# PREDICCIÓN DE RESULTADOS FUTUROS EN PRUEBAS SABER PRO UTILIZANDO MACHINE LEARNING


Simón Correa Henao
Universidad Eafit
Colombia
scorreah@eafit.edu.co

David Gómez Correa
Universidad Eafit
Colombia
dgomezc10@eafit.edu.co

Miguel Correa
Universidad Eafit
Colombia
macorream@eafit.edu.co

Mauricio Toro
Universidad Eafit
Colombia
mtorobe@eafit.edu.co



**RESUMEN**

En la actualidad, en cuanto a educación respecta , podemos encontrar grandes falencias de la educación en Colombia, reflejadas en los resultados de sus pruebas estatales. Tales falencias pueden llegar a afectar, tanto a corto como a largo plazo, la calidad de vida de los ciudadanos en Colombia y el progreso, sea económico, social y o cultural del país. En particular, se derivan estos problemas de aspectos que parten del contexto económico, sociodemográfico y como retroalimentación negativa de la misma calidad académica.



**Palabras clave**

Árboles de decisión, aprendizaje automático, éxito académico, predicción de los resultados de los exámenes


## 1. INTRODUCCIÓN

La educación dentro del panorama nacional representa un pilar fundamental en el progreso del país. Pese a esto la inversión en el sector educativo, en conjunto con la situación sociodemográfica de gran parte del país han resultado en una educación con una calidad que se podría mejorar, en caso de llegar a conocer realmente cuales son los factores influyentes.

Entre la información que se posee actualmente para la medición de la calidad académica, se presentan los resultados de las pruebas Saber 11 y Saber Pro. Estos pueden utilizarse eficientemente aprovechando el tránsito a la educación 4.0. que cuenta con los avances suficientes para presentar un análisis más riguroso de los datos, facilitando la interpretación e importancia de cada parámetro estudiado.

Una vez procesados los datos, interpretado las variables correspondientes y los parámetros realmente influyentes del problema, se espera poder predecir resultados futuros de las pruebas Saber Pro, considerando como caso de éxito a aquel estudiante que obtiene un puntaje total, superior al promedio de su cohorte.

### 1.1. Problema

Considerando de antemano las grandes repercusiones que tiene la calidad del sistema educativo en cada país, el problema radica en que se tiene consciencia de la pésima calidad que posee el sistema colombiano en particular, e igualmente no se toman acciones al respecto.

Entre las muchas repercusiones que proceden de un sistema educativo de baja calidad, se encuentra la falta de mano de obra calificada, escasas oportunidades laborales, una posible baja calidad de vida y un decaimiento de la cultura.

Considerando lo anterior puede llegar a ser oportuno encontrar una solución que sea capaz de predecir el éxito o no de un individuo en las pruebas estatales, es decir las Pruebas Saber y Saber Pro; para de esta manera encontrar las variables más relevantes que afectan el éxito de la educación en Colombia.

### 1.2 Solución

En este trabajo, nos centramos en los árboles de decisión porque proporcionan una gran explicabilidad: *"…decision trees are considered white-box models because of their sequential evaluation nature. Even if a tree is large in size, a human can easily follow its computation step by step by evaluating (simple) decisions at each node from the root to a leaf.*" [2]. Evitamos los métodos de caja negra como las redes neuronales, las máquinas de soporte vectorial y los bosques aleatorios porque carecen de explicabilidad: "*Neural networks and random forests are considered black-box models because of their highly parallel nature: following the execution of neural networks means following sequences of parallel execution steps that result from a complex interplay of the value of all neurons (or nodes). The results of such black-box executions are hard to explain to a human user even for very small examples*." [2].

Por dicho motivo, el algoritmo a implementar para darle solución a este problema es el CART, que genera un árbol de decisión de regresión. Este algoritmo permite con base en la ganancia de información o la impureza de los datos, realizar subdivisiones binarias; igualmente permite trabajar con todo tipo de variables y el corte de los nodos se da por reglas binarias, lo cual facilita el trabajo, y permite una mejor comprensión del proceso realizado. A su vez, por la naturaleza de los datos con que se va a trabajar, dicho algoritmo cumple con los requerimientos pedidos y permite el manejo de todos los datos sin excepción.

