

Análisis de datos para la predicción del éxito académico mediante árboles de decisión

Alejandra Palacio Jaramillo Universidad Eafit Colombia apalacioj@eafit.edu.co	Valentina Moreno Ramírez Universidad Eafit Colombia vmorenor@eafit.edu.co	Miguel Correa Universidad Eafit Colombia macorream@eafit.edu.co	Mauricio Toro Universidad Eafit Colombia mtorobe@eafit.edu.co
--	--	--	--

RESUMEN

El objetivo de este proyecto es crear un algoritmo que permita, mediante una estructura de datos, predecir el éxito académico de una persona en las Pruebas Saber Pro, teniendo en cuenta su resultado en las Pruebas Saber 11 y demás factores relacionados como su condición socioeconómica, el pregrado estudiado, entre otros. La importancia de solucionar este problema radica en que cada día el pronóstico de la información es de mayor relevancia, puesto que tener una idea de lo que sucederá a futuro permite actuar de manera rápida y eficiente frente a lo predicho, gracias a la implementación de las tecnologías emergentes. Con base en soluciones a problemas similares, se hallará la solución más adecuada.

Palabras clave

Árboles de decisión, aprendizaje automático, éxito académico, predicción de los resultados de los exámenes, minería de datos, rendimiento académico.

1. INTRODUCCIÓN

En la actualidad, con la llegada de la recolección de datos, se ha dado espacio a la aparición de problemas relacionados con la predicción y el análisis de los resultados en futuro eventos, con el fin de implementar estrategias que permitan mejorar y optimizar la calidad de estos en años próximos. Para Colombia, y en general para América Latina, es claro que uno de los mayores problemas es la incertidumbre que genera la calidad de la educación y la eficiencia del sistema educativo en relación con el aprendizaje del estudiante. El hecho de pensar en la creación de un algoritmo único que permita anticipar los resultados académicos de los estudiantes para encontrar las falencias que hacen de la educación latinoamericana inoperante respecto al resto del mundo, es la motivación para trabajar en este proyecto arduamente, junto con la idea de llevar a cabo planes que

mejoren este sistema que ha estado fragmentado durante años.

1.1. Problema

De acuerdo con la información personal, familiar, académica, socioeconómica y sociodemográfica que se tiene a disposición de estudiantes que han presentado las Pruebas Saber 11, se desarrollará un algoritmo basado en el uso de

árboles de decisión, el cual predecirá si los resultados totales en las pruebas Saber Pro están por encima del promedio o no. Teniendo en cuenta las herramientas enseñadas en el curso de Estructuras de Datos y Algoritmos I, el concepto y funcionamiento de los árboles de decisión y los conocimientos previos respecto a la temática, en resumen, se logrará presagiar el éxito académico en esta prueba. La importancia de resolver este problema se sintetiza en cuanto puede mejorar el rendimiento académico en las áreas de mayor dificultad si se tiene una idea previa (predicción) de los posibles resultados.

1.2 Solución

Este trabajo está centrado en los árboles de decisión porque proporcionan una gran explicabilidad y precisión en los resultados. Además, los árboles de decisión se asemejan a la manera en la que el ser humano piensa, por tanto, son más fáciles de entender, interpretar y representar gráficamente. Se evitan los métodos de caja negra como las redes neuronales, las máquinas de soporte vectorial y los bosques aleatorios porque carecen de explicabilidad, confiabilidad en el resultado y conocimiento del alcance de escenarios posibles y de la estructura interna.

Para solucionar el problema planteado se implementa un algoritmo de la familia de los árboles de decisión, siendo este el algoritmo CART porque es el más adecuado para problemas que involucren la predicción de éxito o fracaso en

determinada situación ya que emplea el concepto de *impureza de Gini*. Adicionalmente, este algoritmo funciona correctamente con variables numéricas, categóricas y valores atípicos. También, al ser un árbol de clasificación y regresión, se usa para trabajar con datos históricos o pasados para predecir de datos nuevos, como lo es en este caso (uso de datos de Pruebas Saber 11 para predecir el resultado de las Pruebas Saber Pro de varios estudiantes).

1.3 Estructura del artículo

En lo que sigue, en la sección 2, presentamos el trabajo relacionado con el problema. Más adelante, en la sección 3, presentamos los conjuntos de datos y métodos utilizados en esta investigación. En la sección 4, presentamos el diseño del algoritmo. Después, en la sección 5, presentamos los resultados. Finalmente, en la sección 6, discutimos los resultados y proponemos algunas direcciones de trabajo futuras.

