Estructuras de datos y algoritmos

Docente

Gloria Stella Sepulveda Cossio

Proyecto Final

Entrega 2

Índice

Modelo de Machine Learning para predecir resultados en pruebas Saber Pro

Diego Velásquez V Juan Felipe Agudelo V Jose Manuel Ramírez G

Steven Oviedo A

9 de octubre de 2020

Resumen

En la presente entrega daremos revisión a los problemas que se presentaron en la carga de datos, explicaremos las soluciones obtenidas y mencionaremos las opciones de código que tenemos para iniciar con la estructura de nuestro modelo de Machine Learning.

1. Carga de datos

En la anterior entrega, tuvimos algunos inconvenientes al realizar la lectura de los archivos CSV, sin embargo, en la monitoria del martes 6 de septiembre finalmente pudimos leer los datasets de forma exitosa con las siguientes soluciones:

Actualizar pip y pandas: Desde el cmd actualizamos el pip y la librería de pandas, de esta forma logramos solucionar el inconveniente en uno de los computadores.

Cambiar de IDE: Consideramos la posibilidad de que el problema se deba a la forma en que instalamos el Visual Studio Code, por lo que intentamos implementar el mismo código en otros dos editores (Spyder de Anaconda y Sublime Code) y logramos realizar la carga.

La complejidad del código que implementamos para leer los DataSets es de O(n), ya que dado un DataSet de tamaño nxn, el método tendrá que recorrer cada dimensión del archivo para poder leerlo.

2. ¿Que algoritmo utilizaremos para el análisis de datos?

2.1. CART (Classification and Regression Trees)

El algoritmo CART consiste en un árbol binario que clasifica variables tanto categóricas (ordinales o nominales) como continuas o numéricas. Esto se logra por medio de la creación de ramificaciones y nodos para cada una de las hojas hijas que representen una predicción sobre el dato que se evalúa basándose en predicciones dadas por anteriores nodos. Cada nodo representa una pregunta y sus hojas representan una predicción.

Sin embargo, dado que cada clasificación es producto de una predicción. Estas pueden producir resultados incorrectos como clasificar un caso en la clase que no le corresponde. Esto se conoce como impureza y representa la cantidad de casos mal clasificados con respecto a los resultados esperados. La impureza a pesar de que tiene varios métodos para su medición, siempre buscamos minimizarla para obtener las predicciones más fiables de la manera que mostraremos a continuación.

Matemática del algoritmo: En el algoritmo CART, hay tres criterios de decisión para la división y selección de la mejor opción, estos son: índice de Gini, Criterio de Twoing y criterio de ordenación de Towing.

• De una forma muy simplista, el índice de Gini se puede calcular como:

$$1-(Probabilidaddeelegirelladoderechodelahoja)^2-(Probabilidaddeelegirelladoizquierdodelahoja)^2 \tag{1}$$

• El criterio de Twoing se puede calcular con la siguiente formula:

$$\Delta i(s,t) = p_L p_R \left[\sum_j | \, p(j \mid t_L) - p(j \mid t_R) \, | \right]^2 .$$

Donde Delta i(s,t) es la probabilidad de la mejor elección de una solución para un nodo t dada una separación s.

• El criterio de ordenación de Trowing se puede calcular con la siguiente formula:

$$\Delta i(s,t) = i(t) - p_L i(t_L) - p_R i(t_R) = p_L p_R \left[\sum_{j \in C_1} \{p(j \mid t_L) - p(j \mid t_R)\} \right]^2$$

Reglas de parada: Estos son los criterios que se usan para decidir si el algoritmo debe para en algún punto:

• Si todos los casos posibles que tiene un nodo son idénticos, se dice que este es puro y la ejecución del algoritmo debe finalizar.

• Si todos los casos en un nodo tienen la misma probabilidad para cada predictor, el nodo no se separa y el algoritmo debe finalizar.

• Si el tamaño del árbol supera un tope máximo, dado por la capacidad de la maquina en la que se esté trabajando, el proceso de crecimiento del árbol se detendrá al igual que el algoritmo.

• Si el tamaño de un nodo es menor que el tamaño de nodo mínimo que ha sido especificado por el usuario, el nodo no se dividirá.

