de todas las carreras para el Modelo Educativo 420.

Se realizó una nube de palabras con la columna “UA” (Unidad de Aprendizaje) de la base de datos del Departamento de Asesorías, con el fin de observar el predominio de materias en las que los estudiantes solicitaron apoyo y ello reforzó las observaciones que recibimos de los Coordinadores Académicos.

La Tabla 4 muestra las columnas de la base de datos del Departamento de Asesorías, que representan variables empleadas en los análisis, así como su tipo y su descripción.

Tabla 4. Base de datos del Departamento de Asesorías. Fuente: elaboración propia.

Variable	Tipo	Descripción
ID	Numérica	ID para cada estudiante
CARRERA	Texto	Etiquetas LA, LCC, LF, LM, LMAD y LSTI
FECHA	Fecha	Fecha de asistencia a recibir apoyo
HI	Hora	Hora de inicio registrada por el estudiante
HF	Hora	Hora en que finalizó la asesoría registrada
HT	Hora	Tiempo transcurrido de inicio a fin del servicio
MINUTOS	Tipo entero	Tiempo en minutos
CALIDAD	Char	Indicador de la calidad de la asesoría brindada
UA	Texto	Unidad de aprendizaje
TEMA	Texto	Descripción del tema objeto de la asesoría

La Tabla 5 muestra las variables comunes de las bases de datos empleadas en los análisis, así como su tipo y su descripción. El total de características fue diferente de una base de datos a otra debido al escenario de análisis y al período analizado dentro del escenario. Por ejemplo, al realizar un análisis en el escenario I para LMAD, donde se contemplaron las materias del primer semestre, tuvimos 21 características con 20 de ellas como predictoras. Sin embargo, en el mismo escenario y sobre la misma carrera, realizamos el análisis contemplando materias hasta segundo semestre, y en ese caso se obtuvieron 32 predictores.

Tabla 5. Bases de datos generadas para realizar los análisis. Fuente: elaboración propia.

Variable	Tipo	Descripción
ID	Numérica	ID para cada estudiante.
Carrera	Texto	Etiquetas LA, LCC, LF, LM, LMAD y LSTI que identifican las carreras de la FCFM. Aplicando "*OneHotEncoder*", se crearon variables *dummy*.
Calificaciones por materia	Numérica	Cada materia fue representada por una variable que se etiquetó con el prefijo "Cal" unido a la clave asociada según el modelo educativo. *Ejemplo*: Álgebra en el Modelo de Educativo 420, tiene el código "A01", por tanto, la variable referida a las calificaciones de los estudiantes en esa materia es "CalA01". Estas variables se convirtieron a numéricas, ubicando el valor 0 para las etiquetas "CU", "SD", "NC" y "NP" que aparecían registradas.
Número de oportunidades por materia	Numérica	No aparece explícitamente en los Kardex. Toma el valor entero de la oportunidad en la cual el estudiante acreditó la materia. Se etiquetó con el prefijo "TO". Análogo al ejemplo previo.
Min	Fecha	Primer registro de una calificación en el Kardex del estudiante, en una materia determinada.
Max	Fecha	Último registro de una calificación en el Kardex, en una materia determinada. Su valor puede coincidir con el registrado en "Min", cuando el estudiante acredita sus materias en primera oportunidad.
FTIME	Numérica	Fracción de tiempo en años, puede ser 0.
RPM	Numérica	Promedio de calificaciones de matemáticas.
IG	Numérica	Promedio de las materias cursadas.
IOM	Numérica	Razón oportunidades de matemáticas.
AVANCE	Numérica	Razón de las oportunidades mínimas entre las del Kardex.
NP	Numérica	Total, de "NP" reflejados en el Kardex de una materia.
NC	Numérica	Total, de "NC" reflejados en el Kardex de una materia.
SD	Numérica	Total, de "SD" reflejados en el Kardex de una materia.
ET	Numérica	Suma de las variables artificiales "NP", "NC" y "SD".
TO	Numérica	Suma total de oportunidades en una materia.
ASA	Numérica	Asistencia al Departamento de Asesorías de la FCFM.
TPA	Numérica	Tiempo en el Departamento de Asesorías de la FCFM.
AD	Binaria	Variable respuesta: "0" indica que el estudiante se mantiene en el programa y "1" indica que el estudiante desertó o abandonó.

Como resultado de las transformaciones realizadas, la información se estructuró en los dos escenarios planteados; tomando como base las calificaciones de los Kardex de estudiantes:

- para X semestres cursados, $X = 1{,}2$;
- hasta terminar el semestre Y, $Y = 1{,}2$.

Se seleccionó un subconjunto del total de bases de datos posibles a procesar por los algoritmos, tomando como criterio los resultados del análisis descriptivo y la experiencia que nos transmitieron los Coordinadores Académicos.

6.5 Métodos de Análisis

Para la ejecución de los dos análisis, se utilizaron los siguientes algoritmos para realizar la clasificación: *Logistic Regression* (LR)*, K – Nearest Neighbors* (KNN)*, Random Forest* (RF)*, Gradient Boosting* (GB) y *Support Vector Machine* (SVM); además se aplicó un *Ensemble* (ENS) con 4 de estos algoritmos individuales: LR, RF, GB y KNN (los de probabilidad).

6.5.1 Validación de Resultados

Se realizó la partición del conjunto de datos en tres partes: una para entrenamiento, una para validación y otra para prueba. Primeramente, se separó en 80% de entrenamiento y 20% para prueba, y luego el 25% del 80% que teníamos de entrenamiento, fue separado para validación.

6.5.2 Ajustes

Se ajustaron los parámetros de cada algoritmo utilizando las clases *"GridSearchCV"* (búsqueda en rejilla o cuadrícula), "*RandomizedSearchCV*" (explora un subconjunto aleatorio) y "*BayesSearchCV*" (utiliza optimización bayesiana). Apoyados en la documentación de *Python* de cada uno de los clasificadores, se aplicaron penalizaciones y/o regularizaciones a sus atributos (balanceo de clases, profundidad del árbol, número de hojas, *solver*, entre otros), hasta que se encontró el mejor estimador para una métrica (*accuracy*,f_1 o f_1*-macro*). Este proceso fue repetitivo y se ejecutó utilizando validación cruzada sobre los datos de entrenamiento, ajustando los algoritmos.

6.5.3 Comparaciones de los Algoritmos

Para cada algoritmo se obtuvo la matriz de confusión y el reporte de clasificación correspondiente. Ello permitió comparar las matrices de confusión de los diferentes algoritmos para ver cuál tiene el mejor rendimiento en términos de precisión, exhaustividad, exactitud y otras métricas de evaluación.

Para los algoritmos probabilísticos, graficamos sus respectivas curvas ROC y AUC, calculando el valor del área bajo la curva. Estas curvas nos dieron una idea del rendimiento de los algoritmos, al tiempo que nos indicaron la existencia o no de sobreajuste en ellos, desde el punto de vista gráfico. Para detectar el sobreajuste, fue importante comparar la curva ROC de cada modelo en los datos de entrenamiento respecto a la curva en los datos de prueba. Cuando la curva ROC en los datos de entrenamiento fue muy diferente a la curva ROC en los datos de prueba, esto indicó el posible sobreajuste del modelo y que no generalizaría bien ante nuevos datos, por lo que nos regresamos al paso previo: ajustar nuevamente los hiperparámetros.

Por otro lado, se calcularon las curvas ROC de varios modelos simultáneamente, en los datos de prueba y en el subconjunto de validación usando *K-Fold Cross Validation,* lo que permitió observar el comportamiento de las curvas por pliegues, constituyendo otro modo de visualizar el desempeño de los algoritmos ante datos no vistos. Se realizó una prueba MANOVA para comparar los algoritmos y para ello se utilizaron los valores de la media y la desviación estándar de cuatro métricas: *precision, recall, f1-score* y *accuracy;* calculados y almacenados durante la validación cruzada.

6.5.4 Interpretación de la Influencia de los Factores

Concluidos los ajustes de los modelos y la interpretación de sus rendimientos respectivos, se procedió a interpretar la influencia de los factores en la deserción estudiantil. Siempre que fue posible se aplicaron análisis de SHAP (*Shapley Additive exPlanations*) y análisis de PDP (*Parcial Dependence*

Plot). Para el análisis de SHAP se generó un ordenamiento calculando el rango de los valores en puntos "*shap"* para cada factor. Además, se generaron ordenamientos de influencia de los factores usando la importancia de características basada en la ganancia de información (ICBGI) y la importancia de características basada en la permutación (ICBP); asimismo se aplicó esta última técnica para cada algoritmo dentro del *Ensemble*.

Seguidamente, se tomaron los ordenamientos de cada técnica y se aplicó un algoritmo de no dominancia (implementado en R) a cada subgrupo, que devolvió los factores no dominados en cada técnica. Esta información se consolidó y se realizó una suma ponderada que nos permitió decidir cuáles fueron los factores más influyentes en que los estudiantes deserten.

Todas estas técnicas contribuyeron a interpretar los resultados de los algoritmos y sirvieron de apoyo para identificar causas del abandono estudiantil a partir de los factores académicos, siguiendo las estrategias planteadas.

6.6 Entorno de Producción

Una vez concluido todo el procesamiento y confirmada la efectividad de los algoritmos, se exportaron los dos modelos de mejor rendimiento a documentos ($".pkl"$) con la ayuda de la biblioteca "*pickle*" de *Python*, los cuáles contienen por separado el modelo correspondiente al algoritmo de ML elegido. Con estos documentos se realizaron pruebas con datos generados aleatoriamente, de modo que se observó la aplicación de nuestro trabajo al identificar estudiantes en riesgo, en un entorno de producción, y el posible impacto en la mitigación de la deserción estudiantil una vez que las autoridades de la FCFM puedan aplicarlo y proponer estrategias afines.

Las herramientas y funciones que se utilizaron para todo el procesamiento de la información se van a puntualizar en la discusión de los resultados.

CAPÍTULO 7
RESULTADOS Y CONCLUSIONES

En este capítulo se presentan los resultados y conclusiones obtenidos en un caso de estudio sobre deserción estudiantil, el cual fue abordado mediante el análisis de datos y la evaluación de los algoritmos de clasificación, con la finalidad de alcanzar los objetivos de este trabajo. Comienza el capítulo presentando los resultados del análisis descriptivo de los datos, así como los resultados de la evaluación de los cinco algoritmos y un *Ensemble* con cuatro de ellos. Seguidamente se muestran los resultados obtenidos de la comparación entre los algoritmos que se realizó a través de un análisis MANOVA partiendo de datos obtenidos con validación cruzada *K-Fold.* A pesar de que no se encontró un claro ganador entre los algoritmos evaluados, se discutirán las fortalezas y debilidades de cada uno en función de su rendimiento en las diferentes métricas utilizadas. Se muestra el análisis de eficiencia de Pareto que nos permitió identificar los factores académicos más influyentes en la deserción estudiantil. Finalmente, se describe el proceso de exportación de los dos modelos con mejor desempeño, los cuales representan una herramienta valiosa para abordar el problema de la deserción estudiantil y mejorar la calidad de la educación.

7.1 Presentación de los Resultados

El caso de estudio del cual presentamos resultados fue aplicado a la base de datos de todos los estudiantes del Modelo Educativo 420. Para ello se capturó la información de sus dos primeras oportunidades en el primer semestre de su carrera. Se presentan tablas donde resumimos los mejores

resultados de los algoritmos una vez fueron aplicados a las bases de datos por carrera.

7.1.1 Análisis Descriptivo

La Tabla 6 presenta el número total de estudiantes del Modelo Educativo 420 que matricularon y cursaron el primer semestre desde enero – junio de 2015 hasta agosto – diciembre de 2022, que ascendió a 8,336. De estos estudiantes, el 26.52% abandonó su carrera. Al analizar el comportamiento de la deserción por carreras, se observó que LCC y LMAD tuvieron los mayores porcentajes de abandono, con un 35.20% y 35.39%, respectivamente. Por otro lado, LA mostró el menor porcentaje de estudiantes que abandonaron, con un 12.59%.

Tabla 6. Comportamiento de la deserción por carrera. Fuente: elaboración propia.

Carrera	Estudiantes	% Desertores
LA	2169	12.59
LCC	1520	35.20
LF	730	17.40
LM	321	22.43
LMAD	2840	35.39
LSTI	756	26.32
FCFM	8336	26.52

En la Figura 7 se presentan las materias que más se solicitaron en el Departamento de Asesorías. La nube de palabras se generó a partir de la columna que contiene las unidades de aprendizaje en la respectiva base de datos. Al analizarla, se observó que destacaron las solicitudes de materias relacionadas con las matemáticas del primer semestre, como: Cálculo Diferencial, Geometría Analítica y Álgebra. Además, se encontraron materias de semestres superiores, tales como: Probabilidad Básica, Probabilidad Avanzada, Ecuaciones Diferenciales, entre otras. Estos hallazgos fueron consistentes con las descripciones proporcionadas por los Coordinadores Académicos.