2. TRABAJOS RELACIONADOS

3.1 Predicción de factores relacionados al desempeño académico.

Como problema de la investigación se tenía el detectar los factores y patrones asociados al rendimiento académico de estudiantes colombianos de grado undécimo que presentaron las Pruebas Saber 11, teniendo en cuenta la información socioeconómica, académica e institucional de las bases de datos del ICFES. Para obtener esta predicción y así estimar las próximas condiciones de los estudiantes en futuras generaciones, se utilizó la herramienta WEKA y su algoritmo J48, el cual es una implementación del árbol de decisión C4.5, el cual tuvo una precisión del 67% en la clasificación correcta de los datos, pero a su vez tuvo un 33% de imprecisión al clasificar incorrectamente cierta cantidad de instancias. [3]

3.2 Predicción del rendimiento académico con minería de datos.

El problema planteado en esta investigación consistía en predecir el resultado final (ganado o perdido) de los estudiantes matriculados en un curso de Estadística General en la UNALM, tomando en cuenta su información académica previa, con el fin de contar con información adecuada para identificar, posteriormente, los aspectos influyentes en el aprendizaje. Para solucionar el problema, se utilizaron diferentes algoritmos como las redes bayesianas y los árboles de decisión. Como árbol de decisión, se aplicó el método C4.5 con un nivel de confianza de 0.15 y con poda,

el cual tuvo una precisión 68.3%, 2.7% menos que el algoritmo que obtuvo mayor precisión (71%). [4]

3.3 Comparación de técnicas de minería de datos para predecir resultados académicos.

La cuestión en desarrollo en esta investigación es la evaluación comparativa de la funcionalidad en tiempo de ejecución y precisión de algoritmos de clasificación para predecir el rendimiento académico de los estudiantes (bajo, medio y alto), con base en sus datos académicos previos (calificaciones). Dentro de los algoritmos comparados, se encuentran los árboles de decisión y se toma como modelo el algoritmo CART, el cual obtuvo un 75% de precisión y un 25% de error. [5]

3.4 Predicción del rendimiento de los estudiantes utilizando dos tipos de árboles de decisión.

El problema tratado en esta investigación es predecir el rendimiento académico de estudiantes de primer año de ingeniería, con base en los resultados de los estudiantes de ingeniería que en ese momento cursaban el segundo año. Dentro de los datos se tenían el nombre, género, calificaciones en el semestre, calificación obtenida en el examen de ingreso, tipo de admisión, entre otros. Para la predicción, se utilizaron dos árboles de decisión: el algoritmo ID3 y el C4.5, los cuales arrojaron la misma precisión, tanto en las evaluaciones masivas como en las singulares, siendo esta, en promedio, del 75.27% de exactitud. [6]

3. MATERIALES Y MÉTODOS

En esta sección se explica cómo se recopilamos y procesamos los datos y, después, cómo se consideraron diferentes alternativas de solución para elegir un algoritmo de árbol de decisión.

3.1 Recopilación y procesamiento de datos

Obtuvimos datos del *Instituto Colombiano de Fomento de la Educación Superior* (ICFES), que están disponibles en línea en <ftp.icfes.gov.co>. Estos datos incluyen resultados anonimizados de Saber 11 y Saber Pro. Se obtuvieron los resultados de Saber 11 de todos los graduados de escuelas secundarias colombianas, de 2008 a 2014, y los resultados de Saber Pro de todos los graduados de pregrados colombianos, de 2012 a 2018. Hubo 864.000 registros para Saber 11 y 430.000 para Saber Pro. Tanto Saber 11 como Saber Pro, incluyeron, no sólo las puntuaciones sino también datos socioeconómicos de los estudiantes, recogidos por el ICFES, antes de la prueba.

En el siguiente paso, ambos conjuntos de datos se fusionaron usando el identificador único asignado a cada estudiante. Por lo tanto, se creó un nuevo conjunto de datos que incluía a los estudiantes que hicieron ambos exámenes estandarizados. El tamaño de este nuevo conjunto de datos es de 212.010 estudiantes. Después, la variable predictora binaria se definió de la siguiente manera: ¿El puntaje del estudiante en el Saber Pro es mayor que el promedio nacional del período en que presentó el examen?

