APLICACIONES CON ÁRBOLES DE DECISIÓN Y BOSQUES ALEATORIOS EN LA PREDICCIÓN DEL ÉXITO ACADÉMICO EN PRUEBAS SABER PRO

Sebastián Guerra
Universidad Eafit
Colombia
jsguerrah@eafit.edu.co

Jacobo Rave
Universidad Eafit
Colombia
jravel@eafit.edu.co

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

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

RESUMEN

El siguiente texto trata sobre la formulación de un algoritmo para predecir el éxito académico en los resultados de las pruebas saber Pro en la educación superior, teniendo en cuenta variables de gran peso, como lo pueden ser resultados de exámenes anteriores, así como carencias tecnológicas que afectan parcialmente los resultados y su éxito, de manera que la información obtenida se recopile para generar posibles soluciones a futuro.

El algoritmo propuesto para este proyecto ha sido el árbol de decisión CART, junto al uso de bosques aleatorios. Con esto se logró una predicción del 75% utilizando datos de estudiantes que han presentado la prueba anteriormente. Los tiempos obtenidos rondan los 100 segundos para la generación de los árboles. Con la información recopilada para la construcción del algoritmo se evidencia la importancia de los campos lingüísticos en el éxito de una persona, que van desde la comprensión lectora hasta el manejo de una segunda lengua.

Palabras clave

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

1. INTRODUCCIÓN

A lo largo del tiempo, la educación y la investigación han trascendido como factores fundamentales en el crecimiento de la sociedad. Evidentemente, estos han sido determinantes en la evolución exponencial del conocimiento humano. Por estas razones la educación se ha vuelto una necesidad global, un derecho fundamental para cada individuo que, desde su praxis, aporte su 'granito de arena' en la divulgación del conocimiento.

Sin embargo, el sentido de la educación se ha visto afectado por diversos factores que frenan su alcance. Entre estos la accesibilidad, la motivación, la brecha tecnológica y social.

1.1. Problema

Entender las problemáticas anteriores se ha vuelto una labor indispensable para dar pie a la creación de soluciones sostenibles que hagan valer el derecho de la educación.

Una estrategia que permite ahondar en estos aspectos ha sido la implementación de algoritmos en el análisis de las características del entorno de cada individuo y su correlación con el éxito académico. Con esto, podemos identificar los diferentes aspectos que demuestran una incidencia negativa en la dinámica del aprendizaje y, de esta forma, proponer soluciones para ser más eficientes en los principios del derecho de educación.

1.2 Solución

En este trabajo nos centramos en los árboles de decisión porque proporcionan una gran explicabilidad. Según Loyola-González (2019)[11] la explicabilidad se argumenta ya que estos "can be understood by experts in practical applications due to they provide a model closer to the human language" (p. 154096). Por otro lado, evitamos los métodos de caja negra como las redes neuronales, las máquinas de soporte vectorial y los bosques aleatorios porque carecen de explicabilidad "due to the several transformations made to the input data" (Loyola-González, 2019, p. 154098)

El algoritmo que vamos a usar para los árboles de decisión es el CART por su versatilidad a la hora de emplear todo tipo de datos tanto cualitativos como cuantitativos. Su funcionamiento se va a explicar a lo largo de este documento.

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 en la aprobación de un curso

En la Universidad Peruana de Ciencias Aplicadas se planteó la posibilidad de que los estudiantes tuvieran una recomendación sobre que materias cursar en función de su registro académico. Con esto, decidieron llevar a cabo la implementación de un árbol de decisión de tipo C4.5 y el uso de la metodología KKD [1]. Con esta implementación lograron una precisión del 90%, resaltando la necesidad de la mejora del algoritmo para la eliminación de los datos que no representan utilidad. [6]

3.2 Factores asociados en resultados de la prueba Saber 11°

El objetivo principal de la realización de este estudio era la de establecer características que tuvieran relación con el tipo de resultado que se obtuvo en la prueba Saber 11° realizada en 2015 y 2016. Partiendo de la base de datos del ICFES, y con la implementación J48 del algoritmo de árbol de decisión C4.5 encontraron los atributos de mayor ganancia de información tanto en el caso de alto desempeño en la prueba, como en el caso de bajo desempeño. De acuerdo con los resultados del algoritmo, el porcentaje de precisión fue de un 67%. [5]

3.3 Deserción estudiantil

El problema de la deserción estudiantil en América Latina da cuenta de la alta población que carece de educación superior. Con esta motivación, Arnulfo Hernández et al. [3] diseñaron un algoritmo que pueda predecir las probabilidades de que un estudiante deserte. Para este se implementó un árbol de decisión CART que utilizaba atributos tanto académicos como personales, demostrando al final una precisión del 94%. [2]

3.4 Rendimiento de alumno de primer curso

La intención de este estudio fue la de predecir el rendimiento de un alumno de primer curso contando solamente con la información adquirida durante el proceso de matrícula. Con base en lo anterior, se planteó el uso del algoritmo C&R – derivado de CART – para el análisis de los atributos. En este estudio, la precisión fue determinada con el índice de error cuadrático medio, dando un resultado de 17.95. [4]

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 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

En lo que sigue, presentamos diferentes algoritmos usados para construir automáticamente un árbol de decisión binario.