Figura 7. Materias más solicitadas en el Departamento de Asesorías de la FCFM. Fuente: elaboración propia.

Los resultados del análisis descriptivo nos llevaron a centrarnos en los factores relacionados con las materias de matemáticas. Para ello, se incorporaron nuevas medidas (extracción de características), como promedios en dichas materias y razones en relación con la cantidad de oportunidades que los estudiantes presentaron en ellas, lo que aumentó el número de variables predictoras en nuestros análisis. Este enfoque se alineó con la noción de fusionar características existentes con el propósito de generar nuevas que podrían resultar provechosas; similar a la estrategia llevada a cabo por (Kemper et al., 2020), quienes se enfocaron en extraer rasgos que, según sus investigaciones, lograron disminuir notablemente la complejidad de los modelos sin comprometer significativamente su capacidad predictiva.

Dado que el 73.48% de los estudiantes permaneció en el programa de estudio, nos dimos cuenta de que los algoritmos podrían recibir clases desequilibradas. Por lo tanto, en nuestra búsqueda de los mejores hiperparámetros, consideramos atributos y técnicas que nos permitieran equilibrar las clases una vez leída la documentación de los algoritmos. Se activó el atributo correspondiente al balanceo en la búsqueda de hiperparámetros, lo que provoca la asignación de pesos equivalentes al recíproco del total de elementos de las clases.

7.1.2 Preparación de los Datos

En relación con el cumplimiento de los supuestos para los conjuntos de datos, la calidad de los datos, su relevancia y la necesidad de evitar discrepancias en los datos, así como otros conceptos previamente explicados que contribuyen a una mejor interpretación de los algoritmos aplicados sobre ellos, se consideró lo siguiente:

- Para las características seleccionadas que correspondieron a las calificaciones registradas en el Kardex de los estudiantes, se verificó la inexistencia de valores negativos o superiores a 100.
- Se reemplazaron las etiquetas (SD, NC y NP) en las columnas por el valor 0 y se crearon nuevas variables que representan el total por etiqueta presente en cada observación.
- Se verificó que la cantidad de oportunidades por materias en las bases de datos originales no fuera mayor que 6.
- Se generaron nuevas medidas al combinar características existentes, lo que resultó en un aumento en el número de variables predictoras (RPM, IG, IOM y AVANCE).
- Se incorporaron nuevas características basadas en datos provenientes del Departamento de Asesorías (ASA y TPA).
- El número de casos para cada una de las 7 bases de datos estructuradas se muestra en la columna “Estudiantes” de la Tabla 6. Siendo la base de datos correspondiente a LM, la que menor número de observaciones registró.
- La mayoría de los conjuntos de datos estuvieron completos, sin valores faltantes. En las ocasiones en que hubo valores faltantes (escenario I, contemplando 2 semestres), se aplicaron técnicas de imputación de datos, tomando la mediana de la variable predictora para completar calificaciones de materias por cursar. Esto sucedió cuando un estudiante completó los créditos cursando materias de dos semestres.

- Los datos fueron escalados utilizando técnicas de normalización o estandarización. En el caso de datos categóricos (Carrera), se les aplicó la técnica *OneHotEncoder.*
- La independencia entre variables no fue necesaria para la aplicación de los algoritmos. Por tanto, no se realizaron pruebas estadísticas al respecto.
- La independencia entre las observaciones se asumió debido a que las calificaciones de cada estudiante son independientes y es muy improbable que dos estudiantes tengan los mismos registros en su Kardex.
- Para garantizar que los conjuntos de datos de validación y prueba fueran representativos, se asignó un identificador único aleatorio a cada observación, evitando duplicaciones.

7.1.3 Caso de Estudio

El total de observaciones de la base de datos del caso de estudio fue de 8336, correspondiendo a todos los estudiantes de FCFM para el Modelo de Educativo 420. La Tabla 7 muestra ese total, y el resultado de la separación de los datos en entrenamiento, prueba y validación.

Tabla 7. Separación de los datos para el caso de estudio. Fuente: elaboración propia.

Total	Entrenamiento	Prueba	Validación
8336	5001	1668	1667

Como hemos mencionado, se aplicaron cinco algoritmos por separado a los datos: LR, RF, GB, SVM y KNN. Además, creamos un *Ensemble* que incorporó los algoritmos de probabilidad. Para cada uno de los algoritmos individuales, se realizaron ajustes de parámetros seleccionando los mejores hiperparámetros. El estudio incluye este proceso para GB. En el ajuste de los parámetros se utilizaron las clases *"GridSearchCV"* y *"RandomizedSearchCV"* de la biblioteca *scikit-learn*, además de la clase *"BayesSearchCV"* de la

biblioteca *skopt (scikit-optimaze);* ambas blibliotecas de *Python*. Los algoritmos se entrenaron utilizando un conjunto de 5001 observaciones.

7.1.3.1 Métricas de Evaluación

La Tabla 8 presenta los resultados numéricos de las matrices de confusión que se obtuvieron una vez aplicados los seis algoritmos mencionados. Muestra cómo cada algoritmo ha asignado las predicciones en términos de verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos. La evaluación detallada proporcionada por estas matrices permitió una comprensión más profunda de la eficacia de cada algoritmo en el contexto específico de la tarea de clasificación binaria.

Tabla 8. Resultados de las matrices de confusión por algoritmo. Fuente: elaboración propia.

		0	**1**	**Support**
MC-420-OHE		1220	448	1668
	TP	**FP**	**FN**	**TN**
LR	1049	171	83	365
RF	1082	137	97	351
GB	1139	81	135	313
SVM	1058	162	91	357
KNN	1123	97	141	307
ENS	1121	99	112	336

Las clases fueron representadas como sigue:

- “0”: estudiantes que no desertaron.
- “1”: estudiantes que desertaron.

Las matrices de confusión se calcularon utilizando el conjunto de prueba compuesto por 1668 observaciones, una vez que los algoritmos fueron ajustados y entrenados. Se observa que los datos estuvieron desequilibrados, ya que el 73.14% de los datos de prueba correspondían a la clase “0” y sólo el 26.86% pertenecían a la clase “1”.

En la Tabla 9 se muestran los resultados del reporte de clasificación para cada algoritmo. Los mejores valores alcanzados en cada métrica se resaltan en color verde, mientras que los peores valores se resaltan en color rojo.

Tabla 9. Reportes de clasificación por algoritmo. Fuente: elaboración propia.

RC-420-OHE	*P0*	*R0*	*f1-0*	*P1*	*R1*	*f1-1*	*Accuracy*
LR	**0.9267**	**0.8598**	**0.8920**	**0.6810**	**0.8147**	0.7419	**0.8477**
RF	0.9178	0.8877	0.9025	0.7193	0.7835	0.7500	0.8597
GB	0.8940	**0.9336**	0.9134	**0.7944**	0.6987	0.7435	0.8705
SVM	0.9208	0.8672	0.8932	0.6879	0.7969	0.7384	0.8483
KNN	**0.8884**	0.9205	0.9042	0.7599	**0.6853**	**0.7206**	0.8573
ENS	0.9092	0.9188	**0.9140**	0.7724	0.7500	**0.7610**	**0.8735**

Los resultados allí mostrados, no proporcionaron conclusiones rigurosas sobre qué algoritmos obtienen los mejores resultados en nuestro estudio. Sin embargo, nos permiten realizar una selección basada en el criterio de la mayoría simple para mostrar resultados de uno que, bajo este criterio, tuvo buen desempeño. Por ejemplo, el algoritmo GB mostró ser el mejor en dos de las métricas y no fue el peor en ninguna, mientras que el algoritmo ESN resultó ser el mejor en tres de las métricas y tampoco fue el peor en ninguna (resultado esperado según la literatura). Más adelante mostraremos el resultado de la comparación de todos los algoritmos utilizando un MANOVA.

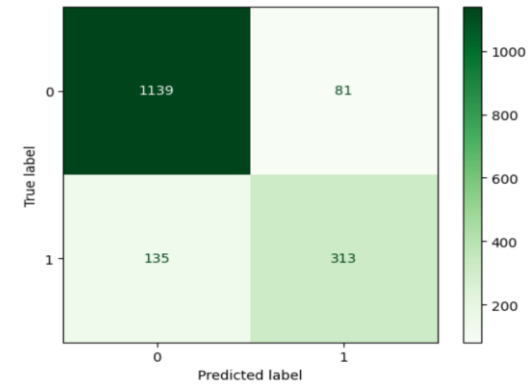
Presentamos, a continuación, los resultados que se obtuvieron para GB. En la Figura 8 se incluye la matriz de confusión, el reporte de clasificación, las curvas ROC tanto para los datos de entrenamiento como para los datos de prueba (incluyen el valor numérico del área bajo estas curvas), y la curva AUC correspondiente.

El reporte de clasificación proporciona una visión detallada del rendimiento del algoritmo aplicado en un problema de clasificación. A continuación, analizaremos cada métrica y su significado para evaluar el rendimiento del algoritmo:

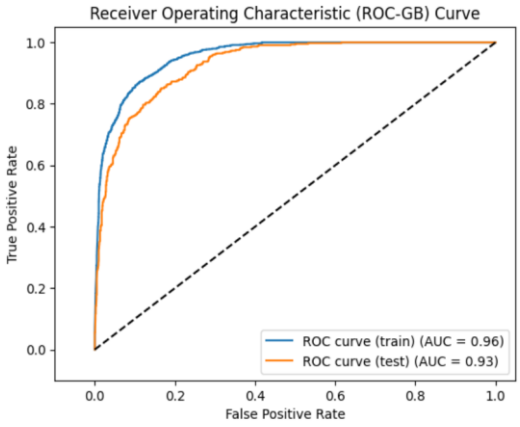
- *Recall*: para la clase ‘0’, el *recall* fue del 93%, lo que indicó que el 93% de las instancias de clase ‘0’ fueron identificadas correctamente. Sin

embargo, para la clase ‘1’, fue del 70%, significando que sólo el 70% de las instancias de clase ‘1’ fueron identificadas correctamente. El algoritmo tuvo un mejor desempeño en términos de *recall* para la clase ‘0’ que para la clase ‘1’.

- *F1-score*: para la clase ‘0’, fue del 91%, lo que indicó un buen equilibrio entre *precision* y *recall*. Sin embargo, para la clase ‘1’, el *f1-score* fue del 74%, lo que sugirió que había margen de mejora en el equilibrio entre *precision* y *recall* para esta clase.
- *Support*: El soporte representó el número de instancias en cada clase, con 1220 instancias de la clase ‘0’ y 448 instancias de la clase ‘1’; lo que mostró el desequilibrio entre las clases.



	precision	recall	f1-score	support
0	0.89	0.93	0.91	1220
1	0.79	0.70	0.74	448
accuracy			0.87	1668
macro avg	0.84	0.82	0.83	1668
weighted avg	0.87	0.87	0.87	1668



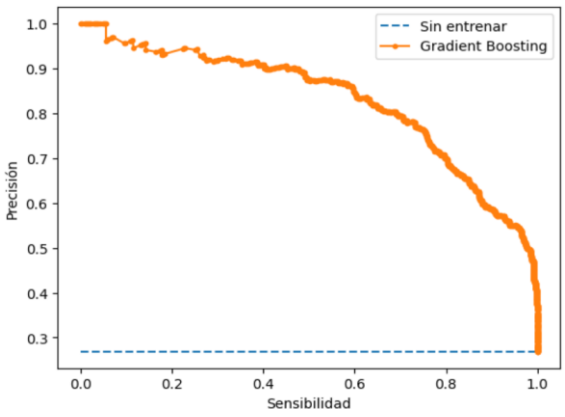


Figura 8. Resultados de *Gradient Boosting* para el caso de estudio. Fuente: elaboración propia.

- *Accuracy*: la exactitud general fue del 87%, lo que indicó que el 87% de las instancias fueron clasificadas correctamente. Aunque la exactitud es

un indicador importante, es necesario tener en cuenta que puede ser engañosa si las clases están desequilibradas en términos de tamaño, como ocurrió en este caso.

- Promedio macro (*Macro Avg*): El promedio macro calcula las métricas promediadas sin considerar el desequilibrio entre las clases. El promedio macro de *precision*, *recall* y *f1-score* fue de 0.84, 0.82 y 0.83 respectivamente. Estos valores indicaron el rendimiento promedio del modelo sin tener en cuenta el tamaño de las clases y se consideró bueno.
- Promedio ponderado (*Weighted Avg*): El promedio ponderado también calcula las métricas promediadas, pero tiene en cuenta el desequilibrio de las clases al ponderarlas según su soporte (cantidad de instancias). El promedio ponderado de *precision*, *recall* y *f1-score* fue de 0.87 en los tres casos. Estos valores indicaron un buen rendimiento promedio del modelo, teniendo en cuenta el desequilibrio de las clases.

