Artículo sobre la predicción del éxito académico en educación superior

Katherin Nathalia Allin Murillo **Alberto Andrés Díaz Mejía**

1) Este artículo nos habla de la búsqueda de un método para predecir la deserción de los alumnos de educación superior por medio de la implementación de árboles de decisión y minería de datos que se define como el proceso de extraer conocimiento útil y entendible, previamente desconocido, desde grandes cantidades de datos almacenados en distintos formatos. Por lo tanto la tarea fundamental de la minería de datos es encontrar modelos inteligibles.
a partir de los datos obtenidos, debido a que las instituciones de educación superior comprenden una gran cantidad de datos se utilizaron los datos de la Universidad Privada Cesar Vallejo comprendidos entre los años 2009 y 2013, con esto se pudo concluir que el algoritmo C5.0 es el más exitoso para lograr conseguir un patrón y así descubrir los alumnos con más probabilidad de deserción de los estudios superiores, a lo largo de la investigación también se pudo concluir que los graduados en carreras universitarias ganan el doble de aquellos que son graduados de educación secundaria y que ganan seis veces más que los desertores.

2) En este artículo se muestra un proceso de predicción para la deserción de los alumnos frente a los estudios de educación superior en América Latina y enfocándose en la región caribe con la implementación de árboles de decisión, para esto se usaron cinco etapas o procesos que se organizaron así respectivamente: selección, procesamiento, transformación, minería de datos y evaluación. Se trataron de implementar muchos algoritmos dentro de los cuales están: EquipAsso, J48, C4.5, ID3 y ADTree, pero el algoritmo que se implementó y que cumplía más con los objetivos fue CART de la herramienta R para hacer un árbol de decisión con cuatro niveles de profundidad y así poder concluir que la idea de deserción tiene mucho que ver con las notas de los alumnos.
Es así como se crea un sistema capaz de predecir y de arroja a aquellos estudiantes con más posibilidad de deserción utilizando su información personal y académica.

3) En este artículo se muestra el proceso de las pruebas con árboles de decisión para predecir los resultados de los alumnos en las pruebas saber 11º ya que Colombia en pruebas nacionales e internacionales está clasificada con un nivel de educación bajo en general, en este proyecto se usó un método descriptivo cuantitativo y no experimental, el ICFES ayudó aportando los datos socioeconómicos y académicos de los alumnos del 2015 y 2016 (datos más actualizados), los cuales se registraron posteriormente en una base de datos en la cual se implementaron herramientas de minería de datos para así poder conseguir un patrón entre los alumnos de buen y mal desempeño, para esto se utilizó el algoritmo J48 de la herramienta WEKA y así los resultados de este proyecto fueron muy satisfactorios.

4) Está investigación tuvo por objetivo estudiar el desempeño de los estudiantes colombianos en las pruebas saber pro implementando árboles de decisión, para esto se hizo un análisis estadístico de los estudiantes de programas profesionales en las pruebas Saber Pro 2011-2 y así poder descifrar un patrón o un ciclo que se forma en base a las pruebas que se practican anualmente, la metodología CRISP-DM, una base de datos limpia y la minería de datos que fueron fundamentales para la ejecución de todo este estudio. El ICFES aportó brindando mucha información de su base de datos y explicando que estas pruebas no se hacen con el objetivo de que todos los participantes estén en un mismo nivel, simplemente es para ver cómo se desenvuelven en cada una de las áreas y si estos resultados llegan a ser genéricos, pero algo que sí se pudo concluir de este estudio es que la posición socioeconómica influye mucho en la preparación para las pruebas y por ende en sus resultados.

Algoritmos para predicción del éxito académico en educación superior

1) El algoritmo ID3 utiliza arboles de forzamiento en donde se obtienen respuestas Si o no, dependiendo de un conjunto de alternativas, este algoritmo hace uso de recursividad ya que se mantiene constantemente devolviéndose para verificar el ejemplo base y a partir de este, crear subconjuntos.