3.2.1 ID3

El algoritmo ID3 basa su operación con el cálculo de la entropía y la ganancia, índices que ayudan a la determinación del nodo con menor pérdida de información. Luego de esto, con el índice de ganancia se busca la asignación de los siguientes nodos en correlación con el atributo de su padre. La complejidad de este algoritmo es de $O(M*N^2)$, donde m representa el tamaño de los datos de entrenamiento, mientras que N el número de atributos.

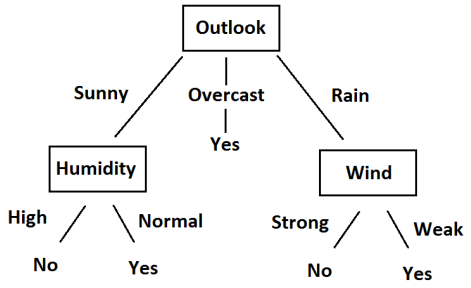

Figura 1. Figura vectorizada de árbol ID3 [7]

3.2.2 C4.5

El algoritmo C4.5, a diferencia del ID3, utiliza el gainRatio como índice para evitar el sobreajuste del árbol. Es decir, evita dar prioridad a los datos de entrenamiento sobre los nuevos datos. Además, incorpora la capacidad de encontrar un umbral para dividir los datos cuando se trata de un atributo continuo. La complejidad de este algoritmo es igual a la del algoritmo ID3.

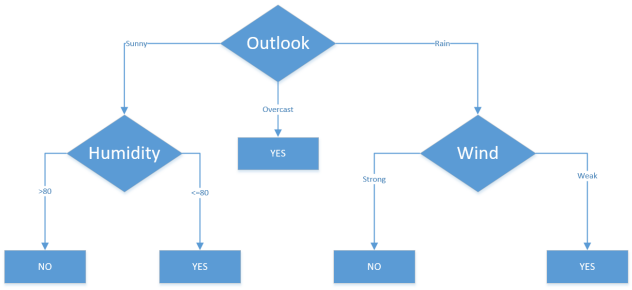

Figura 2. Figura vectorizada de árbol C4.5 [8]

3.2.3 CART

El algoritmo CART es una alternativa a los anteriores. Se basa en el índice de Gini para crear las decisiones en la clasificación. Dependiendo de sus argumentos y valores puede ser Cuantitativo o Cualitativo. Su ventaja es la segmentación recursiva, o sea que hace los nodos terminales tan homogéneos como sea posible. Es más efectivo ante estructuras discriminantes. Su complejidad es O(V*NlogN): V representa el primer ciclo mientras que N representa el segundo y tercer ciclo.

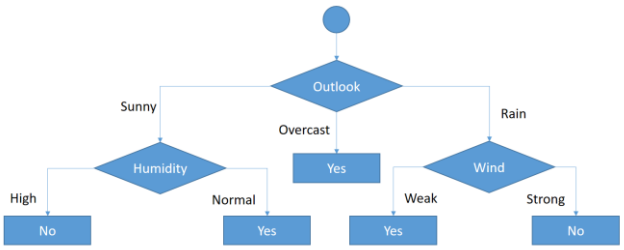

Figura 3. Figura vectorizada de árbol CART [9]

3.2.4 C5.0

El 5.0 es un algoritmo basado en el predecesor C4.5, utiliza la entropía para la selección de los datos. Este ofrece un método del aumento para obtener mayor precisión en las tareas de clasificación. Funciona con un algoritmo inicial eligiendo predictores, para ajustarlos a un modelo final.

