Procesamiento de datasets de entrenamiento y test con árboles de decisión (ID3)

Abad Freddy, Calle Elvis, Cárdenas Paola

Abstract:The realization of this report was given with the purpose of understanding the mechanisms used in Artificial Intelligence such as a decision tree, which are the basis of computer learning, suggesting fundamental issues such as: ID3 trees, pruning of these trees, through the use of Python libraries

Keywords: Trees, ID3, Decision, Sklearn

Resumen: La realización del presente informe se dio con la finalidad de entender los mecanismos usados en Inteligencia Artificial tales como un árbol de decision, que son la base del aprendizaje de computador, sugiriendo temas fundamentales tales como: Árboles ID3, poda de estos árboles, mediante el uso de librerías de Python

Palabras Clave: Árboles, ID3, Decision, Sklearn

1. Análisis de los Datos

Indicar cuántos data points en total (cardinalidad) tiene el input space (|X|) en este problema.

Para conocerlo primero debemos saber la cantidad de los posibles valores que puede tomar cada atributo. Esto es:

a. buying: low, med, high, vhigh (4)
b. maint: low, med, high, vhigh (4)
c. dors: 2, 3, 4, 5more (4)
d. persons: 2, 4, more (3)
e. lug boot: small, med, big (3)
f. safety: low, med, high (3)

Ahora bien, el total de datas points está representado por las combinaciones que se pueden con esta información, esto es:

Número de data points: 4*4*4*3*3*3 = 1728

Indicar el porcentaje de datos que le fueron entrenados del total del input space.
Dado el tamaño del input Space así como el tamaño del Dataset original se puede

calcular el porcentaje de datos entrenados se puede calcular mediante una regla de tres simple.

Porcentaje =
$(Tam\ Dataset\ \div\ Tam\ InputSpace) * 100 = (1157\ \div 1728) * 100 = 66.95\%$
Porcentaje = 66.95%

Encontrar "a mano" el nodo raíz y uno de los nodos de profundidad uno de un árbol de decisión para este problema.
Para encontrar el atributo que mejor clasifica los datos, que sería el nodo raíz se debe obtener la Información de Ganancia de cada uno y ver cual de ellos es el que posee mayor ganancia, que mide que tan bien un atributo separa los datos de entrenamiento dada la función objetivo que poseen los datos.

Para resolver el problema debemos considerar la fórmula:

$$Entropía(S) = \sum_{i=1}^{c}(-p_i * log_2(p_i))$$

y la fórmula de ganancia siguiente

$$\equiv \text{Entropía}(S) - \sum_{v \in Valores(A)} \frac{|S_v|}{|S|}\text{Entropía}(S_v)$$

Procediendo a calcular la entropía para cada uno de los atributos vemos que el atributo que menor entropía tiene en uno de sus valores es el atributo **Safety en el valor low,** estos cálculos se pueden ver con más detalle en el archivo Calculo Entropía.xlsx adjunto a este proyecto.

2. Procesamiento de la información y entrenamiento

A. *Uso de la librería SKLEARN y DECISIONTREE*

Dado el dataset de Entrenamiento, surgió la posibilidad de darle un preprocesamiento, convirtiendo todos los datos alfanuméricos a numéricos, permitiéndose el uso ágil de herramientas de la librería SKLEARN Y PANDAS.
Investigando más a fondo la librería PANDAS ofrece la función ***get_dummies***() que convierte la variable categórica en variables ficticias o indicadoras, evitándose el pre-procesamiento y usando directamente en tiempo de ejecución del arbol de decision. A esta función solo se le pasa el dataset al cual convertir sus parámetros a pseudo-indicadores.

