Modelo Predictivo de Deserción Estudiantil de Educación Preescolar, Básica y Media en el Municipio de Medellín

Celger Paola Chamat Torres

Monografía presentada para optar al título de Especialista en Analítica y Ciencia de Datos

Asesor
Efraín Alberto Oviedo Carrascal, Magíster (MSc) TIC's

Universidad de Antioquia
Facultad de Ingeniería
Especialización en Analítica y Ciencia de Datos
Medellín, Antioquia, Colombia
2021

Cita	(Chamat Torres, 2021)
Referencia **Estilo APA 7 (2020)**	Chamat Torres, C. P. (2021). *Modelo Predictivo de Deserción Estudiantil de Educación Preescolar, Básica y Media en el Municipio de Medellín*, Trabajo de grado especialización. Universidad de Antioquia, Medellín, Colombia.

Especialización en Analítica y Ciencia de Datos, Cohort**e** II.

UNIVERSIDAD DE ANTIOQUIA
Vicerrectoría de Docencia

Sistema de Bibliotecas

Centro de documentación Ingeniería (CENDOI)

Repositorio Institucional: http://bibliotecadigital.udea.edu.co

Universidad de Antioquia - www.udea.edu.co

Rector: John Jairo Arboleda

Decano/Director: Jesús Francisco Vargas Bonilla

Jefe departamento: Diego José Luis Botia Valderrama

TABLA DE CONTENIDO

Pág.

LISTA DE FIGURAS

Pág.

LISTA DE TABLAS

Pág.

1. RESUMEN EJECUTIVO

La deserción escolar entendiéndose como la interrupción, retiro o abandono del estudiante del sistema educativo, es un panorama educativo altamente problemático para el Estado y la sociedad en general, por su relación con la afectación al derecho fundamental del acceso a la educación y al desarrollo normal de un individuo en su etapa de escolaridad, soportado en el Artículo 67 de la Constitución Política de Colombia de 1991. Por ello, es de vital importancia para los entes reguladores velar y garantizar la permanencia educativa de todos los niños, jóvenes y adolescentes en su entorno escolar. Es allí, donde el gran potencial de los datos en conjunto con distintos actores multidisciplinarios permitiría la construcción de estrategias de control innovadoras y oportunas que respondan a las necesidades reales de la ciudadanía, aportando al mejoramiento de la calidad educativa y a los procesos de diagnóstico, planeación, ejecución, seguimiento y evaluación.

Dada la necesidad de disminuir la tasa de estudiantes que abandonan el sistema educativo, se propone desarrollar un modelo predictivo de deserción estudiantil de Educación Preescolar, Básica y Media en el Municipio de Medellín, que permita a partir de técnicas de Machine Learning clasificar en posibles desertores (1) y no desertores (0) a aquellos estudiantes que según sus características académicas, sociodemográficas, socioeconómicas y familiares presentan un mayor riesgo de abandonar la escuela en el sector Oficial.

Para la ejecución del proyecto se recolectó la fuente de información de matrícula al año 2019 suministrada por el Observatorio para la Calidad Educativa de Medellín (OCEM) de la Secretaría de Educación. A partir de la cual, se inició un proceso metodológico que consistió en la preparación de los datos para cruzar con un dataset complementario proveniente de la encuesta del Sisbén en la ciudad, el preprocesamiento de los datos para depurar, limpiar, imputar, transformar y codificar las variables, el balanceo de la clase minoritaria de la variable objetivo a partir de la técnica de sobremuestreo SMOTE, la implementación y entrenamiento de algoritmos de clasificación de aprendizaje supervisado tales como: RandomForestClassifier, StackingClassifier, BaggingClassifier de la librería de Scikit-Learn, así como una red neuronal con autoencoder de la plataforma TensorFlow y un algoritmo de ensemble XGBClassifier de XGBoost, finalizando el proceso con una validación de las métricas obtenidas en cada uno de los modelos y secuencia de iteraciones. Gracias a los resultados obtenidos en cada iteración realizada, fue posible ajustar los parámetros y definir las acciones de mejora en los datos y variables, con el fin de ir de manera progresiva aumentando el porcentaje de verdaderos positivos y disminuyendo la tasa de falsos negativos. Teniendo entonces como resultado final el mejor modelo de XGBClassifier, con una clasificación de los posibles desertores, es decir, de verdaderos positivos del 97% y aproximadamente el 100% de la clasificación de los no desertores.

Palabras Claves: Aprendizaje supervisado, Clasificación, Deserción escolar, Educación, Machine Learning, Modelo predictivo.

GitHub: https://github.com/CelgerpaoCh/UDEA_proyecto_desertores

2. DESCRIPCIÓN DEL PROBLEMA

Teniendo en cuenta la problemática alrededor de la deserción estudiantil y la desescolarización en los niveles de Educación Preescolar, Básica y Media, el Municipio de Medellín desde la Secretaría de Educación ha implementado distintas estrategias, tales como; donatón tecnológica, el programa de Entorno Escolar Protector, entrega de kits escolares, transporte escolar, entre otros, con el fin de fortalecer, velar y garantizar el cumplimiento del derecho al acceso y la permanencia educativa. Sin embargo, se requiere optimizar la priorización de aquellos estudiantes que según la evidencia de los datos tienen el mayor riesgo de desertar del servicio educativo y anticiparse a esta situación con el fin de tomar acciones de control frente a la focalización de las estrategias a esta población que mitiguen este panorama e incida en las necesidades reales de los niños, niñas, adolescentes y establecimientos educativos.

Por lo anterior, con el propósito de disminuir la tasa de deserción en Medellín, es posible implementar modelos predictivos mediante las técnicas de Machine Learning que permitan clasificar a aquellos estudiantes que, según sus características académicas, sociodemográficas, socioeconómicas, subjetivas y familiares presentan un mayor riesgo de abandonar la escuela en el sector Oficial.

2.1 Problema de negocio

Desde la Secretaría de Educación de Medellín, se requiere disminuir la tasa de deserción estudiantil y mejorar los procesos de priorización y selección de estudiantes para el beneficio de las estrategias de permanencia educativa implementada por la misma. Para ello, se cuenta con bases de datos históricas sobre el comportamiento de la matrícula para el sector Oficial, Privado y de Cobertura Contratada, además de la georreferenciación de los estudiantes y características sobre su situación social y económica aportadas por la encuesta del SISBEN.

La importancia de la disminución de la tasa de desertores del sistema educativo se encuentra relacionado directamente con la razón de costo/beneficio desde aspectos como: la reinversión del recurso, riesgos psicosociales, aumento en la tasa de desempleo y el trabajo informal desde el aspecto social.

2.2 Aproximación desde la analítica de datos

Desde el gran potencial de los datos y las técnicas de Machine Learning y sus algoritmos de aprendizaje supervisado, se propone la construcción e implementación de un modelo predictivo de clasificación binaria que logre categorizar en posibles desertores (1) y no desertores (0) a los estudiantes de educación regular de los colegios Oficiales de Medellín a partir de predicciones a futuro basadas en comportamientos o características que se han visto en el histórico de los datos almacenados. Esta propuesta nace, debido a la identificación del alto desbalanceo en las clases de interés; correspondientes a un 2,64% para la clase minoritaria 1 y un 97,36% para la clase 0, por lo cual desde un primer acercamiento en la visualización de las características no es evidente alguna correlación entre los factores y los patrones con mayor influencia incidentes en la deserción educativa.

2.3 Origen de los datos

Los datos utilizados en este proyecto para la creación del modelo predictivo, son proporcionados por el Observatorio para la Calidad Educativa de Medellín (OCEM), de la Secretaría de Educación (sitio web: https://www.medellin.edu.co/secretaria/ocem/). Este conjunto de datos seleccionado corresponde a la información de la matrícula estudiantil al año 2019 (antes del contexto de Pandemia mundial por COVID19) generada por el Ministerio de Educación Nacional (MEN), mediante el Sistema Integrado de Matrícula

(SIMAT); herramienta que permite organizar y controlar el proceso de matrícula en todas sus etapas, así como tener una fuente de información confiable y disponible para la toma de decisiones, permitiendo a las Secretarías de Educación sistematizar, consolidar y analizar la información concerniente a la comunidad educativa.

Adicional a la base de datos principal de matrícula, se cuenta con la base de datos de la encuesta del Sistema de Identificación de Potenciales Beneficiarios de Programas Sociales - Sisbén generada por el Departamento Nacional de Planeación (DNP), a la cual se le extraen unas características en específico con el fin de complementar el perfil del estudiante en cuanto a su situación socioeconómica y de entorno familiar.

3. DATOS

3.1 Datos originales

Los datos originales utilizados en este estudio provienen en una primera parte de los datos generados por el Sistema Integrado de Matrícula – SIMAT del Ministerio de Educación Nacional. Estos se encuentran con fecha de corte del mes de mayo del año 2019 y la razón por la cual se realiza esta elección es debido a que este mes contiene el mayor número de estudiantes matriculados en el año. Este conjunto de datos en formato de valores separados por comas (.csv) contiene información cuantitativa y cualitativa de los estudiantes de los colegios de Medellín, incluyendo los pertenecientes al sector Oficial, Privado y de Cobertura Contratada, y consta de 418.741 registros, 63 características (de las cuales 8 corresponden a información personal) y un tamaño en disco de 96,7 MB. Este primer y principal conjunto de datos se identifica por el nombre "simat_anexo_201905" y la variable a predecir lleva por nombre "ESTADO_DEFINITIVO" la cual fue agregada de manera previa a la base de datos con la información preliminar obtenida del final del año escolar. Las características disponibles en el archivo se describen en la Tabla 1.

Tabla 1. Descripción de base de datos simat_anexo_201905

VARIABLES	DESCRIPCIÓN	REGISTROS
MUN_CODIGO	Códigos DANE Municipios o Distrito	Códigos DANE Municipios (3 posiciones)
CODIGO_DANE	Código DANE de la institución educativa	Código DANE de la institución educativa (12 dígitos)
CODIGO_DANE_SEDE	Código DANE sede educativa en el año 2001 antes de la fusión establecida por la Ley 715	Código DANE de la sede educativa (12 dígitos)
CONS_SEDE	Consecutivo DANE de la sede educativa	Consecutivo DANE de la sede educativa (14 dígitos)
sede	Cantidad de sedes escuelas que contiene una institución educativa	Valor numérico entre 1 y 7
prestacion_servicio	Tipo de prestación de servicio de la institución educativa	Oficial
		Cobertura contratada
		Privado
EXP_DEPTO	Departamento de expedición del documento de identidad del Alumno	Códigos DANE Departamento (2 posiciones)
EXP_MUN	Municipio de expedición del documento de identidad del Alumno	Códigos DANE Municipio (3 posiciones)
RES_DEPTO	Departamento de residencia del Alumno	Códigos DANE Departamento (2 posiciones)
RES_MUN	Municipio de residencia del Alumno	Códigos DANE Municipio (3 posiciones)
ESTRATO	Estrato Socioeconómico del Alumno	0 - Estrato 0
		1 - Estrato 1
		2 - Estrato 2

		3 - Estrato 3
		4 - Estrato 4
		5 - Estrato 5
		6 - Estrato 6
		9 - No Aplica
SISBEN	Nivel de Sisbén del Alumno	1 - SISBEN 1
		2 - SISBEN 2
		3 - SISBEN 3
		4 - SISBEN 4
		5 - SISBEN 5
		6 - SISBEN 6
		9 - NO APLICA
NAC_DEPTO	Departamento de nacimiento del Alumno	Códigos DANE Departamento (3 posiciones)
NAC_MUN	Municipio de nacimiento del Alumno	Códigos DANE Municipio (2 posiciones)
GENERO	Género	F - Femenino
		M - Masculino
POB_VICT_CONF	Población Victima del Conflicto	1 - En Situación de desplazamiento
		2 - Desvinculados de grupos armados
		3 - Hijos de adultos desmovilizados
		4 - Víctimas de Minas
		5 - Responsabilidad Penal
		6 - Acto Terrorista /Atentados/ Combates/ Enfrentamientos / Hostigamientos
		7 - Amenaza
		8 - Delitos contra la libertad e integridad sexual en el marco del conflicto armado
		9 - Desaparición forzada
		10 - Desplazamiento Forzado
		11 - Homicidio
		12 - Secuestro
		13 - Tortura
		14 - Vinculación de Niños Niñas Adolescentes a actividades relacionadas con grupos armados
		15 - Abandono o despojo de tierras
		16 - Perdida de bienes muebles o inmuebles
		17 - Otros

		18 - Sin Información
		19 - Confinamiento
		20 - Lesiones personales físicas
		21 - Lesiones personales psicológicas
		99 - No Aplica
DPTO_EXP	Último Departamento Expulsor	Códigos DANE Departamento (2 posiciones)
MUN_EXP	Último Municipio Expulsor	Códigos DANE Municipio (3 posiciones)
PROVIENE_SECTOR_PRIV	Proviene de Sector Privado	S - Si
		N - No
PROVIENE_OTRO_MUN	Proviene de Otro Municipio	S - Si
		N - No
TIPO_DISCAPACIDAD	Tipo de Discapacidad	3 - Visual - Baja Visión Irreversible
		4 - Visual - Ceguera
		7 - Trastorno del Espectro Autista
		8 - Discapacidad Intelectual
		10 - Discapacidad Múltiple
		12 - Discapacidad Auditiva - Usuario Lengua de Señas Colombiana
		13 - Discapacidad Auditiva - Usuario del Castellano
		14 - Sordoceguera
		15 - Discapacidad Física
		18 - Discapacidad Psicosocial (Mental)
		99 - No Aplica
CAP_EXC	Capacidades Excepcionales	1 - Capacidades excepcionales
		3 -Talento excepcional en ciencias naturales o básicas
		4 - Talento excepcional en artes o en letras
		5 - Talento excepcional en actividad física, ejercicio y deporte
		7 - Talento excepcional en ciencias sociales o humanas
		10 - Talento excepcional en tecnología
		11 - Talento excepcional en liderazgo social y emprendimiento

		9 - No Aplica
ETNIA	Etnia a la que pertenece un Alumno	Valores numéricos del listado de Etnias, donde 0 corresponde a los No Aplica
RES	Resguardo al que pertenece un Alumno	Valores numéricos del listado de Resguardos, donde 0 corresponde a los No Aplica
TIPO_JORNADA	Tipo de Jornada	1 - Completa
		2 - Mañana
		3 - Tarde
		4 - Nocturna
		5 - Fin de semana
		6 - Única
CARACTER	Carácter del Modelo Educativo	1 - Académico
		2 - Técnico
		0 - No Aplica
ESPECIALIDAD	Especialidad del Modelo Educativo Técnico	05 - Académico
		06 - Industrial
		08 - Comercial
		09 - Pedagógico
		10 - Agropecuario
		16 - Promoción social
		07 - Otro
		00 - No Aplica
GRADO	Grado del Alumno	-2 - Pre-Jardín
		-1 - Jardín I o A o Kínder
		0 - Transición o Grado 0
		1 - Primero
		2 - Segundo
		3 - Tercero
		4 - Cuarto
		5 - Quinto
		6 - Sexto
		7 - Séptimo
		8 - Octavo
		9 - Noveno
		10 - Décimo
		11 - Once
		12 - Doce - Normal Superior
		13 - Trece - Normal Superior
		21 - Ciclo 1 Adultos
		22 - Ciclo 2 Adultos
		23 - Ciclo 3 Adultos

