Aplicación de Minería de Datos para Predecir la Deserción Estudiantil en la Facultad de Ingeniera de la Universidad del Magdalena

Janer Alberto Gonzalez Danies

Universidad Magdalena

Facultad de Ingeniería

Ingeniería de Sistemas

Santa Marta, Colombia

2021

Aplicación de Minería de Datos para Predecir la Deserción Estudiantil en la Facultad de Ingeniera de la Universidad del Magdalena

Janer Alberto Gonzalez Danies

Trabajo presentado como requisito parcial para optar al título de:

Ingeniero de Sistemas

Director (a):

Ph.D Idanis Diaz Bolaños

Codirector (a):

Ph.D Jhon Alexander Taborda

Línea de Investigación:

Nombrar la línea de investigación en la que se enmarca el trabajo

Grupo de Investigación:

Nombrar el grupo en caso que sea posible

Universidad del Magdalena

Facultad de Ingenieria

Ingeniería de Sistemas

Santa Marta, Colombia

2021

Nota de aceptación:

Aprobado por el Consejo de Programa en cumplimiento de los requisitos exigidos por la Universidad del Magdalena para optar al título de Ingeniero de Sistemas

_____

Jurado

_____

Jurado

Santa Marta, ____ de ____del _____

Dedico este trabajo mis Padres por estar hay siempre para mi y apoyarme en todo, a mis tutores de tesis Ph. D Idanis Diaz Bolaños y Ph. D Jhon Alexander Taborda por toda su ayuda y contribución al desarrollo del proyecto.

Este apenas es el inicio de este gran sueño

Resumen

El presente trabajo es un reporte final de los resultados obtenidos al emplear métodos de aprendizaje de máquinas para predecir el riesgo de deserción de los estudiantes de la Facultad de Ingeniería de la Universidad del Magdalena. En el desarrollo del proyecto se llevaron a cabo diferentes etapas propias de un proceso de minería de datos, desde la recopilación de datos, limpieza y preparación, extracción de características, y la aplicación de métodos de aprendizaje de máquinas supervisado. La principal fuente de datos utilizada en el proyecto fue el sistema de información de admisiones y registro de la Universidad, del cual se incluyeron tanto variables de tipo socio-demográficas como de tipo académico de una muestra de estudiantes de la Facultad de Ingenieria, registrados en un periodo de tiempo del 2014 hasta 2019. La muestra que se conformó contiene casos de estudiantes que no culminaron sus estudios, y estudiantes cuyo proceso de formación no se había visto en riesgo hasta la fecha de recopilación de los datos. Los datos recopilados fueron sometidos a diferentes procesos de limpieza y transformación para constituir un conjunto de datos que permitiera utilizar métodos de aprendizaje de máquinas no supervisados, y supervisados. En este trabajo se utilizaron métodos de aprendizaje no supervisado para encontrar estructuras y patrones en los datos que no son aparentemente observables; sin embargo, ninguno de los métodos implementados generó grupos compactos u homogéneos. Por otra parte, los métodos de aprendizaje supervisado generaron modelos de predicción prometedores, alcanzado tasas de precisión por encima del 95%, especialmente el random forest, y el método los k-vecinos más cercanos. A partir de estos resultados se puede concluir que sería posible emplear los modelos de predicción generados por estos métodos para predecir de manera temprana el riesgo de un estudiante de desertar en la Facultad de Ingeniería.

Palabras claves: predicción de la deserción académica, minería de datos, aprendizaje supervisado, limpieza de datos, extracción de características, random forest.

ABSTRACT

This document reports the results of applying data mining methods to predict students who are at risk of dropout in the Engineering Faculty at Universidad del Magdalena. The work includes some typical stages of a data mining process, such as data collection, data cleaning and preparation, feature extraction, clustering analysis, and prediction using supervised learning. The main source of the data used in this project was the admission, register, and academic control information system from the University of Magdalena. The collected data consists of socio-demographic and academic variables registered in the system, which were extracted from a sample of engineering student registers, in a sample period. The student registers were selected to include both cases, students who dropped out of the university for any reason, and students who according to the academic information have not been at risk of dropout their programs until the date of the query. The raw data retrieved from the information system underwent different cleaning and conversion methods to fix structural problems and to make possible the implementation of unsupervised and supervised machine learning methods. In this work, we used unsupervised methods to find structures and patterns in data that are not observable apparently; however, no distinct or meaningful groups were formed for any of the implemented methods. On the other hand, the supervised learning yielded promising results, especially, the random forest and the K-nearest neighbors' methods, reaching both precision rates greater than 95%. The prediction models built with these two methods can be helpful to prevent student desertion in the Engineering Faculty at the University of Magdalena.

Keywords: dropout prediction, data mining, machine learning, data cleaning, feature extraction, random forest.

Contenido

Contenido

Introducción **1**

1. Métodos utilizados para predecir la deserción estudiantil en la Facultad de Ingeniería **5**
1.1 Marco Teórico y Estado del Arte 5
1.1.1 Minería de Datos 5
1.2 Aprendizaje de Maquinas 7
1.2.1 Aprendizaje Supervisado 7
1.2.2 Aprendizaje No Supervisado 10
1.2.3 Trabajos Similares 12
1.3 Desarrollo del Proyecto 13
1.3.1 Recopilación de Datos 13
1.3.2 Preparación de los Datos 17
1.3.3 Selección de los Casos Relevantes Para el Estudio 20
1.3.4 Transformación de los Datos 23
1.3.5 Balanceamiento del conjunto de datos 24
1.3.6 Selección de características con el método de aprendizaje de maquinas 25
1.4 Método de Aprendizaje de Maquinas 27
1.4.1 Aprendizaje No Supervisado 27
1.4.2 Aprendizaje Supervisado 31

5. Conclusiones **39**

Lista de figuras

Figura 1 Deserción en la Universidad del Magdalena y la facultad de ingeniería en los periodos académicos 2015-1 a 2019-1 (Fuentes de datos: Oficina de planeación de la Universidad del Magdalena)..... 1
Figura 2 Tasa de deserción por nivel de formación hasta el 2018 en Colombia. Tomado de las Estadísticas de Deserción del Sistema para la Prevención de la Deserción de la Educación Superior (SPADIES) del Ministerio de Educación Nacional. 2
Figura 3 Proceso de descubrimiento de conocimiento en base de datos. Figura tomada de (Hernández et. al, 2004)..... 6
Figura 4 Esquema del proceso de aprendizaje de máquinas supervisado, Figura tomada de (Hernández et. al, 2004)..... 8
Figura 5 resume el objetivo de los algoritmos de aprendizaje no supervisado 10
Figura 6 Distribución de la población de estudiantes consultado y entregado por el Grupo de Admisiones, por programa 14
Figura 7 Distribución de la población de estudiantes consultada y entregada por el Grupo de Admisiones, por año de ingreso 15
Figura 8 Distribución de la población de graduados entregado por Admisiones, por programa 15
Figura 9 Diagramas de barra de variables eliminadas por tener un único o casi un único valor para la mayoría de los registros 19
Figura 10 resume el número de registros encontrados en cada categoría para ambas variables 21
Figura 11 Distribución de la muestra de estudiantes de la Facultad de Ingeniería de acuerdo con las condiciones definidas en la Tabla 4..... 23
Figura 12 Distribución de los conjuntos de datos obtenidos al aplicar las técnicas de over-sampling y sub-sampling para balancear las clases de la variable a predecir condición_estudiante 25
Figura 13 Resultado de la selección de variables obtenidas con Lasso 26
Figura 14 Ranking de las variables predictoras de acuerdo al método Random forest embebido 27
Figura 15 Resultados del agrupamiento utilizando el método K-*Means* 29
Figura 16 Resultados del agrupamiento utilizando el método BDSCAM 30
Figura 17 Resultados del agrupamiento utilizando el método DBSCAM 30
Figura 18 Ejemplo ejecución del método *k-nearest neighobors* sobre el conjunto de datos generado con *over-sampling* y 45 variables 32
Figura 19 Ejemplo ejecución del método r*andom forest* sobre el conjunto de datos generado con *over-sampling* y 45 variables 34
Figura 20 Ejemplo ejecución del método perceptrón multicapas sobre el conjunto de datos generado con *over-sampling* y 45 variables 35
Figura 21 Ejemplo ejecución de máquinas de soporte sobre el conjunto de datos generado con *over-sampling* y 45 variables 36
Figura 22 Ejemplo ejecución del método *XGBClassifier* sobre el conjunto de datos generado con *over-sampling* y 45 variables 37

Lista de tablas

Tabla 1. Revisión de propuestas relacionadas 12
Tabla 2 Información académica de los estudiantes 17
Tabla 3 Variables utilizadas para el entrenamiento 19
Tabla 4 Definición de dos grupos de estudiantes: en riesgo de deserción, y terminación exitosa de acuerdo con los valores de las variables Estado Académico y Situación Académica 22
Tabla 5 Variables categóricas convertidas a binarias y sobre las cuales se aplicó *One hot encoding*..... 23
Tabla 6 Resultados de *K-nearest neighbhors* sobre los conjuntos de datos generados 33
Tabla 7 Resultados del Random Forest sobre los conjuntos de datos generados 34
Tabla 8 Resultados del perceptrón multicapas sobre los conjuntos de datos generados 35
Tabla 9 Resultados de las máquinas de vector de soporte sobre los conjuntos de datos generados 36
Tabla 10 Resultados del XGBClassifier sobre los conjuntos de datos generados 37
Tabla 11 . Mejores resultados de cada uno de los métodos 38

Introducción

La Universidad del Magdalena cuenta con un gran número de datos de estudiantes, los cuales se encuentran almacenados principalmente en el Sistema de Admisiones, Registro y Control Académico (AyRe). Estos datos se utilizaron para evaluar modelos de aprendizaje supervisados y no supervisados, para predecir la deserción en la Facultad de Ingeniería de la Universidad del Magdalena. La cual en los últimos años ha presentado mayores tasas con respecto al promedio universitario, e incluso con la última tasa publicada por el Sistema Nacional para la Prevención de la Deserción de la Educación Superior, en el 2018, que fue de un 8.79% para los programas universitarios. Para mayor ilustración, como se puede observar en la Figura 1, hasta el primer semestre del año 2019 la Facultad de Ingeniería ha presentado mayores tasas de deserción en comparación con las tasas a nivel institucional. Para la Universidad del Magdalena la tasa promedio estimada en los últimos períodos académicos desde el 2015-I al 2019-1 es de un 8.6%. Mientras que para la Facultad de Ingeniería la tasa promedio calculada desde el 2015-I al 2019-1 es el 10.3%. La Figura 2 ilustra las tasas de deserción anual según los niveles de formación en Colombia hasta el 2018.

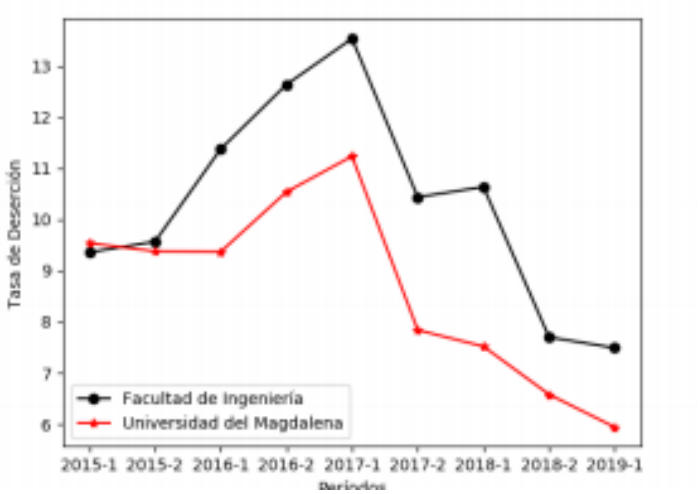


Figura 1 Deserción en la Universidad del Magdalena y la facultad de ingeniería en los periodos académicos 2015-1 a 2019-1 (Fuentes de datos: Oficina de planeación de la Universidad del Magdalena).

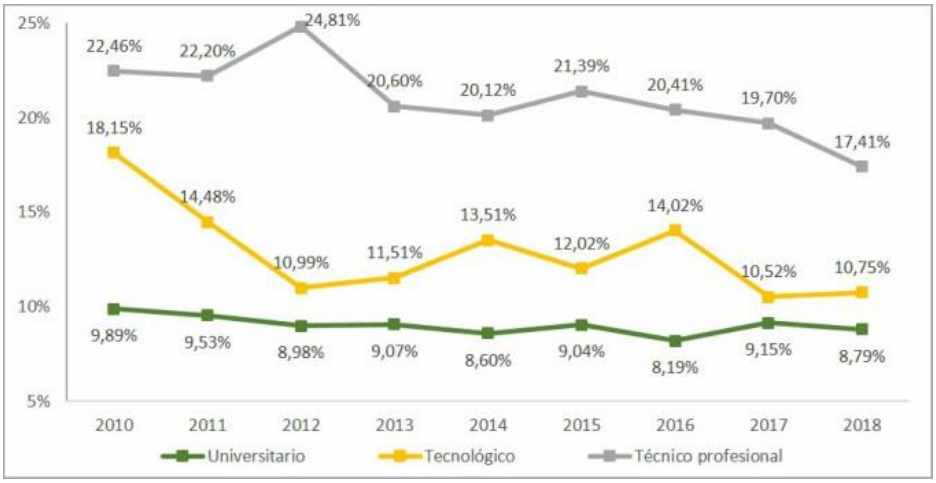


Figura 2 Tasa de deserción por nivel de formación hasta el 2018 en Colombia. Tomado de las Estadísticas de Deserción del Sistema para la Prevención de la Deserción de la Educación Superior (SPADIES) del Ministerio de Educación Nacional.

Los picos más altos de deserción en estudiantes se dan en los primeros semestres de su carrera universitaria. Siendo los factores académicos, económicos, sociales y emociónales los principales que influyen este índice. Las altas tasas de deserción estudiantil tienden a causar varias consecuencias, entre ellas se pueden mencionar: económicas, para los padres, y entidades que ofrecen ayudas de financiación a los estudiantes; en la reputación de las instituciones para quienes una tasa de graduación efectiva es un criterio de evaluación de su calidad; y en las expectativas de los estudiantes y padres quienes escogen una institución y un programa de estudio con el objetivo de mejorar su futuro al obtener un nivel de estudio profesional (Mogollón, 2018).

La Universidad del Magdalena frente al problema de la deserción estudiantil ha desarrollado programas de apoyo para los estudiantes, y ha implementado varias estrategias relacionadas con aspectos de vulnerabilidad económica, académicos en la formación secundaria, situaciones de readmisión, vocacionalidad, déficit de habilidades mentales y cognitivas, baja disposición para el estudio, inadecuado manejo de las emociones, y el consumo de sustancias psicoactivas. Un listado de estas estrategias se puede consultar en el sistema de análisis seguimiento y evaluación de la deserción (SASED). No obstante; pocos trabajos que hagan uso de la información recopilada en un sistema de información como AyRe se han hecho para detectar de manera temprana estudiantes en riesgo de deserción académica, con el propósito de focalizar las estrategias y ayudas a los estudiantes que más lo necesiten en la Institución.

