Optativa IV. Machine Learning

Alumno: Gerardo Hernández Nava
Correo electrónico: ghernandez100@alumnos.uaq.mx

Árbol de decisión ID3 para clasificación de la base de datos TIC-TAC-TOC.

Tarea

I. Introducción.

El juego de tres en línea, también conocido como Tic-tac-toc o gato, es un juego de dos jugadores. Se trata de que se vayan turnando en marcar las nueve casillas de una cuadrícula de 3x3. Uno de ellos marca sus casillas con un tache y el segundo participante con un círculo. Gana el que logre colocar tres de sus marcas juntas en cualquier dirección.

Ahora bien, la base de datos que se va a ocupar en este proyecto es una base de datos que presenta más de 900 tableros llenos e indica si el participante que juega con la marca del tache ganó o no.

Para resolver este problema de clasificación se hará uso de un árbol de decisión ID3, el cual se basa en la estrategia de divide y vencerás.

II. Objetivo

El objetivo principal de esta práctica es implementar un árbol de decisión ID3 el cual ayudará a clasificar las jugadas realizadas en un tablero de tres en línea, también conocido como gato, para saber si el jugador que tira con el símbolo ‘x’ ha ganado o no.

III. Marco teórico.

A. Árboles de decisión

Los árboles de decisión son una técnica de aprendizaje automático supervisado y ha sido utilizada en múltiples ámbitos desde la inteligencia artificial hasta la economía. Esta técnica va tomando las decisiones siguiendo la estructura de un árbol. Los nodos intermedios representan soluciones y las hojas la predicción o clasificación que se está buscando (Martinez, 2020).

Las principales ventajas de este algoritmo es que es fácil de entender y explicar a personas que no están familiarizadas con técnicas de inteligencia artificial, se adapta a cualquier tipo de datos y descubre cuales son los atributos relevantes. Por otro lado, sus desventajas son que no extrapolan bien fuera de su rango de entrenamiento y tienden al sobreajuste (Martinez, 2020).

B. ID3

ID3 es el acrónimo de ‘inducción mediante árboles de decisión. Fue desarrollado por J. Ross Quinlan. La salida del algoritmo se puede presentar como como un grafo en forma de árbol cuyos componentes son:

- Nodo raíz: Es el nodo principal localizado en la parte superior. De este nodo parten ramas hacia otros nodos inferiores, de los cuales pueden salir más ramas hacia otros nodos.
- Nodos terminales: Como su nombre lo indica, son nodos donde termina el flujo y que ya no son raíz de ningún otro nodo, también puede denominarse como hoja. Estos nodos terminales contienen la clasificación a la cual pertenece el objeto que ha conducido hasta dicho nodo.
- Nodos intermedios: representan preguntas con respecto al valor de uno de los atributos.
- Ramas: representan las posibles respuestas que los atributos pueden tomar.

El objetivo principal del algoritmo ID3 es determinar, para un conjunto de datos, el atributo más importante, es decir, aquel que posea el mayor poder discriminatorio para dicho conjunto, el cual será denominado como nodo raíz; este atributo es usado para la clasificación de la lista de objetos, basados en los valores asociados con él mismo. Después de haber hecho la primera prueba de atributo se arrojará un resultado, el cual

es en sí mismo un nuevo problema de aprendizaje de árbol de decisión, con la diferencia que contará con menos ejemplos y un atributo menos, por lo que, cada atributo que se selecciona se descarta para la siguiente prueba (Rahman, 2021).

Este proceso se realiza mediante el apoyo de dos operaciones de probabilidad, la entropía y la ganancia. La entropía es una medida de incertidumbre y es usado para ayudar a decidir qué atributo debe ser el siguiente en seleccionarse. Se calcula de la siguiente forma:

$$\text{Entropia}(S) = \sum_{i=1}^{n} -p_i \log_2 p_i$$

Mientras que la ganancia de información, es una medida de discriminación e indica el siguiente atributo que debe ser seleccionado y se calcula de la siguiente forma:

$$\text{Gan Inf}(S, A) = \text{Entropia}(S) - \sum_{v \in V(A)} \frac{|Sv|}{|S|} \text{Entropia}(Sv)$$

IV. Metodología

1. Se importaron las librerías necesarias para poder implementar el algoritmo, como pandas, math, numpy y operator.
2. Se comienzan a declarar las funciones en las cuales se va a basar nuestro algoritmo. Las funciones fueron las siguientes:
    a. *contar_clases*, como lo dice su nombre esta función cuenta el número de instancias que corresponden a cada clase. Como entrada la función recibe la columna de atributo de decisión de la base de datos a analizar, o si se quiere ver de otra forma 'y', y las observaciones que se tienen en el atributo de decisión o aquellas que se desean contar. Y como salida, la función retorna una arreglo con el número de observaciones contadas.
    b. *calcular_entropía*, está función arroja como resultado la entropía de decisión del sistema que se esté analizando
    c. *ganancia_atributo*, se calcula la ganancia de cada atributo respecto al sistema tomando en cuenta la entropía de decisión.