		24 - Ciclo 4 Adultos
		25 - Ciclo 5 Adultos
		26 - Ciclo 6 Adultos
		99 - Aceleración del Aprendizaje
GRUPO	Grupo del Alumno definido en SIMAT	Valores tipo Object
METODOLOGIA	Metodología de aprendizaje del Alumno	1 - Educación tradicional
		2 - Escuela nueva
		3 - Post primaria
		4 - Telesecundaria
		5 - Ser
		6 - Cafam
		7 - Sat
		8 - Etnoeducación
		9 - Aceleración del aprendizaje
		10 - Programa para jóvenes en extraedad y adultos
		11 - Preescolar escolarizado
		12 - Preescolar no escolarizado/semiescolarizado
		13 - Sat presencial
		14 - Entorno comunidad
		15 - Entorno familiar
		16 - Entorno institucional
		17 - Circulos de aprendizaje
		18 - Media rural
		19 - Transformemos
		20 - Grupos juveniles creativos
		21 - Modalidad virtual asistida UCN
		22 - A crecer
		23 - Bachillerato Pacicultor
		24 - A crecer a través de celulares
		25 - SENA
		26 - Ser humano
		27 - Vamos a poder
		28 - FIMACAF
		29 - Caminar en secundaria
		30 - ESPERE
		31 - Escuela indígena intercultural de jóvenes y adultos - ACIN

		32 - UNAD
		33 - Formación para la reintegración
		34 - Etnoeducativo para comunidades negras – Pacifico colombiano
		35 - Flexible pensar adultos
		36 - Flexible escuela integral
		37 - Flexible pensar
		38 - Retos para gigantes
		39 - Secundaria activa
		40 - La pedagogía del texto cleba
		41 - Todos contamos
		42 - Shur payan
		43 - Escuela nueva activa
		44 - Propuesta para cambiar entornos sociales (paces)
		45 - Caminar en secundaria I
		46 - Caminar en secundaria II
		47 - Comprender y prosperar
		48 - CRIC
SUBSIDIADO	Valida si el Alumno está subsidiado	S - Si
		N - No
REPITENTE	Repitente	S - Si
		N - No
NUEVO	Nuevo en la Institución	S - Si
	Educativa	N - No
SIT_ACAD_ANO_ANT	Situación Académica del Año Anterior	0 - No Estudió Vigencia Anterior, que para este año se refiere a no haber estudiado en vigencia anterior
		1 - Aprobó
		2 - Reprobó
		4 - Pendiente de logros
		6 - Viene de otra Institución Educativa
		7 - Ingresa por primera vez al sistema
		8 - No Culminó Estudios
CON_ALUM_ANO_ANT	Condición del Alumno al Finalizar el Año Anterior	3 - Desertó
		5 - Trasladado a otra Institución Educativa
		9 - No Aplica
FUE_RECU	Fuente de recursos	1 - SGP
		2 - FNR

		3 - Recursos adicionales presupuesto nacional MEN
		4 - Otros Recursos de la Nación
		5 - Recursos Propios de la Secretaría de Educación
ZONA_ALU	Zona Residencial del Alumno	1 - Urbana
		2 - Rural
CAB_FAMILIA	Alumna Madre Cabeza de Familia	S - Si
		N - No
BEN_MAD_FLIA	Beneficiario Hijos Dependientes de Madre Cabeza de Familia	S - Si
		N - No
BEN_VET_FP	Beneficiario Veteranos de la Fuerza Pública	S - Si
		N - No
BEN_HER_NAC	Beneficiario Héroes de la Nación	S - Si
		N - No
INTERNADO	Internado	1 - Internado
		2 - Semi-Internado
		3 - Ninguno
VAL_DES_PERIODO1	Código valoración 1	1 - Superior
		2 - Alto
		3 - Básico
		4 - Bajo
VAL_DES_PERIODO2	Código valoración 2	1 - Superior
		2 - Alto
		3 - Básico
		4 - Bajo
NUM_CONVENIO	Número de contrato de prestación del Servicio Educativo	Valores numéricos
men_per_id	Número único de identificación de persona asignado por el sistema	Valores numéricos
APOYO_ACAD_ESP	Apoyo Académico Especial	1 - No Aplica
		2 - Aula Hospitalaria
		3 - Atención Domiciliaria
		4 - Atención en Institución de apoyo
		5 - Atención en el Establecimiento Educativo
CTE_ID_SRPA	Sistema Responsabilidad Penal	1 - No Aplica
		2 - Privado de la libertad
		3 - No Privado de la libertad
CODIGO_PAIS_ORIGEN	Código del país de origen del Alumno	Acorde con la divipola internacional

tipo_anexo_id	Tipo de anexo del reporte del SIMAT	5A, 5B y 5O - Sector No Oficial
		6A y 6O - Sector Oficial
COMUNA_EST	Comuna en la cual Reside el Estudiante	Valores de las 16 comunas y los 5 corregimientos de Medellín
EDAD	Edad del Estudiante al Año	Valores numéricos correspondientes a las edades de los estudiantes
ESTADO_DEFINITIVO (TARGET)	Estado final del Estudiante en el Corte de Consolidación o Matrícula Definitiva	1 - Matriculado
		2 - Retirado

Como segundo conjunto de datos en formato de valores separados por comas (.csv), se tienen los provenientes de la encuesta más actualizada del Sisbén aplicada en Medellín, la cual clasifica a la población mediante un puntaje dependiendo sus condiciones socioeconómicas. Esta base de datos original consta de 2.020.090 registros, 72 características y un tamaño en disco de 531 MB, sin embargo, previamente se realiza una selección de las características necesarias para el estudio, quedando entonces con una cantidad de 15 variables (de las cuales 6 corresponden a información personal), su descripción se puede observar en la Tabla 2.

Tabla 2. Descripción de base de datos BD_SISBEN

VARIABLES	DESCRIPCIÓN	REGISTROS
ficha	Consecutivo de vivienda	Variable numérica entera (int64)
hogar	Consecutivo del hogar	Variable numérica entera (int64)
ingresos	Ingresos reportados por encuestado	Variable numérica entera (int64)
PUNTAJE	Puntaje Sisbén del encuestado	Variable numérica decimal (float64)
extranjero	Valida si el encuestado es extranjero	1 - Si
		2 - No
telefono	Valida si el encuestado tiene teléfono	1 - Si
		2 - No
computador	Valida si el encuestado tiene computador	1 - Si
		2 - No
embaraza	Valida si la encuestada está embarazada o ha tenido hijos	1 - Si
		2 - No
percibe	Valida si el encuestado percibe ingresos (laborales, arriendos, subsidios, transferencias, en especie)	1 - Si
		2 - No

Es importante mencionar que por motivos de confidencialidad y tratamiento de los datos no es posible dar acceso a los datasets mencionados anteriormente de manera original y con toda la información sensible de los estudiantes. Por lo anterior, se realiza una etapa de preprocesamiento la cual se explica en la sección 3.2, en donde se realizan cruces entre las dos fuentes de información tanto de matrícula como Sisbén y se genera un nuevo y único conjunto de datos con la depuración de aquellas variables con información personal tales como; nombres, apellidos, tipos de documento, documentos de identificación, direcciones y teléfonos, de todos los estudiantes, al cual se puede acceder y descargar para la ejecución del modelo.

3.2 Preparación de los datos

Como proceso inicial luego de la recolección de los conjuntos de datos, se realiza una etapa de preparación la cual consiste en los siguientes pasos:

- Lectura de la base de datos de matrícula SIMAT “simat_anexo_201905.csv” y la base de datos del Sisbén “BD_SISBEN.csv”.
- Selección de las variables del Sisbén que son caso de estudio correspondientes a algunas características socioeconómicas y familiares.
- Creación de una nueva variable “ingresos_promedio” a partir de las existentes en la base de datos del Sisbén "ficha", "hogar" e “ingresos”, con el fin de obtener el ingreso promedio del encuestado según el hogar al que pertenece. Esto se hace, ya que los estudiantes en su mayoría tienen en la variable "ingresos" valores de 0 por no ser trabajadores, lo que conlleva a que información de esta variable no sea relevante. El cálculo realizado consiste en la siguiente fórmula:

$$\text{ingresos\_promedio} = \frac{\text{Ingresos totales en el hogar}}{\text{Cantidad total de habitantes en el hogar}}$$

- Primer cruce o merge entre la base de datos principal de matrícula del SIMAT y la base de datos del Sisbén a partir de los números de documento de identidad de los estudiantes comunes en ambos datasets. Como resultado de este proceso, se obtienen 143.145 registros comunes del total de 418.741 registros de la base principal de matrícula, es decir, el 34,18% de la información.
- Segundo cruce o merge entre la base de datos principal de matrícula del SIMAT y la base de datos del Sisbén de los registros que no cruzaron en el primer procedimiento, a partir de los nombres completos y fechas de nacimiento de los estudiantes comunes en ambos datasets. Como resultado de este proceso, se obtienen 90.872 registros comunes, del restante total de 275.676 registros de estudiantes en la base principal.
- Unión de los datasets resultado de los cruces anteriormente mencionados, obteniendo 234.017 registros en común en ambas bases de datos. Lo que quiere decir que el 55,9% de los estudiantes tienen features adicionales provenientes de la encuesta del Sisbén, por lo cual serán los seleccionados para el modelo predictivo.
- Eliminación de las variables con información personal y sensible de los estudiantes tales como; nombres completos, apellidos, documentos de identidad, fechas de nacimiento, ficha de la encuesta, ficha del hogar encuestado, direcciones de residencia y números telefónicos.

- Se exporta el dataset “BD_anexo_sisben_2019.csv” con un total de 234.017 registros, 60 características y un tamaño en disco de 57,9 MB; fuente de información de entrada del modelo predictivo.

El procedimiento descrito anteriormente, se encuentra consignado en el notebook titulado “Etapa_preparacion_modelo.ipynb”. Debido a la confidencialidad de los datos, este notebook no podrá ser reproducido, ya que las fuentes de información utilizadas contienen datos personales de cada estudiante, necesarios para encontrar sus registros únicos y comunes entre las bases de datos cruzadas para la obtención del dataset exportado con los casos de estudio para el modelo predictivo.

3.3 Ejecución del entorno

Para el acceso y lectura del dataset final “BD_anexo_sisben_2019.csv” con los datos de entrada para la construcción y ejecución del modelo predictivo y, al tratarse de información de carácter confidencial y licenciada, se opta por no colgarla en ningún repositorio, de modo que, el procedimiento para la preparación del entorno de ejecución y la reproducibilidad del notebook desarrollado titulado “proyecto_desertores_educacion.ipynb”, consiste en los siguientes pasos:

- Descargar de la carpeta Drive compartida con el proyecto, el archivo de valores separados por comas llamado “BD_anexo_sisben_2019.csv”.
- Si la ejecución se realizará mediante el entorno de Colab se necesita abrir el notebook “proyecto_desertores_educacion.ipynb”, dirigirse a la carpeta “Archivos” y subir al almacenamiento de sesión la base de datos descargada en el paso anterior.
- Si la ejecución se realizará de manera local se necesita crear una carpeta en la cual se almacene el dataset descargado en el primer paso, junto con el notebook “proyecto_desertores_educacion.ipynb”.
- Una vez la carga de conjunto de datos está completa, es posible iniciar la ejecución del entorno.

Estas instrucciones para la ejecución del entorno se encuentran en el archivo “0_instrucciones.ipynb”.

3.4 Descriptiva

El dataset de estudio en formato csv obtenido en la etapa de preparación (descrita en la sección 3.2) cuenta con un total de 234.017 registros y 61 variables características, los tipos y cantidad de registros nulos de estas últimas se muestran en la Figura 1.