Se descubrió que los conjuntos de datos no estaban equilibrados. Había 95.741 estudiantes por encima de la media y 101.332 por debajo de la media. Realizamos un submuestreo para equilibrar el conjunto de datos en una proporción de 50%-50%. Después del submuestreo, el conjunto final de datos tenía 191.412 estudiantes.

Por último, para analizar la eficiencia y las tasas de aprendizaje de nuestra implementación, creamos al azar subconjuntos del conjunto de datos principal, como se muestra en la Tabla 1. Cada conjunto de datos se dividió en un 70% para entrenamiento y un 30% para validación. Los conjuntos de datos están disponibles en <https://github.com/mauriciotoro/ST0245-Eafit/tree/master/proyecto/datasets>.

	Conjunto de datos 1	Conjunto de datos 2	Conjunto de datos 3	Conjunto de datos 4	Conjunto de datos 5
Entrenamiento	15,000	45,000	75,000	105,000	135,000
Validación	5,000	15,000	25,000	35,000	45,000

Tabla 1. Número de estudiantes en cada conjunto de datos utilizados para el entrenamiento y la validación.

3.2 Alternativas de algoritmos de árbol de decisión

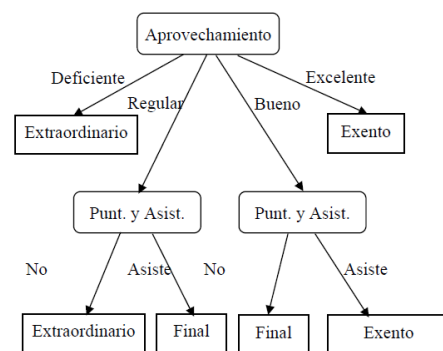
Un árbol de decisión es un clasificador que divide los datos de forma recursiva para formar grupos o clases.

3.2.1 Algoritmo ID3:

Fue el primer árbol de decisión desarrollado por Ross Quinlan. Este construye un árbol de decisión para los datos dados de forma descendente, comenzando por un conjunto de objetos y una especificación de propiedades, recursos e información.[1]

En cada nodo del árbol, se revisan los atributos en función de maximizar la ganancia de información y minimizar el desordenamiento de los datos. Luego de maximizar la ganancia, se forma la ramificación. Cabe destacar que este proceso se realiza de forma recursiva hasta que el conjunto en un subárbol dado contenga objetos de la misma categoría. En sí, el algoritmo ID3 selecciona una prueba usando el criterio de ganancia de información, y no busca otras opciones. [1]

Para n atributos, hay 2^n filas y podemos considerar la salida como una función definida por 2^n . Con esto hay 2^{2^n} posibles funciones diferentes para n atributos.



3.2.2 Algoritmo C4.5:

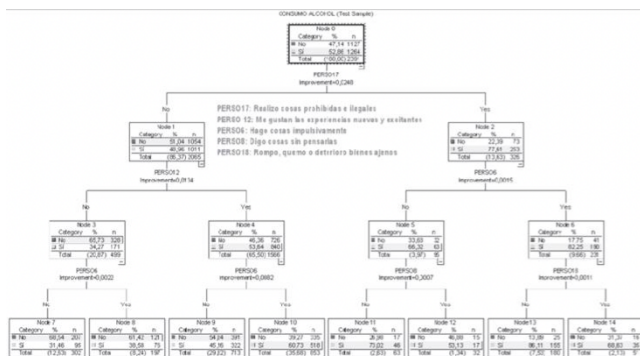
El algoritmo C4.5 es una versión mejorada del algoritmo ID3. C4.5 usa el proceso de *Shannon Entropy* (medir la incertidumbre de una fuente de información), para elegir características con la mayor ganancia de información. Una característica particular de este algoritmo es que construye los nodos con ramas vacías con valores de cero. [1]

El algoritmo considera los casos bases, luego todas las pruebas posibles que pueden dividir el conjunto de datos, posteriormente selecciona la prueba que resulta en la mayor ganancia de información, la toma como parámetro de decisión en el siguiente nodo y sigue así hasta obtener las posibles respuestas. Este proceso de particiones de datos se realiza recursivamente (se puede observar que tiene un comportamiento similar al “divide y vencerás” del algoritmo ID3). A diferencia del algoritmo ID3, el C4.5 puede tomar tanto valores discretos como continuos, tiene menos errores en la poda, es más eficiente, etc. Cabe resaltar que, para cada atributo discreto, se considera una prueba con n resultados, siendo n el número de valores posibles que puede tomar el atributo, mientras que, para cada atributo continuo, se realiza una prueba binaria sobre cada uno de los valores que toma el

atributo en los datos. Tomando en cuenta lo anterior, se puede afirmar que, en cada nodo, el sistema debe decidir cuál prueba escoge para dividir los datos. [2]