### 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

### 2.1 Arboles de decisión para predecir factores asociados al desempeño académico de los estudiantes de bachillerato en las pruebas Saber 11°

El problema principal que aborda Ricardo Timaran y Javier Caicedo durante el reporte técnico, es acerca de cómo con base en los datos recolectados acerca de la situación sociodemográfica de los estudiantes del grado 11° se puede llegar a crear un árbol de decisión capaz de predecir el éxito o no de dichos estudiantes. El algoritmo utilizado para el árbol de decisión fue el J48 de la herramienta WEKA con una precisión del 67%.

### 2.2 Decision trees for predicting the academic success of students

El problema planteado por Josip Mesaric y Dario Sebalj es predecir el éxito o fracaso de los estudiantes en su primer año de estudio universitario, tomando como base de datos los estudiantes que aprobaron dicho año de la facultad de economía de la universidad de Osijek; para solucionar dicho problema aplicaron el uso de distintos algoritmos, como lo fueron el J4.8, ID3, REPTree, RandomTree, RandomForest. Siendo el de mayor precisión el REPTree con un 79.35%.

### 2.3 Comparison of data mining techniques to identify signs of student desertion, based on academic performance

La temática tratada por Rainiero Perez Gutierres en su artículo se centra en el problema de identificar estudiantes en riesgo de deserción académica, partiendo de la información de educandos de una universidad en Colombia del programa de ingeniería de sistemas. En este proyecto se utilizó la minería de datos basada en la metodología CRISP-DM y posteriormente se obtuvieron mejores resultados haciendo uso de los Random Trees, alcanzando una precisión del 83% en el modelo.

### 2.4 Knowledge Capture for the Prediction and Analysis of Results of the Quality Test of Higher Education in Colombia

García Gonzales et al. por medio de la metodología de extracción de datos KDD, para la extracción de la información de las bases de datos, y el uso de redes neuronales, como técnica de minería de datos, buscaron analizar los resultados obtenidos de estudiantes junto con una serie de datos característicos de cada uno, para realizar un modelo predictivo de los posibles futuros resultados, en las pruebas Saber Pro. Finalmente obteniendo una precisión del 82%.

## 3. MATERIALES Y MÉTODOS

En esta sección se explica cómo se recopilaron y procesaron 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 gradudados 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 .

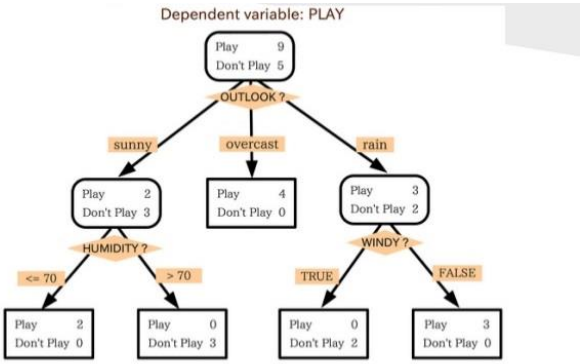
| | Conjunto de datos 1 | Conjunto de datos 2 | Conjunto de datos 3 | Conjunto de datos 4 | Conjunto de datos 5 |
|---|---|---|---|---|---|
| **Entrenamiento** | 15,000 | 45,000 | 75,000 | 105,000 | 135,000 |
| **Validación** | 5,000 | 15,000 | 25,000 | 35,000 | 45,000 |