A raíz de lo expuesto, el modelo mostró un buen rendimiento general con una precisión razonablemente alta y un *f1-score* equilibrado para la clase ‘0’. Sin embargo, no hubo valores similares en el *recall* y el *f1-score* para la clase ‘1’, lo que sugirió que el modelo tenía dificultades para detectar correctamente las instancias de esa clase. En términos de nuestro objetivo de identificar a los estudiantes desertores (clase ‘1’), nos animamos a realizar ajustes y a aplicar otras técnicas para mejorar las métricas.

Asimismo, obtuvimos otras gráficas y tablas que nos permitieron interpretar los factores que influyen en la deserción de los estudiantes. A continuación, presentaremos algunas de ellas junto con nuestras interpretaciones específicas, elaborando conclusiones parciales basadas en el caso de estudio. Siguiendo los análisis sugeridos por Bruce et al. (2020), hemos generado la curva de elevación correspondiente a cada algoritmo. Esta gráfica proporciona información relevante sobre el rendimiento del modelo de clasificación. En la Figura 9 se muestra la curva de elevación correspondiente al algoritmo GB.

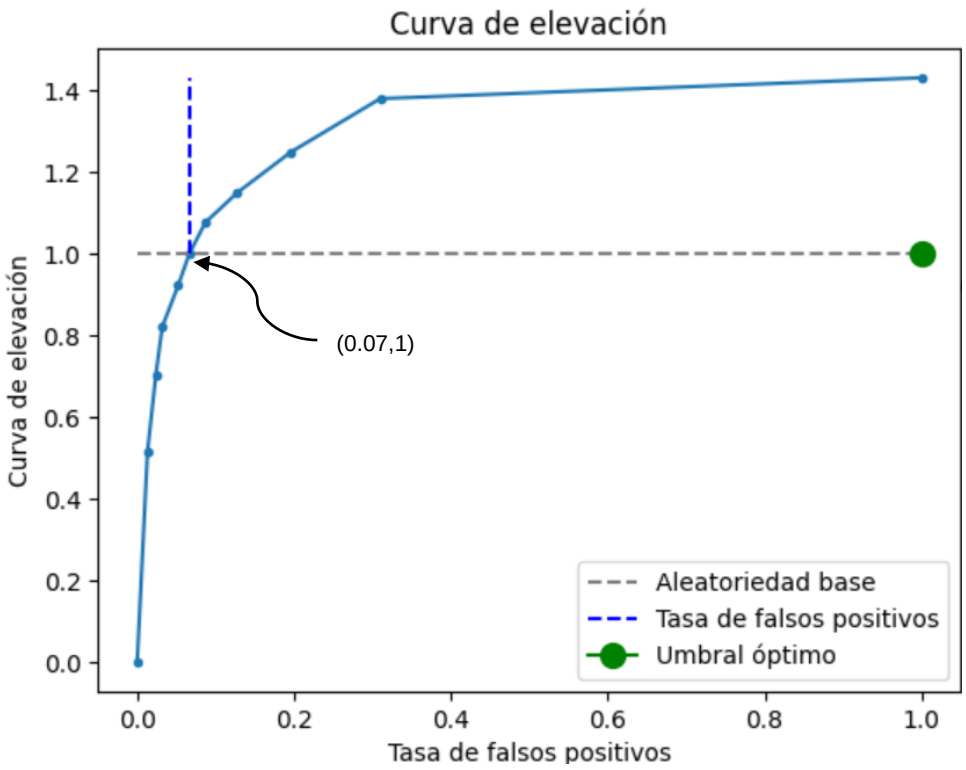


Figura 9. Curva de elevación de *Gradient Boosting* para el caso de estudio. Fuente: elaboración propia.

Los valores en el eje vertical representan la tasa de verdaderos positivos (TPR) o la sensibilidad del modelo. Muestran la proporción de instancias positivas que el modelo ha identificado correctamente en relación con el total de instancias positivas en el conjunto de prueba. A medida que la tasa de verdaderos positivos aumenta, el modelo demuestra una mayor capacidad para identificar de manera correcta a los estudiantes que abandonan.

Los valores en el eje horizontal representan la tasa de falsos positivos (FPR). Indican la proporción de instancias negativas que el modelo ha clasificado incorrectamente como desertores en relación con el total de no desertores en el conjunto de prueba. A medida que la tasa de falsos positivos disminuye, el modelo comete menos errores al clasificar incorrectamente los no desertores como desertores.

La curva de elevación mostró cómo el modelo se desempeñó en comparación con la clasificación aleatoria. A medida que la curva se alejó de la línea de aleatoriedad base, indicó un beneficio adicional en la identificación de desertores. Cuanto más alto esté el punto en la curva, mayor será el beneficio

adicional proporcionado por el modelo en términos de capturar correctamente los “1”. El punto de la curva de elevación que se considera óptimo es aquel que maximiza el beneficio adicional en la identificación de la clase “1”. En nuestro caso, el punto $(0.07, 1.0)$ se identificó como el umbral óptimo. Esto significó que, al establecer un umbral de clasificación en 0.07, el modelo logra capturar correctamente el 100% de los casos positivos en el conjunto de prueba, con una tasa de falsos positivos del 7%. Al ajustar el umbral de clasificación a valores más altos, el modelo logra clasificar correctamente una mayor proporción de TPR sin clasificar erróneamente una gran cantidad de FPR. Esto se considera deseable, ya que indica que el modelo tuvo un buen rendimiento en la identificación de los desertores mientras mantiene un bajo nivel de errores de clasificación.

En resumen, la gráfica de la curva de elevación mostró el desempeño del modelo de clasificación en términos de su capacidad para identificar correctamente los casos positivos y su tasa de falsos positivos. Además, proporcionó información sobre el beneficio adicional obtenido en comparación con la clasificación aleatoria y destacó el umbral óptimo que maximiza dicho beneficio. Con ello concluimos con algunas métricas y técnicas que permitieron evaluar el rendimiento de este modelo. En el siguiente epígrafe abordamos la Kappa de Cohen para todos los algoritmos y profundizamos en la comparación de estos.

7.1.3.2 Comparación de los Algoritmos

A raíz de lo mostrado en la Tabla 9 sospechamos que no hubo diferencias significativas entre los algoritmos, porque los valores de las métricas que identificamos fueron bastante similares. En los estudios realizados por (Solis et al., 2018), (Kemper et al., 2020) y (Vivek Raj, 2020) se realizó el cálculo de la Kappa de Cohen como métrica de evaluación del rendimiento de los modelos, que implica evaluar el acuerdo entre las predicciones generadas por un algoritmo y las clasificaciones de referencia. Los valores de Kappa

cercanos a 1, sugieren que el algoritmo ha sido capaz de clasificar correctamente los elementos en estudio con consistencia. La Tabla 10 muestra los resultados de esta métrica para cada algoritmo.

Tabla 10. Kappa de Cohen para cada algoritmo. Fuente: elaboración propia.

Algoritmos	**LR**	**RF**	**GB**	**SVM**	**KNN**	**ENS**
Kappa de Cohen	0.6351	0.6562	0.6325	0.6341	0.6066	0.6590

Los valores obtenidos para cada algoritmo son bastante similares, siendo el menor de ellos el que correspondió al KNN. Apoyados en la escala presentada en la Tabla 1 (Marco Teórico), los valores obtenidos indican una fuerza de concordancia “Buena”, exceptuando al KNN que se encuentra en el límite entre “Moderada” y “Buena””. Asimismo, estos valores no permitieron definir con claridad cuál algoritmo resultó de mejor desempeño, pero los valores obtenidos para esta métrica resultaron alentadores. Para el cálculo de este índice se utilizó la función *“cohen_kappa_score”* desde el módulo *“sklearn.metrics”*, en *Python*.

La Figura 10 muestra simultáneamente las curvas ROC de todos los algoritmos, con el valor del área bajo cada curva. Todas las curvas fueron similares excepto la de KNN (con menor área bajo la curva). Este procesamiento indicó similitud de desempeño gráficamente. Un análisis análogo fue realizado por (Kemper et al., 2020), comparando los desempeños de los modelos mediante las curvas ROC de los modelos correspondientes al primer semestre, en su estudio, y concluyeron que a simple vista resultó complicado identificar discrepancias significativas en el rendimiento entre ellos.

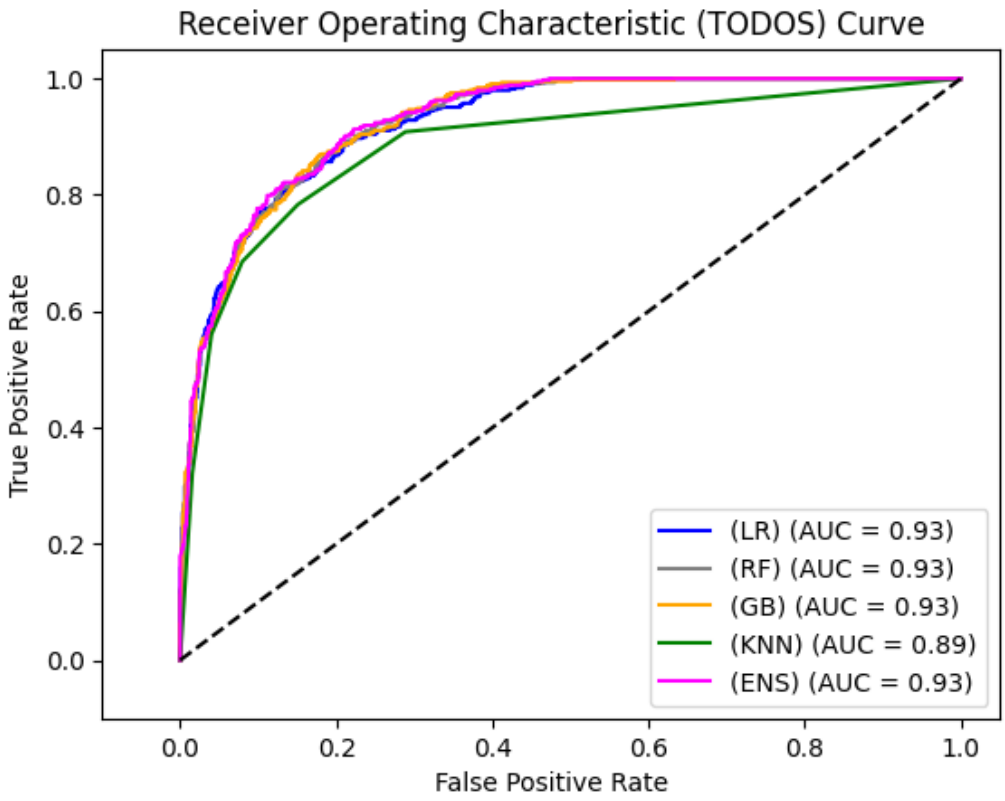


Figura 10. Curvas ROC de los algoritmos en los datos de prueba. Fuente: elaboración propia.

Las conjeturas que pudimos hacernos no superan la metodología rigurosa que nos brindan las pruebas estadísticas para estos fines, por lo que procedimos a apoyarnos en la estadística para corroborar similitudes y/o diferencias entre los algoritmos.

Para el análisis comparativo entre los algoritmos, se realizó un *K-Fold Cross Validation* sobre los datos de validación (usamos "*StratifiedKFold*" con siete pliegues), los cuales no habían sido vistos por los algoritmos. Al término de la ejecución de este código, se obtuvieron las probabilidades de predicción de cada pliegue para la precisión, la exhaustividad, la sensibilidad y la exactitud. La Tabla 11 muestra los valores promedio de la media y la desviación estándar de cada algoritmo, partiendo de resultados de la validación cruzada. Enfocándonos en la media, los mejores valores que se alcanzaron en cada métrica se resaltan en color verde, mientras que los peores valores se resaltan en color rojo.

Tabla 11. Valores promedio de μ y σ^2 para los pliegues. Fuente: elaboración propia.

	precision		recall		f1-score		accuracy	
KF-420-OHE	mean	std	mean	std	mean	std	mean	std
LR	**0.6717**	0.0533	**0.8306**	0.0511	0.7412	0.0407	**0.8494**	0.0276
RF	0.7455	0.0455	0.7656	0.0353	0.7538	0.0216	0.8704	0.0153
GB	**0.7926**	0.0525	**0.7076**	0.0592	0.7449	0.0356	**0.8752**	0.0161
SVM	0.6932	0.0552	0.7866	0.0472	**0.7353**	0.0397	0.8530	0.0263
KNN	0.7660	0.0572	0.7378	0.0519	0.7502	0.0433	0.8728	0.0238
ENS	0.7554	0.0539	0.7749	0.0477	**0.7629**	0.0319	**0.8752**	0.0202

Al realizar una inspección visual, se observó que el *Ensemble* se mantuvo coherente con el razonamiento expuesto en la Tabla 9. Este algoritmo obtuvo los mejores promedios en dos de las cuatro métricas evaluadas y no fue el peor en las dos restantes. Sin embargo, es importante destacar que esta observación no es suficiente para afirmar la existencia de semejanzas y/o diferencias significativas entre los algoritmos.