3.2.3 Algoritmo CART:

Los árboles de Clasificación y regresión CART (*Classification And Regression Trees*) se caracterizan por el hecho de construir el árbol a partir de particiones binarias recursivas (árboles binarios), es decir, cada nodo tiene exactamente dos bordes salientes. Las divisiones se seleccionan utilizando los criterios de dosificación y el árbol obtenido, se poda mediante poda de coste-complejidad. En otras palabras, también se puede decir que se selecciona la variable que minimice la impureza de Gini, para obtener valores más correctos. Es importante resaltar que CART puede manejar variables numéricas y categóricas (objetos). También maneja valores atípicos fácilmente. [1][7]



Como se mencionó anteriormente, el algoritmo utiliza el índice de Gini para calcular la medida de impureza:

$$G(A_i) = M_i \sum_j p(A_{ij}) G\left(\frac{C}{A_{ij}}\right)$$

Siendo:

- A_{ij} : es el atributo empleado para ramificar el árbol,
- J es el número de clases,
- M_i es el de valores distintos que tiene el **bv cfr**

atributo $A_i A_i$

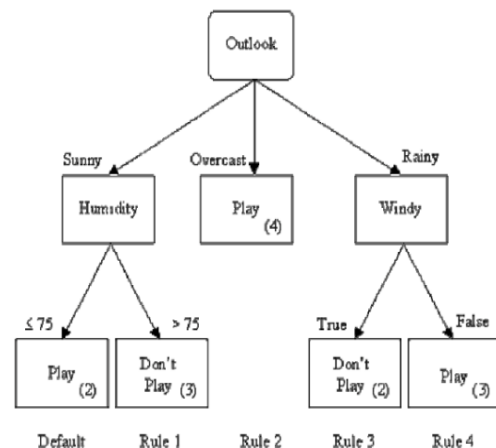
- $p(A_{ij})$ constituye la probabilidad de que $A_i A_i$ tome su j -ésimo valor y representa la probabilidad de que

un ejemplo sea de la clase $C_k C_k$ cuando su atributo $A_i A_i$ toma su j -ésimo valor.

El índice de diversidad de Gini toma el valor cero cuando un grupo es completamente homogéneo y el mayor valor lo alcanza cuando todas las $p(A_{ij})$ son constantes, entonces el valor del índice es $(J-1)J$.

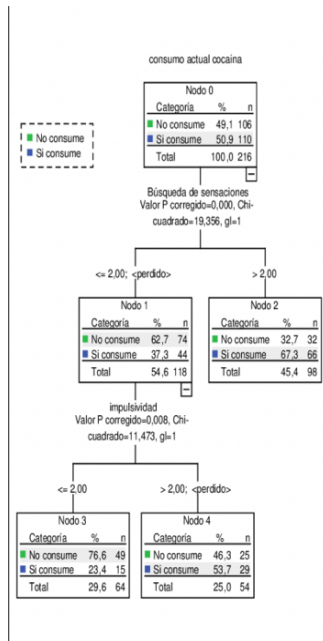
3.2.4 Algoritmo CHAID

CHAID es un algoritmo rápido y eficaz que permite la creación de árboles de decisión mediante la significancia ajustada de los datos. Se caracteriza por el uso de la chi-cuadrado de Pearson para determinar si la variable predictora



tiene una interacción o ajuste significativo respecto a la variable dependiente. Si la interacción es significativa, entonces se dividirá el nodo con base a esa variable y, si no lo es, se fundirá la variable junto con las otras que no son significativas, es decir, junto al grupo al que más se parece la variable, respecto a la variable dependiente. Cabe destacar que este algoritmo solo trata variables discretas. [8][9]





4. DISEÑO DE LOS ALGORITMO

Para la realización de este proyecto se decidió utilizar vectores dinámicos para guardar los datos del archivo, gracias a su complejidad en el acceso a la información ($O(1)$), la cual permite optimizar el tiempo del algoritmo.