En la literatura se han propuesto varios trabajos para predecir de manera temprana estudiantes en riesgo de deserción, a partir de datos, y de esta manera prevenir que el estudiante no culmine sus estudios satisfactoriamente. Trabajos como (Pal, 2012) utiliza

algoritmos de aprendizaje de máquinas para analizar y extraer información a partir de estudiantes de primer año, y así construir un modelo de predicción para identificar aquellos estudiantes que requieren atención especial para reducir las tasas de deserción. Este trabajo se realizó para estudiantes de una Facultad de Ingeniería, y se reportó una alta tasas de predicción utilizando un modelo basado en arboles de decisión. Kovačić (Kovačić, 2010) también plantea que es importante identificar estudiantes vulnerables a la deserción para que las instituciones puedan tomar medidas proactivas a tiempo. En su trabajo este autor explora factores que pueden impactar el desempeño de los estudiantes en el curso de Sistemas de la información en la Universidad Politécnica Abierta de Nueva Zelanda. Es su estudio utiliza datos de registros de los estudiantes y construye modelos para predecir el desempeño de los estudiantes en el curso. Dekker et al. (Dekker; Pechenizkiy; Vleeshouwers, 2009) utiliza técnicas de aprendizaje de minería de datos y aprendizaje de máquinas para predecir deserción en un programa de Ingeniería Eléctrica después del primer semestre. En su trabajo estos autores muestran con resultados experimentales que un modelo de clasificación basado en arboles de decisión produce una buna precisión para el caso de estudio. Otro trabajo que ilustra la posibilidad de construir un predictor de estudiantes en riesgo de deserción es (Denle, 2010) quien propone una investigación cuantitativa para explorar datos históricos institucionales para desarrollar modelos para predecir y explicar el problema de la deserción en las instituciones, también plantea la necesidad de desarrollar instrumentos de predicción temprana de estudiantes en riesgo de deserción. El autor utilizó datos de estudiantes en una ventana de cinco años de universidades en Estados Unidos y evaluó cuatro diferentes técnicas de aprendizaje de máquinas. Los resultados obtenidos mostraron que con datos suficientes y las variables apropiadas los métodos evaluados son capaces de obtener muy buenos porcentajes de precisión al predecir la deserción de estudiantes en los primeros semestres de universidad.

Al realizar la revisión de los trabajos presentadas por los autores, nace la iniciativa de buscar implementar técnicas de aprendizaje de máquinas, el cual ha sido un campo que ha ganado importancia en los últimos años, al caso particular de la deserción en la Facultad de Ingeniería de la Universidad del Magdalena. Algunas de las técnicas implementadas son árboles de decisión, redes neuronales, regresión logística, entre otras, y se tuvieron en cuenta variables de tipo demográficas, socioeconómicas y académicas.

Para finalizar, este trabajo surgió como una iniciativa dentro de un proyecto más general de la Oficina de Planeación que busca implementar métodos de analítica tanto descriptivos como descriptivos para dar soporte a la toma de decisiones frente a algunos factores relevantes de la Institución.

1. Métodos utilizados para predecir la deserción estudiantil en la Facultad de Ingeniería

1.1 Marco Teórico y Estado del Arte

La Facultad de Ingeniería de la Universidad del Magdalena es una dependencia académica y de gobierno encargada de orientar la formación integral en una o varias de las áreas de conocimiento y, en consecuencia de administrar y regentar los procesos de docencia, investigación y extensión, así como las actividades propias de del quehacer universitaria, formada por 7 programas de pregrado; los cuales se encuentran, Ingeniería Agronómica, Ingeniería Pesquera, Ingeniería de Sistemas, Ingeniería Civil, Ingeniería Ambiental y Sanitaria e Ingeniería Electrónica.(Universidad del Magdalena, 2021).Conformada por 5297 estudiantes matriculados para el primer semestre del año 2019 y teniendo una tasa de deserción del 8.6% una cifra baste preocupante debido a que se encuentra, muy por encima del promedio de la universidad.

1.1.1 Minería de Datos

Durante el desarrollo del proyecto de investigación, se utilizó minería de datos, un campo basado en estadísticas y ciencia de la computación que hace referencia al proceso de descubrir patrones en grandes volúmenes de conjuntos de datos. Para la extracción de patrones, que también puede ser visto como extracción de conocimiento se utiliza métodos de la inteligencia artificial, más específicamente de aprendizaje de máquinas, métodos estadísticos y conceptos relacionados con las bases de datos (Rokach,2010).

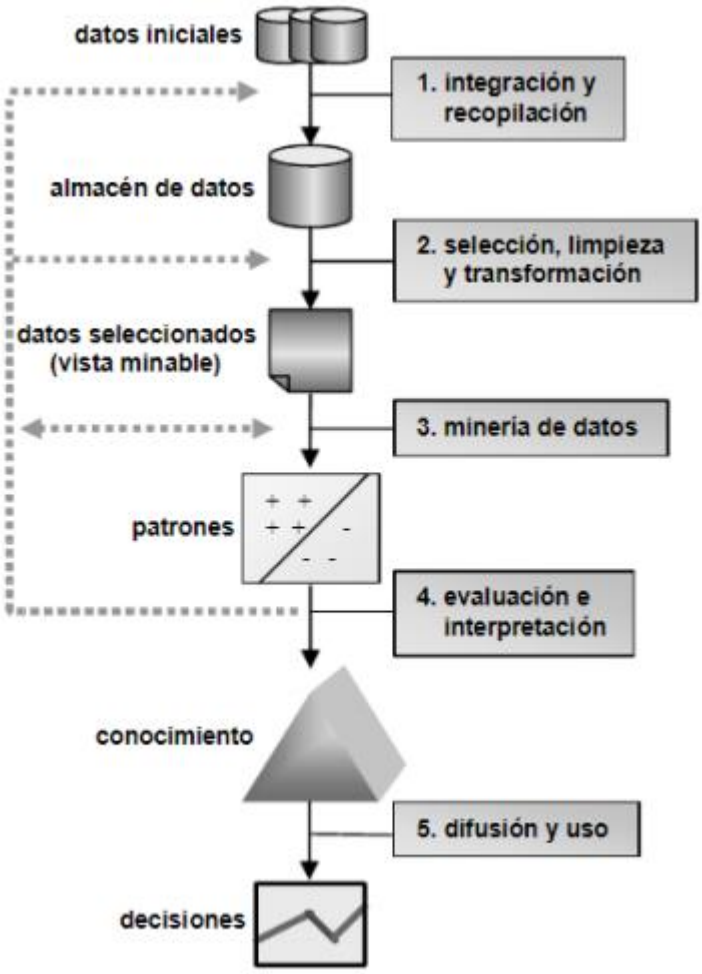


Figura 3 Proceso de descubrimiento de conocimiento en base de datos. Figura tomada de (Hernández et. al, 2004)

El proceso de extracción de conocimiento a partir de bases de datos, o minería de datos consta de cinco pasos tomados de (Hernández et. al, 2004) los cuales son:

1. Fase de integración y recopilación de datos: se determinan las fuentes de información que pueden ser útiles y el lugar donde conseguirlas. luego se transforman todos los datos a un formato común, frecuentemente mediante un almacén de datos que consiga unificar de manera operativa toda la información
2. Selección limpieza y transformación: en esta fase se eliminan o corrigen los datos incorrectos y se decide la estrategia a seguir con los datos incompletos. Además, se proyectan los datos para considerar únicamente aquellas variables

o atributos que van a ser relevantes, con el objetivo de hacer más fácil la tarea propia de minería y para que los resultados de esta sean más útiles

3. Minería de datos: se decide cuál es la tarea para realizar clasificar, agrupar, etc. y se elige el método que se va a utilizar
4. Evaluación e interpretación: se evalúan los patrones y se analizan por los expertos, y si es necesario se vuelve a las fases anteriores para una nueva iteración. Esto incluye resolver posibles conflictos con el conocimiento que se disponía anteriormente.
5. Difusión y uso: se hace uso del nuevo conocimiento y se hace partícipe de él a todos los posibles usuarios

1.2 Aprendizaje de Maquinas

El aprendizaje de máquinas es un campo de las ciencias de la computación y una rama de la inteligencia artificial, cuyo objetivo es desarrollar técnicas que permitan que las computadoras aprendan por si sola. Se dice que un agente aprende cuando su desempeño mejora con la experiencia; es decir, cuando la habilidad no estaba presente en su genotipo o rasgos de nacimiento; pero el agente los aprende a través de interacción con el medio ambiente o a través de datos (Russell, Stuart; Norvig, Peter,2009). Existen tres categorías de métodos de aprendizaje de máquinas: aprendizaje supervisado, aprendizaje no supervisado, y aprendizaje por refuerzo. El trabajo que se propone en esta propuesta está relacionado con los dos primeros tipos de aprendizajes, supervisado y no supervisado; por lo tanto, se describirá cada uno de ellos a continuación:

1.2.1 Aprendizaje Supervisado

Los algoritmos trabajan con datos “etiquetados” (labeled data), intentado encontrar una función que, dadas las variables de entrada (input data), les asigne la etiqueta de salida adecuada. El algoritmo se entrena con un “histórico” de datos y así “aprende” a asignar la etiqueta de salida adecuada a un nuevo valor, es decir, predice el valor de salida (Simeone F., 2018).

Se suele usar en problemas de clasificación, como identificación de dígitos, diagnósticos, o detección de fraude de identidad. También se usa en problemas de regresión, como predicciones meteorológicas, de expectativa de vida, de crecimiento etc. Estos dos tipos principales de aprendizaje supervisado, clasificación y regresión, se distinguen por el tipo de variable objetivo. En los casos de clasificación, es de tipo categórico, mientras que, en los casos de regresión, la variable objetivo es de tipo numérico (Géron, 2017). En este tipo de aprendizaje los algoritmos cuentan con un conjunto de ejemplos etiquetados, o con una variable a predecir ya identificada en los ejemplos. El objetivo es aprender un modelo a partir de estos ejemplos que identifique el valor de la variable a predecir para nuevos casos

que sean presentados (Géron, 2017). La Figura 4 resume el funcionamiento de un algoritmo de aprendizaje de máquinas supervisado:

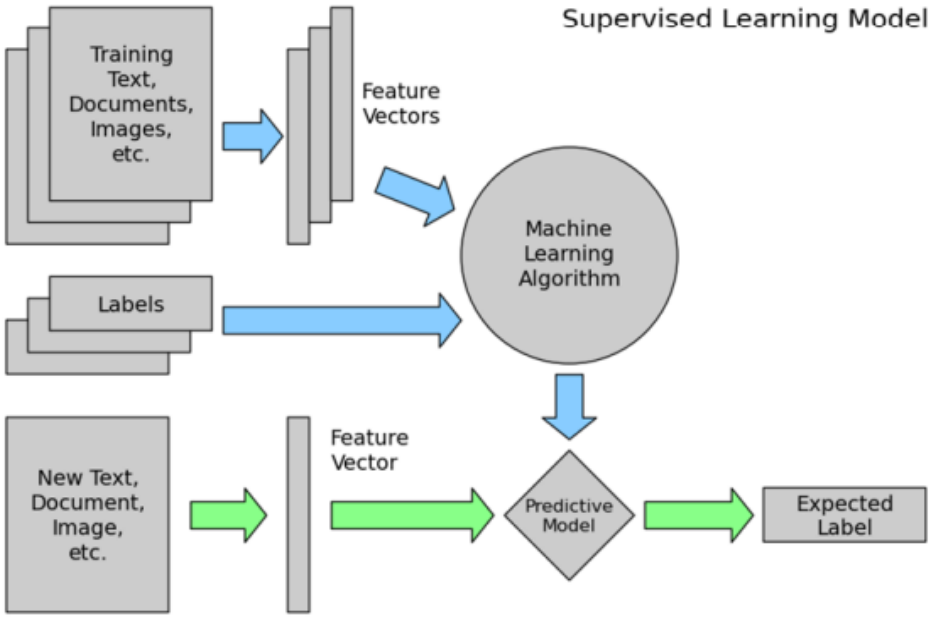


Figura 4 Esquema del proceso de aprendizaje de máquinas supervisado, Figura tomada de (Hernández et. al, 2004).

Entre las técnicas utilizadas de aprendizaje automático podemos encontrar:

- Arboles de Decisión. Un árbol de decisión es un conjunto de condiciones organizadas en una estructura jerárquica, de tal manera que la decisión final a tomar se puede determinar siguiendo las condiciones que se cumplen desde la raíz del árbol hasta alguna de sus hojas. Los árboles de decisión se utilizan desde hace siglos, y son especialmente apropiados para expresar procedimientos médicos, legales, comerciales, estratégicos, matemáticos, lógicos, etc.(Hernández et. al, 2004).

- *Random Forest*. También conocido en castellano como “Bosques Aleatorios” es una combinación de árboles predictores tal que cada árbol depende de los valores de un vector aleatorio probado independientemente y con la misma distribución para cada uno de estos. Es una modificación sustancial de *bagging* que construye una larga colección de árboles no correlacionados y luego los promedia (Betancourt, 2005)

- Método de Regresión Logística. La regresión logística (RL) forma parte del conjunto de métodos estadísticos que caen bajo tal denominación y es la variante que corresponde al caso en que se valora la contribución de diferentes factores en la ocurrencia de un evento simple. En general, la regresión logística es adecuada cuando la variable de respuesta Y es politómica (admite varias categorías de respuesta, tales como mejora mucho, empeora, se mantiene, mejora, mejora mucho), pero es especialmente útil en particular cuando solo hay dos posibles respuestas (cuando la variable de respuesta es dicotómica), que es el caso más común (Géron, 2017).

- Redes Neuronales. Es un método computacional vagamente inspirado en el comportamiento observado en su homólogo biológico. Consiste en un conjunto de unidades, llamadas neuronas artificiales, conectadas entre sí para transmitirse señales. La información de entrada atraviesa la red neuronal (donde se somete a diversas operaciones) produciendo unos valores de salida (Liu,2002).

- Los K Vecinos más Cercanos. Es un método para clasificar casos basándose en su parecido a otros casos. En el aprendizaje automático, se desarrolló como una forma de reconocer patrones de datos sin la necesidad de una coincidencia exacta con patrones o casos almacenados. Los casos parecidos están próximos y los que no lo son están alejados entre sí. Por lo tanto, la distancia entre dos casos es una medida de disimilaría.

- *XGBoost*. Es un algoritmo de impulso basado en el algoritmo de árboles de decisión impulsados por gradientes. Este método aplica una mejor técnica de regularización para reducir el sobreajuste, y es una de las diferencias con el aumento de gradiente. El '*XGBoost'* es una biblioteca de código abierto que proporciona algoritmos de aprendizaje automático bajo los métodos de aumento de gradiente. (Chen, Guestrin, 2016).

1.2.2 Aprendizaje No Supervisado

Son métodos de aprendizaje automático donde el modelo generado se ajusta a las observaciones. Se distingue del aprendizaje supervisado por el hecho de que no hay un conocimiento a priori. En el aprendizaje no supervisado, un conjunto de datos de objetos de entrada es tratado. Así, el aprendizaje no supervisado típicamente trata los objetos de entrada como un conjunto de variables aleatorias, siendo construido un modelo de densidad para el conjunto de datos (Hinton y Sejnowski,1999).