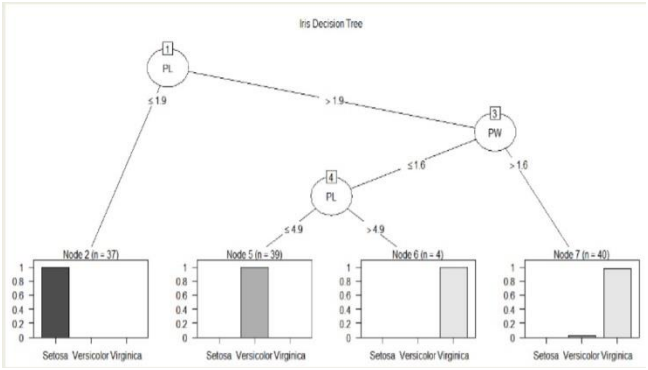

Figura 4. Figura vectorizada de árbol C5.0 [10]

4. DISEÑO DE LOS ALGORITMOS

En lo que sigue, explicamos la estructura de los datos y los algoritmos utilizados en este trabajo. La implementación del algoritmo y la estructura de datos se encuentra disponible en GitHub (https://github.com/jsguerrah/ST0245-001/tree/master/proyecto).

4.1 Estructura de los datos

La estructura de datos utilizada para hacer la predicción es el árbol de decisión binario, y, para unos estándares porcentuales mayores, se hace uso de los bosques aleatorios.

Un árbol binario es el resultado de la distribución de varios nodos provenientes de un nodo raíz: la intención es la de ir recorriendo las ramas que, en función de los atributos de una entrada, se determine su clasificación.

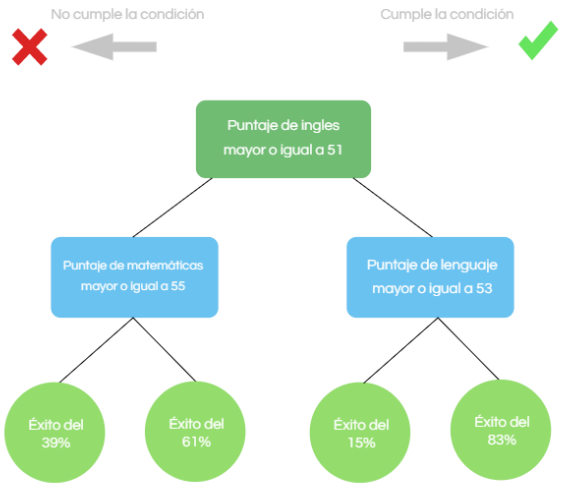

Figura 5. Un árbol de decisión binario para predecir Saber Pro basado en los resultados de Saber 11. Los nodos con color verde claro representan las hojas del árbol, mostrando el éxito por llegar a cada una de ellas.

4.2 Algoritmos

Retomando lo mencionado en la solución, nuestro árbol es de tipo CART, lo que hará que con el índice de Gini se busque la optimización de una elección y distribución. Esto conlleva a resultados más generalizados y con menor entropía.

4.2.1 Entrenamiento del modelo

Lo primero que se hace en el entrenamiento del modelo es la recolección de todos datos del *training set* en una estructura matricial: esto con el fin de tener un orden en la relación de cada atributo con su respectivo estudiante.

Luego de esto, el algoritmo procede a ubicarse sobre una determinada columna para conocer todas las posibles opciones de respuesta y, de esta forma, empezar a evaluar para dar con la mejor pregunta para un nodo. Sin embargo, con motivos de optimización en memoria, se implementó el uso de una lista de indicadores para la identificación de cada estudiante.

Para ser más precisos, la evaluación mencionada en el párrafo anterior consiste en la creación de dos listas a partir de una condición: los que la cumplen y los que no la cumplen. Nuevamente, con objeto de optimizar el consumo, se usaron diccionarios con claves de posibles respuestas a una columna. A cada una de estas subdivisiones del conjunto de datos se le calcula el índice de Gini, para luego encontrar el índice de Gini ponderado (apoyado en la información de la matriz que aún conserva todo el *training set*). Este último será quien se compare con los demás ponderados, siendo el de menor magnitud el que establece la mejor condición.

Partiendo del ponderado y de la condición elegida, se asigna el orden de la construcción de nuestro árbol de forma recursiva.

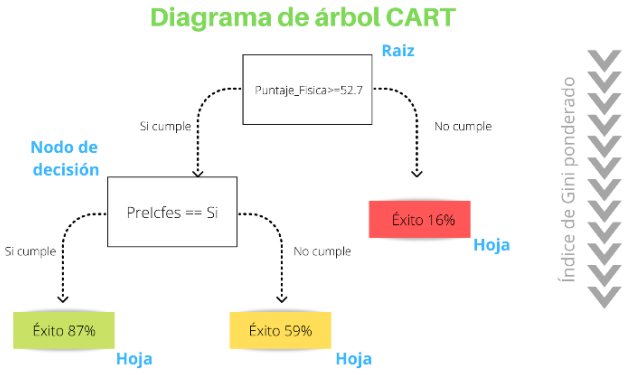

Figura 6. Entrenamiento de un árbol de decisión binario usando árbol *CART.* Los nodos verdes representan a aquellos con una alta probabilidad de éxito, los amarillos con una probabilidad media y los rojos con una baja probabilidad de éxito.