**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

#### 3.2.1 ID3

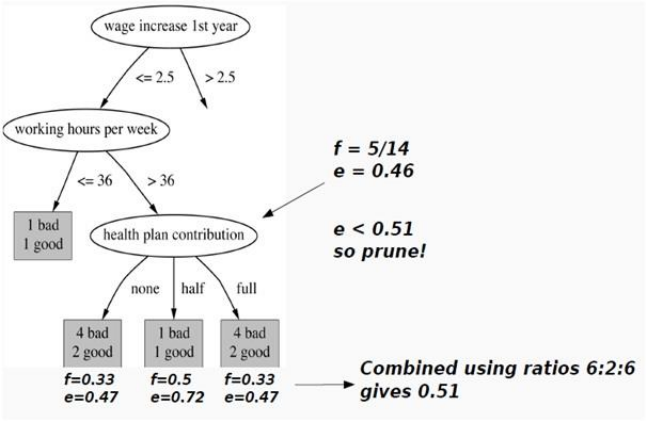
El algoritmo ID3 dado un conjunto de elementos, subdivide o clasifica los elementos de la mejor manera para generar un árbol de decisión. Estas divisiones se hacen con base a la "ganancia de información" obtenida, es decir la diferencia entre la entropía de un nodo y la de sus descendientes, para de esta manera evaluar primero los nodos con información más relevante. El algoritmo parte de las subdivisiones más generales a las más específicas, para ir clasificando sus elementos.




#### 3.2.2 C4.5

El algoritmo C4.5 es la versión mejorada del algoritmo ID3 trabajando con la metodología de "depth-first", es decir realiza la división de los nodos a partir de la ganancia de estos mismos. Entre las diferencias con su predecesor, el algoritmo C4.5 tiene la capacidad de manejar puntos con datos incompletos, añade una nueva técnica conocida como poda, y permite añadirle “pesos” a los parámetros de los datos de entrenamiento.

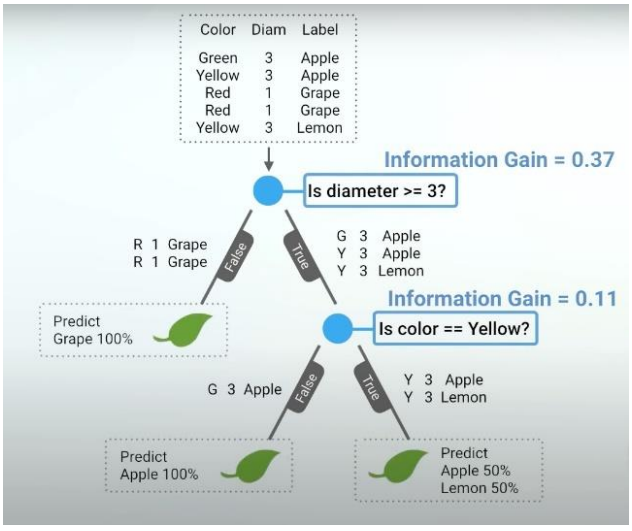
Estas mejoras al algoritmo ID3 posibilitan el determinar la profundidad del árbol de decisión, la reducción errores en la poda y la mejora en la eficiencia computacional.




#### 3.2.3 CART

CART es un algoritmo que crea un árbol de decisión de regresión. El objetivo del algoritmo es encontrar la distribución más pura posible de los elementos, eligiendo las subdivisiones que involucren la mayor ganancia de información en cada nodo.

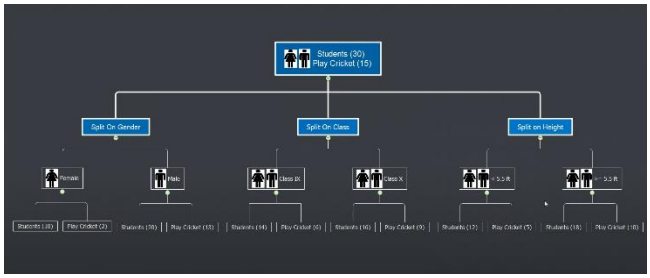
Lo anterior se consigue sirviéndose del índice de Impureza de Gini, para la división del árbol. Con este se toman todas las instancias posibles (o subdivisiones) y las respuestas correspondientes, para después calcular el índice de Gini, sabiendo que entre más cercano a 0 sea su valor, mayor homogeneidad tendrá ese conjunto de datos. Dicho proceso se repite con las variables y los subconjuntos restantes, hasta que no es posible realizar más subdivisiones con la información disponible.