d. *nodo_raiz*, dicha función selecciona el atributo de la base de datos que tenga mayor ganancia respecto a los demás, para que de esta forma sea el nodo raíz del árbol que se está formando o sea el siguiente. Retorna el nombre del nodo (atributo) y la entropía por la que fue seleccionado.
e. *nodos_intermedios*, en esta función se generan sub-bases de datos, esto es que la función recibe como parámetros, la base de datos de atributos con los cuales se debe realizar la clasificación, la columna de atributo de decisión, el atributo bajo el cual se va a realizar el análisis y las observaciones que contiene dicho atributo; para después encontrar todos aquellos índices en los cuales se puede encontrar la primera observación del atributo y una vez que se tienen los índices se extraen de la base de datos original y se haría la primer sub-base de datos y también se extrae de la columna del atributo de decisión. Esto se repite hasta terminar con todas las observaciones.
f. *fin_rama*, ayuda a determinar si la rama que se generó en el árbol es final o mejor dicho termina en una hoja, se dice que es una hoja cuando todas las instancias que llegan hasta ese punto pertenecen a la misma clase de forma que se puede tomar una decisión. Retorna un cero o uno, cero cuando no la rama no es final y el árbol debe continuar y uno cuando la rama termina en una hoja,
g. *arbol_id3*, es una función de recursividad la cual ayudará a ir formando la estructura del árbol id3 mandando a llamar todas las funciones antes descritas y así misma cada que la rama del árbol no termine en una hoja, es decir, se haya tomado una decisión. Retorna la estructura del árbol en un diccionario.
h. Para finalizar se realizan las funciones ya explicadas en trabajos anteriores para evaluar el algoritmo como: exactitud, precisión, sensitividad, especificidad y puntaje F1; así como la función para dividir la base de datos en entrenamiento y prueba e de manera 80% para el entrenamiento y el 20% para la prueba.

3. Una vez teniendo todas la funciones definidas se importó la base de datos con la cual se trabajó mediante pandas y la función read_csv.
4. Se analizó la base de datos, para saber sus dimensiones, atributos, instancias, observaciones por atributos y si habían valores faltantes.
5. Al ver que la base de datos se encontraba completa y no era necesario hacerle algún preprocesamiento se procedió a dividirla en en los atributos que nos brindaran la información para saber a qué clase pertenecen, es decir, 'x' y la columna del atributo de decisión que será igual a 'y'.
6. Se inicializa una variable nueva tipo diccionario la cual será el arból que se construirá y se manda a llamar la función *arbol_id3*.
7. Ya que se diseñó el árbol se imprime para poder realizar una nueva función en la cual se irán anidando sentencias elif para poder construir el árbol.
8. Como la función generada en el paso anterior solo clasifica una instancia de la base de datos a las vez se realiza una función que le auxilie a ir pasando cada una de ellas a la vez y guardar su clasificación en un array para evaluarlo posteriormente.
9. Se evalúa el entrenamiento del árbol y se calculan sus cifras de mérito.
10. Se clasifica el conjunto de prueba y se obtienen sus cifras de mérito.

V. Resultados

Tras un análisis se puede determinar que la base de datos cuenta con nueve atributos, denominados V1, V2, V3, V4, V5, V6, V7, V8, V9, donde cada uno de ellos corresponde a una casilla de un tablero de gato.

Tabla 1. Primeras cinco renglones de la base de datos, donde se pueden observar los atributos, observaciones y clases que contiene.

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
0	x	x	x	x	o	o	x	o	o	positive
1	x	x	x	x	o	o	o	x	o	positive
2	x	x	x	x	o	o	o	o	x	positive
3	x	x	x	x	o	o	o	b	b	positive
4	x	x	x	x	o	o	b	o	b	positive

A su vez cada una de estos atributos contaba con 3 observaciones, ‘x’, ‘o’ y ‘b’. Todas las instancias estaban completas por lo cual no fue necesario realizar imputación alguna.

Tabla 2. Nos muestra información de la base de datos como el número de instancias, número de observaciones por atributo, valor máximo que se presenta en cada atributo y su frecuencia.

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
count	958	958	958	958	958	958	958	958	958	958
unique	3	3	3	3	3	3	3	3	3	2
top	x	x	x	x	x	x	x	x	x	positive
freq	418	378	418	378	458	378	418	378	418	626

La base de datos contaba con décimo atributo el cual es el atributo de decisión denominado V10, este atributo indica cuando el participante que juega con la marca x gana.