Tiene lugar cuando no se dispone de datos “etiquetados” para el entrenamiento, o no se cuenta con una variable a predecir con sus respectivos valores para cada ejemplo en el conjunto de ejemplos. Sólo conocemos los datos de entrada, pero no existen datos de salida que correspondan a un determinado input. Por lo tanto, sólo podemos describir la estructura de los datos, para intentar encontrar algún tipo de organización o patrón que simplifique el análisis. Por ello, tienen un carácter exploratorio (Santos, 2017) Se me puede observar claramente en la figura 5.

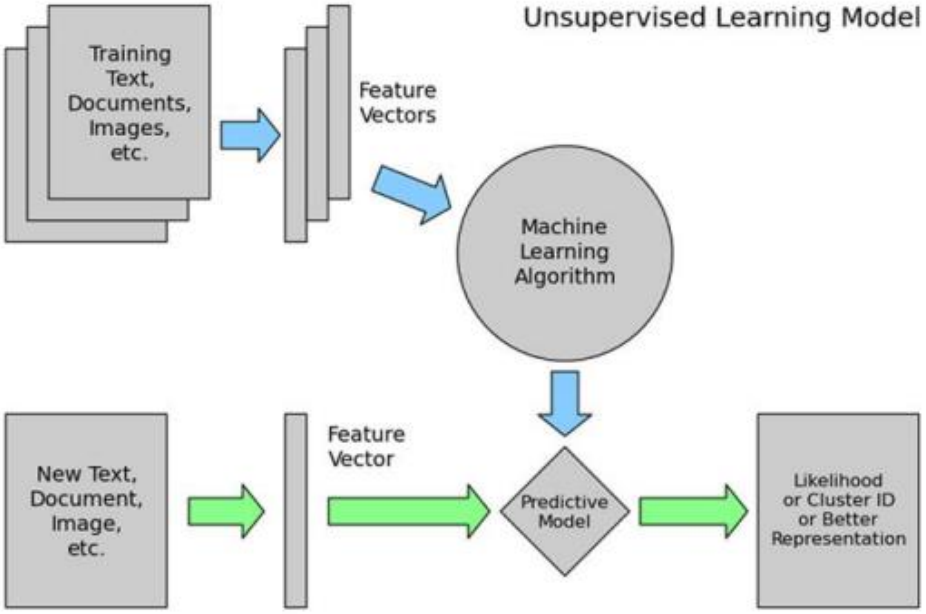


Figura 5 resume el objetivo de los algoritmos de aprendizaje no supervisado

Entre los algoritmos usados para el entrenamiento del modelo, en aprendizaje de máquinas no supervisados, podemos mencionar:

- Análisis de Componentes Principales. Es una técnica utilizada para describir un conjunto de datos en términos de nuevas variables ("componentes") no correlacionadas. Los componentes se ordenan por la cantidad de varianza original que describen, por lo que la técnica es útil para reducir la dimensionalidad de un conjunto de datos (Géron, 2017).

- Algoritmo de *Clustering*. Consiste en agrupar una serie de vectores según un criterio en grupos o clústeres. Generalmente el criterio suele ser la similitud por lo que diremos que agrupa los vectores similares en grupos. Está considerado como un aprendizaje no supervisado dentro de la minería de datos.

- Método K-*Means*. Algoritmo para agrupar los datos tratando de separar las muestras en n grupos de igual varianza, minimizando un criterio conocido como inercia o suma de cuadrados dentro del grupo. Este algoritmo requiere que se especifique el número de clústeres. Se adapta bien a una gran cantidad de muestras y se ha utilizado en una amplia gama de áreas de aplicación en muchos campos diferentes siendo de propósito general a la hora de agrupar (*Skcikit-Learn,2020*).

- DBSCAM. Este algoritmo ve los clústeres como áreas de alta densidad separadas por áreas de baja densidad. Debido a esta visión bastante genérica, los grupos encontrados por DBSCAN pueden tener cualquier forma, a diferencia de k-*Means*, que asume que los grupos tienen forma convexa permitiendo flexibilidad y dinamismo de cada uno de los clústeres (*Skcikit-Learn,2020*).

- *Spectral*. Este tipo de agrupamiento es muy útil cuando la estructura de los grupos individuales es muy no convexa o, más en general, cuando una medida del centro y la extensión del grupo no es una descripción adecuada del grupo completo (*Skcikit-Learn,2020*).

- Agrupamiento Aglomerativo. Es un método de análisis de agrupación que busca construir una jerarquía de agrupaciones. Las estrategias para la agrupación jerárquica en un enfoque "de abajo hacia arriba ": cada observación comienza en su propio grupo y los pares de grupos se fusionan a medida que se asciende en la jerarquía (*Skcikit-Learn,2020*)

1.2.3 Trabajos Similares

En la literatura existen numerosos trabajos que aplican técnicas de aprendizaje de máquinas y minería de datos para predecir el riesgo de deserción de un estudiante. La Tabla 1 resume una revisión de varios de estos trabajos:

Tabla 1. Revisión de propuestas relacionadas

Trabajo	Técnica Utilizada	Atributos o variables
Aulck,2016	• Regresión logística regularizada • Los k vecinos más cercanos • *Random forest*	• Información demográfica (raza, genero, fecha de nacimiento, estatus de residencia e identificación como hispano), • información previa de la entrada a la universidad (puntaje SAT y ACT) • registros completos de transcripciones (clases tomadas, tiempo en que fueron tomadas, calificaciones recibidas especializaciones declaradas)
Dele,2010	• Redes neuronales • Arboles de decisión • Máquinas de soporte vectorial • Regresión logística	Universidad, Grado, Mayor, Horas de otoño registrada, Horas de ganas den otoño, GPA de otoño, GPA acumulativo de otoño, Horas de primavera registradas, Horas de primavera ganadas, GPA de primavera, Etnicidad, Sexo, Código residencial, Estado civil, SAT alto puntaje comprensivo, Edad, Tipo de admisión, Horas de transferencia, Horas ganadas con CLEP
Raju,2015	• Regresión logística • Redes neuronales	• Pre-Universitarios: Etnicidad, Residencia a género, información trabajo, crédito, Opción de universidad • Universidad: ACT, SAT, GPA de matemáticas de escuela segundaria • Objetivo: escuela, segundaria GPA, Distancia a casa
Castro,2018	• Arboles de decisión	Estado civil, Victima del conflicto armado, Estrato, Categoría Sisbén, Núcleo familiar, Trabajo, Motivación, Contribuye a la familia,

Pal,2012	• Arboles de decisión alternativos • Arboles de decisión CART • Arboles de decisión ID3	• Datos demográficos (categoría, genero, etc.) • Datos de rendimiento (SSC o 10 puntaje, HSC, puntaje exámenes, etc.) • Direcciones y numero de contacto
Kovacic,2010	• Arboles de clasificación	• Variables demográficas (edad, genero, nivel educativo, situación laboral, discapacidad) • Variables de entorno de estudio (nombre del programa, bloque del curso, etc.)
Dekker,2009	• Arboles de decisión CART • Clasificador Bayesiano • Modelo lógico • Ramdon forest	• Datos del departamento de ingeniería electrónica referentes al entorno de estudio

La mayoría de los trabajos encontrados que aplican técnicas de minería de datos a información relacionada con la deserción estudiantil en la educación superior utilizan técnicas de aprendizaje supervisado basadas en árboles de decisiones, métodos regresión logística y redes neuronales. Algunas de estas técnicas se encuentran definidas en la Sección de marco teórico También se puede observar que las variables más utilizadas en los trabajos referenciados en la Tabla 1 son: información demográfica del estudiante (raza, genero, nivel educativo, nivel socioeconómico, etc.) e información académica (calificaciones, desempeño, tipo de admisión, etc.)

1.3 Desarrollo del Proyecto

El proyecto se desarrolló siguiendo tres etapas para la realización de minería de datos, descritas en la figura 3. que incluyen la recopilación de datos, la preparación de los datos, y la aplicación de métodos tanto de aprendizaje no supervisado como supervisados con su respectiva evaluación y análisis.

1.3.1 Recopilación de Datos

En esta fase se realizó la recopilación de información de la población de estudiantes de los años 2015 al 2019, pertenecientes a los siete programas académicos de la Facultad de Ingeniería la Universidad del Magdalena. Para la recopilación de esta información se consultó directamente el Grupo de Admisiones y -Registro de la Universidad, quienes proporcionaron la historia académica y algunos campos relacionados con información socio-demográficas de estudiantes y graduados de la Facultad de Ingeniería en una ventana de tiempo, del año 2014 al primer semestre de 2020. La información fue entregada

en archivos separados, por un lado, estudiantes que hasta la fecha de la consulta permanecían en esta condición independientemente de la su situación académica, y por otro lado graduados de los diferentes programas de la Facultad de Ingeniería.

De los archivos entregados con información de estudiantes se eliminaron inicialmente los registros de estudiantes con calificaciones cualitativas, con el fin de trabajar con un conjunto de datos con tipos de datos homogéneos, es decir calificaciones cuantitativas. De esta manera, este trabajo se llevó a cabo con una población de estudiantes de la Facultad de Ingeniería de 6952 estudiantes, y 266 graduados. En las Figuras 6 y 7 se puede observar la distribución de la población de estudiantes por programas, y por año de ingreso, y en la Figura 8 la distribución de los graduados por programa.

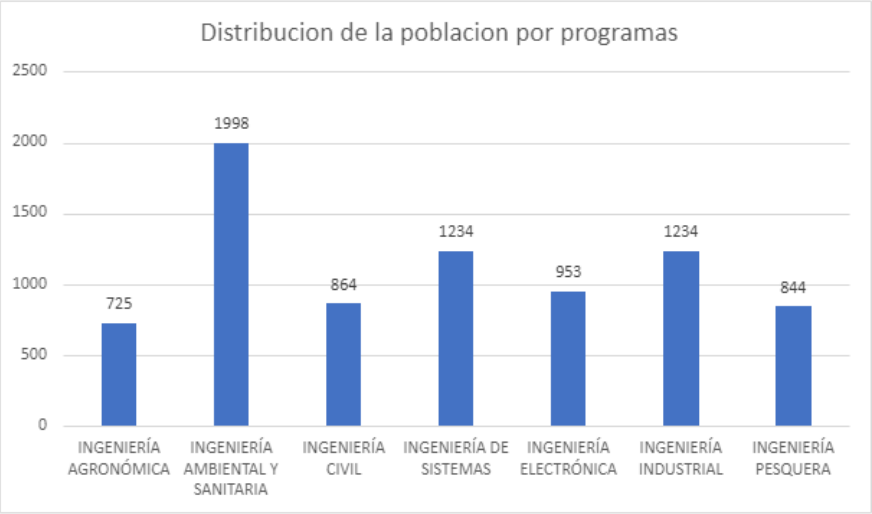


Figura 6 Distribución de la población de estudiantes consultado y entregado por el Grupo de Admisiones, por programa

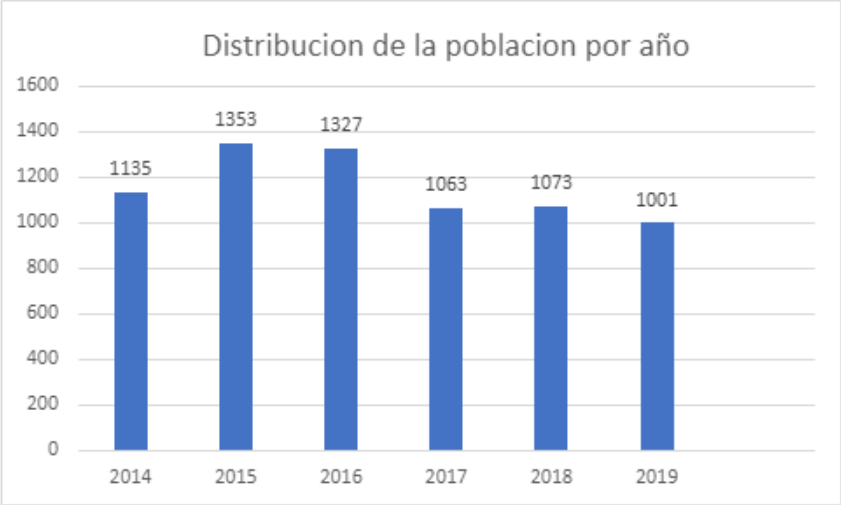


Figura 7 Distribución de la población de estudiantes consultada y entregada por el Grupo de Admisiones, por año de ingreso

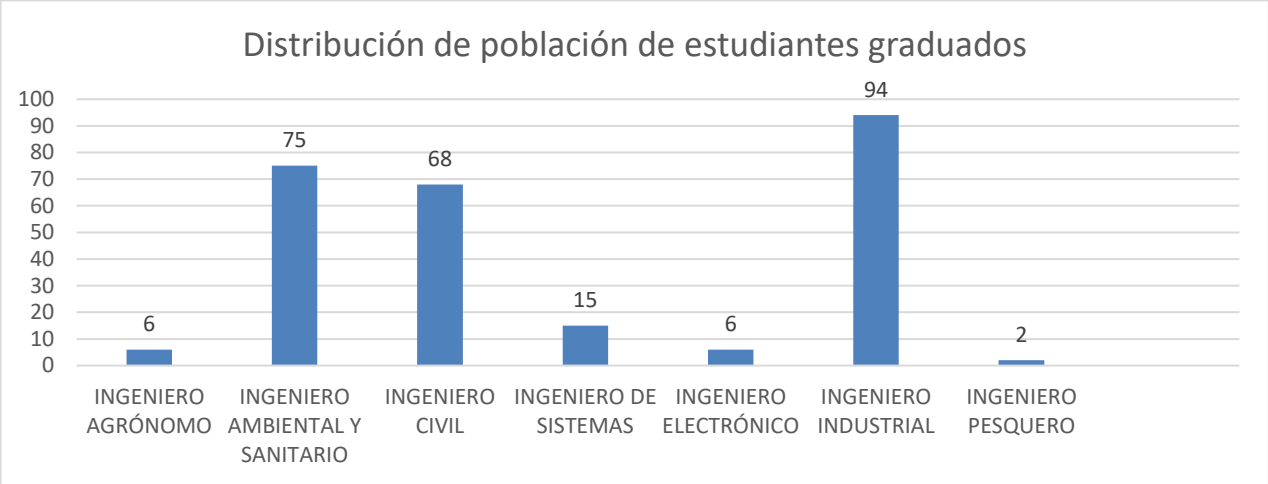


Figura 8 Distribución de la población de graduados entregado por Admisiones, por programa

En los datos suministrado se encontraron 72 variables con información general y socio-demográficas de los estudiantes, aparte de su historial académico. De las 72 variables 37 correspondían a información general tal como el nombre del programa, el promedio semestral, el promedio acumulado, el estado del estudiante, entre otras; y 35 variables correspondían a información socio-demográficas tales como el sexo, el estrato socioeconómico, el tipo de colegio del cual egreso el estudiante, el nombre del colegio, entre otras.

La historia académica de los estudiantes también fue entregada por el Grupo de Admisiones y Registro hasta la fecha de la consulta. Es decir, se contó con el nombre de asignaturas, el número de veces cursadas, y las respectivas calificaciones finales de lo que cada estudiante había cursado hasta el momento de la consulta, es decir hasta el periodo 2020-I.