Uno de las recomendaciones principales es dividir el dataset en un train, validation y test dataset de este, para esta finalidad se usa la función perteneciente a SKLEARN, llamado ***train_test_split***(), el cual nos ayuda a dividir matrices en submatrices aleatorios y subconjuntos de prueba, que nos ayuda a validar la entrada. A esta función se le pasa como parámetros, las filas de la matriz total del dataset, del cual se tomará, la proporción del conjunto de datos para incluir en la división del tren.
Como se anticipó en este literal del informe, no se hizo preprocesamiento de datos, en el caso del uso de librerías y no la herramienta llamada 'WEKA'. Así en este apartado se usa la función ***get_dummies***(subMatriz) así convirtiendo en pseudo-indicadores los datos. Obteniendo así en dos submatrices, los datos convertidos, para emplear la función DesicionTree(), que proporciona un clasificador de árbol de decisión que consiste en pruebas de características que están organizadas en forma de árbol. La prueba de características asociada con el nodo raíz es una que se puede esperar para eliminar la ambigüedad máxima de las diferentes etiquetas de clase posibles para un nuevo registro de datos. Desde el nodo raíz se cuelga un nodo hijo para cada resultado posible de la prueba de características en la raíz. Esta regla de desambiguación de la etiqueta de clase máxima se aplica en los nodos secundarios recursivamente hasta que llegue a los nodos de la hoja. Un nodo hoja puede corresponder a la profundidad máxima deseada para el árbol de decisión o al caso cuando no hay nada más que ganar mediante una prueba de característica en el nodo.
A esta función se le proporciona los parámetros *criterio,* el cual mide la calidad de una división, en este caso se emplea *'entropy'* para la ganancia de información, además de max_depth() que ofrece la profundidad máxima del árbol, si no se define este parámetro, los nodos se expanden hasta que todas las hojas son puras o hasta que todas las hojas contienen menos de muestras min_samples_split.
Definido el arbol de decision se le ajusta con la función ***fit***() , el cual construye un clasificador de árbol de decisión del conjunto de entrenamiento (X, y).
Dado este árbol de decision, finalmente se exporta este árbol de decision en formato ODT, y convirtiéndolo a formato png con las funciones:

tree.***export_graphviz***(ad, out_file = archivo_dot)
check_call(['dot','-Tpng',nombre,'-o',nombre+'.png'])

Brevemente se explica el proceso de entrenamiento del arbol de decision, mediante de herramientas de programacion.
Finalmente, dados los datos encontrados con la herramienta WEKA, se sabe que la profundidad del árbol de decision debe de ser de nivel 6.
Como prueba se hicieron distintos arboles de decision de distinta profundidad de cada una, determinando que un árbol de profundidad máxima llega al nivel 12, con los datos de entrenamiento, todos los archivos de prueba se adjuntan en el archivo .rar para una mejor visualización.

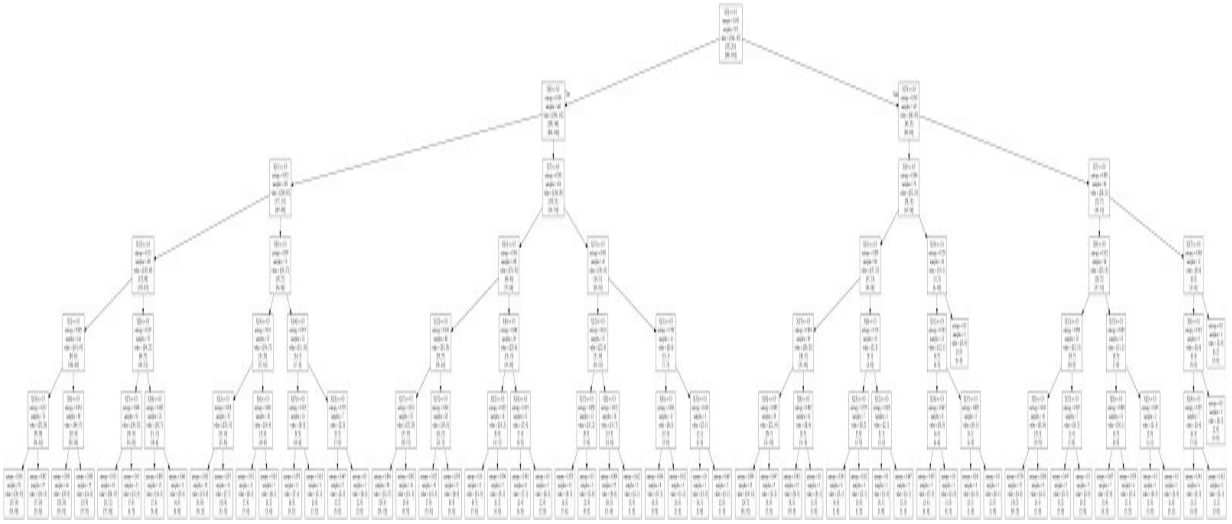


Figura 1. Arbol de decision de profundidad 6, el archivo se adiciona en el .rar para una mejor visualización.

B. Uso de la herramienta **"Decision tree learning applet"**

Esta herramienta para aplicaciones de machine Learning permite el entrenamiento y el test de datos para árboles de decisión con el algoritmo ID3.

El proceso a seguir es cargar los datos de entrenamiento, estos no necesitaron de ningún ajuste manual, pues la herramienta no tiene ningún problema al leer los datos tal y como están, se agregan los nombres de los atributos para una mejor distinción visual de los mismos, una vez que se haya generado el árbol.