4.2.2 Algoritmo de prueba

Durante la construcción de los árboles, cada una de las preguntas utilizadas para fraccionar las listas de estudiantes se fueron almacenando en un atributo correspondiente para cada nodo del recorrido. Es de esta forma que, dado un estudiante, logramos ir haciendo un recorrido a lo largo del árbol con base en sus características. Al final del árbol, nos permitimos hacer la predicción según el porcentaje de éxito que represente la hoja a la que se llegó.

Ahora bien, el uso de los bosques reside en la predicción de la cantidad de árboles generados, siendo la predicción final el valor mas votado por el conjunto de árboles. Finalmente se verifica la certeza de la predicción, sumando al contador de verdaderos positivos, verdaderos negativos, falsos positivos o falsos negativos según el caso del estudiante.

4.3 Análisis de la complejidad de los algoritmos

Algoritmo	Complejidad de tiempo
Entrenar el árbol de decisión	$O(N*M*2^M)$
Validar el árbol de decisión	O(N*(M + N))

Tabla 2: Complejidad temporal de los algoritmos de entrenamiento y prueba. *N* representa el número de estudiante en el *data set*, mientras que *M* es el número de columnas que vamos a utilizar.

Algoritmo	Complejidad de memoria
Entrenar el árbol de decisión	O(N*M)
Validar el árbol de decisión	O(1)

Tabla 3: Complejidad de memoria de los algoritmos de entrenamiento y prueba. *N* representa el número de estudiante en el *data set*, mientras que *M* es el número de columnas que vamos a utilizar.

4.4 Criterios de diseño del algoritmo

El criterio para que los árboles fueran construidos de esa forma fue siempre preservar el espacio en memoria: evitar las matrices con datos repetidos era un gasto abismal, por lo que usar listas de referencias que nos apunten a los estudiantes de una única matriz resultaba en una mejora imprescindible.

Otro aspecto importante que mejorar era el de buscar la mejor condición: entre tantas columnas y respuestas buscar la mejor resultaba recorrer al menos cada columna N^2 veces, por lo que en vez de ver todo el tiempo quienes cumplían, simplemente se contaban asumiendo como claves para diccionarios las posibles respuestas, logrando reducir el recorrido a N veces.

Con estos aspectos, la construcción de cada árbol (sin limitar su altura) era menor a 20 segundos. Es por eso que, con un tiempo de entrenamiento bajo, se hacía viable la implementación de los bosques aleatorios para así poder alcanzar un mayor porcentaje de acierto en nuestros resultados.

En la siguiente sección podrán ver las evaluaciones respectivas después haber calibrado los bosques aleatorios para evitar el *overfitting*.

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. Es pertinente recordar que los resultados están basados en la creación de árboles aleatorios con mil árboles.

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.

	15000	***75000***	***135000***
Exactitud	0.75	0.74	0.74
Precisión	0.73	0.71	0.8
Sensibilidad	0.7	0.76	0.65

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.

	15000	***75000***	***135000***
Exactitud	0.73	0.72	0.75
Precisión	0.71	0.7	0.81
Sensibilidad	0.69	0.73	0.71

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

5.2 Tiempos de ejecución

En la siguiente tabla se registran los promedios en tiempo obtenidos durante la ejecución de los bosques aleatorios para tres conjuntos de datos diferentes.

	15000	***75000***	***135000***
Tiempo de entrenamiento	4.83 s	5.61 s	7.1 s
Tiempo de validación	50.4 s	219.83 s	493.53 s

Tabla 5: Tiempo de ejecución del algoritmo *CART* con bosques aleatorios, 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.

	15000	***75000***	***135000***
Consumo de memoria	93 MB	448 MB	705 MB

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

6. DISCUSIÓN DE LOS RESULTADOS

Para comenzar el análisis de la investigación, se puede decir que los resultados alcanzados fueron óptimos para su análisis, un resultado bastante satisfactorio para un primer paso dentro del campo del *Machine Learning*. Seguido de esto, consideremos los resultados de las métricas bastante asertivos, ya que con una magnitud tan dispersa de información es coherente una precisión relativamente similar. También es importante resaltar que durante el periodo de pruebas se trató de calibrar lo más posible la generación de árboles en el bosque para que alcanzara la mayor cantidad de porcentaje de aciertos posible de forma estable, alcanzando lo visto en las evaluaciones. Ahora bien, el hecho de que podamos haber generado una gran cantidad de árboles de forma rápida nos da seguridad de que el consumo de recursos esta optimizado para este tipo de pruebas.