4.1 Estructura de los datos

Ahora, para la predicción de las Pruebas Saber Pro se optó por la estructura de datos llamada árbol de decisión binario, el cual permitirá realizar elecciones adecuadas entre distintas posibilidades teniendo en cuenta la impureza de Gini. Su estructura permite seleccionar todas las diferentes opciones para conocer los diferentes escenarios.

Inicialmente, existe un nodo raíz el cuál se divide en dos subnodos y cada uno, a su vez, se divide en otros dos subnodos adicionales. Es importante aclarar que este proceso se realiza hasta que se llega a los nodos terminales para obtener el resultado final y que el criterio de división de los nodos depende de la condición que divida mejor los datos.

Algunas ventajas de usar un árbol de decisión binario son [10]:

- Permite la clasificación de nuevos casos
- Facilita la interpretación de la decisión analizada.
- Se puede analizar el comportamiento respecto a una determinada decisión.
- Número de variables reducido

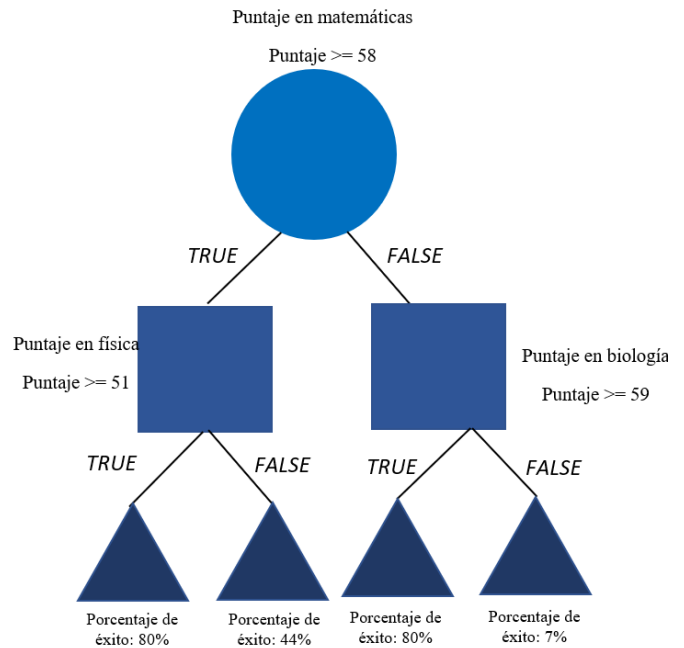


Figura 1: Un árbol de decisión binario para predecir Saber Pro basado en los resultados de Saber 11. El nodo circular es el nodo raíz, los nodos cuadrados son subnodos y los nodos triangulares son nodos terminales. Cuando se llega a los nodos terminales, se dice cuál será el porcentaje de éxito.

4.2 Algoritmos

Para dar solución al problema planteado, se eligió el algoritmo CART, gracias a su simplicidad y eficiencia. En este se generan árboles de decisión binarios, lo que quiere decir que cada nodo se divide en exactamente dos ramas.

Este modelo admite variables de entrada y de salida nominales, ordinales y continuas, por lo que se pueden resolver tanto problemas de clasificación como de regresión.

El algoritmo utiliza el índice de Gini para calcular la medida de impureza.

4.2.1 Entrenamiento del modelo

En primer lugar, para el entrenamiento del modelo, se crea un nodo raíz con una condición determinada, la cual es la que más reduce los datos de la manera más homogénea posible. De acuerdo con la condición, se dividen los datos en 2 subnodos, los cuales también tendrán una condición y dividirán los datos de cada uno de ellos en otros dos subnodos y así sucesivamente hasta llegar a los nodos terminales. Cabe destacar que, para establecer la condición en cada nodo, se implementa el concepto de impureza de Gini, el cual determina la pureza de la división en cada pregunta con las condiciones evaluadas. La condición que al

evaluar tenga una impureza de Gini menor, será la condición utilizada para la división del nodo. Por último, la clave está en establecer cuándo realizar cada pregunta para tener resultados precisos.