1.3.2 Preparación de los Datos

De acuerdo con Hernández *et al.* "*El éxito de un proceso de minería de datos depende, no solo de tener todos los datos necesarios (una buena recopilación), si no que estos estén íntegros, completos y consistentes*" (Hernández, J. & Ramírez, M. & Ferri, C.,2004), por ello es importante su agrupación, búsqueda de datos relevantes y que estos tengan buena limpieza e integración. Para cumplir con este propósito se llevaron a cabo diferentes pasos, los cuales se explicarán a continuación:

- Selección de Características. Con el objetivo de involucrar el historial académico de los estudiantes en este estudio, se incluyeron a manera de variables la siguiente información listada en la Tabla 2:

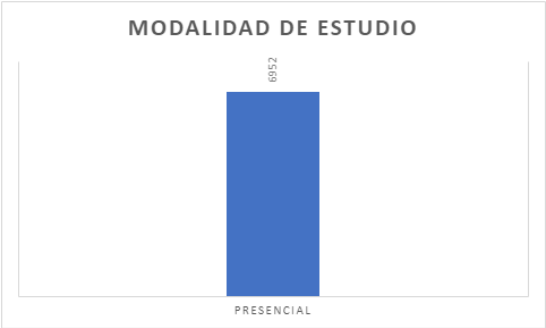
Tabla 2 Información académica de los estudiantes

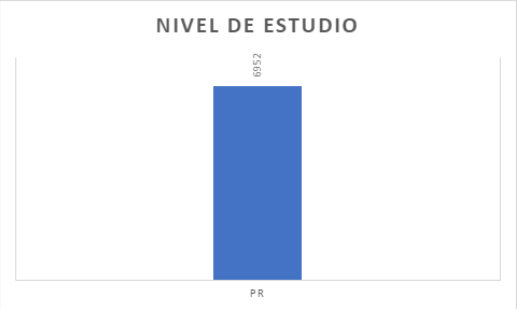
Información	Descripción
Calificaciones de asignaturas comunes en todos los planes de estudios de la Facultad de Ingeniería	Las notas finales obtenidas por los estudiantes en aquellas asignaturas que son comunes para todos los programas de las Facultad.
Número de veces cursadas de las asignaturas	Por cada una de las asignaturas individuales incluida en este estudio, también se adicionó una columna indicando el número de veces que el estudiante la cursó.
Promedio de las ciencias básicas de ingeniería	Se cálculo un promedio con las asignaturas del área de ciencias básicas de ingeniería que no son comunes para todos los programas, para esto se consultaron los Acuerdos Académicos del plan de estudio de cada programa de la Facultad.

Promedio en el área de ingeniería aplicada	Las asignaturas disciplinares de cada programa también se les estimó un promedio por cada estudiante, por ejemplo, las asignaturas del componente circuitos para el caso de ingeniería electrónica, o las asignaturas del componente de diagnóstico ambiental para el caso de ingeniería ambiental.
Máximo de veces en repetir asignatura de ciencias básicas de ingeniería	Teniendo cuenta las asignaturas de ciencia básicas de ingeniería de cada uno de los programas, se obtuvo el número de máximo de veces que el estudiante repitió alguna de las materias del conjunto
Máximo de veces en repetir asignatura de ingeniería aplicada	Tomando consideración las materias pertenecientes al grupo de aplicados de ingeniería, se extrajo el número máximo de veces que el estudiante repitió materias del conjunto

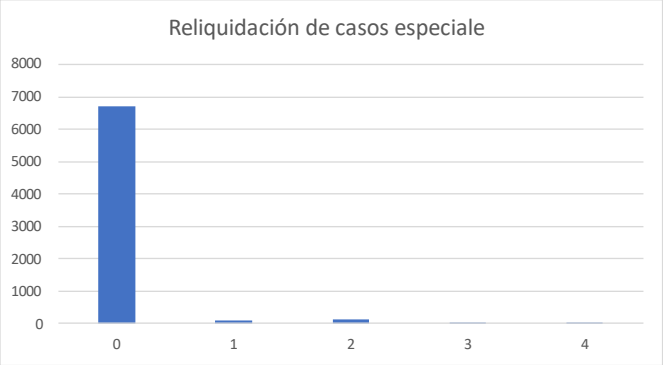
La estimación de promedios por las áreas de ciencias básicas y de ingeniería aplicada se efectuó con el fin de evitar colocar en el conjunto de datos final una columna por cada asignatura existente en la Facultad, teniendo en cuenta que los programas ven muchas asignaturas diferentes. La Tabla 3 lista los nombres de las asignaturas incluidas en este estudio.

Luego de este tratamiento de la información académica de cada estudiante se contaba con 110 variables en el conjunto de datos para trabajar. De estás 110 variables se hizo una selección de un subconjunto de ellas teniendo en cuenta que algunas variables no son apropiadas para trabajar con métodos de aprendizaje de máquinas. Más específicamente, se eliminaron aquellas variables que contenían un valor único o casi único, como por ejemplo tipo de documento, la modalidad de estudio, nivel de estudio, reliquidación casos especiales. También se eliminaron aquellas variables de difícil manejo debido a algún tipo de error en sus valores, o el tratamiento de la misma en los datos recopilados, por ejemplo el nombre del colegio en el que el estudiante terminó la secundaria, este campo contenía errores de escritura un mismo colegio podía aparecer en el conjunto de datos escrito de diferentes formas, de tal manera que en lugar de considerar el nombre del colegio solo se consideró en este estudio el tipo de colegio, público o privado. Otro ejemplo es la variable Saber 11 esta variable no contiene el valor obtenido en la prueba, si no el código con el que fue registrado en el sistema nacional de resultados de la prueba. En la Figura 9 podemos observar diagramas de barra ilustrando variables con un único valor para la población de estudiantes que se recibió del grupo de Admisiones.





a) Modalidad de estudio que cursa el estudiante b) Nivel de estudio del programa



c) Reliquidación de casos especiales

Figura 9 Diagramas de barra de variables eliminadas por tener un único o casi un único valor para la mayoría de los registros

Luego de esta etapa de selección de variables se obtuvo un subconjunto final de 60 variables, tal como se muestra en la Tabla 3.

Tabla 3 Variables utilizadas para el entrenamiento

	Tipo de Información		
	General	**Socio-demográficas**	**Historia académica**
Nombre de la variable	ESTADO SITUACION_ACADEMICA MODALIDAD DE INGRESO CUPO_ESPECIAL PROM_ACUMULADO Porc_Cred_Oblig_Aprobados READMISIONES PLAN	ESTRATO TIPO_COLEGIO SEXO EDAD ORIGEN_MAGDALNA ORIGEN_STA Marta RESIDE_MAGDALENA RESIDE_STA Marta	PROCESOS_LECTORES VECES_PROCESOS RAZONAMIENTO VECES_RAZONAMIENTO VIDA_UNIVERSITARIA VECES_REPTIR_VIDA CALCULO_DIFERENCIAL VECES_CALCULO_DIFERENCIAL

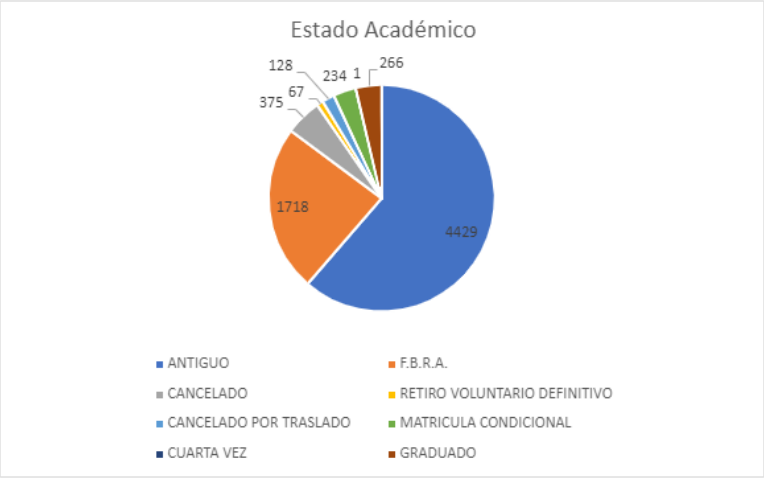
	PROGRAMA AÑO_UNDECIMO TOTAL_PERIODOS_ACADEMICOS CREDPLAN		CALCULO_INTEGRAL VECES_CALCULO_INTEGRAL ECUCACIONES_DIFERENCIALES VECES_ECUACIONES ESTADISTICA_I VECES_ESTADISTICA_I ESTADISTICA_II VECES_ESTADISTICA_II MECANICA VECES_MECANICA ETICA VECES_ETICA FOMACION_HUMANISTICA VECES_FORMACION_HUMANISTICA CATEDRA_DEL_CARIBE VECES_CATEDRA_DEL_CARIBE EXPRESION_ORAL VECES_EXPRESION_ORAL TEORIA_FILOSOFIA VECES_TEORIA_FILOSOFIA METODOLOGIA VECES_METODOLOGIA SEMINARIO_I VECES_SEMINARIO_I SEMINARIO_II VECES_SEMINARIO_II SEMINARIO_III VECES_SEMINARIO_III Ciencias_bascias_ingenieria Veces_ciencias_basicas MAX Ingenieria_aplicada Veces_ingenieria:aplicada MAX

1.3.3 Selección de los Casos Relevantes Para el Estudio

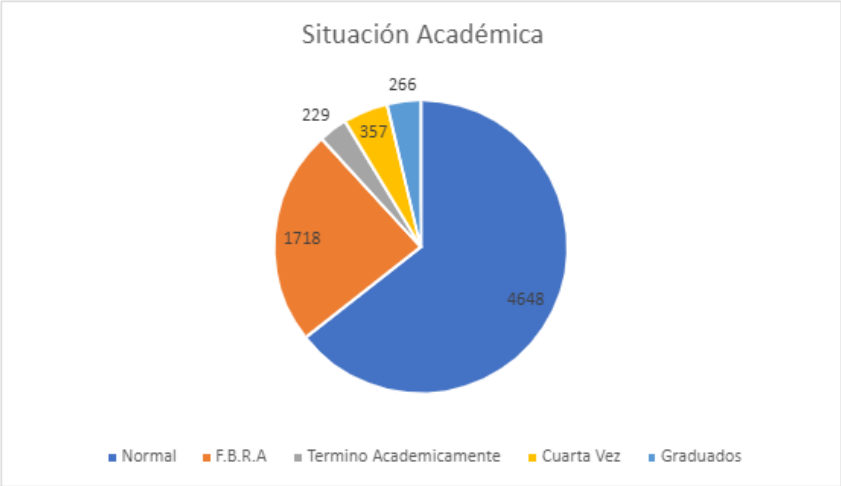
Con el propósito de aplicar técnicas de aprendizaje de máquinas para predecir estudiantes en riesgo de deserción en la Facultad de Ingeniería de la Universidad del Magdalena, se utilizaron las variables ESTADO, y SITUACION_ACADEMICA para definir casos de estudiantes en riesgo de no culminar exitosamente sus estudios, y casos de estudiantes que terminan exitosamente sus estudios.

La variable ESTADO presenta 7 valores en los archivos entregados por el Grupo de Admisiones para los estudiantes de la Facultad de Ingeniería; por su parte para la variable

SITUACION_ACADEMICA se encontraron 4 valores, y sus respectivos valores. La Figura 10 resume el número de registros encontrados en cada categoría para ambas variables.



a) Estado académico de los estudiantes



b) situación académica de los estudiantes

Figura 10 resume el número de registros encontrados en cada categoría para ambas variables

A partir de estas dos variables y sus valores se crearon dos grupos de estudiantes de acuerdo con las condiciones que se pueden observar en la Tabla 4:

Tabla 4 Definición de dos grupos de estudiantes: en riesgo de deserción, y terminación exitosa de acuerdo con los valores de las variables Estado Académico y Situación Académica

Grupos de estudiantes	
En riesgo de deserción	Terminación exitosa
Estudiantes cuyo valor en la variable Situación Académica corresponde a fuera por bajo rendimiento académico, F.B.R.A	Estudiantes cuyo valor en la variable Situación Académica es Normal, que no hayan estado en cuarta vez, y con menos de dos readmisiones
Estudiantes cuyo valor en la variable Situación Académica corresponde a Cuarta Vez	Estudiantes cuyo valor en la variable Situación Académica es Termino Académicamente, y que no hayan estado en cuarta vez y con menos de dos readmisiones
Estudiantes cuyo valor en la variable Estado Académico es Retiro Voluntario Definitivo	Estudiantes cuyo valor en la variable Situación Académica es Graduado, y que no hayan pasado cuarta vez
Estudiantes cuyo valor en la variable Estado Académico es Matricula condicional	
Estudiantes cuyo valor en la variable Estado Académico es Cancelado, y se puede observar que llevaba más de 3 periodos académicos sin estudiar	

Al aplicar este conjunto de condiciones sobre los datos de la muestra de estudiantes de la Facultad que se recopiló para este trabajo se obtuvo la siguiente distribución de casos que muestra la Figura 11.

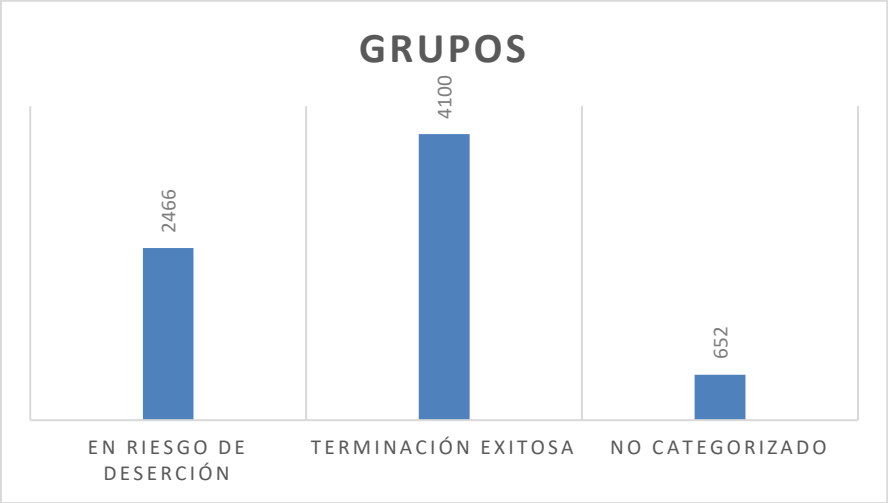


Figura 11 Distribución de la muestra de estudiantes de la Facultad de Ingeniería de acuerdo con las condiciones definidas en la Tabla 4

1.3.4 Transformación de los Datos

Luego de todos los pasos realizados hasta el momento se cuenta con un conjunto de datos de 6566 de estudiantes de la Facultad de Ingeniería, con 58 variables predictoras que corresponden a todas las variables listadas en la Tabla 3 excepto Situación Académica y Estado Académico, de las cuales se derivó la variable a predecir que se le dio el nombre de condicion_estudiante y toma los valores: en riesgo de deserción, y terminación exitosa.

Entre las 59 variables se contaba con variables tanto de tipo numérico, como nominales, y ordinales. Por lo tanto, se procedió a hacer transformaciones para trabajar con un solo tipo de datos, numéricos. Para esto, las variables que tomaban solo dos posibles como el sexo, y condición_estudiante se codificaron directamente como 1 y 0, y para el caso de las variables categóricas se utilizó el método de conversión *One hot encoding* a través de la función *OneHotEncoder* de la librería *Sickit-learn* de Python. La Tabla 5 lista las variables que fueron transformadas de esta manera.