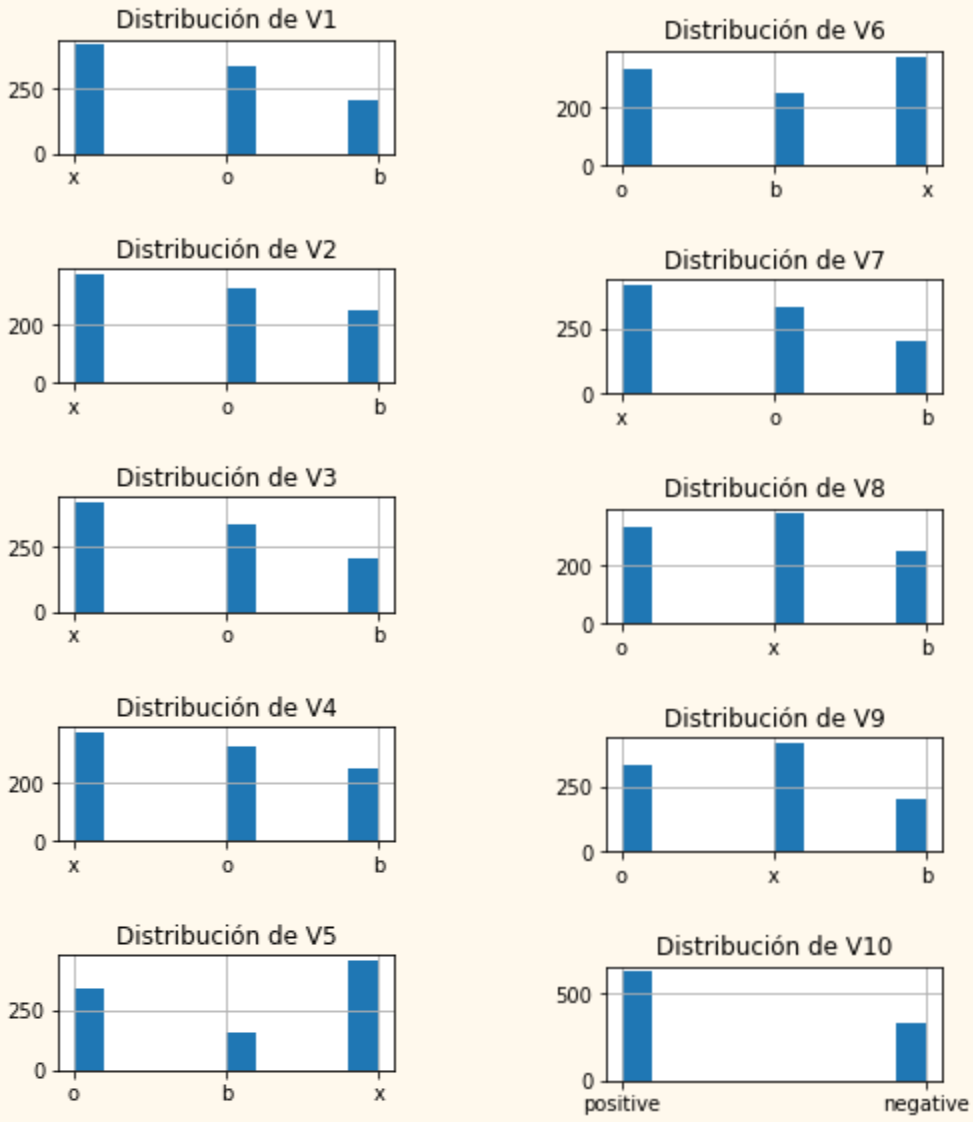

Figura 1. Se puede observar los 10 atributos de la base de datos con sus respectivas observaciones, así como la distribución que presentan.

A continuación, se muestran los resultados obtenidos a partir de la modelación del algoritmo id3.

Tabla 3. En esta tabla se puede apreciar el buen rendimiento que se tuvo al implementar el árbol de decisión tanto en el entrenamiento como en la prueba.

Cifra de mérito	Entrenamiento	Prueba
Exactitud	100 %	96.85 %
Precisión	100 %	95.79 %
Sensitividad	100 %	99.13 %

Especificidad	100 %	93.42 %
Puntaje F1	100 %	97.43 %

VI. Conclusiones

Finalmente, se puede concluir que un árbol de decisión es un algoritmo que resuelve este problema de una manera muy eficiente, ya que tanto en el entrenamiento como en la prueba las cifras de mérito son muy buenas.

Un aspecto negativo, por la forma en la que se programa el árbol de decisión, es la complejidad de pasar el diccionario a una función con comandos elif anidados para poder evaluar la base de datos. Lo cual no facilita el ir modificando el árbol para una evaluación por k fold o algo similar.

VII. Referencias

Martinez, J. (2020, September 19). *Árboles de Decisión con ejemplos en Python*. IArtificial.net. Retrieved June 12, 2022, from https://www.iartificial.net/arboles-de-decision-con-ejemplos-en-python/

Rahman, T. (2021, March 27). *Step by Step Decision Tree: ID3 Algorithm From Scratch in Python [No Fancy Library]*. Medium. Retrieved June 12, 2022, from https://medium.com/geekculture/step-by-step-decision-tree-id3-algorithm-from-scratch-in-python-no-fancy-library-4822bbfdd88f