Figura 1. Dataset en crudo

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 234017 entries, 0 to 234016
Data columns (total 61 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   index                 234017 non-null  int64
 1   MUN_CODIGO            234017 non-null  float64
 2   CODIGO_DANE           234017 non-null  float64
 3   CODIGO_DANE_SEDE      234017 non-null  float64
 4   CONS_SEDE             234017 non-null  float64
 5   sede                  234017 non-null  float64
 6   prestacion_servicio   234017 non-null  object
 7   EXP_DEPTO             233199 non-null  float64
 8   EXP_MUN               233199 non-null  float64
 9   RES_DEPTO             234017 non-null  float64
 10  RES_MUN               234017 non-null  float64
 11  ESTRATO               234017 non-null  float64
 12  SISBEN                188308 non-null  float64
 13  NAC_DEPTO             233241 non-null  float64
 14  NAC_MUN               233241 non-null  float64
 15  GENERO                234017 non-null  object
 16  POB_VICT_CONF         234017 non-null  float64
 17  DPTO_EXP              10890 non-null   float64
 18  MUN_EXP               10890 non-null   float64
 19  PROVIENE_SECTOR_PRIV  234017 non-null  object
 20  PROVIENE_OTRO_MUN     234017 non-null  object
 21  TIPO_DISCAPACIDAD     234017 non-null  float64
 22  CAP_EXC               234017 non-null  float64
 23  ETNIA                 234017 non-null  float64
 24  RES                   234017 non-null  float64
 25  TIPO_JORNADA          234017 non-null  float64
 26  CARACTER              234017 non-null  float64
 27  ESPECIALIDAD          234017 non-null  float64
 28  GRADO                 234017 non-null  float64
 29  GRUPO                 234017 non-null  object
 30  METODOLOGIA           234017 non-null  float64
 31  SUBSIDIADO            234017 non-null  object
 32  REPITENTE             234017 non-null  object
 33  NUEVO                 234017 non-null  object
 34  SIT_ACAD_ANO_ANT      234017 non-null  float64
 35  CON_ALUM_ANO_ANT      234017 non-null  float64
 36  FUE_RECU              234017 non-null  float64
 37  ZON_ALU               234017 non-null  float64
 38  CAB_FAMILIA           234017 non-null  object
 39  BEN_MAD_FLIA          234017 non-null  object
 40  BEN_VET_FP            234017 non-null  object
 41  BEN_HER_NAC           234017 non-null  object
 42  INTERNADO             210280 non-null  float64
 43  VAL_DES_PERIODO1      10 non-null      float64
 44  VAL_DES_PERIODO2      0 non-null       float64
 45  NUM_CONVENIO          23470 non-null   float64
 46  men_per_id            234017 non-null  float64
 47  APOYO_ACAD_ESP        207901 non-null  float64
 48  CTE_ID_SRPA           207901 non-null  float64
 49  CODIGO_PAIS_ORIGEN    197496 non-null  float64
 50  tipo_anexo_id         234017 non-null  object
 51  ESTADO_DEFINITIVO     234017 non-null  float64
 52  COMUNA_EST            234017 non-null  float64
 53  EDAD                  234017 non-null  float64
 54  PUNTAJE               234017 non-null  float64
 55  extranjero            234017 non-null  int64
 56  telefono              234017 non-null  int64
 57  computador            234017 non-null  int64
 58  embaraza              234017 non-null  int64
 59  percibe               234017 non-null  int64
 60  ingresos_promedio     234017 non-null  int64
dtypes: float64(41), int64(7), object(13)
memory usage: 108.9+ MB
```

Como se puede observar en la Figura 1, cuando el dataset es leído se toman en su gran mayoría todas las variables como numéricas, sin embargo, entre ellas se encuentran en una mayor proporción valores que corresponden a categorías, por esta razón, la exploración y visualización de dichas variables categóricas se realizará posteriormente cuando se transforme su tipología (ver sección 4.2.3.1).

A partir del análisis de la Figura 2 acerca de la correlación lineal existente entre las variables de todo el conjunto de datos, se puede evidenciar como algunas se encuentran altamente relacionadas, por lo que aquellas que tienen el cuadro relleno y completamente azul indican una correlación de 1. Un ejemplo significativo de lo mencionado anteriormente se puede ver específicamente en la de variable 'MUN_CODIGO', en la cual se logró identificar que la alta correlación (del rango entre -1 a 1) producida se debe a que solo cuenta con un único registro "1" como se muestra en la Figura 3, dicha correlación se presenta con variables tales como 'sede', 'prestacion_servicio', 'GENERO', 'POB_VICT_CONF', entre otras. La decisión que se toma respecto a esta situación, se describe en la sección 4.2.1.1.

Figura 2. Correlación entre variables

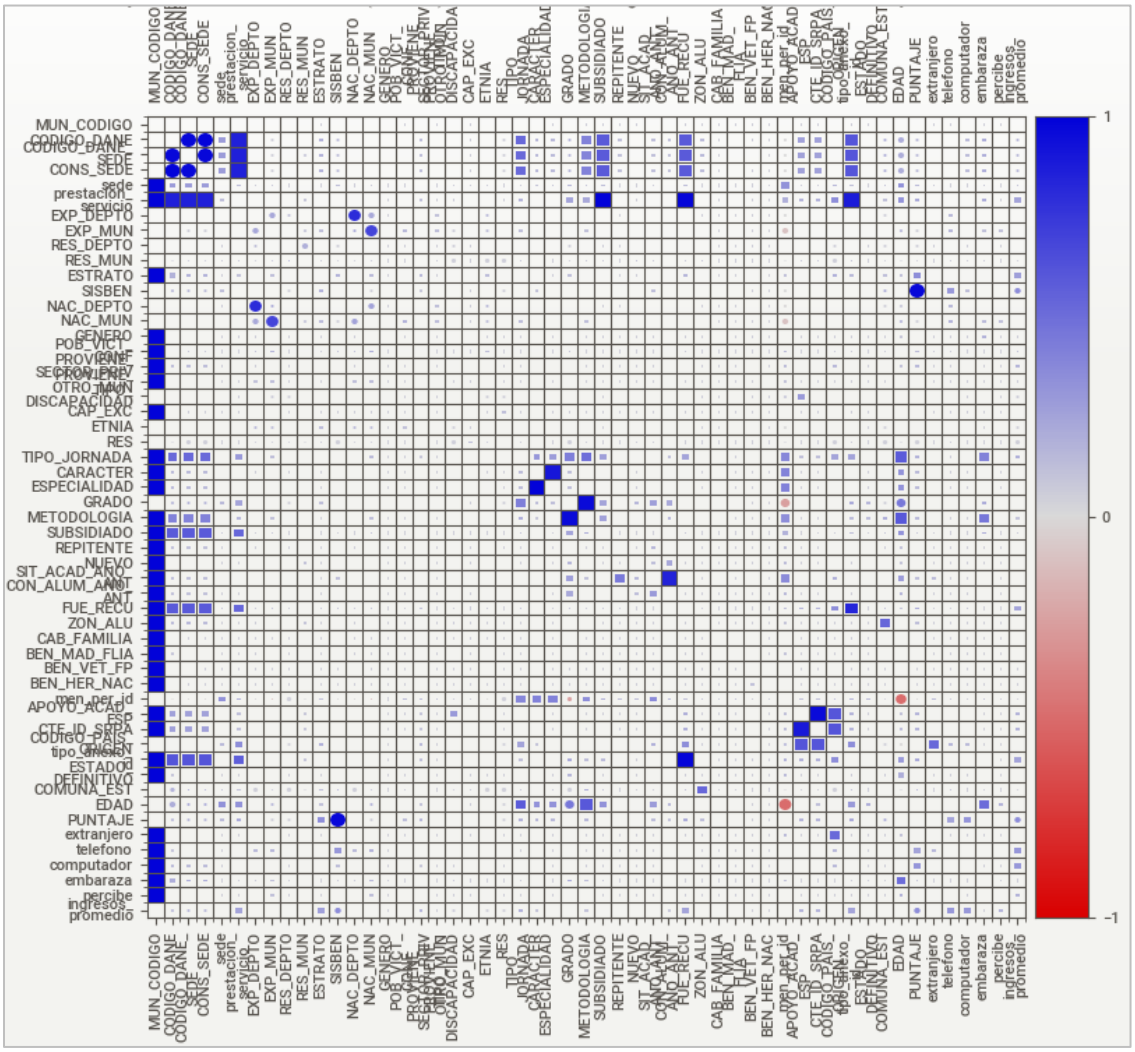


Figura 3. Comportamiento Variable 'MUN_CODIGO'

```
df_simat['MUN_CODIGO'].unique()
```

```
array([1.])
```

3.4.1 Variables categóricas

Una vez es cargado y leído el conjunto de datos, se realiza un recorrido por las columnas para obtener cuales de ellas corresponden a variables categóricas según su exploración inicial y sin ningún tipo de manipulación o transformación de tipología. Esto se puede ver en la Figura 4.

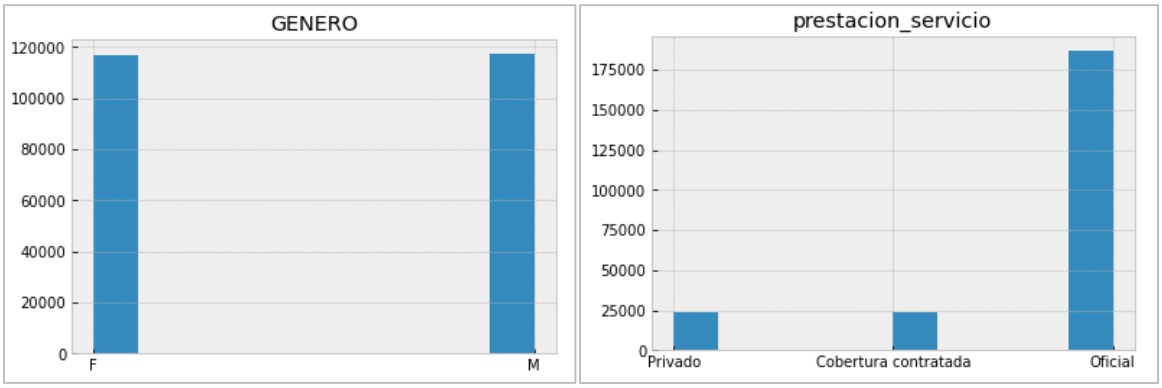
Figura 4. Variables categóricas dataset en crudo

```
cols_categoricas = [i for i in df_simat.columns if not i in df_simat._get_numeric_data()]
print (cols_categoricas)
```

```
['prestacion_servicio', 'GENERO', 'PROVIENE_SECTOR_PRIV', 'PROVIENE_OTRO_MUN', 'GRUPO', 'SUBSIDIADO', 'REPITENTE', 'NUEVO', 'CA
B_FAMILIA', 'BEN_MAD_FLIA', 'BEN_VET_FP', 'BEN_HER_NAC', 'tipo_anexo_id']
```

En la Figura 5, se puede observar el comportamiento de las variables 'GENERO' y 'prestacion_servicio' las cuales corresponden efectivamente a variables categóricas a tratar.

Figura 5. Comportamiento de variables categóricas dataset en crudo



En la variable 'GENERO' se puede ver como la distribución de registros iniciales no se encuentra desproporcionada, para ser más exactos, el género "M" masculino tiene un porcentaje de representatividad del 50,12% frente a un 49,88% correspondiente al género "F" femenino. Con respecto a la variable 'prestacion_servicio' se cuenta con los tipos de sectores educativos principales que ofertan en la ciudad de Medellín, en la cual se puede evidencia que la mayor proporción con un 79,75% de estudiantes se encuentran en el sector Oficial el cual es el caso y año de estudio del presente proyecto.

3.4.2 Variables numéricas

Una vez es cargado y leído el conjunto de datos, se realiza un recorrido por las columnas para obtener cuales de ellas corresponden a variables numéricas según su exploración inicial y sin ningún tipo de manipulación o transformación de tipología. Esto se puede ver en la Figura 6.

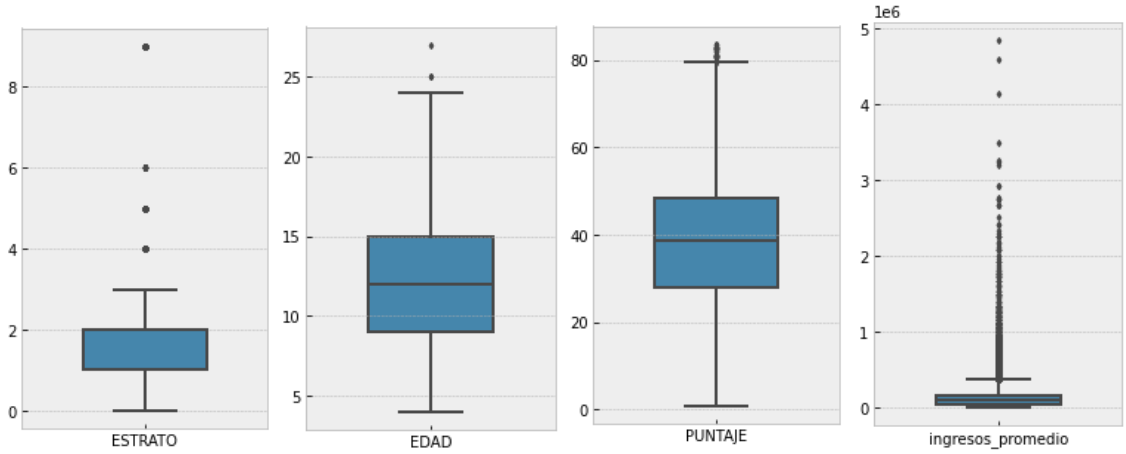
Figura 6. Variables numéricas dataset en crudo

```
cols_numericas = [i for i in df_simat.columns if i in df_simat._get_numeric_data()]
print (cols_numericas)
```

```
['index', 'MUN_CODIGO', 'CODIGO_DANE', 'CODIGO_DANE_SEDE', 'CONS_SEDE', 'sede', 'EXP_DEPTO', 'EXP_MUN', 'RES_DEPTO', 'RES_MUN',
'ESTRATO', 'SISBEN', 'NAC_DEPTO', 'NAC_MUN', 'POB_VICT_CONF', 'TIPO_DISCAPACIDAD', 'CAP_EXC', 'ETNIA', 'RES', 'TIPO_JORNADA',
'CARACTER', 'ESPECIALIDAD', 'GRADO', 'METODOLOGIA', 'SIT_ACAD_ANO_ANT', 'CON_ALUM_ANO_ANT', 'FUE_RECU', 'ZON_ALU', 'men_per_i
d', 'APOYO_ACAD_ESP', 'CTE_ID_SRPA', 'CODIGO_PAIS_ORIGEN', 'ESTADO_DEFINITIVO', 'COMUNA_EST', 'EDAD', 'PUNTAJE', 'extranjero',
'telefono', 'computador', 'embaraza', 'percibe', 'ingresos_promedio']
```

Teniendo en cuenta que se tiene conocimiento sobre cuales de las variables del dataset efectivamente son numéricas, se hace uso del diagrama de caja "box-plot" para representar gráficamente datos numéricos a través de sus cuartiles. Lo anterior, con el fin de identificar cuáles de ellas contienen valores atípicos u outliers. El resultado de este proceso se puede observar en la Figura 7.

Figura 7. Variables numéricas con atípicos



En la Figura 7, se puede evidenciar el comportamiento de las variables numéricas 'ESTRATO', 'EDAD', 'PUNTAJE' e 'ingresos_promedio', de las cuales, las dos últimas contienen la mayor cantidad de valores atípicos, es importante mencionar que estos valores alteran los resultados de las medidas de tendencia central, como el caso del promedio de los datos. Esto se puede constatar en la Figura 8 en donde se presenta la distribución de cada una de las variables mencionadas y como vienen inicialmente del conjunto de datos. En la sección 4.2.2 se realiza una descripción del procedimiento realizado y decisiones tomadas frente a la implicación de valores atípicos en la implementación de algoritmos de Machine Learning.

Figura 8. Distribución de variables numéricas en crudo

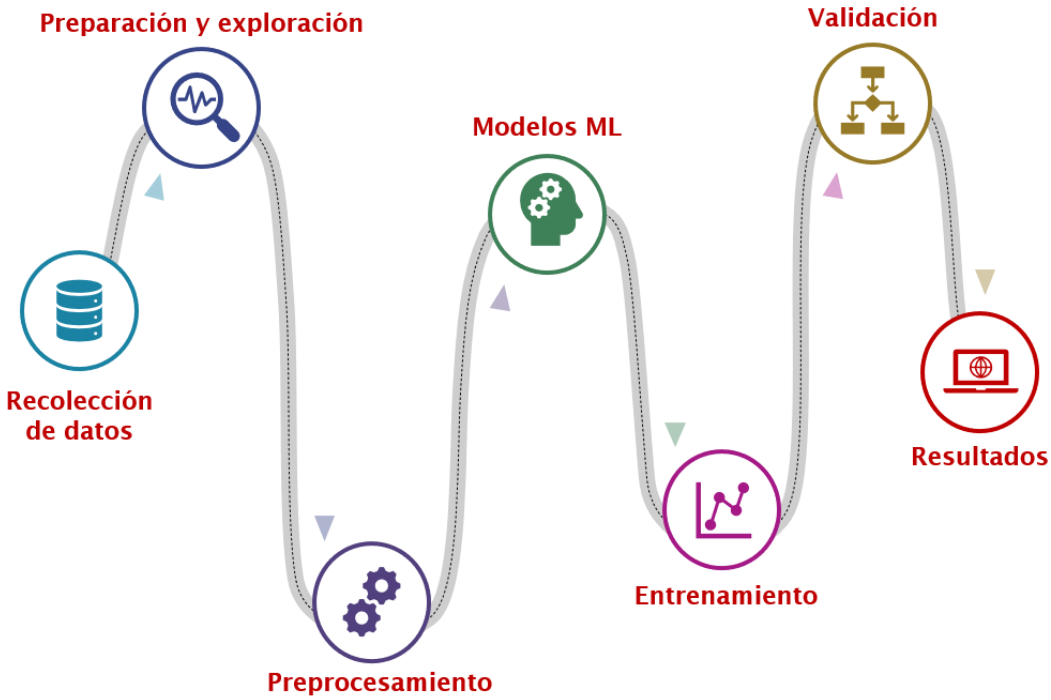
	ESTRATO	GRADO	EDAD	PUNTAJE	ingresos_promedio
count	185894.000000	185894.000000	185894.000000	185894.000000	1.858940e+05
mean	1.892116	5.508462	11.712578	38.153345	1.179022e+05
std	0.862202	3.203228	3.527554	14.227256	1.245006e+05
min	0.000000	0.000000	4.000000	0.810000	0.000000e+00
1%	1.000000	0.000000	5.000000	7.799300	0.000000e+00
10%	1.000000	1.000000	7.000000	18.660000	0.000000e+00
20%	1.000000	2.000000	8.000000	25.110000	1.583300e+04
30%	1.000000	4.000000	10.000000	30.190000	5.000000e+04
40%	2.000000	5.000000	11.000000	34.560000	7.500000e+04
50%	2.000000	6.000000	12.000000	38.670000	9.938000e+04
60%	2.000000	7.000000	13.000000	42.640000	1.240000e+05
70%	2.000000	7.000000	14.000000	46.550000	1.473750e+05
80%	2.000000	9.000000	15.000000	50.830000	1.842250e+05
90%	3.000000	10.000000	16.000000	56.320000	2.500000e+05
97.5%	3.000000	11.000000	18.000000	64.660000	4.056012e+05
max	9.000000	11.000000	27.000000	83.650000	4.844000e+06

4. PROCESO DE ANALÍTICA

4.1 Pipeline principal

En la Figura 9, se describe gráficamente el flujo de trabajo realizado durante el desarrollo del modelo predictivo propuesto, este va desde la recolección de los datos hasta los resultados obtenidos.

Figura 9. Flujo de trabajo modelo predictivo



Como se puede analizar en la figura anterior, para la ejecución iterativa del proyecto, se llevaron a cabo 7 etapas principales, en las cuales se inició con el proceso de recolección de los datos validando cuales serían los medios de acceso y licencias de uso necesarias para su manipulación que lograran responder a la problemática planteada, así como la preparación y exploración del conjunto de datos recolectado, consistente en el cruce previo de fuentes de información y el análisis inicial de su comportamiento, evidenciando aquí la necesidad de una etapa posterior de preprocesamiento que permitiera la estructuración de las variables y el tamaño de la población en estudio a partir de procesos tales como; extracción de características, tratamiento de valores atípicos, definición de la variable objetivo, codificación de variables categóricas y el balanceo de clases. Luego de lo anterior, se implementan los modelos de Machine Learning apropiados al problema planteado de clasificación binaria en aprendizaje supervisado, a partir de la separación del conjunto de datos en entrenamiento y validación que permitieran comprobar las métricas resultantes en cada algoritmo seleccionado. Finalmente, se realiza el análisis de resultados, con el fin de sustentar desde los datos, si la problemática pudo ser de alguna manera contestada positivamente a partir del entrenamiento, prueba y selección del mejor modelo.

4.2 Preprocesamiento

4.2.1 Extracción de características

4.2.1.1 Limpieza e imputación de datos.

A partir de la identificación y análisis de cada una de las variables recolectadas, así como la visualización de los datos nulos contenidos en estas, los cuales se pueden ver en la Figura 1, se toma la decisión de realizar un proceso de limpieza inicial e imputación de datos, los cuales se describen a continuación:

- En un primer momento se depuran aquellas variables con una cantidad de registros nulos superiores al 85% de los datos, estas correspondientes a: 'DPTO_EXP', 'MUN_EXP', 'VAL_DES_PERIODO1', 'VAL_DES_PERIODO2' y 'NUM_CONVENIO'. Además de las anteriores, se elimina la variable INTERNADO por cantidad de registros nulos y la poca relevancia identificada en el proceso.

Figura 10. Variables con gran porcentaje de datos nulos