### 3.2.4 CHAID

CHAID es el acrónimo de "Chi-Squared Automatic Interaction Detector" y se trata de un algoritmo para la creación de árboles de decisión que busca encontrar la relación entre las variables con las que trabaja. Este admite datos nominales, ordinales y continuos, con los cuales realiza todas las permutaciones de divisiones posibles hasta que se logra el mejor resultado y no se puedan realizar más divisiones. Si la variable dependiente es continua utiliza una prueba conocida como "F-test", de lo contrario si es categórica utiliza la técnica del "Chi-Square".

Una vez el árbol esté terminado, se tendrá como nodo raíz, la variable objetivo que se quiere estudiar, y a continuación las variables predictoras organizadas de manera que los nodos que más influencia ejercen sobre la variable objetivo están más cerca del nodo raíz, y aquellas con menos importancia se dispondrán al final del árbol.




## 4. DISEÑO DE LOS ALGORITMOS

En lo que sigue, explicamos la estructura de los datos y los algoritmos utilizados en este trabajo.

### 4.1 Estructura de los datos

El manejo de datos para el análisis de estos mismos es de suma importancia, cabe resaltar que por la naturaleza del algoritmo a trabajar (CART) la estructura de datos es un árbol de decisión binario, es decir que las divisiones generadas son 1 o 0, o en otras palabras si o no, lo que significa que la pregunta que se hace en cada nodo pregunta, siempre tiene solo 2 posibles respuestas.
El árbol se compone por un nodo raíz, que contiene todos los datos de entrada sin subdivisiones. Luego cada nodo tiene como máximo dos nodos hijos, que se dividen al aplicar una pregunta o condición al nodo actual. De esta manera se continúa dividiendo los nodos al hacer las preguntas correspondientes, esto hasta que se llega a un nodo en el que no es posible realizar más subdivisiones. Este ultimo recibe el nombre de hoja.

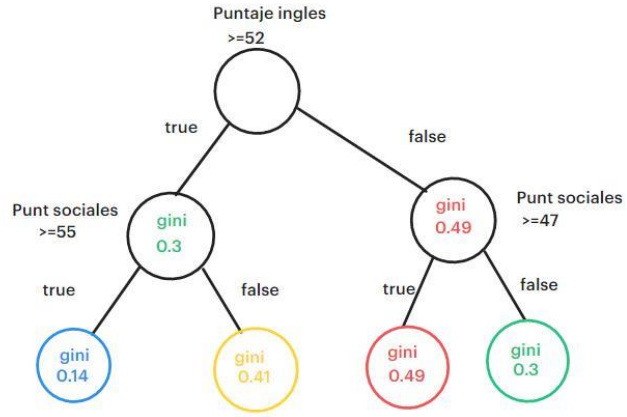



**Figura 1:** Un árbol de decisión binario para predecir Saber Pro basado en los resultados de Saber 11. Los nodos azules representan a aquellos con una alta probabilidad de éxito, los verdes con una probabilidad media, los amarillos una probabilidad media-baja, y los rojos con una baja probabilidad de éxito.

### 4.2 Algoritmos

El algoritmo elegido, CART, se caracteriza por ser un algoritmo de árbol de decisión de regresión o también conocido como *decision tree learning algorithm.* Este utiliza el Indice de Gini y la Ganancia de información, o *Information Gain*, para encontrar las mejores preguntas o condiciones para dividir el árbol, de tal manera que los datos queden lo mejor clasificado posibles. Y finalmente crear un árbol con la capacidad de tomar decisiones propias, a base de predicciones hechas.

#### 4.2.1 Entrenamiento del modelo

El algoritmo para entrenamiento del modelo y construcción del árbol toma en un principio los datos de entrenamiento, los cuales contienen la variable de éxito para cada estudiante, y con estos busca, por medio del índice de Gini, la condición que mejor divide los datos, es decir aquella cuya ganancia de información es mayor. Luego se divide la matriz de datos en dos sub-matrices, una que contiene los individuos que cumplen la condición y la otra con los individuos que no la cumplen. Posteriormente se repite dicho proceso con las nuevas matrices o ramas creadas, hasta que se llega a que cada nodo o rama no puede tener más subdivisiones.