2.2. ID3

El algoritmo ID3 busca la creación de hipótesis o leyes sobre un tema, a partir de un conjunto de ejemplos sobre el mismo. El conjunto de ejemplos debe estar formado por unas tuplas de valores, a los cuales llamaremos atributos, en el que uno de ellos es el objetivo (atributo a clasificar), el cual es de tipo binario (positivo-negativo, sí-no, etc.). De esta forma el algoritmo trata de obtener hipótesis que clasifiquen el fenómeno para próximas instancias. Esto lo hace a través de la construcción de un árbol de decisión, los cuales tendrán 3 elementos:

• Nodos: los cuales contienen los atributos.

• Arcos: que contiene los posibles valores del nodo principal.

• Hojas: nodos que clasifican el ejemplo como positivo o negativo.

Para calcular la entropía se usa:

$$Entropy = \frac{-p}{p+n} log_2(\frac{p}{p+n}) - \frac{n}{p+n} log_2(\frac{n}{p+n})$$

Donde p son lo ejemplos positivos y n los negativos. Cabe recalcar que se debe establecer si el logaritmo es positivo o negativo.

2.3. C4.5

Para hablar del algoritmo C4.5 se hace necesario explicar un poco de la Familia TDIDT (top down decisión trees) que son aquellos que pertenecen a los métodos inductivos del aprendizaje automático que aprenden a partir de ejemplos preclasificados. En minería de datos, se utilizan para modelar las clasificaciones en los datos mediante arboles de decisión.

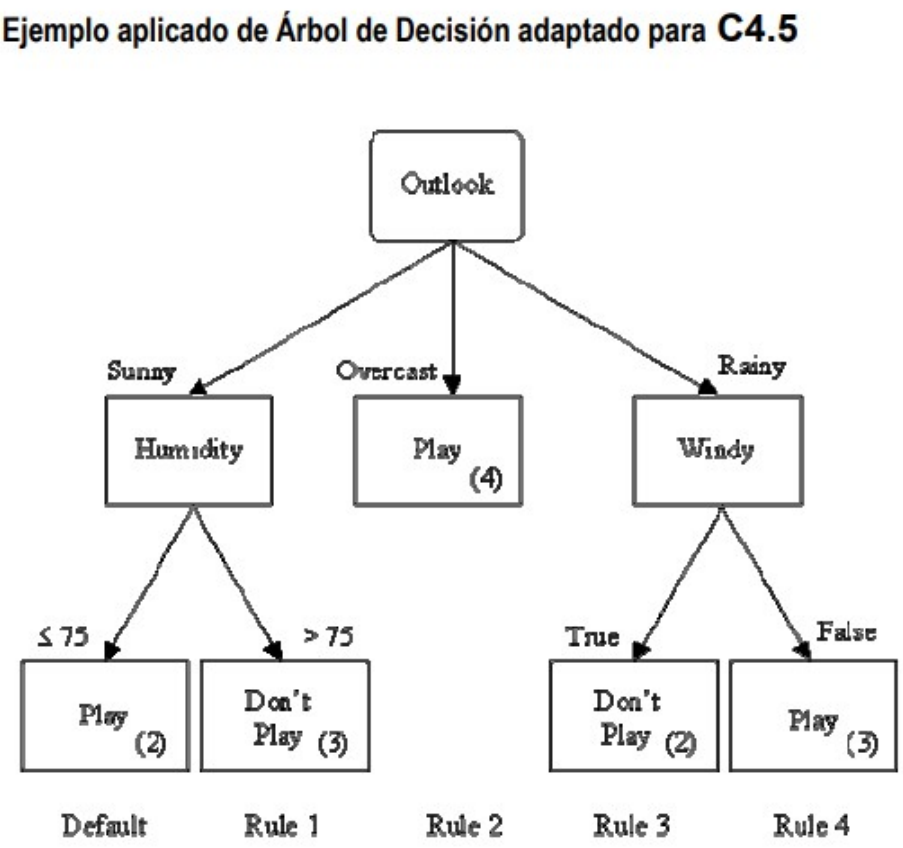
Construcción de los árboles de decisión:

Los árboles TDIDT, a los cuales pertenecen los generados por el ID3 u el C4.5, se construyen a partir del método de Hunt. El esqueleto para construir un árbol de decisión a partir de un conjunto T de datos de entrenamiento es muy simple, por ejemplo: **Sean las clases C1, C2, . . . , Ck. Existen 3 posibilidades:**

• T contiene uno o más casos, todos pertenecientes a una única clase Cj: El árbol de decisión para T es una hoja identificando a la clase Cj.