```
17  DPTO_EXP              10890 non-null   float64
18  MUN_EXP               10890 non-null   float64
43  VAL_DES_PERIODO1      10 non-null      float64
44  VAL_DES_PERIODO2      0 non-null       float64
45  NUM_CONVENIO          23470 non-null   float64
```

- Se realiza la depuración de 24 variables ['index', 'MUN_CODIGO', 'CODIGO_DANE', 'CODIGO_DANE_SEDE', 'CONS_SEDE', 'GRUPO', 'sede', 'men_per_id', 'tipo_anexo_id', 'SISBEN', 'EXP_DEPTO', 'EXP_MUN', 'RES', 'ESPECIALIDAD', 'FUE_RECU', 'RES_DEPTO', 'RES_MUN', 'NAC_DEPTO', 'NAC_MUN', 'CAP_EXC', 'ETNIA', 'BEN_HER_NAC', 'BEN_VET_FP', 'CODIGO_PAIS_ORIGEN'] que no se requieren para el modelo, dado a que son redundantes (están altamente correlacionadas con otras variables del dataset que se mantienen, ver Figura 2) su nivel de importancia es muy bajo y/o contienen valores errados. Para sustentar la toma de esta decisión, se tomaron como referencia los resultados de las primeras iteraciones, la implementación del modelo RandomForestClassifier y la medición del nivel de importancias de todas las variables en el proceso de entrenamiento del mismo, con el fin de validar la hipótesis inicial sobre su poca relevancia en el proceso. Un ejemplo de lo mencionado, se puede observar en la Figura 4.

- Para el caso de las variables 'APOYO_ACAD_ESP' y 'CTE_ID_ SRPA', se decide realizar la imputación de los valores nulos presentes por el valor de "0" dado a que este representaría a los registros desconocidos.

Figura 11. Transformación de variables con valores nulos

```
columns_null_cero = ['APOYO_ACAD_ESP', 'CTE_ID_SRPA']
for i in columns_null_cero:
    df_simat[i] = df_simat[i].replace({np.nan: '0'})
```

Una vez realizadas las transformaciones descritas anteriormente, se obtiene un nuevo número de características de estudio para el modelo predictivos, estas pasan de ser a 61 a una reducción de 31.

Figura 12. Variables seleccionadas para el modelo

```
Luego de la depuración realizada, las variables que se mantienen son:
Index(['prestacion_servicio', 'ESTRATO', 'GENERO', 'POB_VICT_CONF',
       'PROVIENE_SECTOR_PRIV', 'PROVIENE_OTRO_MUN', 'TIPO_DISCAPACIDAD',
       'TIPO_JORNADA', 'CARACTER', 'GRADO', 'METODOLOGIA', 'SUBSIDIADO',
       'REPITENTE', 'NUEVO', 'SIT_ACAD_ANO_ANT', 'CON_ALUM_ANO_ANT', 'ZON_ALU',
       'CAB_FAMILIA', 'BEN_MAD_FLIA', 'APOYO_ACAD_ESP', 'CTE_ID_SRPA',
       'ESTADO_DEFINITIVO', 'COMUNA_EST', 'EDAD', 'PUNTAJE', 'extranjero',
       'telefono', 'computador', 'embaraza', 'percibe', 'ingresos_promedio'],
      dtype='object')
```

4.2.1.2 Extracción del sector de estudio.

Dado a que el estudio sobre la deserción estudiantil se realizará para el sector educativo Oficial incluido el sector privado subsidiado identificados actualmente como el tipo de prestación de servicio de Cobertura Contratada, es necesario eliminar aquellos registros correspondientes al sector educativo Privado y no subsidiado. Para ello, en la variable 'prestacion_servicio' que contiene la información del sector Oficial, Privado y Cobertura Contratada del establecimiento, se eliminarán los registros correspondientes a "Privado", lo anterior se puede observar en la Figura 13.

Figura 13. Filtro sector no oficial

```
sector_no_oficial = df_simat[df_simat['prestacion_servicio'] == 'Privado']
```

La cantidad de registros correspondientes al sector "Privado" corresponden a un total de 23.920, dado a que no son el objeto de estudio, se procede a eliminarlos del dataset. Luego del filtro aplicado, se procede a eliminar la variable prestacion_servicio debido a que el tratamiento requerido con ella, ya fue realizado.

4.2.1.3 Extracción de población de estudio.

La ciudad de Medellín desde la Secretaría de Educación con el objeto misional de garantizar la prestación del servicio educativo a través de políticas y estrategias de acceso, cobertura, permanencia y calidad, ofrece a la población la oportunidad de acceder a distintos niveles o grados, concernientes a la Educación Inicial (grados -1 y -2) Educación Regular (grados del 0 a 11), Ciclos Complementarios (grados 12, 13 y 99) y Ciclos Lectivos Especiales Integrados – CLEI (grados del 21 al 26 para adultos), estos se pueden observar en la Tabla 1 en la variable 'GRADO'. Por lo anterior, debido a que el objetivo del modelo predictivo va dirigido hacia los estudiantes de colegios Oficiales y de Educación Preescolar, Básica y Media, es necesario eliminar aquellos registros con grados escolares diferentes a los mencionados, el procedimiento realizado se puede ver en la Figura 14.

Figura 14. Filtro grados complementarios

```
df_simat['GRADO'].unique()
```

```
array([ 4., 10.,  2.,  3., 24.,  1., 25., 23., 26., 11., 22.,  7.,  8.,
        9.,  6., 99., 21., 13.,  5., 12.,  0., -1., -2.])
```

```
grados_complementarios = df_simat[(df_simat['GRADO'] >= 12.) | (df_simat['GRADO'] < 0.)]
df_simat.drop(grados_complementarios.index, inplace=True)
```

Así mismo, se realiza un filtro adicional para extraer y eliminar aquellos registros que según la variable 'METODOLOGIA' corresponden a Ciclos Complementarios denominados Caminar en Secundaria, es decir, los estudiantes de sexto a noveno grado desfasados respecto a su edad. Se toma esta decisión debido a que estos estudiantes cuentan con situaciones particulares que buscan precisamente la nivelación y continuidad de sus estudios. Con base a lo anterior, se eliminan los registros que sean diferentes a "1" (Educación Tradicional) y "2" (Escuela Nueva). El proceso realizado se puede ver en la Figura 15.

Figura 15. Filtro metodología ciclos complementarios

```
df_simat['METODOLOGIA'].unique()
```

```
array([ 1., 13., 29.,  2.])
```

```
metodologia_exc = df_simat[(df_simat['METODOLOGIA'] == 13.) | (df_simat['METODOLOGIA'] == 29.)]
```

```
df_simat.drop(metodologia_exc.index, inplace=True)
```

4.2.1.4 Creación de nuevas características.

Dentro de las variables incluidas en el conjunto de datos de estudio, se cuenta con la variable 'POB_VICT_CONF' en la cual se tiene la distinción de los estudiantes víctimas del conflicto, en esta los registros "99" corresponden a los estudiantes que no les aplica esta característica. Sin embargo, como es de esperarse, los registros "no" aplica contra los que "sí" aplica tienen una distribución desbalanceada y no uniforme, esto se puede observar en la Figura 16. Por ello, se toma la decisión de crear una nueva variable a partir de esta 'POB_VICT_CONF_IMP, que represente los registros de manera binaria, es decir, "1" para la población víctima del conflicto y "0" a los que no son víctimas. El procedimiento realizado se puede ver en la Figura 17. Además, se procede a eliminar la variable tratada 'POB_VICT_CONF' después de realizado el proceso.

Figura 16. Distribución población víctima

```
df_simat['POB_VICT_CONF'].value_counts()
```

```
99.0    176688
1.0       8981
3.0         93
4.0         67
2.0         49
10.0        15
12.0         1
Name: POB_VICT_CONF, dtype: int64
```

Figura 17. Creación variable binaria población víctima

```
df_simat['POB_VICT_CONF_IMP'] = (df_simat['POB_VICT_CONF'] != 99.0).astype(int)

df_simat['POB_VICT_CONF_IMP'].value_counts()

0    176688
1      9206
Name: POB_VICT_CONF_IMP, dtype: int64

df_simat.drop(columns=['POB_VICT_CONF'], inplace=True)
```

Al igual que en proceso anterior, se tiene la variable 'TIPO_DISCAPACIDAD' en la cual se tiene la distinción de los estudiantes con algún tipo de discapacidad, en esta los registros "99" corresponden a los estudiantes que no les aplica esta característica, teniendo una distribución desbalanceada y no uniforme con respecto a los registros restantes, esto se puede observar en la Figura 18. Por lo tanto, se toma la decisión de crear una nueva variable a partir de esta 'TIPO_DISCAPACIDAD_IMP', que represente los registros de manera binaria, donde "1" corresponderá a la población con algún tipo de discapacidad y "0" a los que no presentan ninguna discapacidad. Además, se procede a eliminar la variable tratada 'TIPO_DISCAPACIDAD' después de realizado el proceso.

Figura 18. Distribución población discapacidad

```
df_simat['TIPO_DISCAPACIDAD'].value_counts()

99.0    178875
18.0      2934
8.0       2344
17.0       366
10.0       350
15.0       263
7.0        224
19.0       129
13.0       124
12.0       116
3.0        101
11.0        39
4.0         23
14.0         6
Name: TIPO_DISCAPACIDAD, dtype: int64
```

Figura 19. Creación variable binaria población discapacidad

```
df_simat['TIPO_DISCAPACIDAD_IMP'] = (df_simat['TIPO_DISCAPACIDAD'] != 99.0).astype(int)

df_simat['TIPO_DISCAPACIDAD_IMP'].value_counts()

0    178875
1      7019
Name: TIPO_DISCAPACIDAD_IMP, dtype: int64

df_simat.drop(columns=['TIPO_DISCAPACIDAD'], inplace=True)
```

Adicionalmente, se crea una nueva variable a partir de la variable 'EDAD', para establecer aquellos estudiantes que se encuentran en extraedad, es decir, que su edad supera la edad ideal para el grado cursado. Para ello, inicialmente se procesa una variable 'EDAD_ideal' para establecer la edad correspondiente por cada grado, esto se puede observar en la Figura 20.

Figura 20. Variable de edad ideal por grado

```
df_simat['EDAD']=df_simat['EDAD'].astype(int)

var_edades = [(df_simat['GRADO'] == 0),
    (df_simat['GRADO'] == 1),
    (df_simat['GRADO'] == 2),
    (df_simat['GRADO'] == 3),
    (df_simat['GRADO'] == 4),
    (df_simat['GRADO'] == 5),
    (df_simat['GRADO'] == 6),
    (df_simat['GRADO'] == 7),
    (df_simat['GRADO'] == 8),
    (df_simat['GRADO'] == 9),
    (df_simat['GRADO'] == 10),
    (df_simat['GRADO'] == 11)]
var_edad_ideal = [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]

df_simat['EDAD_ideal'] = np.select(var_edades, var_edad_ideal, default='')
df_simat['EDAD_ideal'] = df_simat['EDAD_ideal'].astype(int)
```

La extraedad según lo reglamentado por el Ministerio de Educación Nacional (MEN) en sus orientaciones pedagógicas, corresponde al desfase entre la edad y el grado, y se presenta cuando el estudiante tiene 3 o más años por encima de la edad promedio estándar para cursar el determinando grado. Por ello, se crea la variable binaria 'EXTRAEDAD' que evalúa la edad ideal por cada grado frente a la edad real del estudiante que lo cursa, con el fin de establecer si este se encuentra en extraedad "1" o no "0". El proceso realizado se puede ver en la Figura 21.

Figura 21. Variable extraedad

```
df_simat['EXTRAEDAD'] = np.where(((df_simat['EDAD'] - df_simat['EDAD_ideal']) >= 3), 1, 0)
df_simat['EXTRAEDAD'].value_counts()
```

```
0    166404
1     19490
Name: EXTRAEDAD, dtype: int64
```

```
df_simat.drop(['EDAD_ideal'], axis=1, inplace=True)
```

4.2.1.5 Transformación del tipo de variable.

Se realiza la transformación de las tipologías de las variables presentes en el dataset procesado, con el fin de que funcionen de manera correcta dentro del modelo, debido a que en la importación inicial de la base de datos, la mayoría de las variables fueron tomadas como numéricas por ser números (ver Figura 1), sin embargo, estas corresponden a valores categóricos. El procedimiento realizado se puede observar en la Figura 22.

Figura 22. Transformación variables categóricas