Para realizar una comparación más rigurosa entre los modelos, se llevó a cabo una prueba MANOVA utilizando los datos almacenados como resultado de la validación cruzada. A continuación, describiremos en detalle los resultados de esta prueba.

7.1.3.2.1 Prueba MANOVA

Se realizó un *ranking* de las métricas utilizando la función *"JERARQUIA.MEDIA"* de *Microsoft Excel* 365; posteriormente se aplicó el análisis a ambos conjuntos: ranqueados y sin ranqueo. Los resultados que se exponen de la prueba correspondieron a los datos ranqueados, para los que se verificaron los supuestos de la prueba, es decir:

- Prueba de normalidad multivariada.
- Prueba de igualdad de matrices de covarianza.

Se usó el paquete R para ambas pruebas. La prueba de Mardia es una prueba estadística de normalidad multivariada que se utiliza para evaluar si una muestra multivariada proviene de una distribución normal multivariada. Las hipótesis para la prueba de Mardia se plantearon de la siguiente manera:

H_0: La muestra proviene de una distribución normal multivariada.

H_1: La muestra no proviene de una distribución normal multivariada.

Dicha prueba considera la asimetría y la curtosis multivariada como medidas de no normalidad. Por lo tanto, el objetivo de la prueba fue determinar si la muestra presenta asimetría y curtosis significativamente diferentes de los valores esperados en una distribución normal multivariada. La Figura 11 muestra los resultados que se obtuvieron, que indicaron que no hubo evidencia significativa para rechazar H_0 (hubo normalidad).

	Prueba	Medida	Estadístico	valor_p	Conclusión
1	Prueba de Asimetría	0.8973618	9.611375	0.9746839	Normalidad
2	Prueba de Curtosis	20.7588147	3.282901	0.0700051	Normalidad

Figura 11. Prueba de Mardia. Fuente: elaboración propia.

También se realizó una prueba de Henze-Zirkler para verificar normalidad multivariada, que es una prueba no paramétrica cuyo estadístico de prueba se basa en la distancia de Henze-Zirkler y se distribuye asintóticamente según una distribución Chi-cuadrada. Se calculó un valor p que indicó la probabilidad de obtener un valor del estadístico de prueba igual o más extremo que el observado, bajo la suposición de que la hipótesis nula es verdadera. Las hipótesis son análogas a las planteadas anteriormente y la Figura 12 muestra los resultados de la prueba, que indicaron que hubo evidencia significativa para rechazar H_0 (no hubo normalidad), pues el valor p obtenido fue menor que 0.05.

	Test	HZ	p value	MVN
1	Henze-Zirkler	1.107308	0.003401618	NO

Figura 12. *Test* de Henze-Zirkler. Fuente: elaboración propia.

Por otro lado, para realizar la prueba de igualdad de matrices de covarianza se aplicó la prueba de Box's M, que evalúa si las matrices de covarianza de los grupos o condiciones son iguales, lo que implica que las variables dependientes tienen la misma estructura de covarianza en cada grupo o condición. La prueba se basa en el determinante de la matriz de covarianza

combinada y utiliza la distribución de Chi-cuadrada para obtener un estadístico de prueba. Las hipótesis para la prueba de Box's M fueron las siguientes:

H_0: Las matrices de covarianza en cada grupo o condición son iguales.

H_1: Las matrices de covarianza en al menos un grupo o condición son diferentes.

En la Figura 13 mostramos los resultados de prueba, que no proporcionó evidencia para rechazar la hipótesis nula y se concluyó que las matrices de covarianza fueron iguales.

```
        Box's M-test for Homogeneity of Covariance Matrices

data:  datos[, 2:5]
Chi-Sq (approx.) = 53.715, df = 50, p-value = 0.334
```

Figura 13. Prueba de Box's M. Fuente: elaboración propia.

La Figura 14 muestra diagramas de cajas simultáneos de los algoritmos respecto a las métricas (utilizando los resultados de la media en la validación cruzada): gráficamente no se distingue un claro ganador. Siguiendo nuestro análisis, procedimos a realizar la prueba MANOVA, usando la traza de Pillai apoyados del paquete R.

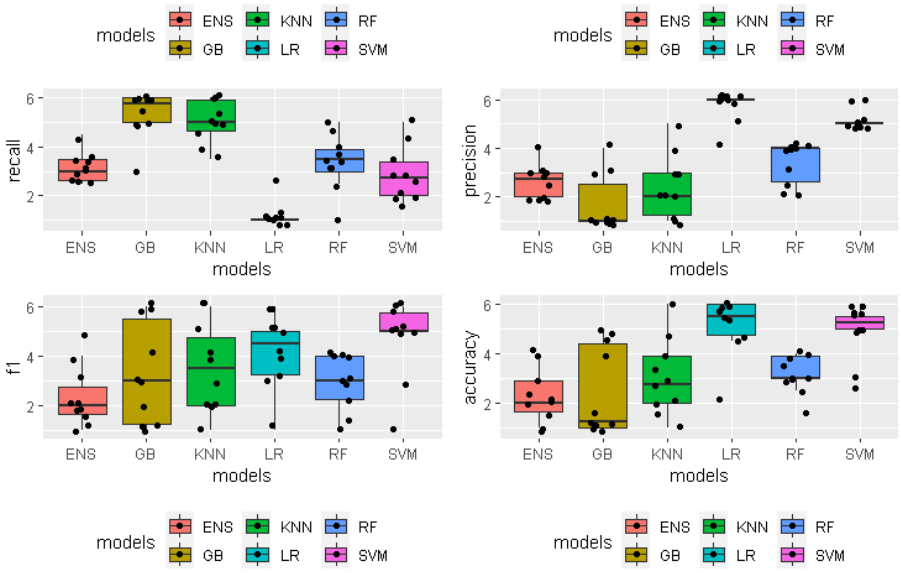


Figura 14. *Boxplots* de algoritmos por métricas. Fuente: elaboración propia.

La prueba calcula un estadístico de prueba llamado "valor de Pillai" o "valor de traza de Pillai", que se basa en el valor de la traza de la matriz de producto cruzado de los residuos. La prueba utiliza una distribución F para obtener un valor p asociado. El p obtenido a partir del valor de Pillai indica la probabilidad de obtener un valor del estadístico igual o más extremo que el observado, bajo la hipótesis nula. Las hipótesis para la prueba fueron las siguientes:

H_0: No hay diferencias estadísticamente significativas en la combinación lineal de las variables dependientes entre los grupos o condiciones.

H_1: Hay diferencias estadísticamente significativas en la combinación lineal de las variables dependientes entre al menos dos grupos o condiciones.