Se ejecuta el algoritmo para entrenar y generar el árbol, seleccionando los atributos por su ganancia, y como resultado se obtiene:

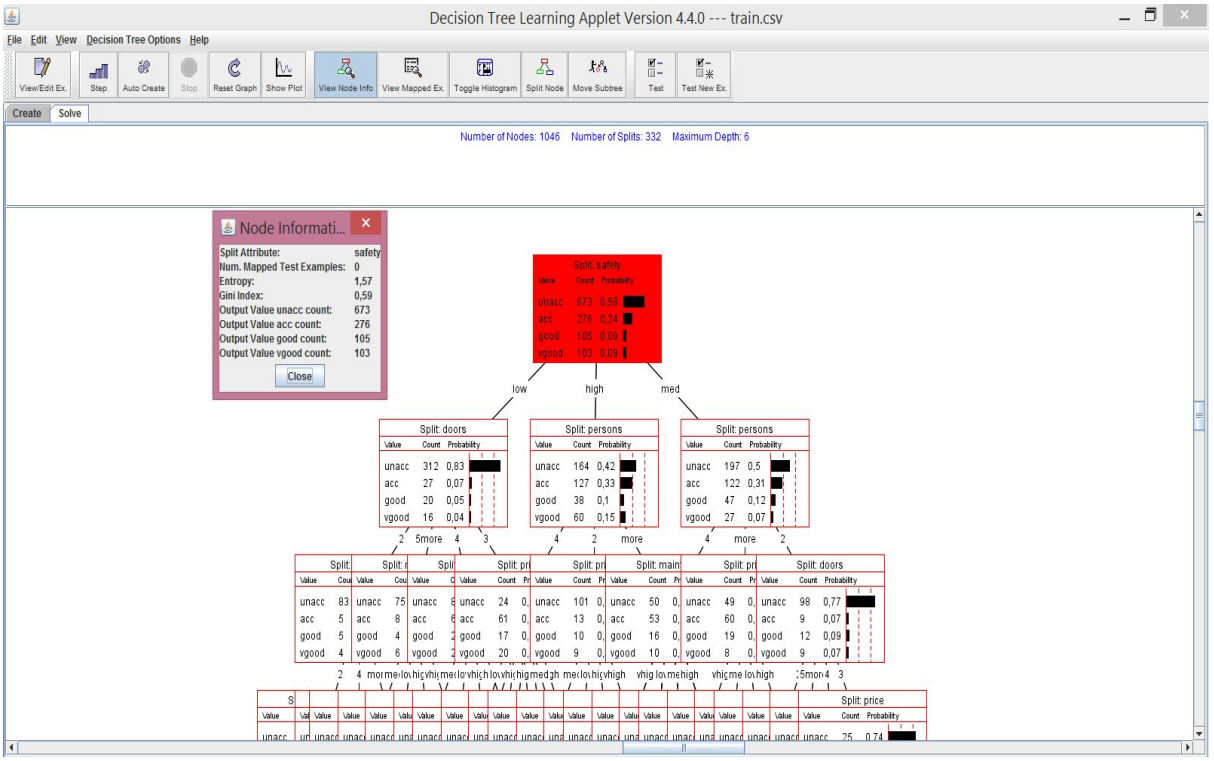


Figura 2. Arbol de decision de profundidad 6

Número de Nodos: 1046
Número de Divisiones: 332
Profundidad:6

Luego se procedió a hacer el test, con datos de la BD original la cual se procesó en excel para poder hacer el match con el test real de datos, otorgado en este curso, al realizar el test se obtuvo la siguiente gráfica de errores.