```
columns_str = ['GENERO', 'POB_VICT_CONF_IMP','TIPO_DISCAPACIDAD_IMP', 'TIPO_JORNADA',
               'PROVIENE_SECTOR_PRIV', 'CARACTER', 'SUBSIDIADO', 'PROVIENE_OTRO_MUN',
               'REPITENTE', 'NUEVO', 'SIT_ACAD_ANO_ANT', 'CON_ALUM_ANO_ANT',
               'ZON_ALU', 'CAB_FAMILIA', 'BEN_MAD_FLIA','APOYO_ACAD_ESP', 'EXTRAEDAD',
               'CTE_ID_SRPA', 'COMUNA_EST', 'extranjero', 'telefono', 'computador',
               'embaraza', 'percibe']
for i in columns_str:
    df_simat[i] = df_simat[i].astype(str)
```

Así mismo, se realiza la transformación de las variables numéricas y flotantes 'ESTRATO' y 'GRADO, a variables numéricas enteras.

4.2.2 Tratamiento de valores atípicos

En la sección 3.4.1 se analizaron e identificaron las variables numéricas con valores atípicos correspondientes a 'ESTRATO', 'PUNTAJE', 'EDAD' e 'ingresos_promedio', a partir de esto y con el propósito de mejorar el desempeño del modelo de clasificación supervisada el cual se ve afectado por la presencia de outliers (valores distantes del resto de los datos), se toma la decisión de realizar una imputación a los valores atípicos de las variables 'ingresos_promedio', 'PUNTAJE' y 'EDAD', asignándole un valor por defecto. Este valor se toma basado en la construcción del diagrama de caja "boxplot", en donde determina que un dato se considera outlier o atípico en el límite superior si se cumple que:

$$Lim_{superior} = Cuartil_3 + 1.5 * IQR$$

Siendo el Rango Intercuartílico (IQR) la diferencia entre el tercer y el primer cuartil de una distribución. Para la variable puntual de 'ingresos_promedio', se le imputa el doble al rango intercuartílico (3 * IQR), para no extraer información que podría ser importante analizar por el apilamiento de estos datos. El procedimiento realizado se puede observar en la Figura 23.

Figura 23. Tratamiento de outliers

```
#Tratamiento variable 'ingresos_promedio'
perc1 = np.percentile(df_simat['ingresos_promedio'], 75)  + 3*(np.percentile(df_simat['ingresos_promedio'], 75)
                                                    - np.percentile(df_simat['ingresos_promedio'], 25))
df_simat.loc[df_simat['ingresos_promedio'] > perc1, 'ingresos_promedio'] = perc1

#Tratamiento variable 'EDAD'
perc2 = np.percentile(df_simat['EDAD'], 75)  + 1.5*(np.percentile(df_simat['EDAD'], 75) - np.percentile(df_simat['EDAD'], 25))
df_simat.loc[df_simat['EDAD'] > perc2, 'EDAD'] = perc2

#Tratamiento variable 'PUNTAJE'
perc3 = np.percentile(df_simat['PUNTAJE'], 75)  + 1.5*(np.percentile(df_simat['PUNTAJE'], 75)
                                                - np.percentile(df_simat['PUNTAJE'], 25))
df_simat.loc[df_simat['PUNTAJE'] > perc3, 'PUNTAJE'] = perc3
```

4.2.3 Definición de la variable objetivo

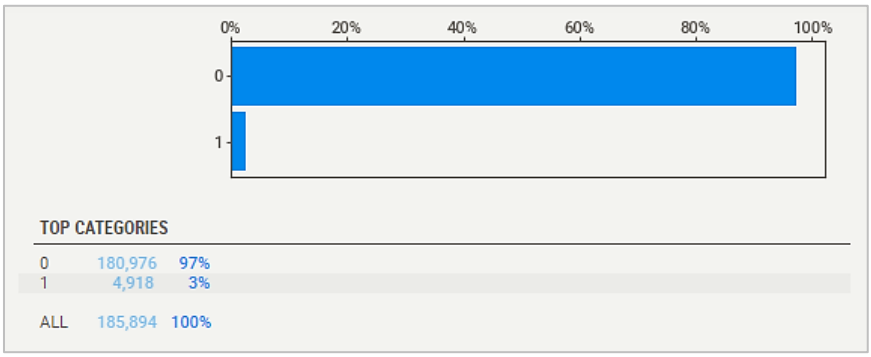
La variable a predecir u objetivo en la cual se tiene como propósito entrenar los modelos supervidos para clasificar a aquellos posibles estudiantes desertores y no desertores, se encuentra implícita en la variable 'ESTADO_DEFINITIVO' del dataset df_simat en la cual, los registros iguales a "1" corresponden a los no desertores (estudiantes que terminaron el año matriculados) y los registros iguales a "2" corresponden a los desertores al final del año en estudio (estudiantes que se retiraron del servicio educativo durante el año académico). Por lo anterior, se crea una nueva variable 'DESERTOR' a partir de la variable 'ESTADO_DEFINITIVO'. El procedimiento realizado se puede observar en la Figura 24.

Figura 24. Variable objeto

```
df_simat['DESERTOR'] = (df_simat['ESTADO_DEFINITIVO']==2).astype(int)

df_simat.drop(columns=['ESTADO_DEFINITIVO'], inplace=True)
```

Posteriormente al proceso realizado, se realiza la validación de la distribución de la variable 'DESERTOR', esto se puede ver en la Figura 25.

Figura 25. Distribución variable objeto

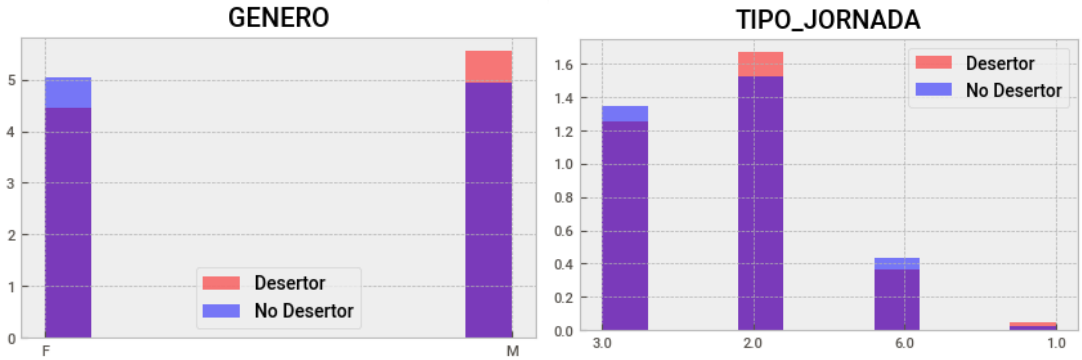


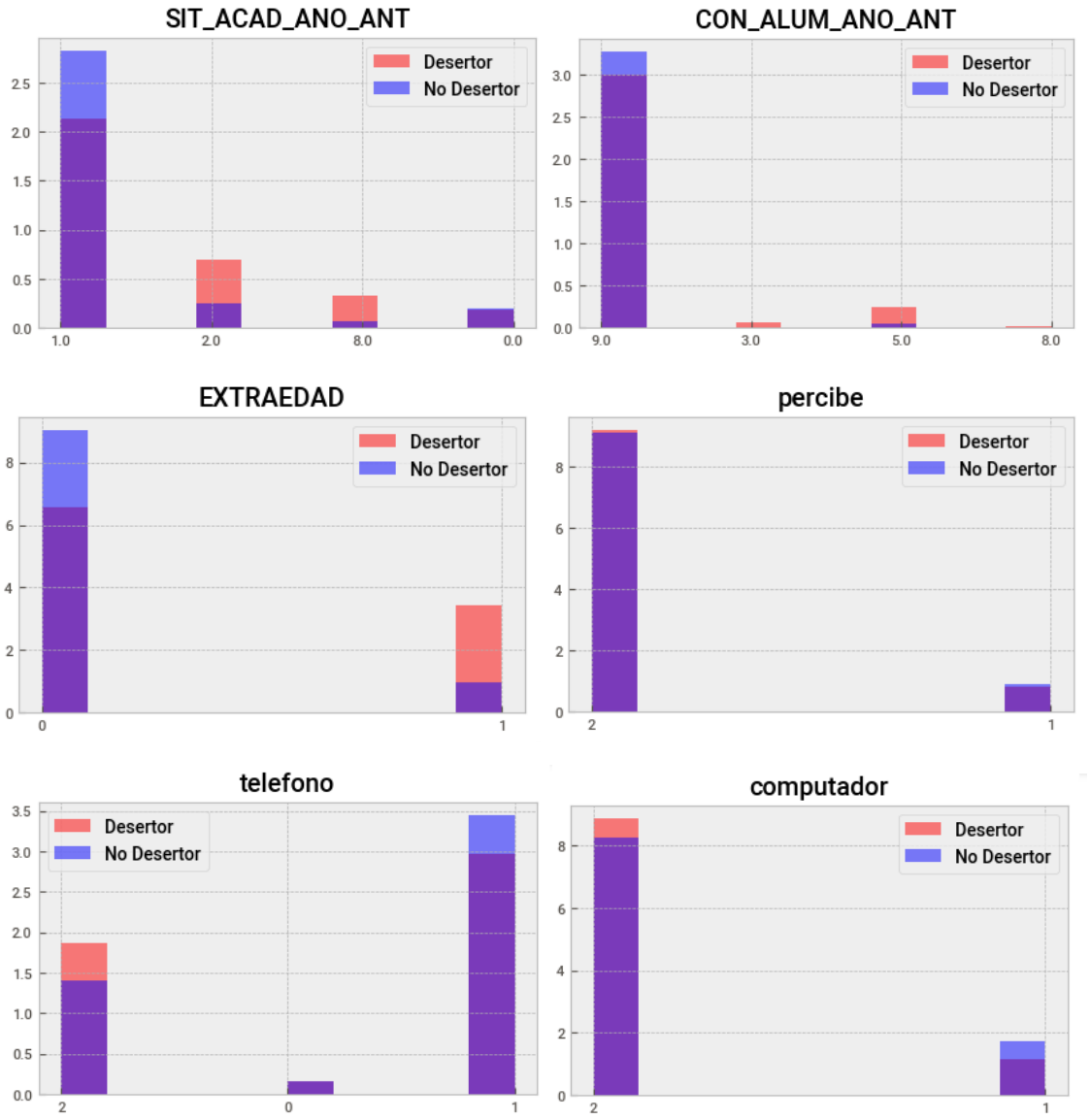
Como se puede observar, la variable a predecir tiene un alto desbalanceo en sus clases; 97% a 3%, situación que es esperada debido a que de lo contrario el sistema educativo estaría enfrentando un déficit en la implementación de políticas públicas y estrategias para velar por el derecho y acceso a la educación. De hecho, se espera que este desbalanceo aumente con el pasar de los años, es decir, que se logre ir disminuyendo el porcentaje de estudiantes que abandonan los colegios en Medellín.

4.2.3.1 Visualización del comportamiento de las variables.

Una vez es realizado el proceso de definición de la variable objetivo y los tratamientos descritos anteriormente, se procede a visualizar el comportamiento de las variables categóricas y numéricas, con el fin de poder validar si existe algún tipo de patrón en los resultados, esto se puede observar en las Figuras 26 y 27.

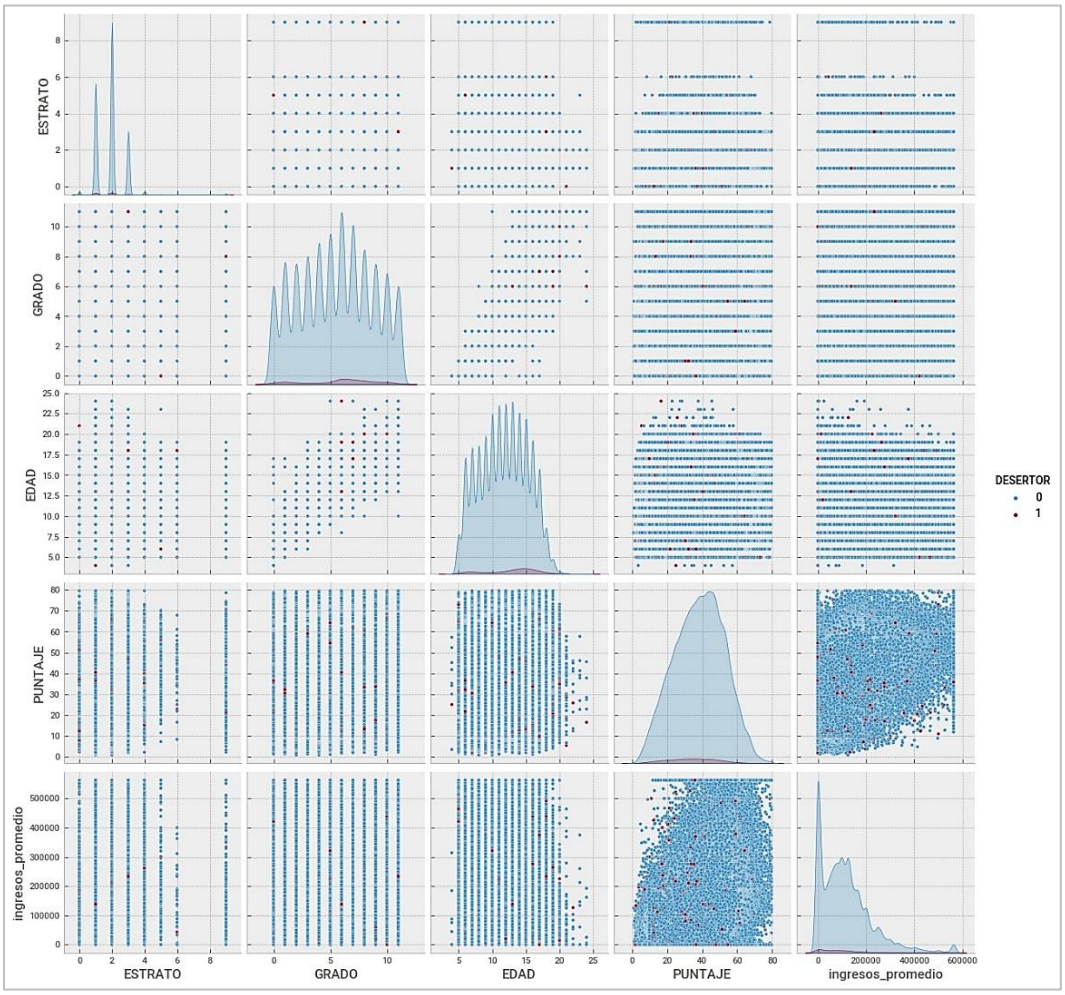
Figura 26. Comportamientos variables categóricas





Una vez analizado el comportamiento de algunas variables categóricas del dataset luego de la definición de la variable objetivo 'DESERTOR' y representando los valores de las clases (Desertor y No Desertor) a una misma escala con el fin de optimizar la visualización, se puede evidenciar un patrón en la variable puntual de 'GENERO', en donde hay un mayor porcentaje de desertores masculinos "M".

Figura 27. Comportamientos variables numéricas



Sin embargo, basado en las imágenes presentadas, se puede analizar como no existe una relación evidente o directa entre los datos según la variable objetivo a nivel general, es por ello, que se sustenta la necesidad de implementar algoritmos de Machine Learning para la predicción de la deserción estudiantil a partir de patrones causales y niveles de importancia en los datos recolectados.

Un ejemplo en particular se encuentra en la variable numérica 'EDAD' respecto a la variable 'GRADO', en las cuales se evidencia la mayor correlación de toda la gráfica con un coeficiente de 0,96, seguido de las variables 'PUNTAJE' frente a la variable 'ingresos_promedio', con un coeficiente de correlación de 0,36 (ver Figura 28), siendo situaciones esperadas frente al contexto de estudio, dado a que la edad de un estudiante condiciona el grado académico en el que se encuentre a menos que esté en extraedad y en el caso del puntaje e ingresos promedios de la encuesta del Sisbén, se espera que exista una relación lineal entre ellas. No obstante, dichos coeficientes no son representativos frente a la variable objetivo 'DESERTOR' y la dispersión presente en los datos, para afirmar que existe algún causal o patrón entre las mismas.

4.2.3.2 Análisis de correlación de los datos.

Luego del preprocesamiento realizado y descrito anteriormente, se genera la correlación existe en los datos numéricos, en los cuales se puede identificar la baja correlación lineal que tienen con respecto a la

variable objetivo. Lo anterior implicaría una mayor intervención del algoritmo para encontrar los patrones en los datos suministrados.

Figura 28. Correlación de variables numéricas

	ESTRATO	GRADO	EDAD	PUNTAJE	ingresos_promedio	DESERTOR
ESTRATO	1.000000	0.009028	-0.014413	0.233543	0.131355	-0.028319
GRADO	0.009028	1.000000	0.961004	0.129205	0.011742	0.014173
EDAD	-0.014413	0.961004	1.000000	0.086182	-0.013697	0.051215
PUNTAJE	0.233543	0.129205	0.086182	1.000000	0.361148	-0.044992
ingresos_promedio	0.131355	0.011742	-0.013697	0.361148	1.000000	-0.029565
DESERTOR	-0.028319	0.014173	0.051215	-0.044992	-0.029565	1.000000

4.2.4 Codificación de variables categóricas

En esta etapa inicialmente se realiza la división de las características (X) y la variable objetivo (y): 'DESERTOR', para luego transformar las variables categóricas y de tipo texto (str u object) de las características de X. Lo anterior, se realiza con el fin de que estas puedan ser codificadas y tratadas, puestos que la mayoría de los algoritmos de Machine Learning solo puede leer valores numéricos debido a que internamente hacen uso de distintas operaciones matemáticas. Es por ello, que se implementa el método de la librería de Pandas llamado get_dummies() el cual convierte los datos categóricos en variables indicadoras o ficticias.

Figura 29. Variables categóricas a dummies