La Figura 15 muestra el resultado que obtuvimos al aplicar la prueba a los datos ranqueados. El valor p asociado con la prueba fue "$1.423e-08$", lo que indica que es extremadamente bajo. Esto sugirió que había evidencia estadísticamente significativa para rechazar la hipótesis nula de que no hubo diferencias significativas en la combinación lineal de las variables dependientes entre los grupos o condiciones. En resumen, indicó la existencia de diferencias en el desempeño de los algoritmos.

```
          Df Pillai approx F num Df den Df    Pr(>F)
models     5 1.1567   4.3937     20    216 1.423e-08 ***
Residuals 54
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figura 15. Prueba MANOVA. Fuente: elaboración propia.

Luego de este resultado, se aplicó análisis discriminante lineal (LDA por sus siglas en inglés) para identificar cuáles fueron distintos. LDA es una técnica estadística utilizada para clasificar observaciones en grupos o categorías predefinidas. El objetivo del LDA es encontrar una combinación lineal de variables predictoras que maximice la separación entre los grupos o categorías conocidas, asume que los grupos o categorías están definidos a priori y que las variables predictoras siguen una distribución normal multivariada en cada

grupo, al tiempo que utiliza la información de las medias y las matrices de covarianza de los grupos para encontrar la mejor combinación lineal.

Resultado de la aplicación de LDA, la Figura 16 muestra puntos representativos de cada algoritmo. Se observa en ella a GB y KNN a la izquierda, el ENS aislado arriba y al centro, prácticamente separado de todos; tanto SVM como LR tuvieron una estructura definida, en la que no se entrelazaron.

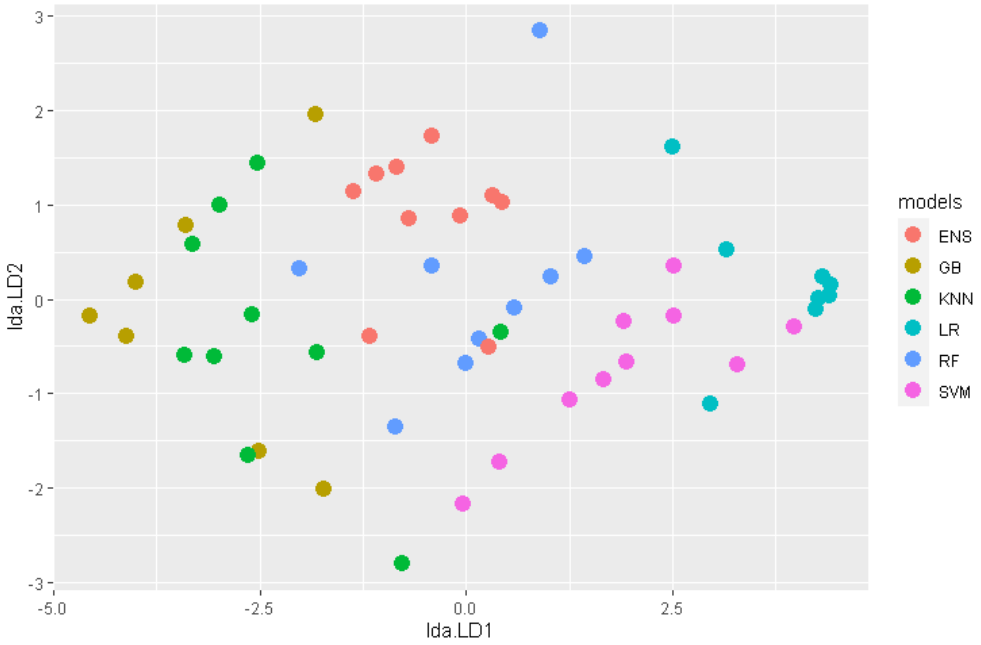


Figura 16. Gráfico de dispersión para identificar diferencias entre los modelos. Fuente: elaboración propia.

Las conclusiones a las que arribamos fueron subjetivas, pero este trabajo nos propició una ayuda para entender mejor la naturaleza de las diferencias entre los algoritmos visualmente. A raíz de esta comparación en las distintas métricas, concluimos que no hubo un claro ganador, aunque sí obtuvimos diferencias entre los modelos.

Por otro lado, la Tabla 12 muestra el tiempo de ejecución que tardaron los algoritmos. La similitud en los tiempos de ejecución de RF y KNN, nos indicó que ambos son igualmente eficientes para resolver el problema de clasificación en cuestión. Asimismo, las diferencias significativas en los tiempos de ejecución de LR y GB, se debió a la complejidad de este último respecto a su par, pues requirió más recursos computacionales para ejecutarse, siendo un resultado

esperado porque GB es un algoritmo tipo *ensemble*. La comparación de los resultados de la precisión y otras métricas entre ellos, expuestas en este capítulo, justifica el tiempo adicional de ejecución de GB.

Tabla 12. Tiempo de ejecución de los algoritmos. Fuente: elaboración propia.

Algoritmos	**Tiempo de ejecución**
Logistic Regression	1 minuto y 59.8 segundos
Random Forest	4 minutos y 33 segundos
Gradient Boosting	10 minutos y 30 segundos
Support Vector Machine	3 minutos y 12 segundos
K-Nearest Neighbors	4 minutos y 2 segundos
Ensemble	6 minutos y 12 segundos

.

7.1.3.3 Interpretación de Resultados

A continuación, analizamos los resultados obtenidos mediante GB utilizando las técnicas mencionadas en la metodología. En la Figura 17 se presenta el primer resultado al aplicar el análisis de SHAP a GB. Partiendo de la gráfica, los factores que más influyeron en la variable a predecir fueron: el Índice General, Metodología de la Programación, Cálculo Diferencial, la Razón Promedio en Matemáticas y la pertenencia a LA, que ocuparon los primeros cinco lugares en términos de puntuación SHAP y su impacto en la salida del modelo. La nube de puntos para cada factor difiere en acumulación y en cercanía a la línea principal. De modo que la técnica sugirió el ordenamiento mencionado (más adelante expondremos el análisis que se realizó basado en esta técnica y el rango de valores en puntos *“shap”*). La librería *"shap"* de *Python* proporciona varias funciones que facilitan la visualización del impacto de las características en la salida de los modelos.

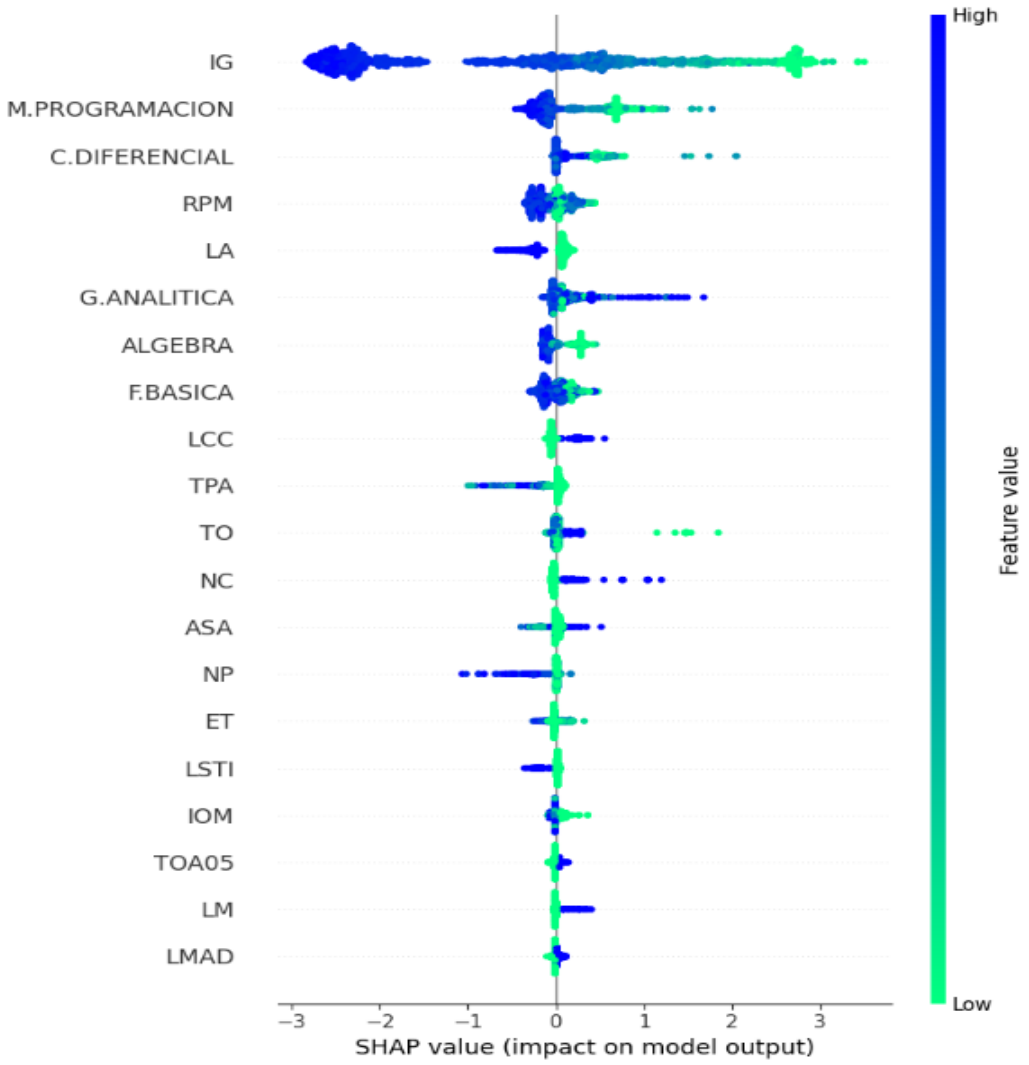


Figura 17. Análisis de SHAP *Gradient Boosting* (impacto en la salida del modelo). Fuente: elaboración propia.

El árbol de decisión (DT) mostrado en la Figura 18 resultó de aplicar el modelo ajustado a los datos de entrenamiento con una profundidad de tres niveles. Ello indicó que en el ajuste de hiperparámetros, antes de entrenar el modelo, se estableció que el atributo *'max_depth'* (profundidad máxima) tuviera un valor de 3. En otras ejecuciones a este atributo se le indicó un valor mínimo de 5. Cada nodo en el árbol representa una condición que se evalúa, y las ramas salientes representan las posibles respuestas a esa condición. El total mínimo de hojas también se configuró en el ajuste de hiperparámetros.

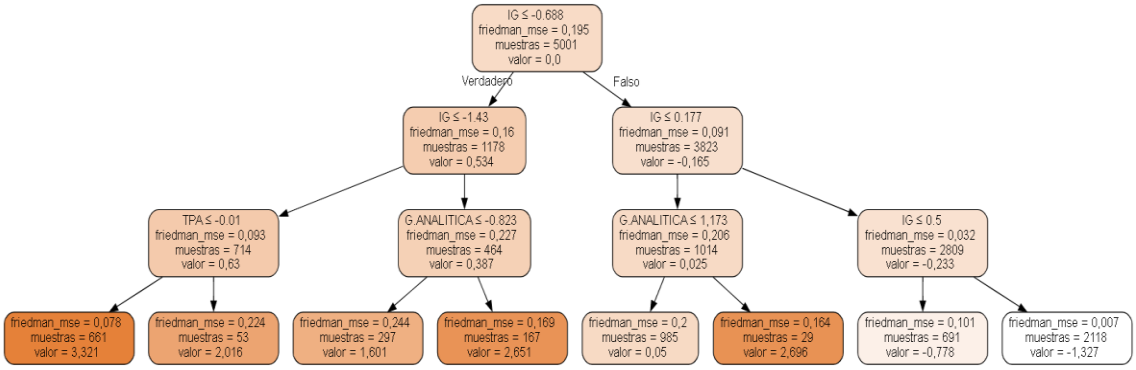


Figura 18. Visualización del árbol de decisión para *Gradient Boosting*. Fuente: elaboración propia.

El árbol comienza con un nodo raíz, identificado como "0". La condición que se evalúa en este nodo es "$IG \leq -0.688$", donde "IG" se refiere al índice general de los estudiantes. Si esta condición es verdadera, se sigue la rama etiquetada como "Verdadero", y si es falsa, se sigue la rama etiquetada como "Falso". El nodo "0" tiene una impureza de Friedman MSE (*mean squared error*) de 0.195 y contiene 5001 observaciones del entrenamiento, todas con un valor de 0.0. Si la condición del nodo "0" es verdadera, se llega al nodo "1". La condición en el nodo "1" es "$IG \leq -1.43$", y su impureza de Friedman MSE es 0.16. Este nodo tiene 1178 muestras, con un valor promedio de 0.534. Análogamente se fue interpretando cada nodo hasta llegar a las hojas. Valores positivos en una hoja indican la probabilidad estimada de pertenecer a la clase positiva, mientras que valores negativos indican la probabilidad estimada de pertenecer a la clase negativa. Estos valores suelen interpretarse como probabilidades, pero también pueden ser transformados en etiquetas binarias utilizando un umbral de decisión. Por ejemplo, si el umbral es 0.5, cualquier valor por encima de ese umbral se puede clasificar como positivo y cualquier valor por debajo se puede clasificar como negativo. Sin embargo, el umbral puede ajustarse según las necesidades del problema y el equilibrio entre los falsos positivos y los falsos negativos.

Los árboles de decisión son modelos que resultan intuitivos y fáciles de interpretar. Siguiendo el flujo de decisiones en el árbol, es posible comprender cómo se llega a una predicción, lo que los convierte en herramientas

especialmente útiles para extraer conocimiento y comprender las relaciones entre las características de los datos. Sin embargo, en nuestro trabajo debido a la profundidad seleccionada por el buscador de hiperparámetros, no se pudo visualizar un top 5 de los factores más influyentes utilizando esta técnica. No obstante, se pudo observar la influencia del Índice General y de Geometría Analítica.

Continuando con nuestro análisis de interpretación de GB, en la Tabla 13 se muestra la importancia de las características basada en la ganancia de información (ICBGI). Los resultados obtenidos fueron consistentes con las demás técnicas de interpretación que hemos aplicado, porque el Índice General destacó como un factor influyente en la clasificación del modelo. Completando este top cinco de factores influyentes, se encontraron Geometría Analítica, Metodología de la Programación, la Razón Promedio en Matemáticas y Álgebra.

Tabla 13. ICBGI para *Gradient Boosting*. Fuente: elaboración propia.

	CARACTERÍSTICA	IMPORTANCIA
12	IG	0.7218
5	G. ANALITICA	0.0305
7	M.PROGRAMACION	0.0222
10	RPM	0.0176
1	ALGEBRA	0.0164
20	LA	0.0119
17	TO	0.0118
19	TPA	0.0105
3	C.DIFERENCIAL	0.0083
9	F. BASICA	0.0082
16	ET	0.0066

En la Tabla 14, se presenta el top cinco para las tres técnicas de interpretación que utilizamos con el algoritmo GB. La columna "INFLUYENTES" se completó de manera ordenada mediante un enfoque de mayoría simple, teniendo en cuenta la cantidad de apariciones y el ranking en dichas apariciones. Más adelante, se detalla el análisis realizado basado en los ordenamientos obtenidos mediante todas las técnicas que aplicamos, y que resultaron válidas para identificar los factores más influyentes hasta el primer

semestre (incluyendo todas las observaciones del Modelo Educativo 420 debido a la existencia del tronco común, más los estudios individuales para las carreras: LA, LCC, LMAD y LSTI); también hasta el segundo semestre donde se consideró sólo para las carreras antes mencionadas.

Tabla 14. Top 5 de factores más influyentes para *Gradient Boosting*.
Fuente: elaboración propia.

SHAP	DT	ICBGI	INFLUYENTES
IG	IG	IG	IG
M. PROGRAMACIÓN	G. ANALÍTICA	G. ANALÍTICA	G. ANALÍTICA
C. DIFERENCIAL	-	M. PROGRAMACIÓN	M. PROGRAMACIÓN
RPM	-	RPM	RPM
LA	-	ÁLGEBRA	C. DIFERENCIAL

Siguiendo la lógica que se aplicó para GB, se realizó el estudio de rendimiento e interpretación de los resultados de: LR, RF, SVM, KNN y ESN (Anexo 2).

Para llegar a conclusiones sobre los factores más influyentes, se tomó la decisión de no utilizar el criterio de mayoría simple, el cual, aunque es una forma común de identificar factores influyentes, carece del rigor propio de la Ciencia de Datos. En su lugar, se llevó a cabo un análisis basado en la no dominancia de Pareto, que será explicado seguidamente.

Se seleccionaron técnicas de ordenamiento de factores, que fueron comunes a los algoritmos: análisis de SHAP, ICBGI, ICBP e importancia de característica basada en la permutación para cada algoritmo dentro del *Ensemble* (ICBP-ENS). La Tabla 15 muestra los valores numéricos que se obtuvieron para ICBGI para cuatro algoritmos de clasificación aplicados al caso de estudio. Los factores fueron ordenados alfabéticamente y cada columna constituye un vector.

Tabla 15. Resultados de ICBGI para el caso de estudio. Fuente: elaboración propia.

FACTORES	LR	RF	GB	SVM
ALGEBRA	0.1340	0.0674	0.0258	1.2436
ASA	0.0001	0.0098	0.0160	0.0135
C.DIFERENCIAL	0.1367	0.0589	0.0267	1.3461
ET	0.0002	0.1088	0.0086	0.0381
F.BASICA	0.0549	0.0418	0.0368	0.9440
G.ANALITICA	0.1408	0.0375	0.0261	1.3858
IG	0.2472	0.2164	0.7218	4.4264
IOM	0.0066	0.0430	0.0030	0.4224
LA	0.0007	0.0085	0.0101	0.0741
LCC	0.0004	0.0045	0.0025	0.0489