Tabla 5 Variables categóricas convertidas a binarias y sobre las cuales se aplicó *One hot encoding*

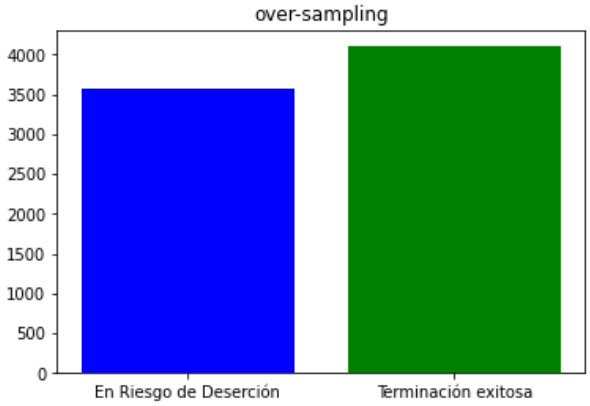
Nombre variable	Categoría	Tipo de transformación
Sexo	Masculino, Femenino	Binaria
Tipo de Colegio	Público, Privado	Binaria
Programa	Ingeniería Agronómica, Ingeniería Ambiental, ingeniería Civil, Ingeniería de sistemas ,Ingeniería Electrónica, ingeniería Industrial, Ingeniería Pesquera	*One hot encoding*
Tipo de ingreso	Nuevo, Pilo paga, Transferencia, Talento Unimagdalena, Afrocolombiana, Acsup 014 2013,Simultaniedad, Traslado, artista acsup 26 de 2009, Artista acsup 26 de 2017, Bachiller pescador costa caribe, Beca Taganga, Beneficiario héroe de la nación, Cupo Icfes, Deportista acsup 26 de 2009, Deportista acsup 26 de 2017,Deportista galardonado ACSUPp 27 de 2009, Desplazado por la violencia, Indígena, Mujer bachiller cabeza familia, Ninguno	*One hot encoding*

Ciudad de Residencia	Santa Marta, Otro	Binaria
Ciudad de origen		
Departamento Origen	Magdalena, Otro	Binaria
Departamento de Residencia		

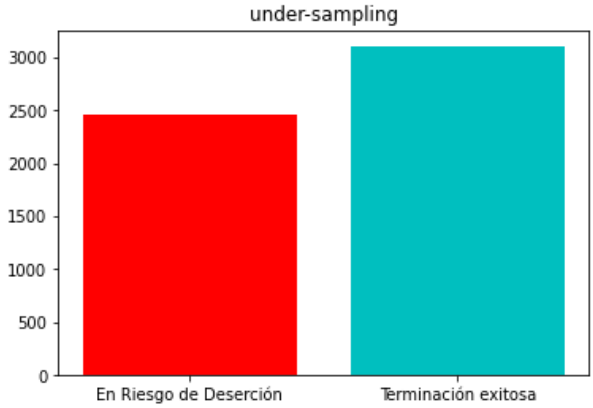
En este paso el conjunto de datos nos quedó con 88 variables incluyendo aquellas generadas por el procedimiento del *One hot encoding*, y la variable a predecir.

1.3.5 Balanceamiento del conjunto de datos

Como se puede observar en la Figura 11 el número de casos en las categorías: en riesgo de deserción, y terminación exitosa está desbalanceado, es decir, el número de estudiantes en el grupo terminación exitosa es mayor que el número de casos en la otra categoría definida. Por lo tanto, se buscó equilibrar las dos categorías usando las técnicas de análisis de datos conocidas como *over-sampling*, y *under-sampling*. El *over-sampling* consiste en muestrear aleatoriamente con reemplazo casos del conjunto de datos que pertenecen a la categoría con menos registros y volverlos a incluir en el conjunto final de datos, con el propósito de aumentar el número de casos que pertenecen a esa categoría. El sub-sampling por su parte consiste en retirar aleatoriamente del conjunto final de datos casos que pertenecen a la categoría con más registros, con el objetivo de disminuir la diferencia entre las clases (Ling, Charles X. y Chenghui Li, 1998). Al aplicar estas técnicas en el conjunto de datos se obtuvo como resultados dos nuevos conjuntos de datos cuya distribución de acuerdo con las categorías de la variable a predecir se pueden observar en la Figura 12.



a) Conjunto de datos por over-sampling



b) Conjunto de datos por under-sampling

Figura 12 Distribución de los conjuntos de datos obtenidos al aplicar las técnicas de over-sampling y sub-sampling para balancear las clases de la variable a predecir condición_estudiante

1.3.6 Selección de características con el método de aprendizaje de maquinas

Con el objetivo de mejorar el desempeño y costo computacional de los métodos de minería de datos a aplicar, en esta etapa del procesamiento de los datos se hizo un análisis de la importancia de cada variable predictora con respecto a la variable a predecir. Para esto se utilizaron métodos de selección de atributos embebidos (Brownlee, 2015), más específicamente se implementaron los métodos *least absolute shrinkage* (Lasso) que se describe como un método de análisis de regresión que realiza selección de variables mediante regularización para mejorar la exactitud e interpretabilidad del modelo (Tibshirani, Robert, 1996), y el método *Random Forest Importance* el cual se conoce

como un método para calcular la importancia de la característica en muestras (Płoński, 2020).

Durante el proceso de regularización Lasso reduce a cero los coeficientes de aquellas variables que no sean relevantes para predecir la clase, y en el caso del *Random Forest Importance* computa un ranking de las variables organizándolas de mayor importancia a menor importancia para predecir la clase. Al aplicar Lasso sobre el conjunto de datos obtuvimos el resultado que se puede observar en la Figura 13.

```
total features: 87
selected features: 55
Index(['EDAD', 'ULT_PER_ACAD', 'CREDPLAN', 'ESTRATO', 'PROM ACU',
       'Porc. Cred. Oblig. Aprobados', 'READMISIONES.1', 'PLAN',
       'TIPO COLEGIO', 'SEXO', 'CIUDAD ORIGEN', 'CIUDAD RESIDENCIA',
       'DEPARTAMENTO ORIGEN', 'DEPARTAMENTO RESIDENCIA', 'AÑO UNDECIMO',
       'TOTAL PERIODOS ACADEMICOS', 'PROCESOS_LECTORES', 'RAZONAMIENTO',
       'VIDA_UNIVERSITARIA', 'CALCULO_DIFERENCIAL', 'CALCULO_INTEGRAL',
       'VECES_CALCULO_INTEGRAL', 'ECUCACIONES_DIFERENCIALES',
       'VECES_ECUACIONES', 'ESTADISTICA_II', 'VECES_ESTADISTICA_II',
       'MECANICA', 'VECES_MECANICA', 'ETICA', 'FOMACION_HUMANISTICA',
       'VECES_FORMACION_HUMANISTICA', 'EXPRESION_ORAL', 'VECES_EXPRESION_ORAL',
       'TEORIA_FILOSOFIA', 'VECES_TEORIA_FILOSOFIA', 'METODOLOGIA',
       'VECES_METODOLOGIA', 'SEMINARIO_I', 'SEMINARIO_II', 'SEMINARIO_III',
       'basicas', 'veces_basicas', 'ingenieria', 'veces_aplicados',
       'INGENIERIA AGRONOMICA', 'INGENIERIA AMBIENTAL Y SANITARIA',
       'INGENIERIA CIVIL', 'INGENIERIA DE SISTEMAS ',
       'INGENIERIA ELECTRONICA ', 'INGENIERIA INDUSTRIAL ', 'NUEVO',
       'TALENTO MAGDALENA', 'TRASLADO', 'Ninguno', 'TALENTO MAGDALENA.1'],
      dtype='object')
features with coefficients shrank to zero: 11
```

Figura 13 Resultado de la selección de variables obtenidas con Lasso

Lasso seleccionó 55 variables predictoras de las 87 que se tenían, y de las variables eliminadas 11 de ellas fueron reducidas colocando sus coeficientes en cero.

En el caso del Random forest obtuvimos un ranking de las variables con grados de importancia entre 0.23 y 0.0001, y 9 variables con valores de 0 en el ranking. La Figura 14 se ilustra con un diagrama de barras el ranking de las variables predictoras obtenido de acuerdo a su grado de importancia.

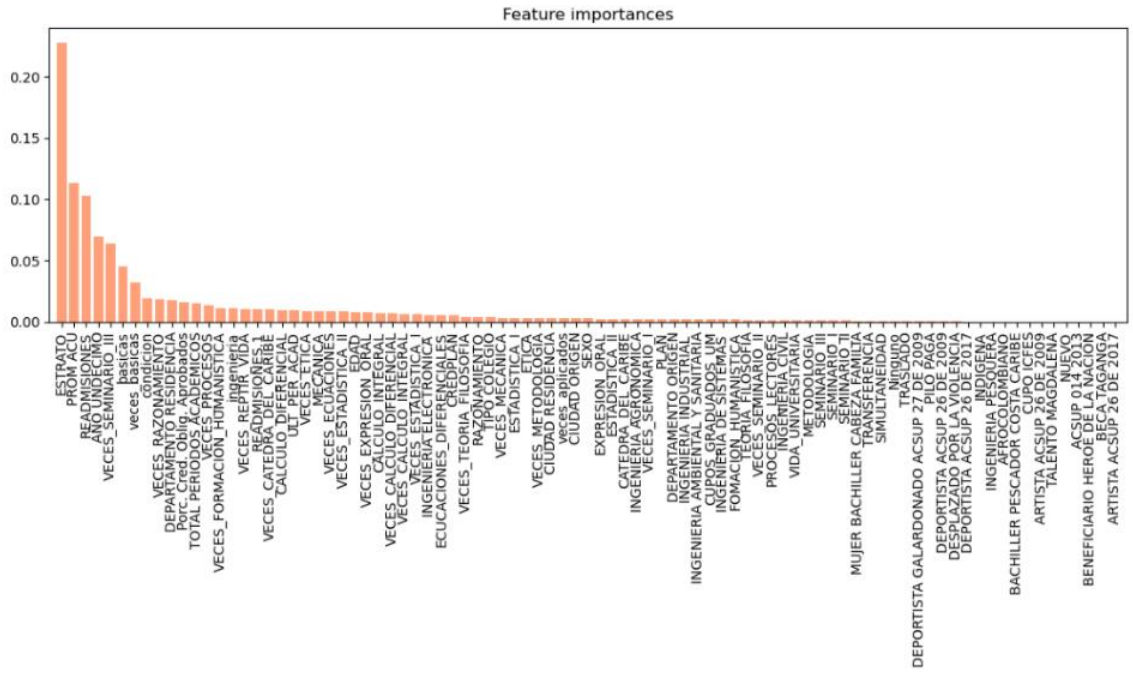


Figura 14 Ranking de las variables predictoras de acuerdo al método Random forest embebido

De acuerdo a los resultados obtenidos, ilustrados en la Figura 15 entre las 55 primeras variables en el ranking de importancia 45 concuerdan con las 55 características seleccionados por Lasso. Las variables coincidentes están listadas en el Anexo 1.

1.4 Método de Aprendizaje de Maquinas

Luego de limpiar los datos, y seleccionar características, se aplicaron métodos de aprendizaje de máquinas no supervisados y supervisados utilizando la librería de Python de código libre *Scikit-Learn*, en su versión 0.24. A continuación se presentarán los resultados obtenidos con ambos tipos de métodos.

1.4.1 Aprendizaje No Supervisado

Se utilizaron diferentes métodos de agrupamientos con el objetivo de observar si existían patrones de similitud que pudiera separar la población de estudio en diferentes grupos de estudiantes. Para la medición de la calidad de las particiones de los métodos se utilizaron como métricas *silhoutte_score*, *homogeneity*, y *completeness*.

La métrica *silhoutte_score* interpreta y valida la coherencia dentro del análisis de grupos.se calcula con base en la distancia media entre grupos y la distancia de los elementos más cercanos entre grupos. Esta métrica toma valores entre 1 y -1, siendo 1 el mejor resultado y -1 el peor. un valor de 0 o cercano de 0 significa que hay solapamiento entre los grupos,

y un valor negativo significa que algunos casos del conjunto de datos fueron asignados al grupo equivocado o que existe un grupo diferente al asignado con el cual tiene más similitud (Rousseuw,1987).

La métrica *homogeneity* es una medida de la homogeneidad de los datos que caen dentro de cada grupo, esta métrica dará un mejor resultado si los datos agrupados en cada grupo coinciden al pertenecer realmente a una misma clase. Esta métrica produce un valor entre 0 y 1, 0 significa que no hay homogeneidad en los grupos, es decir los grupos incluyen elementos de diferentes clases, y el valor de 1 significa que los casos agrupados en cada partición pertenecen todos a una misma clase.

Por su parte, *completeness* mide la completitud de las particiones generadas por el método de agrupamiento, si todos los casos que pertenecen a una misma clase caen dentro de un mismo grupo, entonces la métrica tendrá su máximo valor, de igual manera que con la métrica anterior el mejor valor que toma la métrica es 1, y 0 es el peor valor. Tanto la métrica de homogeneidad como la de completitud requieren un *ground-truth* o unos valores de referencia para comparar el grupo o clase real de cada caso, con el grupo asignado por el método de aprendizaje.