```
X = df_simat.drop(columns=['DESERTOR'])
y = df_simat["DESERTOR"]
print (X.shape, y.shape)
```

```
(185894, 29) (185894,)
```

```
columns_str_dummies = ['GENERO', 'POB_VICT_CONF_IMP','TIPO_DISCAPACIDAD_IMP', 'TIPO_JORNADA',
                       'PROVIENE_SECTOR_PRIV', 'CARACTER', 'SUBSIDIADO', 'PROVIENE_OTRO_MUN',
                       'REPITENTE', 'NUEVO', 'SIT_ACAD_ANO_ANT', 'CON_ALUM_ANO_ANT',
                       'ZON_ALU', 'CAB_FAMILIA', 'BEN_MAD_FLIA','APOYO_ACAD_ESP',
                       'CTE_ID_SRPA', 'COMUNA_EST', 'extranjero', 'telefono', 'computador',
                       'embaraza', 'percibe', 'EXTRAEDAD']
X = pd.get_dummies(X, columns = columns_str_dummies)
```

Figura 30. Columnas a partir de get_dummies()

```
X.columns
```

```
Index(['ESTRATO', 'GRADO', 'EDAD', 'PUNTAJE', 'ingresos_promedio', 'GENERO_F',
       'GENERO_M', 'POB_VICT_CONF_IMP_0', 'POB_VICT_CONF_IMP_1',
       'TIPO_DISCAPACIDAD_IMP_0', 'TIPO_DISCAPACIDAD_IMP_1',
       'TIPO_JORNADA_1.0', 'TIPO_JORNADA_2.0', 'TIPO_JORNADA_3.0',
       'TIPO_JORNADA_6.0', 'PROVIENE_SECTOR_PRIV_N', 'PROVIENE_SECTOR_PRIV_S',
       'CARACTER_0.0', 'CARACTER_1.0', 'CARACTER_2.0', 'SUBSIDIADO_N',
       'SUBSIDIADO_S', 'PROVIENE_OTRO_MUN_N', 'PROVIENE_OTRO_MUN_S',
       'REPITENTE_N', 'REPITENTE_S', 'NUEVO_N', 'NUEVO_S',
       'SIT_ACAD_ANO_ANT_0.0', 'SIT_ACAD_ANO_ANT_1.0', 'SIT_ACAD_ANO_ANT_2.0',
       'SIT_ACAD_ANO_ANT_8.0', 'CON_ALUM_ANO_ANT_3.0', 'CON_ALUM_ANO_ANT_5.0',
       'CON_ALUM_ANO_ANT_8.0', 'CON_ALUM_ANO_ANT_9.0', 'ZON_ALU_1.0',
       'ZON_ALU_2.0', 'CAB_FAMILIA_N', 'CAB_FAMILIA_S', 'BEN_MAD_FLIA_N',
       'BEN_MAD_FLIA_S', 'APOYO_ACAD_ESP_0', 'APOYO_ACAD_ESP_1.0',
       'APOYO_ACAD_ESP_2.0', 'APOYO_ACAD_ESP_3.0', 'APOYO_ACAD_ESP_4.0',
       'APOYO_ACAD_ESP_5.0', 'CTE_ID_SRPA_0', 'CTE_ID_SRPA_1.0',
       'CTE_ID_SRPA_2.0', 'CTE_ID_SRPA_3.0', 'COMUNA_EST_1.0',
       'COMUNA_EST_10.0', 'COMUNA_EST_11.0', 'COMUNA_EST_12.0',
       'COMUNA_EST_13.0', 'COMUNA_EST_14.0', 'COMUNA_EST_15.0',
       'COMUNA_EST_16.0', 'COMUNA_EST_2.0', 'COMUNA_EST_3.0', 'COMUNA_EST_4.0',
       'COMUNA_EST_5.0', 'COMUNA_EST_50.0', 'COMUNA_EST_6.0',
       'COMUNA_EST_60.0', 'COMUNA_EST_7.0', 'COMUNA_EST_70.0',
       'COMUNA_EST_8.0', 'COMUNA_EST_80.0', 'COMUNA_EST_9.0',
       'COMUNA_EST_90.0', 'extranjero_1', 'extranjero_2', 'telefono_0',
       'telefono_1', 'telefono_2', 'computador_1', 'computador_2',
       'embaraza_1', 'embaraza_2', 'percibe_1', 'percibe_2', 'EXTRAEDAD_0',
       'EXTRAEDAD_1'],
      dtype='object')
```

Como se puede observar en la Figura 30, mediante el método get_dummies() es posible realizar la codificación de una variable categórica para convertirla en varias columnas con el identificador del registro al que corresponde, obteniendo 1 o 0 en el caso de que se cumpla la condición en el registro. Un ejemplo de esto se observa en la Figura 31, en la cual se muestra la codificación de la variable 'SIT_ACAD_ANO_ANT' que contiene 4 posibles valores en sus registros "0.0", "1.0", "2.0" y "8.0", quedando ahora a nivel de 4 columnas características. Es importante comentar que para que este método no se vea afectado a futuro, se deben recibir los mismos registros con los que se implementa.

Figura 31. Columnas correspondientes a SIT_ACAD_ANO_ANT

	SIT_ACAD_ANO_ANT_0.0	SIT_ACAD_ANO_ANT_1.0	SIT_ACAD_ANO_ANT_2.0	SIT_ACAD_ANO_ANT_8.0
222	0	1	0	0
1982	0	1	0	0
5040	0	1	0	0
6170	0	1	0	0
8236	0	1	0	0
...	...	...	...	...
233235	0	1	0	0
233236	0	1	0	0
233239	0	1	0	0
233240	0	1	0	0
233242	0	1	0	0

185894 rows × 4 columns

4.2.5 Balanceo de clases

En la sección 4.2.3 específicamente en la Figura 25, se identificó el desbalanceo de las clases en la variable objetivo, lo cual como se analizó en la primera iteración del modelo afecta el desempeño de los algoritmos de clasificación dado a que se perjudica a la clase minoritaria (1) y que en este estudio corresponde al posible desertor (el de mayor importancia para predecir). A partir de esto, se realiza la generación de datos sintéticos para balancearlas, añadiendo muestras a la clase minoritaria perteneciente al dataset de entrenamiento, mediante la técnica de sobremuestreo SMOTE (Synthetic Minority Oversampling Technique) de la librería imbalanced-learn. El funcionamiento de esta técnica consiste en la selección una muestra de la clase minoritaria aleatoriamente y sus vecinos más cercanos, sintetizando nuevos datos minoritarios entre los reales existentes. El procedimiento realizado se puede observar en la Figura 32.

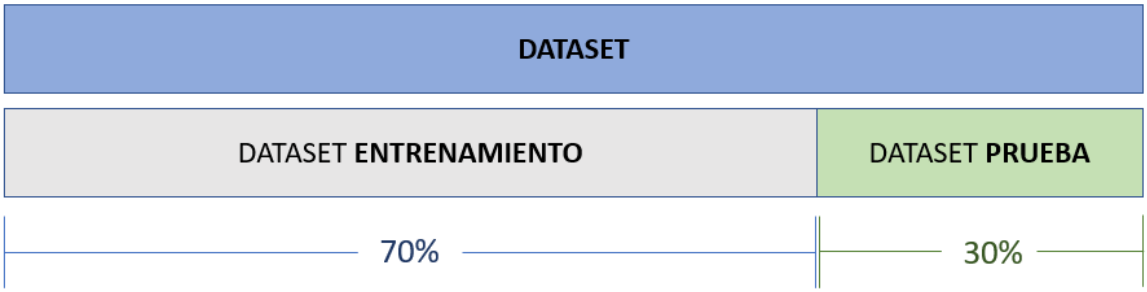
Figura 32. Balanceo de clase minoritaria con SMOTE

```
sm = SMOTE(random_state=20)
X_res, y_res = sm.fit_resample(X,y)

print('Forma del dataset sobremuestreado: %s' % Counter(y_res))