Retomando, las variables utilizadas durante el desarrollo del proyecto demostraron que tanto los puntajes de inglés como castellano tienen una estrecha relación con el éxito de un estudiante. Es por esto que un mayor enfoque en el pensamiento crítico, análisis textual y el bilingüismo serian a futuro antecedentes de mayores índices de éxito en los estudiantes colombianos. Además, la predicción del algoritmo se puede aplicar tanto para dar becas como ayudar a los estudiantes con baja de probabilidad de éxito. Sin embargo, con base en los resultados de precisión, el algoritmo es mucho más eficiente para predecir con éxito que un estudiante no le irá bien en la prueba, lo que lo hace más útil para ayudar a esos estudiantes en riesgo.

6.1 Trabajos futuros

La mejora de los algoritmos se vuelve una necesidad para el desarrollo del perfil de ingeniero, así que – como lo fue durante el desarrollo de este proyecto – optar por una mayor ergonomía en cuanto al consumo y la obtención de mejores porcentajes de acierto han de ser pilares en el continuo aprendizaje para la aplicación de estructuras de *Machine Learning.*

AGRADECIMIENTOS

Agradecemos la colaboración con la estructura de bosques aleatorios a Juan David Echeverri y Octavio Vásquez, compañeros de curso en Estructuras de datos y algoritmos 1. De igual forma, agradecemos la asistencia en la selección de criterios para evitar el *overfitting* brindada por Kevin Sossa, compañero de curso en Estructuras de datos y algoritmos 1.

REFERENCIAS

1. Apolaya C., Espinosa A. and Barrientos A., Predicción del Rendimiento Académico en carreras de Computación utilizando Árboles de Decisión., IEEE transactions on journal name, mindata, 1-7
2. Cuji B., Gavilanes W., Sánchez R., Modelo predictivo de deserción estudiantil basado en arboles de decisión, Revista Espacios 38 (55) 17
3. Hernandez Gonzalez A. G., Melendez Armenta R. A., Luis Morales Rosales A., Garcia Barrientos A., Tecpanecatl Xihuitl J. L., Algredo I., Comparative Study of Algorithms to Predict the Desertion in the Students at the ITSM-Mexico, Cornell Chronicle., Retrieved diciembre 22, 2016, fromITSM-Mexico: DOI: 10.1109/TLA.2016.7795831
4. R. Alcover , J. Benlloch, P. Blesa, M. A. Calduch1 , M. Celma, C. Ferri, J. Hernández-Orallo, L. Iniesta, J. Más , M. J. Ramírez-Quintana, A. Robles , J. M. Valiente , M. J. Vicent, L. R. Zúnica, Análisis del rendimiento académico en los estudios de informática de la Universidad Politécnica de Valencia aplicando técnicas de minería de datos., Universidad Politécnica de Valencia 164-169
5. 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° Rev.investig.desarro.innov., 363-378
6. Vialard C., Chue J., Barrientos A., Victoria D., Estrella J., Peche J. P. and Ortigosa A., A Case Study: Data Mining Applied to Student Enrollment, researchgate. Retrieved June 11-13, 2010, from Universidad de Lima, Escuela Politecnica Superior, Universidad Autonoma de Madrid: https://www.researchgate.net/publication/221570513_A_Case_Study_Data_Mining_Applied_to_Student_Enrollment
7. Sefik Ilkin Serengil. (2017). A Step by Step ID3 Decision Tree Example. Recuperado de: https://sefiks.com/2017/11/20/a-step-by-step-id3-decision-tree-example/
8. Sefik Ilkin Serengil. (2018). A Step by Step C4.5 Decision Tree Example. Recuperado de: https://sefiks.com/2018/05/13/a-step-by-step-c4-5-decision-tree-example/
9. Sefik Ilkin Serengil. (2018). A Step by Step CART Decision Tree Example. Recuperado de: https://sefiks.com/2018/08/27/a-step-by-step-cart-decision-tree-example/
10. Nguyen A., Comparative Study of C5.0 and CART algorithms, Recuperado de: http://mercury.webster.edu/aleshunas/Support%20Materials/C4.5/Nguyen-Presentation%20Data%20mining.pdf
11. Loyola-Gonzalez O. (2019). Black-Box vs. White-Box: Understanding Their Advantages and Weaknesses From a Practical Point of View. IEEE Access, Access, IEEE, 7, 154096–154113. https://doi-org.ezproxy.eafit.edu.co/10.1109/ACCESS.2019.2949286