En este trabajo se aplicaron los siguientes métodos:

- Método K-*Means*. Este método recibe como parámetro el número de grupos que se desea encontrar. En este trabajo se evaluó un conjunto de valores para este parámetro que se puede observar en la Figura 15. También se puede observar los valores de las métricas estimadas con cada valor evaluado con los conjuntos de datos balanceados con *over-sampling* y *under-sampling*, y las 45 variables coincidentes como relevantes generadas en el análisis realizado en la Sección 1.2.5.

```
For n_clusters = 2 , silhouette score: 0.374, homogeneity: 0.012, Completeness: 0.012
For n_clusters = 6 , silhouette score: 0.416, homogeneity: 0.031, Completeness: 0.013
For n_clusters = 10 , silhouette score: 0.462, homogeneity: 0.045, Completeness: 0.015
For n_clusters = 20 , silhouette score: 0.578, homogeneity: 0.073, Completeness: 0.019
For n_clusters = 30 , silhouette score: 0.636, homogeneity: 0.081, Completeness: 0.018
For n_clusters = 40 , silhouette score: 0.667, homogeneity: 0.089, Completeness: 0.019
For n_clusters = 50 , silhouette score: 0.617, homogeneity: 0.100, Completeness: 0.020
For n_clusters = 60 , silhouette score: 0.615, homogeneity: 0.107, Completeness: 0.021
For n_clusters = 70 , silhouette score: 0.631, homogeneity: 0.115, Completeness: 0.022
For n_clusters = 80 , silhouette score: 0.632, homogeneity: 0.128, Completeness: 0.023
For n_clusters = 90 , silhouette score: 0.605, homogeneity: 0.128, Completeness: 0.023
For n_clusters = 100 , silhouette score: 0.598, homogeneity: 0.137, Completeness: 0.024
```

a) Resultados del método K-*Means* sobre el conjunto de datos con over-sampling y 45 variables

```
For n_clusters = 2 , silhouette score: 0.382, homogeneity: 0.020, Completeness: 0.020
For n_clusters = 6 , silhouette score: 0.446, homogeneity: 0.047, Completeness: 0.020
For n_clusters = 10 , silhouette score: 0.521, homogeneity: 0.064, Completeness: 0.020
For n_clusters = 20 , silhouette score: 0.620, homogeneity: 0.078, Completeness: 0.020
For n_clusters = 30 , silhouette score: 0.686, homogeneity: 0.088, Completeness: 0.020
For n_clusters = 40 , silhouette score: 0.737, homogeneity: 0.104, Completeness: 0.023
For n_clusters = 50 , silhouette score: 0.758, homogeneity: 0.097, Completeness: 0.021
For n_clusters = 60 , silhouette score: 0.718, homogeneity: 0.109, Completeness: 0.022
For n_clusters = 70 , silhouette score: 0.675, homogeneity: 0.115, Completeness: 0.022
For n_clusters = 80 , silhouette score: 0.648, homogeneity: 0.129, Completeness: 0.023
For n_clusters = 90 , silhouette score: 0.653, homogeneity: 0.132, Completeness: 0.024
For n_clusters = 100 , silhouette score: 0.637, homogeneity: 0.136, Completeness: 0.024
```

b) Resultados del método K-*Means* sobre el conjunto de datos con under-sampling y 45 variables

Figura 15 Resultados del agrupamiento utilizando el método K-*Means*

- DBSCAN. Este método trata de encontrar grupos como áreas altas densidad separadas por áreas de baja densidad. Este método requiere dos parámetros esenciales, el radio que define un vecindario, llamado *eps*, y el número de casos o ejemplos del conjunto de datos que deben caer dentro de cada vecindario de puntos definido por *eps* para considerarse realmente como un vecindario. En este trabajo utilizamos para *eps* el valor por defecto que viene en la implementación del método en la librería *Sickit-Learn* de Python, y el número de vecinos lo variamos como lo muestra la Figura 16

```
For min_samples = 5 , silhouette score: 0.637, homogeneity: 0.043, Completeness: 0.025
For min_samples = 10 , silhouette score: 0.637, homogeneity: 0.043, Completeness: 0.025
For min_samples = 15 , silhouette score: 0.637, homogeneity: 0.043, Completeness: 0.025
For min_samples = 20 , silhouette score: 0.637, homogeneity: 0.043, Completeness: 0.025
For min_samples = 35 , silhouette score: 0.637, homogeneity: 0.043, Completeness: 0.025
For min_samples = 40 , silhouette score: 0.637, homogeneity: 0.042, Completeness: 0.025
For min_samples = 45 , silhouette score: 0.637, homogeneity: 0.042, Completeness: 0.025
For min_samples = 50 , silhouette score: 0.637, homogeneity: 0.042, Completeness: 0.025
```

a) Resultados del método BDSCAN sobre el conjunto de datos con *over-sampling* de 45 y 45 variables

```
For min_samples = 5 , silhouette score: 0.651, homogeneity: 0.043, Completeness: 0.025
For min_samples = 10 , silhouette score: 0.651, homogeneity: 0.043, Completeness: 0.025
For min_samples = 15 , silhouette score: 0.651, homogeneity: 0.043, Completeness: 0.025
For min_samples = 20 , silhouette score: 0.651, homogeneity: 0.043, Completeness: 0.025
For min_samples = 35 , silhouette score: 0.651, homogeneity: 0.043, Completeness: 0.025
For min_samples = 40 , silhouette score: 0.651, homogeneity: 0.042, Completeness: 0.025
For min_samples = 45 , silhouette score: 0.651, homogeneity: 0.042, Completeness: 0.025
For min_samples = 50 , silhouette score: 0.651, homogeneity: 0.042, Completeness: 0.025
```

b) Resultados del método BDSCAM sobre el conjunto de datos con *under-sampling* y 45 variables

Figura 16 Resultados del agrupamiento utilizando el método BDSCAM

- Agrupamiento aglomerativo. Es un algoritmo que crea una jerarquía al momento de realizar las agrupaciones utilizado un enfoque de abajo hacia arriba, como se describe en la Sesión 1.2.2. El método implementado en *Sickit-Learn* requiere como parámetro el número de grupos o clústeres que se quiere encontrar. En este trabajo evaluamos el método con los mismos valores para número de grupos que se utilizaron con el método K-*Means*.

```
For n_clusters = 2 , silhouette score: 0.375, homogeneity: 0.030, Completeness: 0.038
For n_clusters = 6 , silhouette score: 0.425, homogeneity: 0.041, Completeness: 0.018
For n_clusters = 10 , silhouette score: 0.467, homogeneity: 0.049, Completeness: 0.016
For n_clusters = 20 , silhouette score: 0.606, homogeneity: 0.077, Completeness: 0.019
For n_clusters = 30 , silhouette score: 0.700, homogeneity: 0.083, Completeness: 0.019
For n_clusters = 40 , silhouette score: 0.745, homogeneity: 0.089, Completeness: 0.019
For n_clusters = 50 , silhouette score: 0.754, homogeneity: 0.101, Completeness: 0.021
For n_clusters = 60 , silhouette score: 0.769, homogeneity: 0.105, Completeness: 0.022
For n_clusters = 70 , silhouette score: 0.677, homogeneity: 0.111, Completeness: 0.021
For n_clusters = 80 , silhouette score: 0.677, homogeneity: 0.113, Completeness: 0.021
For n_clusters = 90 , silhouette score: 0.660, homogeneity: 0.126, Completeness: 0.023
For n_clusters = 100 , silhouette score: 0.651, homogeneity: 0.131, Completeness: 0.023
```

a) Resultados del método Agglomerative sobre el conjunto de datos con *over-sampling* y 45 variables

```
For n_clusters = 2 , silhouette score: 0.375, homogeneity: 0.030, Completeness: 0.038
For n_clusters = 6 , silhouette score: 0.425, homogeneity: 0.041, Completeness: 0.018
For n_clusters = 10 , silhouette score: 0.467, homogeneity: 0.049, Completeness: 0.016
For n_clusters = 20 , silhouette score: 0.606, homogeneity: 0.077, Completeness: 0.019
For n_clusters = 30 , silhouette score: 0.700, homogeneity: 0.083, Completeness: 0.019
For n_clusters = 40 , silhouette score: 0.745, homogeneity: 0.089, Completeness: 0.019
For n_clusters = 50 , silhouette score: 0.754, homogeneity: 0.101, Completeness: 0.021
For n_clusters = 60 , silhouette score: 0.769, homogeneity: 0.105, Completeness: 0.022
For n_clusters = 70 , silhouette score: 0.677, homogeneity: 0.111, Completeness: 0.021
For n_clusters = 80 , silhouette score: 0.677, homogeneity: 0.113, Completeness: 0.021
For n_clusters = 90 , silhouette score: 0.660, homogeneity: 0.126, Completeness: 0.023
For n_clusters = 100 , silhouette score: 0.651, homogeneity: 0.131, Completeness: 0.023
```

b) Resultados del método Agglomerative sobre el conjunto de datos con *under-sampling* y 45 variables

Figura 17 Resultados del agrupamiento utilizando el método DBSCAM

- Análisis de los resultados. Como se puede observar en las Figuras 16-18 los valores de las métricas indican que ninguno de los métodos encontró una buena partición de los datos. Para los tres métodos ensayados la métrica *silhoutte_score* tendió a mejorar de una manera incremental a medida que se aumentan el número de grupos obtenidos. Sin embargo, de acuerdo a las métricas de homogeneidad y completitud las particiones generadas con muchos grupos no mejoran mucho en cuanto a la homogeneidad y la completitud. No podemos concluir que haya patrones de similitud que se puedan identificar de los valores que toman las variables predictoras para cada caso.

1.4.2 Aprendizaje Supervisado

A pesar de que los métodos de agrupamientos ensayados en la sección anterior no mostraron patrones de similitud para identificar grupos de estudiantes homogéneos que compartieran algunas características en común, esos resultados no son concluyentes para descartar que a partir de las variables independientes podemos predecir la condición que hemos definido para cada caso de estudiante. Por lo tanto, en esta sección se describe la implementación de diferentes métodos de aprendizaje supervisado para clasificación.

De igual manera que en la sección anterior se utilizaron métodos implementados en la librería *Scikit-Learn* de Python, tales como *k-nearest neighbors*, random forest, perceptron multicapas, máquina de vector de soporte, XGBoost, estos métodos fueron descritos anteriormente en la Sección 1.2.1 aprendizaje supervisado. Para medir y comparar el desempeño de cada uno de estos métodos utilizamos las métricas: *accuracy* la cual mide el porcentaje de casos que el modelo ha acertado, *recall* que nos da la cantidad de información que cada modelo es capaz de identificar, y *precision* que no es más que la precisión en dar la respuesta correcta y nos ayuda a medir la calidad de los modelos (Martinez,2020). Además, también analizamos las matrices de confusión de cada método para observar el desempeño de cada predictor con respecto a cada clase.

Cada método fue evaluado sobre un conjunto de datos reservados para prueba equivalente al 30% del conjunto de datos. Los parámetros requeridos por cada uno de los métodos *fueron ajustados utilizando los procedimientos de búsqueda Grid-Search y Random-*Search, ambos implementados en la librería *Sickit-Learn*, ambos métodos permiten evaluar los clasificadores utilizando validación cruzados con 10 bloques. A continuación, se mostrarán los resultados obtenidos con cada método implementado.

- *k-Nearest Neighbor*. Es un algoritmo de propósito general bástate utilizado, En este trabajo fue implementado para clasificación, utilizando la clase *KNeighborsClassifier* de la librería *Sickit-learn*. El parámetro más importante de definir para este método es el número de vecinos a considerar por cada caso del conjunto de datos. con los vecinos más cercanos se estiman las etiquetas que se le dan a los datos.Para este método se consideró un conjunto de 10 valores diferentes que van desde 10 hasta 100. También se evaluaron diferentes valores para el parámetro *weights* siendo estos '*uniform*' que significa que cada uno de los puntos del vecindario se pondera por igual, y por otro lado el parámetro puede tomar el valor '*distance*' que significa que los vecinos más cercanos tienen una mayor influencia que los vecinos más lejanos. El parámetro *algorithm* que implica el tipo de algoritmo utilizado para calcular los vecinos más cercanos presenta las opciones: '*auto*', '*ball tree*', '*kd-tree*', y '*brute*', las cuales también fueron evaluadas. La Tabla 6 muestra un resumen de los resultados obtenidos por el *KNeighborsClassifier* sobre los dos conjuntos de datos generados con *over-sampling*, y *under-sampling* en el proceso de balanceo del conjunto de datos, en la Sección 1.2.4, y teniendo en cuenta la selección de variables realizada en la Sección 1.2.5 donde se aplicaron los métodos Lasso y *random forest* obteniendo como resultados 55 variables relevantes para el caso de Lasso, de las cuales el *random forest* coincidió en 45 de ellas. Adicionalmente, la Figura 18 muestra los valores para todos los parámetros que fueron evaluados. Con la función *GridSearchCV* se obtuvo como resultado el conjunto de parámetros que mejor resultado arrojó para el método basado en la métrica *precision_score* para cada una de las pruebas. Como se observa en la figura 18, en la cual se encuentran el conjunto de datos generado con *over-sampling* y con 45 variables.

```
[22] import numpy as np

     from sklearn.model_selection import GridSearchCV, RandomizedSearchCV
     from sklearn.neighbors import KNeighborsClassifier
     from sklearn.metrics import accuracy_score, precision_score, recall_score, confusion_matrix

     n_neighbors = [10,20,30,40,50,60,70,80,90,100]
     weights = ['uniform', 'distance']
     algorithm  = ["auto","ball_tree","kd_tree","brute"]
     p  = [1,2]

     random_grid = {'n_neighbors': n_neighbors,
                    'weights': weights,
                    'algorithm': algorithm,
                    'p': p}

     knn_clf = KNeighborsClassifier()
     knn_random = GridSearchCV(estimator = knn_clf, param_grid = random_grid, cv = 10, verbose=1,scoring='precision')
     knn_random.fit(X_train, y_train)
     predictions = knn_random.predict(X_test)

     print("accuracy score: %0.3f, precision score: %0.3f, recall score: %0.3f" %(accuracy_score(y_test, predictions),precision_score(y_test, predictions),recall_score(y_t
     print("best parameters:",knn_random.best_params_ )
     print("confusion matrix: ", confusion_matrix(y_test, predictions))

     Fitting 10 folds for each of 160 candidates, totalling 1600 fits
     [Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
     [Parallel(n_jobs=1)]: Done 1600 out of 1600 | elapsed:  3.8min finished
     accuracy score: 0.801, precision score: 0.790, recall score: 0.857
     best parameters: {'algorithm': 'auto', 'n_neighbors': 10, 'p': 1, 'weights': 'uniform'}
     confusion matrix:  [[ 782  281]
      [ 177 1060]]
```

Figura 18 Ejemplo ejecución del método *k-nearest neighobors* sobre el conjunto de datos generado con *over-sampling* y 45 variables

Tabla 6 Resultados de *K-nearest neighbhors* sobre los conjuntos de datos generados

Conjunto de datos/Métricas	*Accuracy*	*Precision*	*Recall*	Matriz de confusión
Over-sampling, 45 variables	0.8	0.79	0.85	[782, 281] [177, 1060]
Over-sampling, 55 variables	0.79	0.79	0.83	[788, 275] [204, 1022]
Under-sampling, 45 variables	0.78	0.79	0.81	[567, 194] [171, 738]
Under-sampling, 55 variables	0.973	0.954	0.998	[1003, 60] [2, 1235]

- *Random Forest*. Este es un método que usa varios árboles clasificadores que toman la decisión en varias submuestras del conjunto de datos. En este proyecto fue implementado utilizando la clase *RandomForestClassifier* de *Sickit-learn*. Uno de los parámetros con mayor importancia para ajustar en este método es el número de árboles a generar (*n_estimators*). Para este parámetro se evaluaron valores en el rango de 5 a 55, con incrementos de 5. Además, se ajustaron los parámetros *criterion* que toma valores como '*gini*' y '*entropy*' para determinar la función criterio para la partición de cada nodo al conformar los árboles de decisión, y el parámetro *max_depth* que especifica la profundidad máxima de cada árbol, este último parámetro fue evaluado en el rango (8,12). La Figura 19 muestra un ejemplo de implementación de este método y los parámetros que se ajustaron; y por otro lado podemos observar los resultados obtenidos sobre cada uno de los 4 subconjuntos de datos en la Tabla 7.

```
from sklearn.model_selection import RandomizedSearchCV, GridSearchCV
from sklearn.ensemble import RandomForestClassifier
from scipy.stats import uniform, truncnorm, randint
from sklearn.metrics import accuracy_score, precision_score, recall_score, confusion_matrix
n_neighbors = [5,10,15,20,25,30,35,40,45,50,55]

model_params = {

    'n_estimators': n_neighbors,
    'criterion': ['gini','entropy'],
    'max_depth' : [8, 9, 10, 11, 12],

}

rf_model = RandomForestClassifier()

clf = GridSearchCV(rf_model, model_params,  cv=10,verbose=1, scoring='precision')

model = clf.fit(X, y)

predictions = clf.predict(X_test)
print("accuracy score: %0.3f, precision score: %0.3f, recall score: %0.3f" %(accuracy_score(y_test, predictions),precision_score(y_test, predictions),recall_score(y_t
print("best parameters:",clf.best_params_ )
print("confusion matrix: ", confusion_matrix(y_test, predictions))

Fitting 10 folds for each of 110 candidates, totalling 1100 fits
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 1100 out of 1100 | elapsed:  5.1min finished
accuracy score: 0.963, precision score: 0.940, recall score: 0.995
best parameters: {'criterion': 'gini', 'max_depth': 9, 'n_estimators': 55}
confusion matrix:  [[ 984   79]
 [   6 1231]]
```