Una vez que el árbol no puede ser dividido en ninguna de sus ramas, se considera como terminado y se retorna una referencia al nodo raíz del árbol.

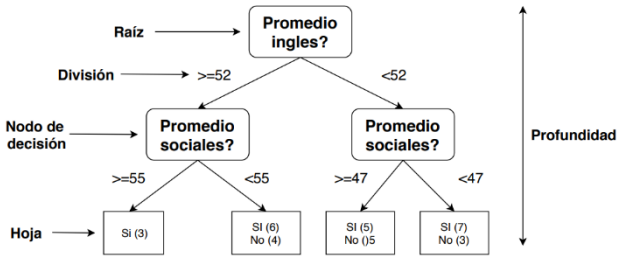



**Figura 2:** El Algoritmo usado para construir un árbol de decisión binario y para predecir el éxito de un individuo fue el CART. En este ejemplo, se muestra un modelo para predecir si un estudiante va a tener éxito en las Pruebas Saber Pro, basado en sus resultados de las pruebas Icfes

### 4.2.2 Algoritmo de prueba

El algoritmo de prueba se encarga de testear el modelo creado o el árbol de decisión con datos que no fueron utilizados durante su entrenamiento. Este ingresa los nuevos datos por medio del árbol, y para cada individuo evalúa y predice si esté tendrá éxito o no en un futuro. Luego compara ese resultado obtenido del árbol, con una serie de respuestas ya preestablecidas de cada estudiante, para así saber cual es la tasa de acierto del algoritmo creado anteriormente.

### 4.3 Análisis de la complejidad de los algoritmos

Para el algoritmo de entrenamiento y creación del árbol de decisión, considerando N como el número de filas y M como el número de columnas, en el peor de los casos se debió recorrer la matriz de la siguiente manera:

Primero se recorrieron M columnas para hacer todos los cálculos necesarios para encontrar la condición que mejor dividía el árbol o la matriz de datos. Luego, internamente además, se recorrieron N filas para evaluar todos los posibles valores que tomaba cada columna. Y finalmente, para calcular el índice de Gini con cada condición probada, se debieron recorrer N filas, en el peor de los casos, para encontrar el mejor.

Dado que estas operaciones fueron anidadas, se explican los términos $N^2*M$ que aparece en el cálculo de complejidad en tiempo.

Para explicar el termino $2^M$ se debe tener en consideración que las divisiones máximas en el peor de los casos que puede tomar el árbol, es preguntar cada condición, es decir cada columna, para que divida el árbol, por lo menos una vez. Siendo así, el numero de subdivisiones finales del árbol, dependería del número M de columnas o condiciones que se tuvieran, y puesto que el algoritmo trabajó sobre un árbol binario, la complejidad terminó siendo $2^M$ subdivisiones en el peor de los casos.

En cuanto a lo que la complejidad de la validación del árbol se refiere, este se realizó de tal manera que solo se tuviera que recorrer las N filas y M columnas una vez, apoyándonos del árbol de decisión ya creado, para saber cual sería el resultado de cada uno de los individuos presentes en los datos de validación.

| Algoritmo | La complejidad del tiempo |
|---|---|
| Entrenar el árbol de decisión | $O(N^2*M*2^M)$ |
| Validar el árbol de decisión | O(N*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.

La complejidad en memoria, al momento de entrenar el modelo resultó por ser N * M, dado que en el peor de los casos se crean un par de matrices de cada nodo del tamaño unificado de la matriz original. Esto para cada una de las subdivisiones o nodos, que finalmente toma el número de $2^M$, porque se trata de un árbol binario con M subdivisiones en el peor de los casos.

La complejidad en memoria para la validación del modelo es una complejidad constante, puesto que al momento de validar ya se encuentra creado todo el árbol binario y los elementos necesarios para la validación o la obtención de resultados a partir de nuevos datos ingresados.

| Algoritmo | Complejidad de memoria |
|---|---|
| Entrenar el árbol de decisión | $O(N*M*2^M)$ |
| Validar el 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.