Forma del dataset sobremuestreado: Counter({0: 180976, 1: 180976})
```

4.3 Dataset de entrenamiento y prueba

A partir del dataset preprocesado y balanceado, y la necesidad de evaluar el rendimiento del modelo predictivo desarrollado así como su validación, se realiza la separación de los datos, haciendo uso de la función sklearn.model_selection.train_test_split de la librería Scikit-learn (Sklearn), la cual permite dividir un dataset en dos bloques; dataset de entrenamiento “train” (se usa para entrenar o ajustar el modelo) y dataset de prueba “test”. Lo anterior se aplica con el fin de hacer una evaluación final imparcial que no se encuentre sesgada y el modelo sea evaluado con datos nuevos que no fueron vistos a la hora del entrenamiento. La proporción en porcentaje de la separación realizada se puede observar en la Figura 33.

Figura 33. Separación de dataset de entrenamiento y prueba



- Forma del dataset de entrenamiento (registros, variables); X_train: (253.366, 86), y_train: (253.366,).
- Forma del dataset de prueba (registros, variables); X_test: (108.586, 86), y_test: (108.586,).

4.4 Modelos

Con el fin de dar respuesta a la problemática de clasificación supervisada planteada para el desarrollo del modelo predictivo de deserción escolar, se consideran e implementan los siguientes algoritmos de Machine Learning:

4.4.1 Random Forest

Ranfom Forest (Bosque aleatorios) es un algoritmo de aprendizaje supervisado implementado en problemas de clasificación y regresión, el cual está conformado por un conjunto de árboles de decisión individuales los cuales son entrenados cada uno con una muestra aleatoria perteneciente a los datos originales de entrenamiento, es por ello que, la predicción de una nueva observación resulta de agregar las predicciones de todos los árboles individuales presentes en el modelo.

La librería Scikit-Learn (Sklearn) (disponible en https://scikit-learn.org/) cuenta con el clasificador de bosque aleatorio "RandomForestClassifier" el cual ajusta varios clasificadores de árboles de decisión en varias submuestras del dataset y usa el promedio para mejorar la predicción y controlar el sobreajuste. Este metaestimador cuenta con parámetros que pueden ser variados para un mejor desempeño.

4.4.2 Red Neuronal con Autoencoder

Un Autoencoder es un modelo de red neuronal que busca aprender una representación comprimida de una entrada, el cual consta de dos partes: un codificador y un decodificador. El codificador aprende a interpretar la entrada y comprimirla a una representación interna definida por la capa de cuello de botella. El decodificador toma la salida del codificador (la capa de cuello de botella) e intenta recrear la entrada. Se trata de un método utilizado particularmente para aprendizaje no supervisado, sin embargo, este se entrena mediante métodos de aprendizaje supervisados, denominados auto-supervisados.

La plataforma TensorFlow (disponible en https://www.tensorflow.org/) cuenta con las capas y módulos necesarios para la implementación de redes neuronales.

4.4.3 Stacking

Stacking (Generalización apilada) es un algoritmo de Machine Learning de estimadores con clasificador final, el cual utiliza un metaaprendizaje para poder aprender a combinar en mejor manera las predicciones de dos o más algoritmos básicos. Tiene como beneficio que gracias al apilamiento puede aprovechar las bondades de una variedad de modelos con buen desempeño ya sea en tareas de clasificación o regresión.

La librería Scikit-Learn (Sklearn) (disponible en https://scikit-learn.org/) cuenta con el generador apilado "StackingClassifier" el cual consiste en apilar la salida del estimador individual y usar un clasificador para calcular la predicción final. Este modelo cuenta con parámetros que pueden ser variados para un mejor desempeño.

4.4.4 Bagging

Bagging (ensacado) un algoritmo conjunto (ensemble) de Machine Learning que combina las predicciones de distintos clasificadores. Aquí se ajustan múltiples modelos, cada uno con un subconjunto distinto de los datos de entrenamiento. Para realizar la predicción, todos los modelos participan aportando su predicción. Como valor final, se toma la media de todas las predicciones (variables continuas) o la clase más frecuente (variables categóricas).

La librería Scikit-Learn (Sklearn) (disponible en https://scikit-learn.org/) cuenta con el modelo "BaggingClassifier" el cual ajusta los clasificadores base mediante subconjuntos aleatorios del conjunto de datos original de entrenamiento y luego agrega sus predicciones individuales (por votación o promediando) para formar una predicción final. Este metaestimador es fácil de implementar debido a que cuenta con pocos hiperparámetros que pueden ser configurados para un mejor desempeño.

4.4.5 XGBoost

XGBoost son las siglas de eXtreme Gradient Boosting (Aumento de Gradiente Extremo) y es un algoritmo de aprendizaje supervisado que se puede utilizar para tareas de clasificación o regresión según sea el problema. Este es una implementación de código abierto del algoritmo de árboles impulsados por gradientes. El aumento de gradiente es llamado así porque minimiza la función de pérdida utilizando un algoritmo de descenso de gradiente. Su implementación fue diseñada para un rendimiento y velocidad óptimos.

Para el uso de este algoritmo es necesario instalar xgboost en la computadora o el entorno de ejecución de preferencia (disponible en https://xgboost.readthedocs.io/) y de esta manera hacer uso del modelo "XGBClassifier". Este algoritmo cuenta con hiperparámetros que son de importante elección para ajustar el rendimiento y desempeño de las métricas.

4.5 Métricas de desempeño

Con el propósito de evaluar el rendimiento del modelo predictivo de deserción estudiantil para los algoritmos entrenados de clasificación binaria en el aprendizaje supervisado, es necesario evaluarlos en función de distintas métricas de desempeño. En este caso de estudio, la métrica principal implementada de Machine Learning es la matriz de confusión "Confusion Matrix" de la librería de Sklearn, la cual permite evaluar la precisión de la predicción en el proceso de clasificación binaria (1 - desertores y 0 - no desertores) usando una tabla cruzada, arrojando información relevante como verdaderos positivos (True positive - TP), falsos positivos (False positive - FP), verdaderos negativos (True negative - TN) y falsos negativos (False negative - FN). Esta es usada debido a la gran importancia de visualizar el rendimiento en la predicción de los verdaderos positivos y falsos negativos.

Tabla 3. Matriz de confusión

		PREDICCIÓN	
		Negativo (0)	**Positivo (1)**
REAL	**Negativo (0)**	Verdaderos negativos - TN	Falsos positivos - FP
	Positivo (1)	Falsos negativos - FN	Verdaderos positivos – TP

Adicionalmente, se realiza el cálculo del puntaje de clasificación de precisión "Accuracy" como la proporción de predicciones correctas sobre el número total de las predicciones realizadas por el modelo

con respecto a la matriz de confusión. La sensibilidad “Recall” o tasa de verdaderos positivos, como la proporción de aquellos predichos como positivos (en este caso a los posibles desertores) entre los verdaderos positivos con respecto a la matriz de confusión. La puntuación promedia ponderada de la precisión y recuperación “F1 Score” la cual corresponde al cálculo de la media armónica de la precisión y la sensibilidad. Y, el AUC siendo el área bajo la curva ROC “Receiver Operating Characteristic”, correspondiente a una gráfica que enfrenta la tasa de falsos positivos (1 - especificidad) como la coordenada en el eje X, con la tasa de verdaderos positivos (sensibilidad) como la coordenada en el eje Y, en varios umbrales del modelo definidos entre 0 y 1, por lo tanto, el modelo con AUC más cercano a 1 se considera un mejor modelo. A continuación, se muestran las fórmulas correspondientes a las métricas seleccionadas para la evaluación del desempeño de los algoritmos de clasificación:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

$$Precision = \frac{TP}{TP + FP}$$

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

$$F1\ Score = \ 2 * \frac{Precision\ x\ Recall}{Precision + Recall}$$

Todas las métricas descritas anteriormente, fueron implementadas para complementar la validación a partir de datos nuevos, en cada uno de los modelos utilizados de Machine Learning.

5. METODOLOGÍA

5.1 Baseline

Para el proceso de codificación de la primera iteración del modelo predictivo, se elaboró el notebook titulado “1ra_iteracion_proyecto_desertores.ipynb”, el cual contiene la documentación del proceso realizado para obtener un primer resultado del estudio de las variables del conjunto de datos inicial. Este notebook está constituido por las siguientes partes:

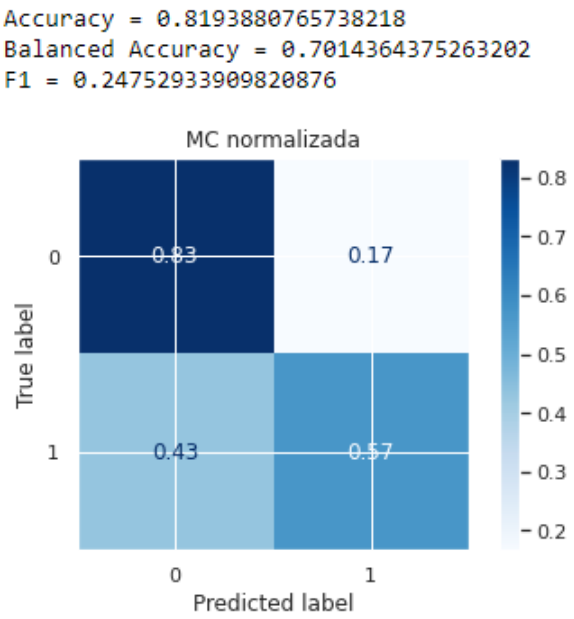
- Importación de librerías.
- Carga de la base de datos.
- Identificación de variables.
- Manipulación de las variables.
- Definición de variable a predecir de deserción.
- Transformación del tipo de variable.
- Visualización del comportamiento de las variables.
- Métrica de evaluación del modelo.
- Predicciones.
- Selección del mejor modelo en la primera iteración.
- Siguientes iteraciones del modelo predictivo.

Como resultado de la selección del mejor modelo para la primera iteración, se obtuvo el modelo metaestimador de árboles aleatorios “Random Forest” de clasificación mediante la búsqueda de hiperparámetros, el cual comparado con los resultados obtenidos con otros modelos tales como; regresión logística “Logistic Regression” y árbol de decisión “Decision Tree” de la librería de Sklearn, tuvo un mejor puntaje en la evaluación de desempeño.

La métrica principal implementada para medir el desempeño del modelo, fue la matriz de confusión “Confusion Matrix de Sklearn” la cual permitió evaluar la precisión de la predicción en el proceso de clasificación binaria (desertores y no desertores), arrojando información relevante como verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos. Sin embargo, se acompañó con el cálculo del puntaje de clasificación de precisión “Accuracy”, la precisión equilibrada “Balanced Accuracy” y la puntuación promedia ponderada de la precisión y recuperación “F1 Score”, para ver su influencia en los resultados con datos desbalanceados, en donde solo el 5,26% de los datos originales estudiados correspondían a desertores (1).

De la primera iteración realizada con el conjunto de datos "matricula_simat_prel_2019", se pudo concluir que las predicciones obtenidas con el mejor modelo “Random Forest Classifier” tuvieron un alto porcentaje de falsos negativos conllevando a que no fuese un buen modelo predictivo. Las métricas obtenidas con este modelo, se pueden observar en la Figura 34.

Figura 34. Métricas primera iteración



Con base a lo anterior, se tuvieron las consideraciones mencionadas a continuación para la realización de las siguientes iteraciones:

- Incluir progresivamente nuevas variables cruzadas con otras fuentes de datos con factores importantes tales como; aspectos socioeconómicos, sociodemográficos, subjetivos y familiares en el proceso de permanencia escolar en el municipio de Medellín, que podrían tener mayor incidencia en la deserción.
- Realizar un proceso de sobremuestreo de datos para solucionar problemas de desbalanceo con la variable a predecir.
- Implementar nuevas técnicas de clasificación con búsqueda de hiperparámetros.

5.2 Iteraciones y evolución

En el desarrollo del modelo predictivo se llevaron a cabo una serie de iteraciones que permitieran ir progresivamente mejorando el desempeño de los modelos de clasificación seleccionados para la tarea objetivo, las tareas realizadas en cada uno de ellas se describen a continuación de manera sintetizada:

- **Iteración 1 (Baseline):** La descripción de esta iteración se puede ver en la sección 5.1 de manera más detallada. Aquí, se utilizó la base de datos de matrícula de todos los estudiantes de los colegios de Medellín, con todas sus variables, sin realizar extracción de características, con un primer proceso de limpieza e imputación de datos y con las clases de la variable objetivo desbalanceadas. Se implementaron los modelos de clasificación tales como; LogisticRegression (Regresión Logística), DecisionTreeClassifier (Árbol de decisión de Clasificación) y RandomForestClassifier (Bosques Aleatorios de Clasificación). Una vez seleccionado el mejor modelo según sus métricas de desempeño, se realizó la generación gráfica de la importancia de

las variables del modelo, con el fin de que fuese un primer indicio para el tratamiento posterior de dichas variables, así como las acciones de mejora.

- **Iteración 2:** En esta iteración se realiza uno de los primeros procesos establecidos en la primera iteración, el cual consistía en agregar ocho nuevas variables al dataset con información socioeconómica, sociodemográfica y familiar de los estudiantes proveniente de la base de datos de Sisbén, sin embargo, se siguieron manejando las clases de la variable objetivo desbalanceadas para confirmar o anular la importancia de balancearlas debido al gran porcentaje de diferencia. Aquí, se implementó el mejor modelo obtenido en el baseline RandomForestClassifier con búsqueda de hiperparámetros.

- **Iteración 3:** En esta iteración se realiza el balanceo de las clases pertenecientes a la variable objetivo, mediante la técnica de sobremuestreo SMOTE, adicionalmente, se realiza un filtro de los grados en estudio correspondientes a los de Educación Regular (grado 0 a 11). Como resultado de este procedimiento, se obtiene una mejora sustancial en las métricas de desempeño.

- **Iteración 4:** En esta iteración se realiza la implementación del modelo de Red Neuronal con Autoencoder con las mismas variables utilizadas en la 3ra iteración.

- **Iteración 5:** En esta iteración se realiza la implementación del modelo ensemble StackingClassifier en donde se colocó a votación un modelo de LogisticRegression con un DecisionTreeClassifier, con las mismas variables utilizadas en la 3ra iteración e hiperparámetros por defecto.

- **Iteración 6:** En esta iteración se realiza un proceso de extracción de características el cual consistió en las siguientes tareas; limpieza e imputación de datos, extracción del sector de estudio en el cual se eliminó el sector correspondiente al Privado, extracción de población de estudio en donde se filtró el grado y la metodología de aprendizaje del estudiante, la creación de nuevas variables entre ellas unas variables binarias a partir de otras con registros que presentaban distribuciones heterogéneas, se realizó tratamiento a los valores atípicos (outliers) haciendo uno del rango intercuartílico (IQR) para su imputación. Adicionalmente se realizó la depuración de 9 variables del dataset, debido a que con iteraciones anteriores y el análisis de correlación, no le aportaban nada al modelo. Además de todo lo anterior, se implementa el modelo ensemble BaggingClassifier en el cual el modelo interno que mejor rendimiento dio fue el DecisionTreeClassifier.

- **Iteración 7:** En esta iteración se realiza la implementación del modelo ensemble XGBClassifier en el cual se entrenó únicamente con las variables numéricas del dataset.