Figura 19 Ejemplo ejecución del método r*andom forest* sobre el conjunto de datos generado con *over-sampling* y 45 variables

Tabla 7 Resultados del Random Forest sobre los conjuntos de datos generados

Conjunto de datos/Métricas	*Accuracy*	*Precision*	*Recall*	Matriz de confusión
Over-sampling, 45 variables	0.963	0.940	0.995	[984, 79] [6, 1231]
Over-sampling, 55 variables	0.97	0.951	0.994	[1000, 63] [7, 1230]
Under-sampling, 45 variables	0.976	0.962	0.996	[725, 36] [4, 905]
Under-sampling, 55 variables	0.958	0.932	0.996	[695, 66] [4, 905]

- Perceptrón Multicapas Este método consiste básicamente en ajustar un conjunto de pesos para que el modelo obtenga como resultado la salida esperada al ser introducido cada ejemplo del conjunto de datos de entrenamiento. En este proyecto fue implementado utilizando la clase *MLPClassifier* de la librería *Sickit-Learn*. Entre los parámetros de configuración más importante para este método se encuentran el número de capas (*hidden_layer_sizes*), y el número de neuronas por capa, en nuestro caso se probaron estas configuraciones para un modelo de dos capas (10, 10), (20, 20), (30, 20), (40, 30), (50, 40). También se ajustaron la tasa de aprendizaje la cual fue variada de 0.0001 a 0.05 con incrementos de 0.005, y el máximo número de interacciones del método, para el cual se tuvo en cuenta un rango de 100 a 500, con incrementos de 100. La Figura 20 muestra las configuraciones de estos parámetros que fueron ensayadas, y los resultados obtenidos al ejecutar el método sobre uno de los conjuntos de datos considerados en este trabajo. También, la Tabla 8 resume los mejores resultados obtenidos de las pruebas realizadas con los distintos conjuntos de datos.

```
from sklearn.neural_network import MLPClassifier

from sklearn.model_selection import RandomizedSearchCV
mlp = MLPClassifier(max_iter=100)

parameter_space = {
    'hidden_layer_sizes': [(10, 10), (20, 20), (30, 20), (40, 30), (50, 40)],
    'learning_rate_init':[0.0001, 0.001, 0.01, 0.0005, 0.005, 0.05],
    'max_iter':[100, 200, 300, 400, 500]
}

from sklearn.model_selection import GridSearchCV

clf = RandomizedSearchCV(mlp, parameter_space,  cv=10,verbose=1, scoring='precision')
clf.fit(X_train,y_train)
predictions = clf.predict(X_test)

print("accuracy score: %0.3f, precision score: %0.3f, recall score: %0.3f" %(accuracy_score(y_test, predictions),precision_score(y_test, predictions),recall_score(y_test, predictior
print("best parameters:",clf.best_params_ )
print("confusion matrix: ", confusion_matrix(y_test, predictions))
```

Figura 20 Ejemplo ejecución del método perceptrón multicapas sobre el conjunto de datos generado con *over-sampling* y 45 variables

Tabla 8 Resultados del perceptrón multicapas sobre los conjuntos de datos generados

Conjunto de datos/Métricas	*Accuracy*	*Precision*	*Recall*	Matriz de confusión
Over-sampling, 45 variables	0.802	0.878	0.733	[937,126] [330, 907]
Over-sampling, 55 variables	0.782	0.736	0.928	[651, 412] [89, 1148]
Under-sampling, 45 variables	0.752	0.707	0.928	[412, 349] [65, 844]
Under-sampling, 55 variables	0.798	0.833	0.787	[618, 143] [194, 715]

- *Support Vector Machine.* Este método construye un hiperplano o conjunto de hiperplanos en un espacio de dimensionalidad muy alta (o incluso infinita) que puede ser utilizado en problemas de clasificación o regresión, tal y como se describe en la Sección 1.2.1 del presente documento. Para su implementación se utilizó la clase SVM de la librería *Sickit-Learn*. y se configuró el parámetro *Kernel* siendo esta la especificación del tipo de función de interpolación que utiliza el método internamente para separar los datos en clases. Este parámetro se varió con los valores disponibles en la implementación de la librería *Sickit-Learn*, que son *rbf, linear* y *poly*. Como se observa en la Figura 21.

```
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.datasets import load_breast_cancer
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, precision_score, recall_score, confusion_matrix

from sklearn.model_selection import GridSearchCV

param_grid = {
              'kernel': ['rbf','linear', 'poly']}

grid = GridSearchCV(SVC(), param_grid,verbose = 1,cv=10,scoring='precision')

grid.fit(X_train, y_train)

grid_predictions = grid.predict(X_test)
predictions = grid.predict(X_test)

print("accuracy score: %0.3f, precision score: %0.3f, recall score: %0.3f" %(accuracy_score(y_test, predictions),precision_score(y_test, predictions),recall_score(y_test, predictions)))
print("best parameters:",grid.best_params_ )
print("confusion matrix: ", confusion_matrix(y_test, predictions))
```

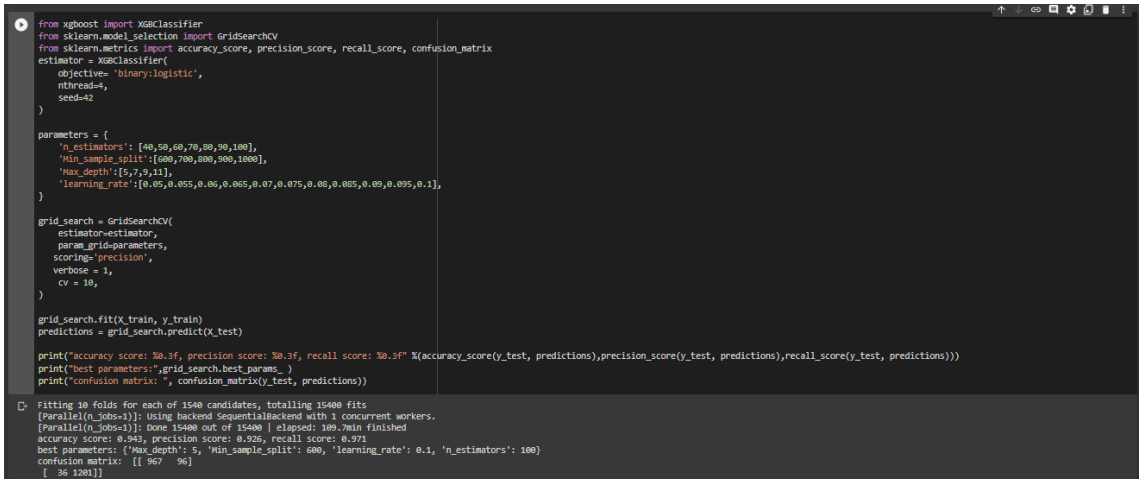
```
Fitting 10 folds for each of 3 candidates, totalling 30 fits
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  30 out of  30 | elapsed: 159.9min finished
accuracy score: 0.887, precision score: 0.878, recall score: 0.916
best parameters: {'kernel': 'linear'}
confusion matrix:  [[ 906  157]
 [ 104 1133]]
```

Figura 21 Ejemplo ejecución de máquinas de soporte sobre el conjunto de datos generado con *over-sampling* y 45 variables

Tabla 9 Resultados de las máquinas de vector de soporte sobre los conjuntos de datos generados

Conjunto de datos/Métricas	*Accuracy*	*Precision*	*Recall*	Matriz de confusión
Over-sampling, 45 variables	0.887	0.878	0.916	[906, 157] [104, 1133]
Over-sampling, 55 variables	0.880	0.883	0.896	[916, 147] [129,1108]
Under-sampling, 45 variables	0.884	0.887	0.902	[657,104] [89, 820]
Under-sampling, 55 variables	0.877	0.884	0.890	[655, 106] [100,809]

- XGBOOST. Este método implementa arboles de decisión impulsados por gradiente como se encuentra descrito en la Sección 1.2.1 del presente trabajo. El método se implementó mediante la clase *XGBClassifier* de la librería *Sickit-Learn* de Python. Para la realización de las pruebas se utilizaron la configuración de las viables a continuación mencionadas, número de árboles (*n_estimators*) en un rango de 40 a 100 con aumentos de 10, así mismo la tasa de aprendizaje variada en 0.05 y 0.1 en escala de 0.05 , además de del número mínimo de muestras (*Min_sample_split*) que se requieren en cada nodo estando configurado en escalas de 100 que va desde 600 a 1000, igualmente la profundidad máxima del árbol (*Max_depth*) como se presenta en la Figura 22.



```
from xgboost import XGBClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import accuracy_score, precision_score, recall_score, confusion_matrix
estimator = XGBClassifier(
    objective= 'binary:logistic',
    nthread=4,
    seed=42
)

parameters = {
    'n_estimators': [40,50,60,70,80,90,100],
    'Min_sample_split':[600,700,800,900,1000],
    'Max_depth':[5,7,9,11],
    'learning_rate':[0.05,0.055,0.06,0.065,0.07,0.075,0.08,0.085,0.09,0.095,0.1],
}

grid_search = GridSearchCV(
    estimator=estimator,
    param_grid=parameters,
   scoring='precision',
   verbose = 1,
    cv = 10,
)

grid_search.fit(X_train, y_train)
predictions = grid_search.predict(X_test)

print("accuracy score: %0.3f, precision score: %0.3f, recall score: %0.3f" %(accuracy_score(y_test, predictions),precision_score(y_test, predictions),recall_score(y_test, predictions)))
print("best parameters:",grid_search.best_params_ )
print("confusion matrix: ", confusion_matrix(y_test, predictions))
```

```
Fitting 10 folds for each of 1540 candidates, totalling 15400 fits
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 15400 out of 15400 | elapsed: 109.7min finished
accuracy score: 0.943, precision score: 0.926, recall score: 0.971
best parameters: {'Max_depth': 5, 'Min_sample_split': 600, 'learning_rate': 0.1, 'n_estimators': 100}
confusion matrix:  [[ 967   96]
 [  36 1201]]
```

Figura 22 Ejemplo ejecución del método *XGBClassifier* sobre el conjunto de datos generado con *over-sampling* y 45 variables

Tabla 10 Resultados del XGBClassifier sobre los conjuntos de datos generados

Conjunto de datos/Métricas	*Accuracy*	*Precision*	*Recall*	Matriz de confusión
Over-sampling, 45 variables	0.943	0.926	0.971	[967, 96] [36, 1201]
Over-sampling, 55 variables	0.943	0.926	0.971	[967,96] [36, 1201]
Under-sampling, 45 variables	0.935	0.915	0.971	[679,82] [26, 883]
Under-sampling, 55 variables	0.935	0.915	0.970	[679, 82] [27, 882]

- Análisis de resultados. Luego de las pruebas realizadas con cada uno de los métodos mencionados en la Sección 1.2.1, el análisis expuesto nos indica que los mejores resultados para clasificar casos de estudiantes en riesgo de deserción se obtuvieron con los métodos *random forest*, y el *k-nearest neighbors* quienes alcanzaron porcentajes de precisión superior al 95%. Al observar las matrices de confusión de ambos métodos tienen mayor precisión al identificar la clase de los estudiantes que terminarían sus estudios exitosamente. Por otro lado, el peor desempeño los obtuvo el perceptrón multicapas y el método de máquinas de vector de soporte, con precisiones no mayores al 88%.También se puede observar que el número de variables exceptuando el método de perceptrón multicapas, no presentó mayores diferencias en los resultados obtenidos con los demás métodos; es decir, no se observan grandes diferencias, por ejemplo, al utilizar los conjuntos de datos generado con *under-sampling* y 45 variables, y el conjunto de datos *under-Sampling* y 55 variables al entrenar un modelo de clasificación y evaluar su desempeño. La Tabla 11 resume la comparación de los diferentes métodos utilizados en este estudio entre sí. Cada fila de la tabla corresponde a los mejores resultados obtenidos con cada método, extraído de las Tablas 6-10.

Tabla 11 . Mejores resultados de cada uno de los métodos

Método	Conjunto de dato	*Accuracy*	*Precision*	*Recall*	Matriz de confusión
Random Forest.	*Under-sampling*, 45 variables	0.976	0.962	0.996	[725, 36] [4, 905]
K-Nearest Neighbors	*Under-sampling*, 55 variables	0.973	0.954	0.998	[567, 194] [171, 738]
XGBOOST	*Over-sampling*, 55 variables	0.943	0.926	0.971	[967,96] [36, 1201]
Support vector machine	*Over-sampling*, 55 variables	0.880	0.883	0.896	[916, 147] [129,1108]
Perceptron multicapas.	*Over-sampling*, 45 variables	0.802	0.878	0.733	[937,126] [330, 907]

5. Conclusiones

Para el desarrollo del presente proyecto de investigación se tuvo como motivación el gran número de estudiantes en deserción pertenecientes a la Facultad de Ingeniería de la Universidad del Magdalena, cuya tasa de deserción es incluso más alta que el promedio de tasas a nivel nacional. Este problema de la deserción estudiantil se presenta con mayor frecuencia en los primeros semestres de los programas que conforman la Facultad. Se tomó la decisión de aportar en la solución a esta problemática con la utilización de técnicas de aprendizaje de máquinas y minería de datos que con base a trabajos encontrados en la literatura anteriormente realizados mostraban predecir de manera temprana el riesgo de un estudiante a desertar. En la revisión de literatura que se realizó se encontró que las variables más utilizadas para este tipo de predicción son especialmente de tipo demográficas y académicas de los estudiantes.

Teniendo en cuenta lo anterior se recurrió a consultar el sistema de admisiones y registros de la Universidad (AyRe) como la fuente principal de ambos tipos de información tanto socio-demográficas, como académica de los estudiantes de la Facultad de Ingeniería. Debido a la cantidad de datos que se encuentra en el sistema de información AyRe se definió una muestra comprendida en una ventana de tiempo comprendida entre el año 2014 y el primer semestre del 2020 que fue el momento en el que se hizo la consulta. Los registros de estudiantes recopilados contenían información de estudiantes de la Facultad de Ingeniería, principalmente sobre la modalidad de ingreso, el programa académico del estudiante, el total de periodos académicos, algunas variables socio-demográficas entre las cuales se encuentra estrato, sexo, tipo de colegio, y variables pertenecientes al historial académico del estudiante, tales como las calificaciones cuantitativas de las asignaturas agrupadas por componentes de cada una de las ingenierías, así como también las veces que se repitieron asignaturas. Un listado completo de estas variables se puede consultar en la Tabla 3.