### 4.4 Criterios de diseño del algoritmo

El algoritmo fue escogido pensando principalmente en la forma como, tanto el índice de Gini como la estructura de datos utilizada por el CART, se adaptaban perfectamente al problema inicial. Esto considerando que el algoritmo CART ofrece una buena complejidad en tiempo, se sirve de una excelente estructura de datos como los árboles binarios y además de posee una gran explicabilidad. Igualmente, se tomaron las matrices como punto de partida, aprovechando su particular acceso en O(1), característica que fue muy útil para priorizar el tiempo de ejecución del algoritmo.

## 5. RESULTADOS

### 5.1 Evaluación del modelo

En esta sección, presentamos 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. Precisión es la proporción de estudiantes exitosos identificados correctamente por el modelo y estudiantes exitosos identificados por el modelo. Por último, 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 presentamos las métricas de evaluación de los conjuntos de datos de entrenamiento en la Tabla 3.

| | ***Conjunto de datos 1*** | ***Conjunto de datos 3*** | ***...Conjunto de datos 5*** |
|---|---|---|---|
| *Exactitud* | 0.78 | 0.78 | 0.78 |
| *Precisión* | 0.76 | 0.72 | 0.72 |
| *Sensibilidad* | 0.79 | 0.82 | 0.82 |

**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 presentamos las métricas de evaluación para los conjuntos de datos de validación en la Tabla 4.

| | ***Conjunto de datos 1*** | ***Conjunto de datos 3*** | ***...Conjunto de datos 5*** |
|---|---|---|---|
| *Exactitud* | 0.77 | 0.78 | 0.78 |
| *Precisión* | 0.76 | 0.72 | 0.72 |
| *Sensibilidad* | 0.78 | 0.81 | 0.81 |

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

### 5.2 Tiempos de ejecución

| | ***Conjunto de datos 1*** | ***Conjunto de datos 3*** | ***...Conjunto de datos n*** |
|---|---|---|---|
| *Tiempo de entrenamiento* | 2.8 s | 16.4 s | 28.2 s |
| *Tiempo de validación* | 0.4 s | 4.5 s | 10.3 s |

**Tabla 5:** Tiempo de ejecución del algoritmo CART para diferentes conjuntos de datos.

### 5.3 Consumo de memoria

Presentamos el consumo de memoria del árbol de decisión binario, para diferentes conjuntos de datos, en la Tabla 6.

| | ***Conjunto de datos 1*** | ***Conjunto de datos 3*** | ***...Conjunto de datos 5*** |
|---|---|---|---|
| Consumo de memoria | 15 MB | 26 MB | 78 MB |

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

## 6. DISCUSIÓN DE LOS RESULTADOS

Los resultados obtenidos tanto en términos de exactitud, precisión y o sensibilidad del algoritmo fueron muy prometedores. Esto hablando en términos de margen de error del algoritmo y haciendo una comparación con los trabajos relacionados.

Por otro lado, con los conjuntos de datos presentados para realizar el entrenamiento y validación del modelo, no se presentó sobreajuste, pues para los datos de validación se conservaron estadísticas similares a las medidas anteriormente. Cabe aclarar que mientras más datos de entrenamiento se utilicen, mejores resultados se obtienen con el modelo.

Finalmente, observando la complejidad en tiempo y en memoria. Respecto a la complejidad en tiempo se obtuvieron resultados aceptables, aunque pueden aún ser optimizados. Mientras que la complejidad en memoria, teniendo en cuenta que se le dio prioridad al tiempo de ejecución, puede llegar a ser de bastante consumo.

Con respecto a la utilidad del algoritmo, considerando que el margen de error de este se ubica entre el 20 y 30%, no sería adecuado para ser utilizado como criterio al momento de otorgar becas académicas; por el contrario, dicho algoritmo y sus resultados si son convenientes para identificar y de esa manera reforzar estudiantes que tengan una baja probabilidad de éxito en sus estudios.

### 6.1 Trabajos futuros

Algunas cosas que se podrían mejorar del algoritmo en un futuro son:

Primero, optimizar el código del algoritmo, depurarlo y hacerlo más consistente y sencillo, dado que hay procedimientos repetitivos en el código.