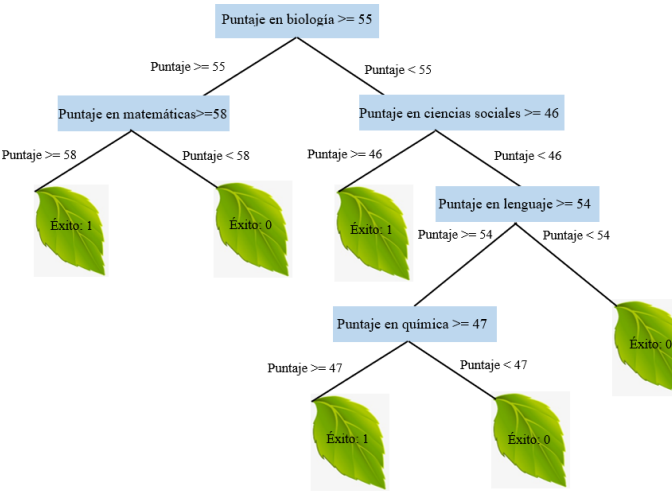


Figura 2 y 3: Entrenamiento de un árbol de decisión binario usando el algoritmo CART. En estos ejemplos, se muestra un modelo para predecir el éxito académico de un estudiante en las Pruebas Saber Pro con base en su resultado en las pruebas Saber 11.

4.2.2 Algoritmo de prueba

Luego de construir el árbol de decisión con el algoritmo CART y con los datos de entrenamiento, se utilizará un nuevo conjunto de datos para validar las predicciones que genera el árbol creado. Inicialmente, se crea una matriz de tamaño $n \times 2$, donde n es el número de estudiantes. Cabe destacar que, las dos columnas de esta matriz almacenarán en la posición cero el éxito del estudiante y en la posición 1 la predicción obtenida con el árbol de entrenamiento. Ahora, se comparan los valores que hay en estas columnas para cada estudiante y se obtienen la cantidad de éxitos que se predijeron bien, los que se predijeron mal, los fracasos que se predijeron bien y los que se predijeron mal para así calcular la precisión, exactitud y sensibilidad del árbol. De acuerdo con los datos arrojados por las variables mencionadas anteriormente, se define si el árbol predice correctamente o no.

4.3 Análisis de la complejidad de algoritmos:

Para el algoritmo de entrenamiento y creación del árbol de decisión, la complejidad asintótica para el peor de los casos del algoritmo es $O(N^2 \cdot M \cdot 2^M)$, dicho de otra, la complejidad

es exponencial y depende principalmente del número de columnas de la matriz de datos.

En primera instancia, se tiene que la complejidad del método que calcula la menor impureza de Gini para una condición determinada es $O(n^2)$ y, como en el peor de los casos el número de condiciones puede ser igual al número de columnas, entonces el método se ejecuta por cada columna, quedando así la complejidad en $O(n^2 \cdot m)$. Por último, como crear un árbol binario requiere de la creación de dos nodos hijos, entonces la complejidad asintótica para este caso es $O(2^n)$. De acuerdo con esto, como para cada nodo del árbol se calcula la mejor impureza de Gini y la mejor condición que divide el nodo más homogéneamente, entonces la complejidad final para la construcción del árbol binario de decisión implementando el algoritmo CART con los datos de entrenamiento es de $O(N^2 \cdot M \cdot 2^M)$.

Por otro lado, la complejidad asintótica en el peor de los casos para validar el árbol de decisión es $O(N \cdot M)$, donde N es el número de filas y M el número de columnas de la matriz que contiene los datos del test para verificar si el árbol predice bien o no.

Algoritmo	Complejidad del tiempo
Entrenar árbol de decisión	$O(N^2 \cdot M \cdot 2^M)$
Validar árbol de decisión	$O(N \cdot M)$

Tabla 2: Complejidad temporal de los algoritmos de entrenamiento y prueba. Siendo N el número de filas y M el número de columnas de los datos ingresados.

Ahora bien, al hablar de la complejidad en memoria al momento de entrenar el modelo, se tiene que hablar principalmente de la memoria que ocupa la matriz de datos de entrenamiento. Teniendo en cuenta lo anterior, la complejidad en memoria para el peor de los casos para este archivo sería $O(N \cdot M)$, donde N es el número de filas y M el número de columnas. Adicionalmente, como en cada nodo del árbol se tiene un subconjunto de este archivo, el cual cumple con ciertas condiciones para llegar hasta ese nodo, entonces la complejidad en memoria en el peor de los casos para la construcción del árbol será de $O(N \cdot M \cdot 2^M)$, por lo que se explicó anteriormente del comportamiento de los nodos al crearse el árbol (es exponencial porque en el peor de los casos, por ser un árbol binario, se crean 2 a la n nodos).