LF	0.0001	0.0017	0.0013	0.0072
LM	0.0002	0.0015	0.0007	0.0154
LMAD	0.0005	0.0078	0.0072	0.0468
LSTI	0.0003	0.0031	0.0043	0.0333
M.PROGRAMACION	0.0570	0.0690	0.0340	0.8511
NC	0.0003	0.0041	0.0060	0.0112
NP	0.0003	0.0404	0.0055	0.0479
RPM	0.2118	0.1417	0.0294	1.1578
SD	0.0000	0.0000	0.0000	0.0030
TO	0.0010	0.0641	0.0098	0.0488
TOA01	0.0018	0.0220	0.0016	0.1120
TOA02	0.0016	0.0083	0.0031	0.0952
TOA03	0.0018	0.0066	0.0002	0.1291
TOA04	0.0005	0.0082	0.0017	0.0464
TOA05	0.0011	0.0128	0.0047	0.1246
TPA	0.0002	0.0119	0.0129	0.0033

Estos valores se incorporaron como entrada al análisis de no dominancia de Pareto, que se procesó usando R, con los pasos siguientes:

- Se creó una matriz de dominancia.
- Se asignó dominancia a la comparación de los factores.
- Se calculó la fuerza de la dominancia de cada factor y se almacenó en un vector.
- Se generó la matriz de *ranking* de alternativas.

La Tabla 16 muestra la matriz de *ranking* de alternativas que se obtuvo para ICBGI, manteniendo el orden alfabético de los factores. Este subproceso

constituyó uno de los cuatro resultados que nos permitieron arribar a conclusiones.

Tabla 16. *Ranking* de alternativas para ICBGI en el caso de estudio.
Fuente: elaboración propia.

FACTORES	FORTALEZA	DEBILIDAD
ALGEBRA	18	1
ASA	2	7
C.DIFERENCIAL	17	1
ET	2	2
F.BASICA	16	1
G.ANALITICA	15	1
IG	25	0
IOM	7	5
LA	8	7
LCC	3	11
LF	1	21
LM	1	18
LMAD	5	9
LSTI	3	11
M.PROGRAMACION	18	1
NC	2	10
NP	3	7
RPM	19	1
SD	0	25
TO	8	4
TOA01	3	8
TOA02	5	7
TOA03	1	8
TOA04	3	12
TOA05	6	7
TPA	1	7

Se obtuvo el ranking de alternativas para ICBP-ENS, para análisis de SHAP y para ICBP. La Tabla 17 muestra la sección superior de la tabla donde se consolidaron cada uno de los *rankings* de alternativas, y una columna “INFLUYENTES” que permitió ordenar los factores utilizando una suma ponderada, que penaliza las debilidades cuando su valor numérico es mayor o igual a 2. En el Anexo 3 se muestra la expresión de la suma ponderada que

utilizamos para el cálculo de los valores en dicha columna. Mientras menor es el valor obtenido en sus celdas, mayor influencia tiene el factor en la variable a predecir: son los factores más influyentes para identificar los estudiantes en riesgo.

Tabla 17. Consolidación del *Ranking* de alternativas para el caso de estudio. Fuente: elaboración propia.

FACTORES	FOR-ICBGI	DEB-ICBGI	FOR-ICBP_ENS	DEB-ICBP_ENS	FOR_SHAP_R	DEB_SHAP_R	FORT_ICBP	DEB_ICBP	INFLUYENTES
IG	**25**	**0**	**15**	**0**	**25**	**0**	**15**	**0**	**0**
M.PROGRAMACION	**18**	**1**	**14**	**0**	**17**	**1**	**22**	**0**	**0.5**
G.ANALITICA	**15**	**1**	**14**	**0**	**16**	**1**	**19**	**0**	**0.5**
ALGEBRA	**18**	**1**	**14**	**0**	**16**	**1**	**15**	**1**	**0.75**
C.DIFERENCIAL	**17**	**1**	**1**	**0**	**16**	**1**	**6**	**1**	**0.75**
IOM	7	5	5	1	7	1	13	2	13.5
RPM	19	1	0	1	12	1	0	14	18.25
TO	8	4	0	1	8	1	0	16	22.5
TOA05	6	7	0	2	9	5	11	2	25.25
F.BASICA	16	1	0	2	17	1	0	20	26.5

Como resultado de nuestro análisis del caso de estudio, se concluyó que el Índice General, así como las materias: Metodología de la Programación, Geometría Analítica, Álgebra y Cálculo Diferencial, por ese orden, fueron los factores más influyentes en el abandono estudiantil. De modo independiente se realizó este procesamiento para las carreras: LA, LCC, LMAD y LSTI; utilizando la estrategia I, e información correspondiente al primer y segundo semestre (Anexo 3).

En la Tabla 18 se muestra una tabla de frecuencias que reflejó, en la columna “SUMA”, el total de aparición de los factores en los cinco estudios que se realizaron utilizando información de los Kardex hasta primer semestre. Es decir, se consideran los factores influyentes de la Tabla 17 y las tablas de la 24 a la 27 (Anexo 3).

Tabla 18. Frecuencia de los factores más influyentes en los casos de estudiados. Fuente: elaboración propia.

FACTORES	TODAS	LA	LCC	LMAD	LSTI	SUMA
G.ANALITICA	1	1	0	1	1	4
C.DIFERENCIAL	1	1	1	1	0	4
RPM	0	1	1	1	1	4
IG	1	0	1	1	0	3
M.PROGRAMACION	1	0	1	1	0	3
ET	0	1	0	1	1	3
NP	0	1	0	1	1	3
ALGEBRA	1	1	0	0	0	2
F.BASICA	0	1	0	0	0	1
NC	0	0	0	0	1	1

El enfoque utilizado para combinar los resultados obtenidos permitió que el análisis fuera sólido y, al mismo tiempo, proporcionó elementos para concluir que Geometría Analítica, Cálculo Diferencial y la Razón Promedio en Matemáticas son los factores más influyentes basándonos en las calificaciones del Kardex hasta el primer semestre. Además, la inclusión de Total de Etiquetas (ET), NP y NC en esta tabla enfatiza su relevancia estadística, lo que indica la necesidad de realizar un análisis exhaustivo del sistema de etiquetado, las normativas para asignar etiquetas y temas relacionados. Esta última conclusión estuvo respaldada por los resultados que obtuvimos en los análisis hasta el segundo semestre, los cuales se presentan en el Anexo 3.

Finalmente, se optó por seleccionar dos algoritmos para el entorno de prueba: LR (Regresión Logística) entre los individuales, y GB (*Gradient Boosting*) entre los de tipo *ensemble*.

7.2 Conclusiones

- Los algoritmos de *Machine Learning* utilizados en el proyecto permitieron clasificar de manera efectiva, con niveles aceptables de precisión, a estudiantes en riesgo de abandono tomando sus calificaciones del primer semestre y del primer año.
- La predicción del abandono escolar es posible utilizando los Kárdex de los estudiantes. Ello concuerda con lo expresado por (Heublein, 2010) y

(Kemper et al., 2020), en el hecho de que el enfoque centrado en los registros académicos es adecuado para identificar posibles casos de estudiantes en riesgo de deserción.

- La “Razón Promedio en Matemáticas” (resultado de promediar las calificaciones de Geometría Analítica, Cálculo Diferencial y Álgebra) y el “Índice General” (promedio de todas las materias), así como la calificación de “Geometría Analítica” y la de “Cálculo Diferencial” fueron los factores influyentes en la clasificación y, por ende, los que inciden en la deserción estudiantil.
- También se identificaron como influyentes factores relacionados con el sistema de etiquetas: SD (sin derecho), NC (no cumplió), NP (no presentó) y ET (suma total de las tres anteriores); hechos reflejados en la medida en que los estudiantes avanzan en sus carreras. Este conocimiento podría ser utilizado por la institución para implementar estrategias y medidas preventivas que ayuden a retener a los estudiantes, mejorando sus desempeños académicos.
- Un bajo desempeño en la "Razón Promedio en Matemáticas" y el "Índice General", así como por reiteradas etiquetas de SD (sin derecho), NC (no cumplió), NP (no presentó) o una alta frecuencia de la etiqueta ET (suma total de las tres anteriores), son indicadores que revelan que un estudiante se encuentra en riesgo de deserción académica. Además, bajas calificaciones en "Geometría Analítica" y "Cálculo Diferencial" refuerzan aún más esta advertencia.

7.3 Trabajo Futuro

- Para potenciar y ampliar los resultados obtenidos en este proyecto, se recomienda realizar análisis adicionales incorporando nuevas variables explicativas combinadas bajo criterio de expertos y considerando un análisis más detallado por carrera, dentro de la Facultad de Ciencias

Físico Matemáticas. Esto podría facilitar una visión más completa y exacta de los patrones de deserción en diferentes contextos.

- Incluir variables sociodemográficas en el análisis, asumiendo la dificultad de recopilar información adicional, posiblemente relacionada con la privacidad, y siendo cuidadosos de no discriminar previamente entre estudiantes en función de estos datos no relacionados con sus estudios. Estos datos podrían ofrecer una comprensión más profunda de las circunstancias individuales de los estudiantes, lo que podría facilitar la personalización de las intervenciones para prevenir la deserción.
- Una vez identificados los estudiantes en riesgo de deserción, con la aplicación de nuestro enfoque, es fundamental implementar medidas correctivas de manera efectiva. Se recomienda que las autoridades académicas elaboren estrategias de retención orientadas a los estudiantes en riesgo. Esta colaboración podría proporcionar un apoyo integral a los estudiantes y abordar las causas subyacentes que contribuyen a la deserción.
- Una vez aplicadas las estrategias de retención, continuar con la aplicación de estos algoritmos para medir la eficacia de dichas acciones correctivas.

Finalizando nuestra exposición, este proyecto ha puesto de manifiesto el potencial de las herramientas de *Machine Learning* y la Ciencia de Datos para abordar el problema de la deserción estudiantil utilizando datos académicos. Las conclusiones extraídas y las futuras líneas de trabajo ofrecen una base sólida para seguir investigando estrategias efectivas que contribuyan a mejorar la retención y el éxito académico de los estudiantes en la Facultad de Ciencias Físico Matemáticas de la Universidad Autónoma de Nuevo León.

REFERENCIAS

Alvarado-Uribe, J., Mejía-Almada, P., Masetto Herrera, A. L., Molontay, R., Hilliger, I., Hegde, V., Montemayor Gallegos, J. E., Ramírez Díaz, R. A., & Ceballos, H. G. (2022). Student Dataset from Tecnológico de Monterrey in Mexico to Predict Dropout in Higher Education. Data, 7(9). https://doi.org/10.3390/data7090119

Betancourt, G. A. (2005). Las máquinas de soporte vectorial (SVMs). Scientia et technica, 1(27).

Bernal, D. M. M. (2013). La Deserción Escolar: Un problema de Carácter Social. Revista In Vestigium Ire., 6, 115–124.

Boden, M. A. (2017). Inteligencia artificial. Turner.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. Proceedings of the Fifth Annual Workshop on Computational Learning Theory, 144–152. https://doi.org/10.1145/130385.130401

Breiman, L. (2001). Method Random Forest.

Bruce, P., Bruce, A., Gedeck, P., & Safari, an O. M. Company. (2020). Practical Statistics for Data Scientists, 2nd Edition.

Carmona Suárez, E. J. (2016). Tutorial sobre Máquinas de Vectores Soporte (SVM).

Chitarroni, H. (2002). La regresión logística. http://www.salvador.edu.ar/csoc/idicso

de Ullibarri Galparsoro, L., & P. F. S. (1999). Medidas de concordancia: el índice de Kappa. Cad Aten Primaria, 6, 169-171.

Díaz Peralta, C. (2008). Modelo conceptual para la Deserción Estudiantil Universitaria Chilena. Estudios Pedagógicos (Valdivia), 34(2), 65–86. https://doi.org/10.4067/S0718-07052008000200004

Dicovskiy Riobóo, L. M., & Pedroza Pacheco, M. E. (2018). Predicción de deserción y éxito en estudiantes. Caso de estudio: ingeniería agroindustrial de la UNI Norte, Nicaragua, 2011-2015. Nexo Revista Científica, 31(01), 16–27. https://doi.org/10.5377/nexo.v31i01.6451