Como conclusión de esta primera etapa del desarrollo de este trabajo cabe resaltar que, para futuros proyectos de minerías de datos, o big data sobre el sistema de información de los estudiantes de la Universidad del Magdalena, se debe estructurar el sistema de tal forma que minimice los pasos de integración y limpieza que son requeridos para poder explotar la información allí registrada. Es sabido que en la actualidad se apunta en esta dirección, uno de los bienes más preciados de cualquier compañía, empresa o institución son los datos, si estos son valorados y explotados para la toma de decisiones, se pueden obtener mejoras significativas en el diseño de estrategias que permita cada vez perfeccionar y enriquecer los objetivos de dicha entidad.

Los registros recopilados con información de estudiantes de la Facultad de Ingeniería fueron agrupados como estudiantes en situación normal, estudiantes en condición de F.B.R.A., estudiantes que terminaron académicamente, estudiantes en situación de cuarta vez, y graduados (ver Figura 10). Estos grupos fueron posteriormente reagrupados en dos categorías: estudiantes en riesgo de deserción, y estudiantes con terminación exitosa como se puede observar en la Tabla 4. Esta categorización de los registros en dos clases nos permitió tratar el problema como un problema de clasificación binaria, y enfocarnos en el tratar de predecir cuándo un estudiante podría estar en la categoría de estar en riesgo de deserción.

En el desarrollo del presente proyecto se tuvieron en cuenta dos tipos de métodos de aprendizaje de máquinas, el no supervisado y supervisado. El primer tipo se ve reflejado en la implementación de algoritmos de agrupamiento (clustering), tales como método *k-Means*, DBSCAN, *spectral*. Por su parte, el aprendizaje supervisado se ve reflejado en la utilización de métodos de clasificación tales como los k vecinos más cercanos, redes neuronales, métodos de regresión logística, y arboles de decisión. Previamente a la aplicación de los métodos de aprendizaje de máquinas, especialmente los métodos supervisados se implementaron otras técnicas más, típicas de un proceso de minería de datos para la limpieza y preparación de los datos y para balancear el conjunto de datos recopilados, debido a que la mayoría de los registros pertenecían a estudiantes en condiciones normales, es decir en la segunda categoría definida en el trabajo, estudiantes que no estaban en riesgo de deserción sino que apuntaban más a una terminación exitosa. El balanceo de los datos se llevó a cabo mediante las técnicas de muestreo, *Over-Sampling* y *Under-Sampling* quedando dos conjuntos de datos los cuales pueden observar en la Figura 12. Seguidamente se realizó una selección de características relevantes con métodos de Lasso permitiendo pasar de 87 variables a 55 y el método de ranking *random forest* el cual pasa de 87 variables a 45 como se observa en la Figura 13 y Figura 14. De todo este proceso Para finalmente quedar con 4 grupos *Over-Sampling* para 55 y 45 variables, seguido de *Under-Sampling* para 45 y 55 variables.

De lo anterior en el transcurso de la aplicación de los distintos métodos de aprendizaje de máquinas evaluados, no se observó una diferencia significativa en los resultados de acuerdo al tipo de muestreo utilizado, esto fue aún más remarcable con los métodos de aprendizaje de máquinas no supervisado, para los cuales no hubo ningún tipo de diferencia en los resultados arrojados por cada método. Sin embargo, se debe mencionar que los métodos de aprendizaje no supervisado implementados, entre ellos k-*Means*, DBSCAN y agrupamiento aglomerativo, no generaron métricas de evaluación de agrupamiento que correspondieran o indicaran la existencia de patrones de concordancia entre los registros, o en otras palabras patrones de similitud entre los datos que permitiera la conformación de grupos de estudiantes con condiciones similares. Por ejemplo, la métrica *silhoutte,* obtuvo valores pobres no mayor al 40% para los resultados de los distintos métodos ensayados. De igual manera las métricas de

homogeneidad y completitud indicaron que los resultados de los métodos de agrupamiento no fueron los mejores.

Por otra parte, en el momento de emplear aprendizaje supervisado se seleccionaron métodos muy reconocidos en la literatura tales como k-*nearest neighbors*, *random forest*, percentrón multicapas, *Support Vector Machine* y *XGBOOST*. A diferencia del conjunto de métodos anterior, no supervisados, se obtuvieron muy buenos resultado logrando obtener porcentajes de precisión mayores al 96% en las mejores corridas de los métodos. Los métodos que arrojaron el mejor desempeño fueron *random forest* y *k-nearest neighbors*. Es decir, los resultados obtenidos muestran que es posible predecir el riesgo de un estudiante a desertar, teniendo en cuenta las variables de tipo socio-demográficas, y académicas que se lograron recopilar y estudiar en este proyecto.

En conclusión, los resultados obtenidos con los métodos de aprendizaje supervisado indican que con un alto grado de confianza se pueden utilizar los modelos de predicción obtenidos en este estudio para detectar casos de deserción de manera temprana. De esta manera, se podrá ofrecer al estudiante estrategias de mejoras de su condición, que permitan a la vez reducir la tasa de deserción en la Facultad de Ingeniería de la Universidad del Magdalena.

Como trabajo futuro de este trabajo se resalta que una mejor estructuración del sistema de información AyRe de la Universidad permitirá mejorar futuras investigaciones como la que se ha desarrollado, mejorando la calidad de los datos, disminuyendo el tiempo requerido para integración y limpieza de los datos, y por lo tanto reduciendo la probabilidad de incurrir en errores de tratamiento de los datos. Un nuevo diseño, y restructuración del sistema AyRe debe ir dirigido a soportar la aplicación de métodos de minería de datos para la toma de decisiones.

Además, se buscaría integrar todo el proceso que se realizó de minería de datos, con los sistemas de información de la Universidad del Magdalena, que manejen datos de estudiantes, por ejemplo, el sistema de información de Bienestar Universitario, que ofrece ayudas a los estudiantes en condiciones socioeconómicas especiales. También se podría integrar con sistemas de información de las oficinas de Desarrollo Estudiantil, y la de Relaciones Internacionales, etc. Siendo este el objetivo del proyecto original recopilar e integrar datos de los distintos sistemas de información de la universidad para hacer un despliegue de indicadores que permitan visualizar en cifras la labor que se realiza, y también poder utilizar toda esta información para proceso de minería de datos y toma de decisiones.

A medida que en las diferentes dependencias de la universidad se maduren en la recolección mantenimiento y administración de la información, será posible la integración de este proyecto con otros sistemas de información adicionales al sistema de admisiones, lo cual permitirá involucrar y explotar otras variables predictoras relacionadas con los estudiantes.

También, se prevé las construcciones de modelos de predicción del éxito, y no éxito de los estudiantes en las otras facultades de la universidad. Este proyecto se podrá integrar a un sistema de alertas tempranas de estudiantes en riesgo de deserción, y puede tomarse como base para la planeación de estrategias que ayuden a mitigar este flagelo en la Facultad de Ingeniería.

A. Anexo: Método de selección de características

	Métodos de selección de características	
	Least Absolute Shrinkage	*Random Forest Importance*
Nombre de la variable	ESTRATO READMISIONES PROM ACU AÑO UNDECIMO VECES_SEMINARIO_III basicas veces_basicas Porc. Cred. Oblig. Aprobados VECES_RAZONAMIENTO TOTAL PERIODOS ACADEMICOS DEPARTAMENTO RESIDENCIA VECES_PROCESOS VECES_FORMACION_HUMANISTICA ingenieria CALCULO_DIFERENCIAL VECES_CATEDRA_DEL_CARIBE VECES_REPTIR_VIDA	EDAD READMISIONES ULT_PER_ACAD CREDPLAN ESTRATO PROM ACU Porc. Cred. Oblig. Aprobados READMISIONES.1 PLAN TIPO COLEGIO SEXO CIUDAD ORIGEN CIUDAD RESIDENCIA DEPARTAMENTO ORIGEN DEPARTAMENTO RESIDENCIA AÑO UNDECIMO

	VECES_ETICA VECES_ECUACIONES READMISIONES.1 EDAD CALCULO_INTEGRAL VECES_EXPRESION_ORAL ULT_PER_ACAD VECES_ESTADISTICA_II VECES_CALCULO_DIFERENCIAL MECANICA VECES_CALCULO_INTEGRAL VECES_ESTADISTICA_I CREDPLAN INGENIERIA ELECTRONICA ECUCACIONES_DIFERENCIALES RAZONAMIENTO ETICA VECES_TEORIA_FILOSOFIA ESTADISTICA_I TIPO COLEGIO VECES_METODOLOGIA SEXO CIUDAD RESIDENCIA CIUDAD ORIGEN VECES_SEMINARIO_I ESTADISTICA_II	TOTAL PERIODOS ACADEMICOS PROCESOS_LECTORES RAZONAMIENTO VIDA_UNIVERSITARIA VECES_REPTIR_VIDA CALCULO_DIFERENCIAL VECES_CALCULO_DIFERENCIAL CALCULO_INTEGRAL VECES_CALCULO_INTEGRAL ECUCACIONES_DIFERENCIALES VECES_ECUACIONES VECES_ESTADISTICA_I ESTADISTICA_II VECES_ESTADISTICA_II MECANICA VECES_MECANICA ETICA FOMACION_HUMANISTICA VECES_FORMACION_HUMANISTICA CATEDRA_DEL_CARIBE VECES_CATEDRA_DEL_CARIBE TEORIA_FILOSOFIA VECES_TEORIA_FILOSOFI METODOLOGIA

	VECES_MECANICA PLAN DEPARTAMENTO ORIGEN TEORIA_FILOSOFIA CATEDRA_DEL_CARIBE CUPOS_GRADUADOS_UM INGENIERIA AMBIENTAL Y SANITARIA INGENIERIA CIVIL FOMACION_HUMANISTICA INGENIERIA AGRONOMICA EXPRESION_ORAL INGENIERIA DE SISTEMAS	VECES_METODOLOGIA Básicas veces_basicas Ingeniería veces_aplicados INGENIERIA AGRONOMICA INGENIERIA AMBIENTAL Y SANITARIA INGENIERIA CIVIL INGENIERIA ELECTRONICA INGENIERIA INDUSTRIAL NUEVO TALENTO MAGDALEN TRASLADO Ninguno TALENTO MAGDALENA.1

BIBLIOGRAFÍA

Aulck, L. & Velagapudi, N. & Blumenstock, J., West, J. (2016). Predicting Student Dropout in Higher Education. ICML Workshop on #Data4Good: Machine Learning in Social Good Applications, New York, NY, USA

Betancourt, G. A. (2005). Las Máquinas de Soporte Vectorial (SVMs). Scientia Et Technica, XI(27).

Brownlee, (2015). 8 Tactics to Combat Imbalanced Classes in Your Machine Learning Dataset. Machine Learning Mastery. https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/

Castro R. L.F., Espitia P. E., Montilla A.F. (2018). Applying CRISP-DM in a KDD Process for the Analysis of Student Attrition. In: Serrano C. J., Martínez-Santos J. (eds) Advances in Computing. Communications in Computer and Information Science, 885, Springer, Cham.

Chien, Guestrin (2016). XGBoost: un sistema de aumento de árboles escalable. ACM Digital Librery. https://dl.acm.org/doi/abs/10.1145/2939672.2939785

Dekker, G. W. &Pechenizkiy, M. & Vleeshouwers, J. M. (2009). Predicting Students Drop Out: A Case Study, Rev Educational Data Mining. International Working Group on Educational Data Mining. Congreso llevado a cabo en Cordoba, España.

Delen, D. (2010). A comparative analysis of machine learning techniques for student retention management, Rev Decision Support Systems, 49, 498-506.

Geoffrey Hinton, Terrence J. Sejnowski (1999). Unsupervised Learning and Map Formation: Foundations of Neural Computation

Géron, A. (2017). Hands-on Machine Learning With Scikit-Learn and TensorFlow: Concepts Tools and Techniques to Build Intelligent Systems, Sebastopol, CA, USA: Reilly.

Hernández, J. & Ramírez, M. & Ferri, C. (2004). Introducción a la minería de datos, Madrid ,España : Madrid Pearson Educación

Kovačić, Z. J. (2010). Early Prediction of Student Success: Mining Students Enrolment Data. Proceedings of Informing Science & IT Education Conference (InSITE), 647-665.

Linear and Quadratic Discriminant Analysis — scikit-learn 0.24.1 documentation. (2020). scikit learn. **https://scikit-learn.org/stable/modules/lda_qda.html#id2**

Ling, Charles X. y Chenghui Li. "Minería de datos para marketing directo: problemas y soluciones". Kdd . Vol. 98. 1998.

Liu, B. (2002). Theory and Practice of Uncertain Programming. Studies in Fuzziness and Soft Computing, Warsaw, Polonia, Physica, Heidelberg

Martínez (2010). Precision, Recall, F1, Accuracy en clasificación. IArtificial.net. visible body: https://www.iartificial.net/precision-recall-f1-accuracy-en-clasificacion/

Mogollón, P. A. (2018). El problema no es solo plata: 42 % de los universitarios deserta. Visible body: Revista semana: **https://www.elespectador.com/noticias/educacion/el-problema-no-es-solo-plata-42- de-los-universitarios-deserta-articulo-827739**.

Pal, S. (2012). Mining Educational Data to Reduce Dropout Rates of Engineering Students, Rev I.J. Information Engineering and Electronic Business, 2, 1-7.

Peter J. Rousseeuw (1987). «Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis». Computational and Applied Mathematics 20: 53-65

Płoński. (2020) Random Forest Feature Importance Computed in 3 Ways with Python. Visible Body: **https://mljar.com/blog/feature-importance-in-random-forest/#:~:text=Random%20Forest%20Built%2Din%20Feature,a%20set%20of%20Decision%20Trees**.

Raju, D., & Schumacker, R. (2015). Exploring student characteristics of retention that lead to graduation in higher education using data mining models. Journal of College Student Retention, 16, 563–591.

Rokach, L. (2010). Ensemble-Based Classifiers, Artif. Intell. Rev., 33, 1-39.

Santos, P. R. (2017). Los 2 tipos de aprendizaje en Machine Learning: supervisado y no supervisado. Visible body: Think Big Empresas Recuperado de **https://empresas.blogthinkbig.com/que-algoritmo-elegir-en-mlaprendizaje/**

Scikit-learn(2020). Visible body: documentation Clustering . **https://scikit-learn.org/stable/modules/clustering.html#mean-shift**

Scikit-learn(2020). Visible body: documentation DBSCAN . **https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html**

Scikit-learn(2020). Visible body: documentation KMeans . **https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html**

Scikit-learn(2020). Visible body: documentation MeanShift. **https://scikit-learn.org/stable/modules/generated/sklearn.cluster.MeanShift.html**

Scikit-learn(2020). Visible body: documentation recall_score . https://scikit-learn.org/stable/modules/generated/sklearn.metrics.recall_score.html#:~:text=The%20recall%20is%20the%20ratio,the%20worst%20value%20is%200.

Scikit-learn(2020). Visible body: documentation Spectral Clustering. **https://scikit-learn.org/stable/modules/generated/sklearn.cluster.SpectralClustering.html**

Scikit-learn(2020).Visible body: documentation One Hot Encoder.https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html

Simeone F.(2018). A Very Brief Introduction to Machine Learning With Applications to Communication Systems

sklearn.preprocessing.OneHotEncoder — scikit-learn 0.24.2 documentation. (2020). Retrieved 30 April 2021, from **https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html**