Por último, la complejidad en memoria para la validación del modelo es complejidad constante puesto que al momento de validar los nuevos datos, el árbol ya se encuentra creado y los elementos necesarios para la validación o la obtención de los predicciones con base en los nuevos datos ingresados.

Algoritmo	Complejidad de memoria
Entrenar árbol de decisión	$O(N \cdot M \cdot 2^M)$
Validar árbol de decisión	$O(1)$

Tabla 3: Complejidad de memoria de los algoritmos de entrenamiento y prueba. Siendo N el número de filas y M el número de columnas de los datos ingresados

5. RESULTADOS

5.1 Evaluación del modelo

En esta sección, se presentan algunas métricas para evaluar el modelo. La exactitud es la relación entre el número de predicciones correctas y el número total de datos de entrada. La precisión es la proporción de estudiantes exitosos identificados correctamente por el modelo sobre estudiantes exitosos identificados por el modelo. Por último, la sensibilidad es la proporción de estudiantes exitosos identificados correctamente por el modelo y estudiantes exitosos en el conjunto de datos.

5.1.1 Evaluación del modelo en entrenamiento

A continuación, se presentan las métricas de evaluación de los conjuntos de datos de entrenamiento en la Tabla 3.

Altura del árbol = 5	Conjunto de datos 1 (15.000 datos)	Conjunto de datos 2 (45.000 datos)	Conjunto de datos 3 (135.000 datos)
<i>Exactitud</i>	0.81	0.79	0.71
<i>Precisión</i>	0.83	0.81	0.79
<i>Sensibilidad</i>	0.78	0.77	0.72

Tabla 3. Evaluación del modelo con los conjuntos de datos de entrenamiento.

5.1.2 Evaluación de los conjuntos de datos de validación

A continuación, se presentan las métricas de evaluación para los conjuntos de datos de validación en la Tabla 4.

	Conjunto de datos 1 (15.000 datos)	Conjunto de datos 2 (45.000 datos)	Conjunto de datos (135.000 datos)
<i>Exactitud</i>	0.77	0.77	0.77
<i>Precisión</i>	0.8	0.8	0.79
<i>Sensibilidad</i>	0.72	0.72	0.74

Tabla 4. Evaluación del modelo con los conjuntos de datos de validación.

5.2 Tiempos de ejecución

A continuación, se presentan los tiempos de ejecución del algoritmo, tanto en su etapa de entrenamiento como de validación.

Altura del árbol = 5	Conjunto de datos 1 (15.000 datos)	Conjunto de datos 2 (45.000 datos)	Conjunto de datos (135.000 datos)
<i>Tiempo de entrenamiento</i>	10 s	43 s	155 s
<i>Tiempo de validación</i>	0 s	4 s	53 s

Tabla 5: Tiempo de ejecución del algoritmo (Por favor, escriba el nombre del algoritmo, C4.5, ID3) para diferentes conjuntos de datos.

5.3 Consumo de memoria

A continuación, se presenta el consumo de memoria del árbol de decisión binario, para diferentes conjuntos de datos, en la Tabla 6.

Altura del árbol= 5	Conjunto de datos 1 (15.000 datos)	Conjunto de datos 2 (75.000 datos)	...Conjunto de datos (135.000 datos)
Consumo de memoria	302 MB	346 MB	886 MB

Tabla 6: Consumo de memoria del árbol de decisión binario para diferentes conjuntos de datos.

6. DISCUSIÓN DE LOS RESULTADOS

La precisión de los datos del algoritmo es acorde a lo que se esperaba ya que, según los ensayos realizados y las medidas tomadas, esta se encuentra entre un 74% y 83% de exactitud, por lo tanto el modelo se está actualizando cada vez con el archivo de datos dado.

El consumo de memoria y tiempo son aspectos que se pueden mejorar pero, a pesar de esto, es bastante rápido para la cantidad de datos que son entregados al programa. Además, teniendo en cuenta la intención y el futuro esperado para este algoritmo, se espera que al manejar grandes bases de datos y ejecutar el código en la nube, el programa sea más rápido y eficaz.

Para un proyecto o aplicación futura, se pretende optimizar el funcionamiento del árbol para que este produzca resultados en una menor cantidad de tiempo, para que así sea posible calcular las predicciones cuando los datos en un nodo no se puede dividir más, en otras palabras, cuando se tienen datos más homogéneos en los nodos terminales del árbol, y que así tener como resultado un árbol altamente preciso. Además, se busca implementar el random forest ya que este sirve como un a técnica de reducción de la dimensionalidad y la creación de una estructura mucho más robusta que la generada por CART.