Dietterich, T. G. (2000). Ensemble Methods in Machine Learning (pp. 1–15). https://doi.org/10.1007/3-540-45014-9_1

Donoso, S., & Schiefelbein, E. (2007). Análisis de los modelos explicativos de retención de estudiantes en la universidad: una visión desde la desigualdad social. Estudios Pedagógicos (Valdivia), 33(1). https://doi.org/10.4067/S0718-07052007000100001

Dutt, A., Ismail, M. A., & Herawan, T. (2017). A Systematic Review on Educational Data Mining. IEEE Access, 5, 15991–16005. https://doi.org/10.1109/ACCESS.2017.2654247

EL, L. P. L. E. Y., D. M. S. D. P., & Ú. C. (2017). Instituto Nacional para la Evaluación de la Educación.

Fernández, O. (2014). La mediación escolar como alternativa en la prevención de la deserción escolar. http://eprints.uanl.mx/id/eprint/13710

Ferreyra, M. M., Avitabile, C., Botero Álvarez, J., Haimovich Paz, F., & Urzúa, S. (2017). At a Crossroads: Higher Education in Latin America and the Caribbean. World Bank, Washington, DC. https://doi.org/10.1596/978-1-4648-1014-5

Fleiss, J. L., Cohen, J., & Everitt, B. S. (1969). Large sample standard errors of kappa and weighted kappa. Psychological Bulletin, 72(5), 323–327. https://doi.org/10.1037/h0028106

Gadea Cavazos, E. A., Medina Villanueva, M., Duelos Martínez, J. E., & Ruiz Ponce, M. del C. (2011). Modelo de gestión académico deportivo motivos de deserción escolar en el club Deportivo Tigres sinergia deportiva S.A. de C.V. Revista Del Ciencias Del Ejercicio FOD, 7(7), 22–28.

García, A. J. C. (2020). Modelo para la predicción de la deserción de estudiantes de pregrado, basado en técnicas de minería de datos.

García, V., Sánchez, J. S., & Mollineda, R. A. (2012). On the effectiveness of preprocessing methods when dealing with different levels of class imbalance. Knowledge-Based Systems, 25(1), 13–21. https://doi.org/10.1016/j.knosys.2011.06.013

Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow Second Edition Concepts, Tools, and Techniques to Build Intelligent Systems. http://oreilly.com

Gómez Triana, F. J. (2013). El impacto del programa mexicano de becas PRONABES en el rendimiento académico de los alumnos de licenciatura de la UANL, generación 2007-2012. Universidad Autónoma de Nuevo León.

Guerra Turrubiates, J. I. (2020). Deserción escolar en pacientes adolescentes embarazadas del noreste de México. Universidad Autónoma de Nuevo León.

Hastie, T., Tibshirani, R., & Friedman, J. (2017). The Elements of Statistical Learning Data Mining, Inference, and Prediction (Second Edition). Springer.

Heublein, U., H. C., & S. J. (2010). Ursachen des Studienabbruchs in Bachelor- und in herkömmlichen Studiengängen. Forum Hochschule. Hannover: Deutsches Zentrum für Hochschulund Wissenschaftsforschung.

Hilliger, I., Ortiz-Rojas, M., Pesántez-Cabrera, P., Scheihing, E., Tsai, Y.-S., Muñoz-Merino, P. J., Broos, T., Whitelock-Wainwright, A., & Pérez-Sanagustín, M. (2020). Identifying needs for learning analytics adoption in Latin American universities: A mixed-methods approach. The Internet and Higher Education, 45, 100726. https://doi.org/10.1016/j.iheduc.2020.100726

Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied Logistic Regression. Wiley. https://doi.org/10.1002/9781118548387

Jia, J.-W., & Mareboyana, M. (2014). Predictive Models for Undergraduate Student Retention Using Machine Learning Algorithms. In Transactions on Engineering Technologies (pp. 315–329). Springer Netherlands. https://doi.org/10.1007/978-94-017-9115-1_24

Kemper, L., Vorhoff, G., & Wigger, B. U. (2020). Predicting student dropout: A machine learning approach. European Journal of Higher Education, 10(1), 28–47. https://doi.org/10.1080/21568235.2020.1718520

Kiss, B., Nagy, M., & Molontay, R. (2019). Predicting Dropout Using High School and First-semester Academic Achievement Measures. 2019 17th International Conference on Emerging ELearning Technologies and Applications (ICETA), 383–389. https://doi.org/10.1109/ICETA48886.2019.9040158.

Lundberg, S., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. http://arxiv.org/abs/1705.07874

Manrique, R., Nunes, B. P., Marino, O., Casanova, M. A., & Nurmikko-Fuller, T. (2019). An Analysis of Student Representation, Representative Features and Classification Algorithms to Predict Degree Dropout. Proceedings of the 9th International Conference on Learning Analytics & Knowledge, 401–410. https://doi.org/10.1145/3303772.3303800

Marins, M. A. (2016). Classificação de falhas em máquinas rotativas utilizando métodos de similaridade e random forest.

Marler, R. T., & Arora, J. S. (2004). Survey of multi-objective optimization methods for engineering. In Structural and Multidisciplinary Optimization (Vol. 26, Issue 6, pp. 369–395). https://doi.org/10.1007/s00158-003-0368-6

Matos, L. F. A. (2021). Diseño de una metodología multicriterio de apoyo a la decisión para la gestión de la permanencia estudiantil de educación superior (pp. 1–176).

Moreno Salinas, J. G. (2017). Científico de datos: codificando el valor oculto e intangible de los datos. Revista Digital Universitaria, 18(7). https://doi.org/10.22201/codeic.16076079e.2017.v18n7.a2

Moreno Torres, M., Ortiz, Y. O., & González, M. G. (2016). Capacitación de Docentes en Procesos Neurocognitivos. Revista puertorriqueña de Psicología, 27(2), 304–318.

Namoun, A., & Alshanqiti, A. (2020). Predicting Student Performance Using Data Mining and Learning Analytics Techniques: A Systematic Literature Review. Applied Sciences, 11(1), 237. https://doi.org/10.3390/app11010237

Palacios-Pacheco, X., Villegas-Ch, W., & Luján-Mora, S. (2019). Application of data mining for the detection of variables that cause university desertion. Communications in Computer and Information Science, 895, 510–520. https://doi.org/10.1007/978-3-030-05532-5_38

Peralta, C. D. (2008). Modelo conceptual para la Deserción Estudiantil Universitaria Chilena. Estudios Pedagógicos (Valdivia), 34(2), 65–86. https://doi.org/10.4067/S0718-07052008000200004

Rochin Berumen, F. L. (2021). Deserción escolar en la educación superior en México: revisión de literatura. RIDE Revista Iberoamericana Para La Investigación y El Desarrollo Educativo, 11(22). https://doi.org/10.23913/ride.v11i22.821

Rodríguez Pérez, V. M. (2013). Puntaje del concurso de ingreso al nivel medio superior como instrumento para prevenir la reprobación escolar en estudiantes de la preparatoria 5 de la UANL. Universidad Autónoma de Nuevo León.

Rowtho, V. (2017). Early Detection of At-Risk Undergraduate Students through Academic Performance Predictors. Higher Education Studies, 7(3), 42. https://doi.org/10.5539/hes.v7n3p42

Ruiz, Carolina. G., Muriel, D. Marcela. D., Gallego, Jorge. F., Vélez, Elkin. C., Gómez, Santiago. G., & Portilla, Karoll. G. (2009). Deserción estudiantil en la educación superior colombiana: metodología de seguimiento, diagnóstico y elementos para su prevención. Ministerio de Educación Nacional.

RUIZ, M. J. E. (2018). Abandono escolar en la educación media superior de México, políticas, actores y análisis de casos. Universidad de Guanajuato.

Santamaría, F. A., & Bustos, A. (2013). Permanence and Dropout Rates in Higher Education: A Research Experience Based on Young Students' Voices (Vol. 12, Issue 2, pp. 73–80).

Schaeffer, P. E. R. (2000). Análisis de las causas de deserción en los estudios de licenciatura en la Facultad de Ingeniería Mecánica y Eléctrica y estrategias para su abatimiento.

SEP. (2019). Lineamientos para la formulación de indicadores educativos septiembre 2019. https://www.planeacion.sep.gob.mx/indicadorespronosticos.aspx

Shi, L., Weng, M., Ma, X., & Xi, L. (2010). Rough Set Based Decision Tree Ensemble Algorithm for Text Classification. In Journal of Computational Information (Vol. 6). http://www.JofCI.org1553-9105/

Silva Fuente-Alba, C., & Molina Villagra, M. (2017). Likelihood ratio (razón de verosimilitud): definición y aplicación en Radiología. Revista Argentina de Radiología, 81(3), 204–208. https://doi.org/10.1016/j.rard.2016.11.002

Silva Laya, M. (2011). El primer año universitario: Un tramo crítico para el éxito académico. Perfiles educativos, 33(SPE), 102-114.

Solís, M., Moreira, T., González, R., Fernández, T., & Hernández, M. (2018). Perspectives to Predict Dropout in University Students with Machine Learning. 2018 IEEE International Work Conference on Bioinspired Intelligence (IWOBI), 1–6. https://doi.org/10.1109/IWOBI.2018.8464191

Spady, W. G. (1970). Dropouts from higher education: An interdisciplinary review and synthesis. Interchange, 1(1), 64–85. https://doi.org/10.1007/BF02214313

Steuer, R. E. (1989). Trends in Interactive Multiple Objective Programming (pp. 107–119). https://doi.org/10.1007/978-3-662-22160-0_15

Tinto, V. (1982). Limits of Theory and Practice in Student Attrition. The Journal of Higher Education, 53(6), 687. https://doi.org/10.2307/1981525

Torres Castillo, M. I. (2012). El trabajo colaborativo como estrategia de gestión académica en el fortalecimiento de la reorganización curricular por ciclos. https://hdl.handle.net/10901/10063

Tukey, J. W. (1962). The Future of Data Analysis. The Annals of Mathematical Statistics, 33(1), 1–67. https://doi.org/10.1214/aoms/1177704711

UANL. (2022). Plan de Desarrollo Institucional 2022-2030.

UNESCO. (2020). Towards-universal-access-to-higher-education-international-trends.

Vivek Raj, S. N., & M. S. K. (2020). Predicting student failure in university examination using machine learning algorithms. forest, 84(66.14), 0-24.

Wolpert, D. H. (1996). The lack of a priori distinctions between learning algorithms. Neural computation, 8(7), 1341-1390.

ANEXOS

ANEXO 1

Asociación Clave/Materia por carrera, hasta segundo semestre del Modelo Educativo 420.

CIENCIAS COMPUTACIONALES	
A01	Álgebra
A02	Cálculo Diferencial
A03	Geometría Analítica
A04	Metodología de la Programación
A05	Física Básica
A07	Cálculo Integral
801	Matemáticas Discretas
301	Programación Estructurada
302	Física
A06	Tópicos de Álgebra
303	Laboratorio de Programación Estructurada
304	Laboratorio de Física

MULTIMEDIA Y ANIMACIÓN DIGITAL	
A01	Álgebra
A02	Cálculo Diferencial
A03	Geometría Analítica
A04	Metodología de la Programación
A05	Física Básica
500	Programación Básica
501	Matemáticas para Videojuegos I
502	Tecnologías Multimedia
503	Dibujo de la Anatomía Humana
504	Expresiones Artísticas y Socioculturales
008	Responsabilidad Social y Desarrollo Sustentable

FÍSICA	
A01	Álgebra