• T no contiene ningún caso: El árbol de decisión es una hoja, pero la clase asociada debe ser determinada por toda la información que no pertenece a T. Por ejemplo, una hoja puede escogerse de acuerdo a conocimientos de base del dominio, como ser la clase mayoritaria.

• T contiene casos pertenecientes a varias variables: en este caso, la idea es refinar T en subconjuntos de casos que tiendan, o aparenten tender a una colección de datos pertenecientes a una única clase. Se elige una prueba basada en un único.



El algoritmo C4.5 genera un árbol de decisión a partir de los datos mediante particiones realizadas recursivamente. El árbol se construye mediante la estrategia de profundidad primero (depth-first)

El algoritmo considera todas las pruebas posibles que pueden dividir el conjunto de datos y selecciona la prueba que resulta en la mayor ganancia de información. Para cada atributo discreto, se considera una prueba con n resultados. Siendo n el número de valores posibles que puede tomar el atributo. Por otra parte, para cada atributo continuo se realiza una prueba binaria sobre cada uno de los valores que toma el atributo en los datos. En cada nodo, el sistema debe decidir cual prueba escoge para dividir los datos.

Los tres tipos de pruebas posibles propuestas por el C4.5 son:

• La prueba estándar para las variables discretas, con un resultado y una rama para cada valor posible de la variable.

• Una prueba más compleja, basada en una variable discreta, en donde los valores posibles son asignados a un número de variable de grupos con un resultado posible para cada grupo,

en lugar de para cada valor.

• Si una variable A tiene valores numéricos continuos, se realiza una prueba binaria con resultados A<=Z y A>Z, para lo cual debe determinarse el valor límite Z.

Características del algoritmo:

• Permite trabajar con valores continuos para los atributos, separando los posibles resultados en 2 ramas Ai<=N y Ai>N.

• Los árboles son menos frondosos, ya que cada hoja cubre una distribución de clases no una clase en particular.

• Utiliza el método "divide y vencerás"para generar el árbol de decisión inicial a partir de un conjunto de datos de entrenamiento.

• Se basa en la utilización del criterio de proporción de ganancia (gain ratio), definido como I(Xi,C)/H(Xi). De esta manera se consigue evitar que las variables con mayor número de posibles valores salgan beneficiadas en la selección.

• Es Recursivo.

El algoritmo C4.5 puede mejorarse evitando el sobreajuste de los datos, mejorando la eficiencia computacional, manejando atributos continuos, entre otros.

3. Conclusiones

Como equipo, conciliamos trabajar con el algoritmo CART, puesto que además de ser un método bastante flexible, CART no hace asunciones sobre la distribución de ningún tipo (ya sea de las variables predictoras como de la variable criterio) Además. no está afectado por valores extremos,colinealidad, heterocedasticidad que afecten los procedimientos establecidos. Los valores outliers pueden ser aislados en un nodo y no tienen ningún efecto en la división. El algoritmo CART (con el conjunto de variables predictoras o independientes) nos dará los resultados usando solo las variables más importantes, lo que con el objetivo que tiene el proyecto, consideramos lo más apropiado.

4. Bibliografía - Web-grafía

• Breiman, L., Friedman, J.H., Olshen, R., and Stone, C.J., 1984. Classification and Regression Tree Wadsworth Brooks/Cole Advanced Books Software, Pacific California.

• Breiman, L., Friedman, R. Olshen, A. y Stone, C.(1984). Classification and regression trees. Belmont. 1er. Ed. Calif: Wadsworth. USA.

• https://www.youtube.com/watch?v=7VeUPuFGJHkt=734s

• https://www.youtube.com/watch?v=7VeUPuFGJHkt=734s

• https://www.youtube.com/watch?v=oSWTXtMglKE

• https://stackoverflow.com/questions/20092632/can-someone-explain-me-the-difference-between-id3-and-cart-algorithm

• https://towardsdatascience.com/decision-trees-for-classification-id3-algorithm-explained-89df76e72df1

• https://www.youtube.com/watch?v=8-vHunc4k8s

• http://www2.cs.uregina.ca/ dbd/cs831/notes/ml/dtrees/c4.5/tutorial.html

• http://ciberconta.unizar.es/Biblioteca/0007/arboles.html

• http://www.itnuevolaredo.edu.mx/takeyas/apuntes/Inteligencia20Artificial/Apuntes/tareas alumnos/C4.5/C4.5(2005-II-B)