AGRADECIMIENTOS

Agradecemos a la universidad EAFIT por brindar espacios de conocimiento en los cuales, gracias al docente, obtuvimos las bases para llevar a cabo este proyecto; el realizar este proyecto es una gran oportunidad para ir preparándonos para la vida profesional con herramientas de Machine Learning y predicción y estimación de situaciones, las cuales al día de hoy son parte vital en la vida de un profesional de las ciencias de la computación. Además, agradecemos al profesor Helmut Trefftz por brindarnos de los fundamentos en el lenguaje de programación de Java.

REFERENCIAS

[1] Anónimo. *Cuáles son las diferencias entre ID3, C4.5 y CART?*. Organización PresMaryTwen, Abril del 2020, from <https://presmarymethuen.org/es/dictionary/what-are-the-differences-between-id3-c4-5-and-cart/>

[2] López, B. *ALGORITMO C4.5*. Instituto Tecnológico Nuevo Laredo, Nuevo Laredo, Tamaulipas, Noviembre del 2005, from [http://www.itnuevolaredo.edu.mx/takeyas/apuntes/Inteligencia%20Artificial/Apuntes/tareas_alumnos/C4.5/C4.5\(2005-II-B\).pdf](http://www.itnuevolaredo.edu.mx/takeyas/apuntes/Inteligencia%20Artificial/Apuntes/tareas_alumnos/C4.5/C4.5(2005-II-B).pdf)

[3] Timarán-Pereira, R., Caicedo-Zambrano, J., & Hidalgo-Troya, A. Árboles de decisión para predecir factores asociados al desempeño académico de estudiantes de

bachillerato en las pruebas Saber 11°. *Revista de Investigación Desarrollo e Innovación*, 9(2), 363-378, from https://revistas.uptc.edu.co/index.php/investigacion_duitama/article/view/9184/7721

[4] Menacho, C.H. Predicción del rendimiento académico aplicando técnicas de minería de datos. *Anales científicos*, 78(1), 26-33, from <https://dialnet.unirioja.es/servlet/articulo?codigo=6171237>

[5] Ochoa, L.L., Rosas, K., Baluarte, C. *Evaluación de Técnicas de Minería de Datos para la Predicción del Rendimiento Académico*. Global Partnerships for Development and Engineering Education: Proceedings of the 15th LACCEI International Multi-Conference for Engineering, Education and Technology, Boca Raton, FL, Estados Unidos, 2017, from http://www.laccei.org/LACCEI2017-BocaRaton/full_papers/FP368.pdf

[6] Adhatrao, k., Gaykar, A. Dhawan, a., Jha, R. & Honrao, V. Predicting Students' Performance Using Id3 And C4.5 Classification Algorithms. *International Journal of Data Mining & Knowledge Management Process*, 3(5), 39-52, from <https://github.com/mauriciotoro/ST0245-Eafit/blob/master/proyecto/problemas-relacionados/PREDICTING%20STUDENTS%E2%80%99%20PERFORMANCE%20USING%20ID3%20%26%20C4.5.pdf>

[7] Anónimo. *Aprendizaje automatizado: árboles de clasificación*. Departamento de Sistemas e Informática, from https://www.dsi.fceia.unr.edu.ar/downloads/ing_conocimiento/Presentaciones/ArbDec09.pdf

[8] Teixeira, O. Integración del algoritmo CHAID y una adaptación de este, CHAID*, en la plataforma Weka. Universidad de País Vasco, 2016, from https://addi.ehu.es/bitstream/handle/10810/21808/TFG_TeixeiraMartin.pdf?sequence=1&isAllowed=y

[9] Berlanga, V., Rubio, M.J., Vilà, R. Cómo aplicar árboles de decisión en SPSS. *REIRE*, 6(1), 65-79, from <https://revistes.ub.edu/index.php/REIRE/article/viewFile/5155/7229#:~:text=La%20funci%C3%B3n%20C%20A%20de%20decisi%C3%B3n,se%20ajuste%20a%20nuestros%20datos.>

[10] Parra, F. Estadística y Machine Learning con R. 110-125, from <https://bookdown.org/content/2274/metodos-de-clasificacion.html>