En este algoritmo se llega a una solución cuando algún subconjunto está formado por un tipo de ejemplos, que luego sería clasificado como positivo o negativo, es decir, que se habrá llegado a un nodo hoja. Los atributos contenidos en este algoritmo, se le llaman nodos, pero cada nodo que contiene al menos 2 nodos(hijos) se le considera nodo padre. Los nodos hijos contienen posibles valores del nodo padre, que son llamados arcos.

En este algoritmo se deberá ingresar inicialmente Ejemplos, Atributo-objetivo y Atributos, de los cuales surgirán respuestas:

```
Si todos los ejemplos son positivos devolver un nodo positivo
   Si todos los ejemplos son negativos devolver un nodo negativo
   Si Atributos está vacío devolver el voto mayoritario del valor del
atributo objetivo en
Ejemplos
   En otro caso
         Sea A Atributo el MEJOR de atributos
         Para cada v valor del atributo hacer
               Sea Ejemplos(v) el subconjunto de ejemplos cuyo valor de
atributo A es v
               Si Ejemplos(v) está vacío devolver un nodo con el voto
mayoritario del
Atributo objetivo de Ejemplos
            Sino Devolver Id3(Ejemplos(v), Atributo-objetivo,
Atributos/{A}
```

2) *El algoritmo C4.5 es una extensión del algoritmo ID3, que al igual que él, es usado para generar arboles de decisión y además es usualmente referido por ser un clasificador estadístico que utiliza un criterio para la ganancia de información, a diferencia de la entropía utilizada en el ID3 que es usada para elegir a un atributo y de ahí dividir los datos.*

Este algoritmo lleva algunas mejoras como por ejemplo el manejo de atributos (continuos y discretos), ya que se crea un lumbral y en listas dividas ya anteriormente se ingresan los atributos cuyos valores son superiores, inferiores e iguales al lumbral. Otra de sus mejoras más importantes esta la capacidad de eliminar arboles después de su creación, logrando solo eliminar las ramas del árbol que no aportan, remplazándolos con nodos de hoja.

3) El algoritmo C5 es descendiente del C4.5, e igualmente generan arboles de decisión, sus modelos de C5 son capaces de dividir la muestra para así obtener una mayor ganancia de información obtenida en sus nodos, además solo es capaz de predecir un objetivo categórico (nominales u ordinales).Su proceso es debe a la división de las submuestras originales, que se vuelven a dividir basándose en otro campo y este termina solo y cuando le resulta imposible, volver a dividir a las submuestras (Nodos hijos). Este modelo solo puede ser entrenado mediante un campo categórico objetivo y uno o más campos de entrada y los campos que serán utilizados debe estar instanciados.

Esta generación comprende una mayor velocidad que se beneficia en la habilitación del proceso en paralelo.

4) El algoritmo CART, es capaz de generar arboles de decisión, procede del ámbito de la estadística y es capaz de trabajar con variables de todo tipo. El corte que realiza en cada nodo es generado por reglas binarias.

Su algoritmo está basado en la idea de impureza, ya que CART elige el corte que conducirá a un mayor decrecimiento de la impureza, y así conseguir descendientes homogéneos en una variable de respuesta.

Su funcionamiento es muy parecido al ID3, C4.5 y C5 y hace uso de, la entropía, el índice de Gini y el criterio de Twoing para el problema de clasificación, y para el problema de regresión se usa las varianzas de todos los nodos terminales.

Este algoritmo tiene como alternativa, el no parar.

Se le llama terminal al nodo que tiene un tamaño inferior a un umbral preestablecido.

Descripción diseño de algoritmo

Tenemos un método para obtener el gini ponderado, recibimos una matriz, una variable tipo int *y una tipo* String *que serán la columna actual. Cuando obtenemos la impureza de gini esta generaría un valor tipo* float*, debemos ver el tamaño de la matriz que nos entró anteriormente y procedemos a recorrerla; la variable fila en la cual se encuentra un* ArrayList *colocaríamos adentro como índice la columna actual y especificaríamos una condición la cual indicaría que si se encuentra un igual el tamaño de la derecha aumenta, de lo contrario no hace nada. Al terminar el proceso anterior obtendríamos el gini ponderado gracias a el recorrido para conseguir el tamaño de la izquierda multiplicado por el tamaño de la derecha.*