A02	Cálculo Diferencial
A03	Geometría Analítica
A04	Metodología de la Programación
A05	Física Básica
A06	Tópicos de Álgebra
A07	Cálculo Integral
400	Mecánica Traslacional y Rotacional
401	Lenguajes de Programación
010	Cultura de Paz

SEGURIDAD EN TECNOLOGÍAS DE INFORMACIÓN	
A01	Álgebra
A02	Cálculo Diferencial
A03	Geometría Analítica
A04	Metodología de la Programación
A05	Física Básica
A07	Cálculo Integral
200	Fundamentos de la Seguridad Informática
201	Fundamentos de Sistemas Operativos
202	Introducción a la Programación
801	Matemáticas Discretas
A06	Tópicos de Álgebra

MATEMÁTICAS	
A01	Álgebra
A02	Cálculo Diferencial
A03	Geometría Analítica
A04	Metodología de la Programación
A05	Física Básica
A06	Tópicos de Álgebra
A07	Cálculo Integral
301	Programación Estructurada
801	Matemáticas Discretas
400	Mecánica Traslacional y Rotacional
303	Laboratorio de Programación Estructurada

ACTUARÍA	
A01	Álgebra
A02	Cálculo Diferencial
A03	Geometría Analítica
A04	Metodología de la Programación
A05	Física Básica
A06	Tópicos de Álgebra
A07	Cálculo Integral
A08	Programación I
A09	Matemáticas Financieras
A10	Seguro de Vida
A11	Análisis de Datos

ANEXO 2

Características más influyentes bajo el criterio de mayoría simple.

COEFICIENTES	SHAP	INFLUYENTES
IG	IG	IG
RPM	RPM	RPM
G. ANALÍTICA	G. ANALÍTICA	G. ANALÍTICA
ALGEBRA	ALGEBRA	ALGEBRA
M. PROGRAMACIÓN	M. PROGRAMACIÓN	M. PROGRAMACIÓN

Tabla 19. Top 5 de características más influyentes para *Logistic Regression*: Índice General, Razón Promedio en Matemáticas, Geometría Analítica, Álgebra y Metodología de la Programación, en orden descendente. Fuente: elaboración propia.

DT	ICBGI	ICBP	INFLUYENTES
IG	IG	IG	IG
M. PROGRAMACIÓN	RPM	ET	ET
IOM	ET	NC	M. PROGRAMACIÓN
G. ANALÍTICA	TO	ASA	ÁLGEBRA
ÁLGEBRA	ÁLGEBRA	M. PROGRAMACIÓN	RPM

Tabla 20. Top 5 de características más influyentes para *Random Forest*: Índice General, Total de Etiquetas, Metodología de la Programación, Álgebra y Razón Promedio en Matemáticas, en orden descendente. Fuente: elaboración propia.

SHAP	ICBGI	INFLUYENTES
IG	IG	IG
G. ANALÍTICA	G. ANALÍTICA	G. ANALÍTICA
C.DIFERENCIAL	C.DIFERENCIAL	C.DIFERENCIAL
RPM	RPM	RPM
ALGEBRA	ALGEBRA	ALGEBRA

Tabla 21. Top 5 de características más influyentes para *Support Vector Machine*: Índice General, Geometría Analítica, Cálculo Diferencial, Razón Promedio en Matemáticas y Álgebra, en orden descendente. Fuente: elaboración propia.

ICBP-LR	ICBP-RF	ICBP-GB	ICBP-KNN	ICBP-ENS	INFLUYENTES
IG	IG	IG	MP	IG	GA
RPM	NC	GA	GA	GA	IG
IOM	MP	CD	NC	RPM	MP
GA	ET	MP	FB	CD	RPM
CD	GA	RPM	A	A	CD

Tabla 22. Top 5 de características más influyentes para el *Ensemble*: Geometría Analítica, Índice General, Metodología de la Programación, Razón Promedio en Matemáticas y Cálculo Diferencial, en orden descendente. Fuente: elaboración propia.

ICBP	INFLUYENTES
M.PROGRAMACIÓN	M.PROGRAMACIÓN
G.ANALÍTICA	G.ANALÍTICA
NC	NC
F.BÁSICA	F.BÁSICA
TOA01	TOA01

Tabla 23. Top 5 de características más influyentes para *K-Nearest Neighbors*: Metodología de la Programación, Geometría Analítica, No Cumplió, Física Básica y Total de Oportunidades en Álgebra, en orden descendente. Fuente: elaboración propia.

ANEXO 3

Consolidación del *Ranking* de alternativas hasta los primeros dos semestres.

$$INFLUYENTES = (SI([@[DEB-ICBGI]] < 2{,}1, SI([@[DEB-ICBGI]] < 5{,}2{,}10)) * [@[DEB-ICBGI]] + SI\left([@[DEB-ICBP_{ENS}]] < 2{,}1, SI\left([@[DEB-ICBP_{ENS}]] < 5{,}2{,}10\right)\right) * [@[DEB-ICBP_{ENS}]] + SI\left([@[DEB_{SHAP_R}]] < 3{,}1{,}5\right) * [@[DEB_{SHAP_R}]] + SI([@[DEB\_ICBP]] < 3{,}1{,}5) * [@[DEB\_ICBP]])/4$$

Suma ponderada utilizada para el cálculo de valores de la columna INFLUYENTES, una vez aplicada la no dominancia de Pareto y obtenidos los *Ranking*. En ella se penalizan las debilidades y se eligen los factores cuyos resultados en dicha columna están más cercanos a "0", en orden ascendente.

FACTORES	FOR-ICBGI	DEB-ICBGI	FOR-ICBP_ENS	DEB-ICBP_ENS	FOR_SHAP_R	DEB_SHAP_R	FORT_ICBP	DEB_ICBP	INFLUYENTES
C.DIFERENCIAL	**17**	**0**	**1**	**0**	**13**	**0**	**6**	**0**	**0**
ALGEBRA	**6**	**1**	**4**	**0**	**6**	**1**	**12**	**0**	**0.5**
G.ANALITICA	**13**	**1**	**0**	**0**	**6**	**1**	**1**	**1**	**0.75**
RPM	**0**	**1**	**2**	**0**	**0**	**0**	**5**	**2**	**0.75**
ET	**0**	**2**	**1**	**0**	**1**	**1**	**9**	**0**	**1.25**
F.BASICA	**1**	**2**	**1**	**0**	**1**	**1**	**8**	**0**	**1.25**
NP	**0**	**2**	**1**	**0**	**0**	**0**	**3**	**2**	**1.5**
IOM	0	1	2	0	0	0	2	4	5.25
TO	0	4	2	0	0	5	2	1	8.5
M.PROGRAMACION	4	2	2	0	0	1	3	6	8.75

Tabla 24. Consolidación del *Ranking* de alternativas para LA hasta primer semestre. Los factores más influyentes para este escenario fueron: Cálculo Diferencial, Álgebra, Geometría Analítica, Razón Promedio en Matemáticas, Total de Etiquetas, Física Básica y No Presentó, en orden descendente. Fuente: elaboración propia.

FACTORES	FOR-ICBGI	DEB-ICBGI	FOR-ICBP_ENS	DEB-ICBP_ENS	FOR_SHAP_R	DEB_SHAP_R	FORT_ICBP	DEB_ICBP	INFLUYENTES
IG	**19**	**0**	**13**	**0**	**19**	**0**	**13**	**0**	**0**
C.DIFERENCIAL	**17**	**1**	**0**	**0**	**15**	**1**	**1**	**1**	**0.75**
M.PROGRAMACION	**5**	**2**	**1**	**0**	**2**	**1**	**15**	**0**	**1.25**
RPM	**13**	**1**	**3**	**1**	**14**	**1**	**4**	**2**	**1.25**
ALGEBRA	14	2	6	0	12	2	1	3	5.25
G.ANALITICA	12	3	4	0	7	4	8	1	6.75
TO	2	4	4	1	3	5	5	1	8.75
ET	9	5	3	1	9	3	5	2	17
TOA04	7	3	0	4	2	6	0	8	21
F.BASICA	9	5	0	1	2	6	5	3	24

Tabla 25. Consolidación del *Ranking* de alternativas para LCC hasta primer semestre. Los factores más influyentes para este escenario fueron: Índice General, Cálculo Diferencial, Metodología de la Programación y Razón Promedio en Matemáticas, en orden descendente. Fuente: elaboración propia.

FACTORES	FOR-ICBGI	DEB-ICBGI	FOR-ICBP_ENS	DEB-ICBP_ENS	FOR_SHAP_R	DEB_SHAP_R	FORT_ICBP	DEB_ICBP	INFLUYENTES
IG	**1**	**0**	**1**	**0**	**0**	**0**	**7**	**0**	**0**
M.PROGRAMACION	**16**	**0**	**15**	**0**	**14**	**0**	**15**	**0**	**0**
RPM	**15**	**0**	**10**	**1**	**12**	**0**	**5**	**2**	**0.75**
G.ANALITICA	**10**	**1**	**4**	**0**	**10**	**2**	**5**	**0**	**0.75**
C.DIFERENCIAL	**8**	**1**	**1**	**1**	**9**	**2**	**0**	**2**	**1.5**
NP	**9**	**3**	**1**	**0**	**11**	**0**	**11**	**0**	**1.5**
ET	**11**	**3**	**5**	**0**	**14**	**0**	**10**	**1**	**1.75**
ALGEBRA	13	1	11	1	8	4	13	1	5.75
F.BASICA	3	4	1	3	1	3	0	4	12.25
IOM	3	7	0	3	0	6	5	4	31.5

Tabla 26. Consolidación del *Ranking* de alternativas para LMAD hasta primer semestre. Los factores más influyentes para este escenario fueron: Índice General, Metodología de la Programación, Razón Promedio en Matemáticas, Geometría Analítica, Cálculo Diferencial, No Presentó y Total de Etiquetas, en orden descendente. Fuente: elaboración propia.

FACTORES	FOR-ICBGI	DEB-ICBGI	FOR-ICBP_ENS	DEB-ICBP_ENS	FOR_SHAP_R	DEB_SHAP_R	FORT_ICBP	DEB_ICBP	INFLUYENTES
RPM	**15**	**0**	**6**	**0**	**14**	**0**	**18**	**0**	**0**
G.ANALITICA	**4**	**0**	**0**	**0**	**12**	**1**	**6**	**0**	**0.25**
NC	**7**	**0**	**1**	**0**	**7**	**0**	**15**	**2**	**0.5**
ET	**7**	**1**	**2**	**1**	**7**	**0**	**13**	**1**	**0.75**
NP	**2**	**2**	**2**	**0**	**6**	**2**	**16**	**1**	**1.75**
C.DIFERENCIAL	0	4	0	0	5	4	0	4	12
TOA03	0	2	0	1	0	6	9	5	15
ASA	1	2	1	1	0	4	6	8	16.25
F.BASICA	2	2	0	3	5	2	0	11	16.75
IOM	0	4	1	0	0	5	1	7	17

Tabla 27. Consolidación del *Ranking* de alternativas para LSTI hasta primer semestre. Los factores más influyentes para este escenario fueron: Razón Promedio en Matemáticas, Geometría Analítica, No Cumplió, Total de Etiquetas y No Presentó, en orden descendente. Fuente: elaboración propia.

FACTORES	FOR-ICBGI	DEB-ICBGI	FOR-ICBGI_ENS	DEB-ICBGI_ENS	FOR_SHAP_R	DEB_SHAP_R	FORT_ICBP	DEB_ICBP	INFLUYENTES
ET	27	0	28	0	27	0	29	0	0
NC	16	2	15	0	24	1	19	0	1.25
NP	21	2	1	1	20	1	23	1	1.75
RPM	30	0	9	0	28	0	9	4	5
C.DIFERENCIAL	9	3	17	1	8	4	13	2	7.25
ALGEBRA	17	1	0	1	18	3	9	5	10.5
TOA06	2	4	1	1	1	8	11	1	12.5
TOA09	0	4	0	2	0	7	15	3	15.5
IOM	5	7	1	2	6	1	16	3	22.5
ASA	1	3	0	5	2	0	2	8	24

Tabla 28. Consolidación del Ranking de alternativas para LA hasta segundo semestre. Los factores más influyentes para este escenario fueron: Total de Etiquetas, No Cumplió y No Presentó, en orden descendente. Fuente: elaboración propia.

FACTORES	FOR-ICBGI	DEB-ICBGI	FOR-ICBGI_ENS	DEB-ICBGI_ENS	FOR_SHAP_R	DEB_SHAP_R	FORT_ICBP	DEB_ICBP	INFLUYENTES
ET	19	1	19	0	22	1	34	0	0.5
P.ESTRUCTURADA	20	2	3	0	21	1	33	1	1.5
IOM	19	0	3	0	23	0	14	4	5
NP	13	1	1	2	14	1	21	3	5.25
IG	33	0	8	0	30	0	10	5	6.25
G.ANALITICA	6	4	1	1	1	3	3	2	6.5
RPM	26	1	1	0	21	0	12	5	6.5
TO303	13	2	0	0	7	2	13	7	10.25
TO301	2	3	1	2	2	8	14	4	17.5
NC	13	4	3	1	17	1	6	13	18.75

Tabla 29. Consolidación del *Ranking* de alternativas para LCC hasta segundo semestre. Los factores más influyentes para este escenario fueron: Total de Etiquetas y Programación Estructurada, en orden descendente. Fuente: elaboración propia.

FACTORES	FOR-ICBGI	DEB-ICBGI	FOR-ICBGI_ENS	DEB-ICBGI_ENS	FOR_SHAP_R	DEB_SHAP_R	FORT_ICBP	DEB_ICBP	INFLUYENTES
IG	**32**	**0**	**14**	**0**	**32**	**0**	**14**	**1**	**0.25**
NP	**19**	**1**	**27**	**0**	**18**	**1**	**32**	**0**	**0.5**
ET	**11**	**1**	**21**	**1**	**16**	**1**	**30**	**1**	**1**
M.PROGRAMACION	**12**	**1**	**13**	**0**	**6**	**1**	**14**	**2**	**1**
C.DIFERENCIAL	**16**	**2**	**21**	**0**	**11**	**2**	**27**	**2**	**2**
RPM	28	1	14	0	22	1	16	4	5.5
ALGEBRA	5	4	16	1	5	1	18	3	6.25
TOA02	2	3	16	1	1	4	22	2	7.25
TO501	1	3	6	0	1	4	12	6	14
F.BASICA	12	2	10	4	2	5	19	4	14.25

Tabla 30. Consolidación del *Ranking* de alternativas para LMAD hasta segundo semestre. Los factores más influyentes para este escenario fueron: Índice General, No Presentó, Total de Etiquetas, Metodología de la Programación y Cálculo Diferencial, en orden descendente. Fuente: elaboración propia.

FACTORES	FOR-ICBGI	DEB-ICBGI	FOR-ICBGI_ENS	DEB-ICBGI_ENS	FOR_SHAP_R	DEB_SHAP_R	FORT_ICBP	DEB_ICBP	INFLUYENTES
IG	**32**	**0**	**3**	**0**	**29**	**0**	**5**	**0**	**0**
I.PROGRAMACION	**22**	**1**	**22**	**0**	**26**	**0**	**23**	**0**	**0.25**
ET	**22**	**1**	**6**	**0**	**28**	**1**	**21**	**0**	**0.5**
C.INTEGRAL	**4**	**4**	**2**	**1**	**8**	**0**	**20**	**2**	**2.75**
TO	11	4	0	1	11	5	19	1	8.75
TOA02	6	4	0	1	4	5	20	2	9
AVANCE	9	1	0	1	10	4	9	5	11.75
TO202	10	4	0	1	11	5	17	5	14.75
IOM	2	2	3	1	3	7	19	4	15
FSI	12	3	13	1	13	5	16	7	16.75

Tabla 31. Consolidación del *Ranking* de alternativas para LSTI hasta segundo semestre. Los factores más influyentes para este escenario fueron: Índice General, Introducción a la Programación, Total de Etiquetas y Cálculo Integral, en orden descendente. Fuente: elaboración propia.