Luego de esto, algo importante sería mejorar la complejidad en memoria, puesto que puede llegar a afectar la utilización del modelo, y se trata de algo prioritario.

Y finalmente aún se podría optimizar un poco más la complejidad en tiempo del modelo, cosa que podrá resultar más fácil una vez que el programa se optimizado y depurado.

Por otra parte, el uso de bosques aleatorios en trabajos futuros podría llegar a ser beneficioso para mejorar la diversidad y tal vez las estadísticas del árbol o modelo. Sería algo adecuado para hacer más adelante.

**AGRADECIMIENTOS**

Los agradecimientos van dirigidos principalmente al profesor de la materia *Estructura de Datos y Algoritmos I* de la Universidad EAFIT, que nos acompañó tanto resolviendo dudas como guiándonos con ayudas oportunas para el correcto desarrollo de todo el proyecto.

Agradecemos a los estudiantes David Gómez y Simón Correa por su esfuerzo y dedicación en llevar este proyecto a cabo y por culminarlo de la mejor manera posible, obteniendo óptimos resultados.

**REFERENCIAS**

1. García-González, J.R., Sánchez-Sánchez, P.A., Orozco, M., Obredor, S. Knowledge capture for the prediction and analysis of results of the quality test of higher education in Colombia. *Formacion Universitaria, 12* (4), 55-62.

2. Gossen, F., Margaria, T. and Steffen, B., 2020. Towards Explainability in Machine Learning: The Formal Methods Way. IT Professional, 22(4), 8-12. Retrieved October 10, 2020 doi: 10.1109/MITP.2020.3005640.

3. Josh Gordon. 2017. Let's Write a Decision Tree Classifier from Scratch - Machine Learning Recipes #8. Video. (13 September 2017). Retrieved August 16, 2020 from https://www.youtube.com/watch?v=LDRbO9a6XPU&t=432s

4. Kabakus, T., 2020. ID3 Algorithm & ROC Analysis. [online] www.slideshare.net/. Available at: <https://image.slidesharecdn.com/id3algorithmrocanalysis-121101120125-phpapp02/95/id3-algorithm-roc-analysis-6-638.jpg?cb=1351771495> [Accessed 16 August 2020].

5. López, B. Algoritmo ID3. Instituto Tecnologico Nuevo Laredo. Recuperado Agosto 12, 2020: http://www.itnuevolaredo.edu.mx/takeyas/Apuntes/Inteligencia%20Artificial/Apuntes/IA/ID3.pdf

6. López, B. Inteligencia Aritificial Algoritmo C4.5. Instituto Tecnologico de Nuevo Laredo, Nuevo Laredo, 2005.

7. Mesaric, J. Sebalj, D. Decision trees for predicting the academic success of students, University of Josip Juraj Strossmayer in Osijek, Osijek, 2016, 367-388.

8. octaviansima.wordpress.com. 2020. Octavian's Blog. [online] Available at: <https://octaviansima.files.wordpress.com/2011/03/c45-sample1.jpg> [Accessed 16 August 2020].

9. Orellana, J. Arboles de decisión y Random Forest. Bookdown. Recuperado Agosto 16, 2020 : https://bookdown.org/content/2031/

10. Rainiero, B. Comparison of data mining techniques to identify signs of student desertion, based on academic performance. *Revistas UIS Ingenierias, 19* (1), 193-204.

11. Splunk & Machine Learning. 2019. Decision Tree : Construction of Classification Tree Using Chi-Square Algorithm. Video. (4 July 2019). Retrieved August 16, 2020 from https://www.youtube.com/watch?v=J3QwOjSVH8k

12. Statistics Solutions. 2020. CHAID. Retrieved August 17, 2020, from : <https://www.statisticssolutions.com/non-parametric-analysis-chaid/> [Accessed].

13. Timarán, R. Calcedo, J. Hidalgo, A. Arboles de decisión para predecir factores asociados al desempeño académico de los estudiantes de bachillerato en las pruebas Saber 11°. *Rev.investig.desarro.innov, 9* (2), 363-378.