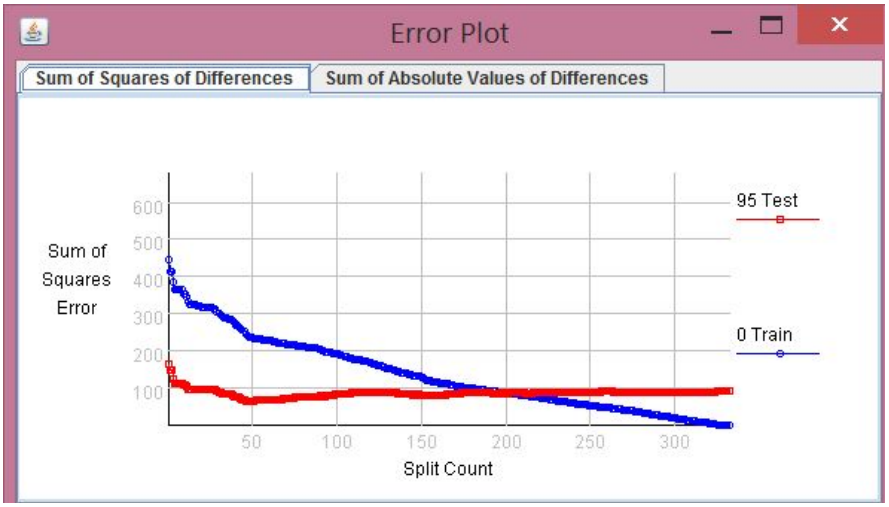
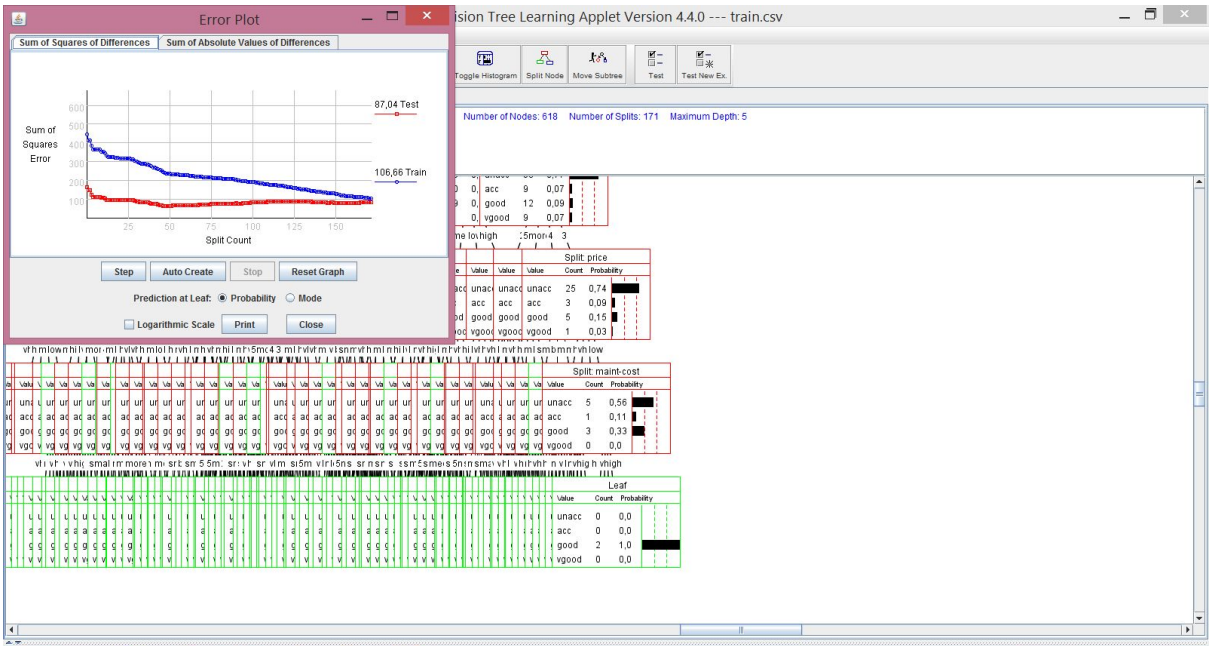


Figura 3. Gráfica de error

Al ver este resultado vemos que hubo algo de sobreajuste (overfitting) con la profundidad 5 por lo que limitamos esta profundidad para obtener mejores resultados.



Bibliográficas:

[1]Anónimo, “Decision Tree”, 2016. Ref: https://www.lucidchart.com/pages/es/qu%C3%A9-es-un-diagrama-de-%C3%A1rbol-de-decisi%C3%B3n

[2]Colaboradores de Wikipedia, “Decision Tree” , 2017. Ref: https://es.wikipedia.org/w/index.php?title=%C3%81rbol_de_decisi%C3%B3n&oldid=101577422

[3]Aprendizaje basado en árboles de decisión https://www.slideshare.net/angenio2/machine-learning-aprendizaje-basado-en-rboles-de-decisin

[4]Decision tree learning applet http://www.aispace.org/dTree/help/tutorial2.shtml

[5] Editores SCIKIT LEARN, “Documentation of scikit-learn 0.19.1”, 2017. Ref: http://scikit-learn.org/stable/documentation.html

[6] Editores Pandas Python, “pandas: powerful Python data analysis toolkit”, 2017. Ref: https://pandas.pydata.org/pandas-docs/stable/