

Universidad Nacional San Agustín de Arequipa

FACULTAD DE INGENIERIAS DE PRODUCCION Y SERVICIOS

ESCUELA PROFESIONAL DE INGENIERIA
DE SISTEMAS

Inteligencia Artificial

Alumno:

Fuentes Paredes Nelson Alejandro

Mayo 2020

```
[ ]:
```

1 Importado Librerias

```
[1]: import pprint
import math
import pandas as pd
```

2 Metodo para calcular la Entriopia

```
[2]: def entropy(data):
    size = len(data)
    positive = len(data.loc[data['Clase'] == '+'])
    negative = len(data.loc[data['Clase'] == '-'])
    if positive > 0:
        positive = (positive / size) * math.log2(positive / size)
    else:
        positive = 0
    if negative > 0:
        negative = negative / size * math.log2(negative / size)
    else:
        negative = 0
    return - positive - negative
```

3 Metodo Para calcular la ganancia de un atributo

Este metodo también retorna una diccionario con la entriopia de cada valor posible del atributo, recibe la data y las columnas sobre la que tiene que trabajar y la entropia del atributo anterior.

```
[3]: def earning(data, column, entropy_current):
    earn = entropy_current
    unique_values = data[column].unique()
    entropies = {}
    for value in unique_values:
        data_value = data.loc[data[column] == value]
        entropy_value = entropy(data_value)
        entropies[value] = entropy_value
        earn = earn - len(data_value) / len(data) * entropy_value
    return earn, entropies
```

4 Metodo para la contrucción del árbol y nodo

Es un metodo recursivo que recibe los datos y columnas que van quedando y aplica el algoritmo ID3

```
[4]: def build_node(data, columns):
    entropy_data = entropy(data)
    dictionary = {}
    earnings = pd.DataFrame(columns=('Column', 'Earning'))
    entropies = {}

    # Obtención de la ganancia y entropías de los valores del atributo
    for column in columns:
        earn_result = earning(data, column, entropy_data)
        earn = earn_result[0]
        entropies[column] = earn_result[1]
        earnings.loc[len(earnings)] = [column, earn]

    # Selección del atributo con más ganancia
    earnings = earnings.set_index('Column')
    major = earnings[['Earning']].idxmax().to_list()[0]

    # Creación del nuevo nodo con el valor resultado o un nuevo nodo recursivo
    node = entropies[major]
    columns = columns.drop(major)
    for value in node:
        new_data = data.loc[data[major] == value]
        if node[value] == 0:
            node[value] = new_data['Clase'].mode()[0]
        else:
            node[value] = build_node(new_data, columns)
    dictionary[major] = node
    return dictionary
```

5 Metodo Principal

Recibe como parametro el nombre del archivo que va a usar

```
[5]: def main(file):
    data = pd.read_csv(file, index_col=0)
    print(data)
    columns = data.columns.drop('Clase')
    return build_node(data, columns)
```

6 Ejecución del Algoritmo

```
[6]: file = 'data.csv'
    three = main(file)
    print("\nÁRBOL RESULTADO\n")
    pprint.pprint(three, width=1)
```

	Color	Forma	Tamaño	Clase
Ej				
1	rojo	cuadrado	grande	+
2	azul	cuadrado	grande	+
3	rojo	redondo	pequeño	-
4	verde	cuadrado	pequeño	-
5	rojo	redondo	grande	+
6	verde	cuadrado	grande	-

ÁRBOL RESULTADO

```
{'Color': {'azul': '+',
            'rojo': {'Tamaño': {'grande': '+',
                                'pequeño': '-'}},
            'verde': '-'}}}
```