- **Iteración 8:** Debido a los resultados obtenidos en la séptima iteración en la cual se entrenó un modelo únicamente con las variables numéricas y algunas categóricas como dummies, se decide realizar una nueva depuración de variables y evaluar el desempeño de los modelos ante esta disminución de dimensionalidad de las variables de entrada (antes de la categorización en dummies), así como la búsqueda de hiperparámetros en los modelos implementados.

5.3 Herramientas

Para la realización de este proyecto, se utilizaron las siguientes herramientas, lenguaje de programación y librerías:

- **Computador** con un procesador intel core i7 de 8th Generación con una memoria RAM de 16 GB + 4GB DRAM.
- **Lenguaje de programación PYTHON** (Versión 3.8.8)

Así mismo, dentro del proyecto y mediante Python, se utilizaron las siguientes librerías:

- **Pandas:** es una librería especializada en el manejo y análisis de estructuras de datos.
- **Numpy:** es una librería especializada en el cálculo numérico y el análisis de datos.
- **SciPy:** es una biblioteca libre y de código abierto para Python, la cual está compuesta por herramientas y algoritmos matemáticos.
- **Matplotlib:** es una librería especializada en la creación de gráficos en trazado 2D.
- **Seaborn:** es una librería basada en matplotlib que permite generar fácilmente elegantes gráficos.
- **Scikit-Learn:** es una librería de código abierto construida sobre SciPy, siendo actualmente la más utilizada para modelos de Machine Learning en Python.
- **TensorFlow:** es una librería especializada en el uso de algoritmos de redes neuronales.

- **IDEs**
  - **Google Colab:** extensión de Google para programar y ejecutar código en línea desde un navegador en lenguaje Python.
  - **Jupyter Notebook:** servidor para la ejecución de notebooks.

6. RESULTADOS

6.1 Métricas

Se pueden observar los resultados obtenidos con los datos de prueba en cada una de las iteraciones realizadas en el desarrollo del modelo predictivo.

Tabla 4. Métricas iteración 1 (Baseline)

MODELO	HIPER-PARÁMETROS	ACCURACY	RECALL	F1 SCORE	MATRIX CONFUSION
Logistic Regression	Default	0.95	0.03	0.05	**TN** = 1 FN = 0.97 **TP** = 0.03 FP = 0.0001
Decision Tree Classifier	max_depth=3	0.95	0.09	0.17	**TN** = 1 FN = 0.91 **TP** = 0.09 FP = 0
Random Forest Classifier	{max_depth=3, n_estimators=50, class_weight='balanced'}	0.82	0.57	0.25	**TN** = 0.83 FN = 0.43 **TP** = 0.57 FP = 0.17

Tabla 5. Métricas iteración 2

MODELO	HIPER-PARÁMETROS	ACCURACY	RECALL	F1 SCORE	MATRIX CONFUSION
Random Forest Classifier	{max_depth=4, n_estimators=50, class_weight='balanced'}	0.80	0.52	0.13	**TN** = 0.81 FN = 0.48 **TP** = 0.52 FP = 0.19

Tabla 6. Métricas iteración 3

MODELO	HIPER-PARÁMETROS	ACCURACY	RECALL	F1 SCORE	MATRIX CONFUSION
Random Forest Classifier	{max_depth=5, n_estimators=40, class_weight='balanced'}	0.87	0.84	0.87	**TN** = 0.91 FN = 0.16 **TP** = 0.84 FP = 0.09

Tabla 7. Métricas iteración 4

MODELO	HIPER-PARÁMETROS	ACCURACY	RECALL	F1 SCORE	MATRIX CONFUSION
Random Forest Classifier	{max_depth=5, n_estimators=40, class_weight='balanced'}	0.87	0.84	0.87	**TN** = 0.91 FN = 0.16 **TP** = 0.84 FP = 0.09
Neural Network with Autoencoder	{epochs=10, batch_size=16, verbose=2} Model = LogisticRegression()	0.88	0.81	0.88	**TN** = 0.96 FN = 0.19 **TP** = 0.81 FP = 0.04

Tabla 8. Métricas iteración 5

MODELO	HIPER-PARÁMETROS	ACCURACY	RECALL	F1 SCORE	MATRIX CONFUSION
Random Forest Classifier	{max_depth=5, n_estimators=40, class_weight='balanced'}	0.87	0.84	0.87	**TN** = 0.91 FN = 0.16 **TP** = 0.84 FP = 0.09
Neural Network with Autoencoder	{epochs=10, batch_size=16, verbose=2} Model = LogisticRegression()	0.88	0.81	0.88	**TN** = 0.96 FN = 0.19 **TP** = 0.81 FP = 0.04
Stracking Classifier	clf_a = LogisticRegression (random_state=0) clf_b = DecisionTreeClassifier (max_depth=3, random_state=0) estimators = [('DT1', clf_a), ('DT2', clf_b)]	0.83	0.72	0.81	**TN** = 0.94 FN = 0.28 **TP** = 0.72 FP = 0.06

Tabla 9. Métricas iteración 6

MODELO	HIPER-PARÁMETROS	ACCURACY	RECALL	F1 SCORE	MATRIX CONFUSION
Random Forest Classifier	{max_depth=5, n_estimators=50, class_weight='balanced'}	0.87	0.83	0.86	**TN** = 0.90 FN = 0.17 **TP** = 0.83 FP = 0.09

Neural Network with Autoencoder	{epochs=10, batch_size=16, verbose=2} Model = LogisticRegression()	0.90	0.83	0.89	**TN** = 0.97 FN = 0.17 **TP** = 0.83 FP = 0.03
Stracking Classifier	{estimators=[('DT1', LogisticRegression(random_state=0)), ('DT2', DecisionTreeClassifier(max_depth=3, random_state=0))], final_estimator=LogisticRegression(random_state=42)}	0.83	0.73	0.81	**TN** = 0.94 FN = 0.27 **TP** = 0.73 FP = 0.06
Bagging Classifier	{base_estimator= DecisionTreeClassifier(), bootstrap=False, max_features=4, random_state=22}	0.94	0.88	0.93	**TN** = 1 FN = 0.12 **TP** = 0.88 FP = 0.005

Tabla 10. Métricas iteración 7

MODELO	HIPER-PARÁMETROS	ACCURACY	RECALL	F1 SCORE	MATRIX CONFUSION
Random Forest Classifier	{max_depth=5, n_estimators=50, class_weight='balanced'}	0.87	0.83	0.86	**TN** = 0.90 FN = 0.17 **TP** = 0.83 FP = 0.09
Neural Network with Autoencoder	{epochs=10, batch_size=16, verbose=2} Model = LogisticRegression()	0.90	0.83	0.89	**TN** = 0.97 FN = 0.17 **TP** = 0.83 FP = 0.03
Stracking Classifier	{estimators=[('DT1', LogisticRegression(random_state=0)), ('DT2', DecisionTreeClassifier(max_depth=3, random_state=0))], final_estimator=LogisticRegression(random_state=42)}	0.83	0.73	0.81	**TN** = 0.94 FN = 0.27 **TP** = 0.73 FP = 0.06
Bagging Classifier	{base_estimator= DecisionTreeClassifier(), bootstrap=False, max_features=4, random_state=22}	0.94	0.88	0.93	**TN** = 1 FN = 0.12 **TP** = 0.88 FP = 0.005

XGBClassifier	{max_depth = 4, n_estimators=100, colsample_bytree=0.7, subsample=0.7, objective='binary:logistic', eval_metric='auc', min_child_weight=1, base_score = np.mean(y_train_bal)}	0.97	0.95	0.97	**TN** = 1 FN = 0.05 **TP** = 0.95 FP = 0.0004

En la Tabla 11, se muestran los resultados generales obtenidos de la octava y última iteración del modelo predictivo el cual contiene las mejores métricas por cada uno de los algoritmos implementados luego de haberse ejecutado cada uno de los pasos del pipeline (ver Figura 9).

Tabla 11. Métricas generales de modelos implementados

MODELO MACHINE LEARNING	HIPERPARÁMETROS (GridSearchCV)	MEJOR ESTIMADOR	MÉTRICAS OBTENIDAS
Random Forest Classifier	parameteres = {'n_estimators': [40,50,80], 'max_depth': [2,3,4,5], 'class_weight': ['None','balanced']}	{class_weight='balanced', max_depth=6, n_estimators=50, random_state=0}	Matriz de confusión: **TN** = 0.94 FN = 0.14 **TP** = 0.86 FP = 0.06 Accuracy = 0.90 Recall = 0.86 F1 score = 0.89 **ROC AUC** = 0.97
Neural Network with Autoencoder	{epochs=10, batch_size=16, verbose=2} Model = LogisticRegression()	{epochs=10, batch_size=16, verbose=2} Model = LogisticRegression()	Matriz de confusión: **TN** = 1 FN = 0.03 **TP** = 0.97 FP = 0.001 Accuracy = 0.98 Recall = 0.97 F1 score = 0.98
Stacking	cv=5, estimators=[('lr', LogisticRegression()), ('knn', KNeighborsClassifier()), ('cart', DecisionTreeClassifier())],final_estimator=LogisticRegression()	cv=5, estimators=[('lr', LogisticRegression()), ('knn', KNeighborsClassifier()), ('cart', DecisionTreeClassifier())],final_estimator=LogisticRegression()	Matriz de confusión: **TN** = 0.98 FN = 0.03 **TP** = 0.97 FP = 0.02 Accuracy = 0.97 Recall = 0.97 F1 score = 0.97

			ROC AUC = 0.99
Bagging	cv=3, estimator=BaggingClassifier (n_jobs=-1, random_state=1), n_jobs=-1, param_grid={'base_estimator': [LogisticRegression(), DecisionTreeClassifier()], 'bootstrap': [True, False], 'bootstrap_features': [True, False], 'max_features': [0.5, 1.0], 'max_samples': [0.5, 1.0], 'n_estimators': [4, 8, 10]}	{'base_estimator': DecisionTreeClassifier(), 'bootstrap': False, 'bootstrap_features': True, 'max_features': 1.0, 'max_samples': 1.0, 'n_estimators': 10}	Matriz de confusión: **TN** = 1 FN = 0.03 **TP** = 0.97 FP = 0.02 Accuracy = 0.99 Recall = 0.97 F1 score = 0.98 **ROC AUC** = 0.99
XGBClassifier	{'max_depth': [2,4,5], 'n_estimators': [80,100], 'colsample_bytree': [0.7], 'subsample': [0.7], 'objective':['binary:logistic'], 'eval_metric':['auc'], 'min_child_weight':[1], 'n_jobs': [-1]}	{'max_depth': 5, 'n_estimators': 100, 'colsample_bytree': 0.7, 'subsample': 0.7, 'eval_metric':'auc', 'min_child_weight':1, 'n_jobs': -1}	Matriz de confusión: **TN** = 1 FN = 0.03 **TP** = 0.97 FP = 0.0004 Accuracy = 0.98 Recall = 0.97 F1 score = 0.98 **ROC AUC** = 0.99

De los resultados obtenidos mediante la validación de las métricas de desempeño seleccionadas (ver sección 4.5), se puede identificar que el modelo XGBClassifier obtuvo un alto rendimiento, lo cual sería posible su selección como el mejor modelo implementado, dando cumplimiento al objetivo de la problemática sobre la clasificación de los posibles desertores (TP) y el bajo porcentaje de falsos negativos (FN). Adicionalmente, cuenta con un ROC AUC de 0.99, lo que da constancia de la sensibilidad frente a la especificidad de la respuesta.

Figura 35. Matriz de confusión modelo XGBClassifier

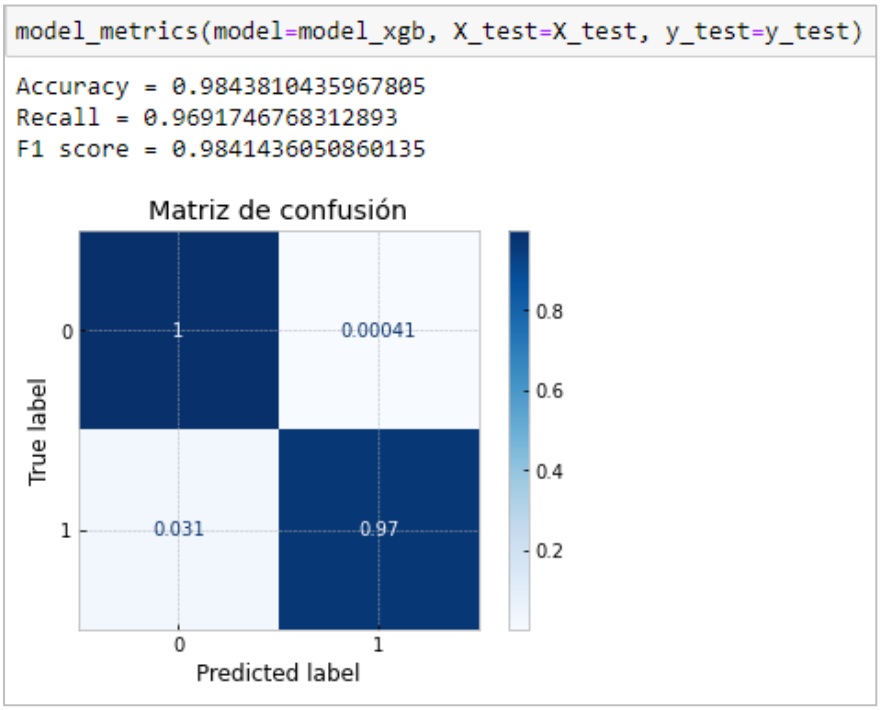
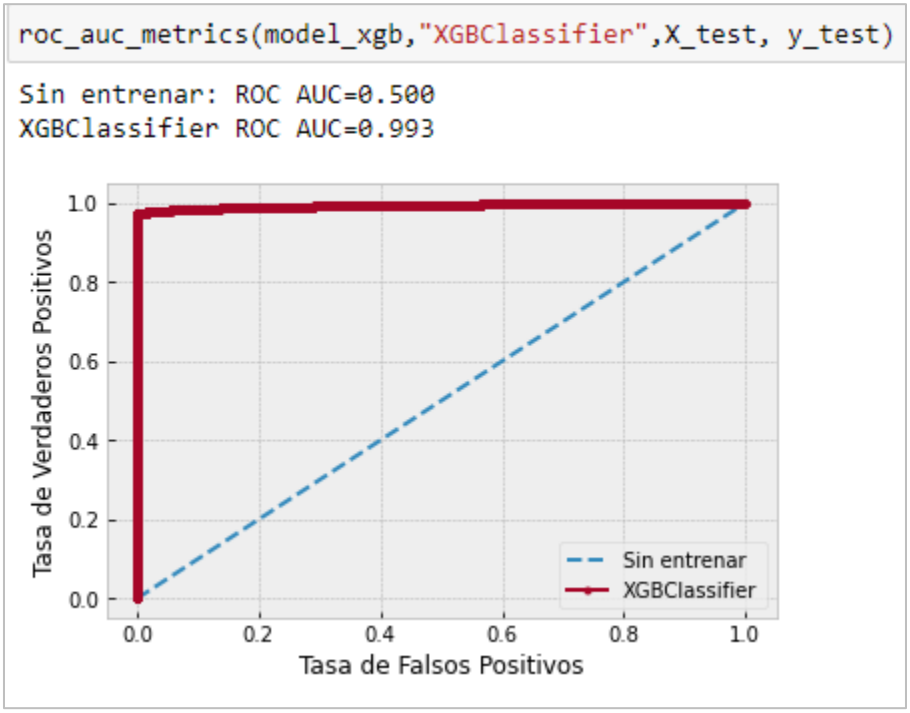


Figura 36. Gráfica ROC AUC modelo XGBClassifier



6.2 Evaluación cualitativa

Una vez realizados los procesos de recolección, preparación, preprocesamiento, implementación de modelos de Machine Learning, entrenamiento y validación para el desarrollo de un modelo predictivo de

deserción escolar, se puede identificar a partir de los resultados obtenidos, que las métricas de desempeño de los algoritmos atienden positivamente a la problemática central de la clasificación de los posibles desertores del sistema educativo. De igual manera, se atiende a la métrica de negocio concerniente a la obtención de un bajo porcentaje de falsos negativos que, de ser utilizado el mejor modelo para la toma de decisiones de la población objetivo para la aplicación de estrategias de permanencia, implicaría dejar por fuera de la priorización a aquellos estudiantes que según el modelo no desertarían, pero en la realidad sí presentaran un riesgo particular de abandonar la escuela, siendo este un escenario problemático.

Es importante mencionar que, como uno de los beneficios más relevantes frente al sistema educativo oficial, es que se busca mejorar la calidad de la educación media vocacional, dado que es necesario incrementar las habilidades de los estudiantes, para que en el momento de graduarse puedan aspirar a obtener un mejor salario en el momento de ingresar al mercado laboral. Por esto, la importancia de la disminución de la tasa de desertores en las instituciones educativas se encuentra relacionada directamente con la razón de costo/beneficio desde aspectos como: la reinversión del recurso, riesgos psicosociales, aumento en la tasa de desempleo y el trabajo informal desde el aspecto social.

7. CONCLUSIONES

Del trabajo realizado sobre el desarrollo de un modelo predictivo de deserción escolar en los colegios Oficiales de Medellín y a partir de los datos suministrados por el Observatorio para la Calidad Educativa de Medellín (OCEM) de la Secretaría de Educación para la ejecución del mismo, se presentan las principales conclusiones:

- En la etapa inicial de recolección y exploración de los datos, se logró evidenciar el gran desbalance de las clases de la variable objetivo, situación esperada dado a su significado en el contexto problemático. Por ello, se tomó la decisión de aplicar una técnica de sobremuestreo para el entrenamiento de los modelos.

- Dada la evidencia de la falta de datos que dieran cuenta de algunos factores socioeconómicos, sociodemográficos y familiares, se realizó el cruce con la fuente de información del Sisbén que permitieran complementar y aportar al perfil del estudiante y encontrar un patrón en sus características. El proceso mencionado, se realizó de manera previa e independiente dada la confidencialidad de los datos de los estudiantes.

- A partir del análisis del nivel de importancia relativa de las variables obtenido de la implementación y entrenamiento del modelo de RandomForestClassifier en las primeras iteraciones, se logró obtener un panorama de cuales podrían ser las características más relevantes en el proceso de predicción y cuales por el contrario no aportaban en el mismo. De esta manera, se definieron acciones de continuidad tales como; la reducción de dimensionalidad, creación de nuevas variables, imputación de datos, entre otros, en la etapa de preprocesamiento de los datos. Como resultado de lo anterior, se pudo evidencia un buen comportamiento en las métricas de desempeño de los modelos, validando así, lo arrojado por el estudio de las variables, su importancia y correlación con la variable objetivo.

- Mediante la implementación del modelo de ensemble XGBClassifier de XGBoost el cual solo acepta variables numéricas para su entrenamiento, se tomó la decisión de codificar aquellas variables categóricas faltantes y con un nivel de importancia relevante en el proceso. Además, este modelo presentó un alto rendimiento en su desempeño y sus métricas.

- Con las métricas obtenidas de los diferentes modelos de Machine Learning para clasificación supervisada para el proceso en estudio, se puede decir que los algoritmos de votación tales como StackingClassifier y BaggingClassifier, presentaron mejoras considerables en comparación con el modelo de RandomForestClassifier.

Para la continuación y futuros trabajos alrededor de la problemática planeada, se tienen las siguientes recomendaciones:

- Debido a que el modelo predictivo desarrollado en este trabajo, fue pensado para analizar y utilizar un dataset del año 2019, antes del contexto de pandemia por COVID19, los patrones de los datos pueden diferir en los que se presenten luego de este año, debido al cambio y alteración de las situaciones académicas, familiares y territoriales de los estudiantes. Por ello, se recomienda analizar nuevos factores cuantitativos y cualitativos que se puedan extraerse a partir de trabajo en territorio, fuentes de información secundaria, encuestas escolares, instrumentos institucionales, reportes de estrategias de permanencia y población beneficiada.

- Dada la confidencialidad de los datos, se requiere un análisis técnico sobre la estrategia más óptima y segura para desplegar el modelo en la nube, según los lineamientos operativos de la entidad.

8. REFERENCIAS

Ministerio de Educación Nacional – MEN (s.f.) Orientaciones pedagógicas. Disponible en: https://www.mineducacion.gov.co/1621/article-82787.html

Osorio, I., y Hernández, M. (2011). Prevalencia de deserción escolar en embarazadas adolescentes de instituciones educativas oficiales del Valle del Cauca, Colombia, 2006. Colombia médica, 42(3), 303-308.

Mello, A. (2020). XGBoost: theory and practice. Towards Data Science. Disponible en: https://towardsdatascience.com/xgboost-theory-and-practice-fb8912930ad6

Singh Chauhan, N. (2020). Métricas De Evaluación De Modelos En El Aprendizaje Automático. Disponible en: https://www.datasource.ai/es/data-science-articles/metricas-de-evaluacion-de-modelos-en-el-aprendizaje-automatico

Boschetti, A., y Massaron, L. (2015). Python data science essentials. Packt Publishing Ltd.

Amat Rodrigo, J. (2020). Random Forest con Python. Ciencia de datos. Disponible en: https://www.cienciadedatos.net/documentos/py08_random_forest_python.html

Brownlee, J. (2020). Autoencoder Feature Extraction for Classification. Disponible en: https://machinelearningmastery.com/autoencoder-for-classification/