El segundo método se encarga de calcular la media de atributo como dice su nombre, va a recibir dos datos que serían: una matriz y una variable entera que va a ser la posición de la variable, vamos a usar un contador *que va a ser una variable tipo* float *y también usaremos un* acumulador *que sería la suma de atributos, ambas variables se inician en 0; luego, vamos a buscar en la fila hasta que encontremos m y cuando encontremos m se va a detener, es decir que dejará de contar. Si hay en la fila dos variables iguales se usa el acumulador y se va a ingresar esa variable. Luego de todo el proceso retornará la media que sería la suma del atributo divido entre el tamaño de este.*

Tenemos una clase Reader *que se encargará de leer nuestro árbol completo para arrojarnos los resultados de este, es decir que se encargará de mostrarnos nodos tanto positivos como negativos y todo el proceso elaborado.*

Por último, tenemos una clase llamada Tree *que tiene la mayor parte del funcionamiento de nuestro programa de predicción, ya que posee los métodos más importantes para que este se ejecute de la mejor manera, por ejemplo: hay un método llamado* countLabelsMatrix *que posee el contador, se encarga del conteo de longitudes en general y las probabilidades; el siguiente método se encarga de mostrar los Nodos, como el nodo izquierdo positivo, el nodo derecho positivo, el nodo izquierdo negativo y el nodo derecho negativo, esto lo hace mediante la anidación de ciclos; tenemos otros método que se encarga de añadir filas y columnas, actualizando al instante nuestro contador; el siguiente método llamado* promedioSimple *se encarga de encontrar el promedio simple de una matriz de forma simple o común, esto lo hace utilizando una función de contador y de suma para al final del método retornarnos el promedio; el método* AlertIsNumeric *se encarga de alertar si encuentra algún número que no corresponde a la secuencia.*

Para nuestro código usamos de base el algoritmo CART y los conocimientos adquiridos durante el semestre mediante el profesor, talleres, clases, laboratorios, parciales y otras dinámicas utilizadas por el docente.

Referencias

Daza Vergaray, A. (2016). Un modelo basado en árboles de decisión para predecir la deserción estudiantil en la Educación Superior Privada. Faculta de Ingeniería de Sistemas. Universidad Cesar Vallejo. Lima

Modelo predictivo de deserción estudiantil basado en arboles de decisión. (2017). Revista ESPACIOS. 38(55). ISSN 0798 1015

Timarán-Pereira, R., Caicedo-Zambrano, J., y Hidalgo-Troya, A. (2019). Árboles de decisión para predecir factores asociados al desempeño académico de estudiantes de bachillerato en las pruebas Saber 11º. Revista de Investigación, Desarrollo e Innovación. 9(2). 363-378. Doi: 10.19053/20278306.v9.n2.2019.9184

Timarán-Pereira, S. R., Hernández-Arteaga, I., Caicedo-Zambrano, S. J., Hidalgo-Troya, A. y Alvarado-Pérez, J. C. (2016). Descubrimiento de patrones de desempeño académico con árboles de decisión en las competencias genéricas de la formación profesional. Bogotá: Ediciones Universidad Cooperativa de Colombia. Doi: http://dx.doi.org/10.16925/9789587600490

Data Mining. Data Mining con Arboles de Decisión [online] Avaliable at: <https://web.fdi.ucm.es/posgrado/conferencias/JorgeMartin-slides.pdf>

Es.wikipedia.org. 2020. Algoritmo ID3. [Online] Available at: <https://es.wikipedia.org/wiki/Algoritmo_ID3> [Accessed 25 Abril 2020].

Es.wikipedia.org. 2020. C4.5. [online] Available at: <https://es.wikipedia.org/wiki/C4.5> [Accessed 8 February 2020].

IBM Knowledge Center. Nodo C5.0 [online] Available at: <https://www.ibm.com/support/knowledgecenter/es/SS3RA7_sub/modeler_mainhelp_client_ddita/clementine/c50node_general.html>