INSTITUTO TECNOLÓGICO BELTRÁN
Centro de Tecnología e Innovación

Algoritmos de Machine Learning

“Algoritmo FOIL”

Alumno = Nicolas Mesquiatti

Materia = Aprendizaje Automático

Instituto Tecnológico Beltrán

Algoritmo de FOIL

3.Algoritmo de FOIL

Ejercicio 1: Identificar empleados en formación Este ejercicio permite practicar cómo inducir reglas lógicas simples a partir de diferencias entre ejemplos positivos y negativos, tal como lo hace el algoritmo FOIL en su forma más básica.

Se tiene un conjunto de datos con personas que trabajan en una empresa. Algunas están en formación (aprendices) y otras no.

Cada persona tiene atributos como:

Edad,departamento, nivel_educativo,en_formacion(booleano: True o False)

Dataset de ejemplo:

datos = [{"edad": 22, "departamento": "IT", "nivel_educativo": "terciario", "en_formacion": True}, {"edad": 24, "departamento": "IT", "nivel_educativo": "universitario", "en_formacion": True}, {"edad": 21, "departamento": "RRHH", "nivel_educativo": "terciario", "en_formacion": True}, {"edad": 35, "departamento": "IT", "nivel_educativo": "universitario", "en_formacion": False}, {"edad": 40, "departamento": "Finanzas", "nivel_educativo": "maestría", "en_formacion": False}, {"edad": 29, "departamento": "RRHH", "nivel_educativo": "universitario", "en_formacion": False}, {"edad": 23, "departamento": "IT", "nivel_educativo": "terciario", "en_formacion": True}, {"edad": 38, "departamento": "Finanzas", "nivel_educativo": "universitario", "en_formacion": False}]

Objetivo del ejercicio:

1. Separar los ejemplos positivos (en_formacion == True) y negativos (en_formacion == False).

2. Para cada atributo, identificar los valores que aparecen en los positivos, pero no en los negativos.

3. Desarrollar un programa en Python que muestre la Regla inducida para identificar a personas en formación:

Regla inducida para identificar a personas en formación: Respuestas a las preguntas guía:

¿Qué valores de departamento aparecen solo en los positivos?

- Rta : Ninguno, Ya que IT también aparece en algún caso en falso

¿Qué niveles educativos son comunes en los positivos, pero no en los negativos?

- Rta : El nivel educativo común en los positivos y no en los negativos es el nivel terciario

¿Qué edades aparecen solo en los positivos?

- Rta : 21,22,23 y 24

Ejercicio 2:

a) Desarrollar un programa en Python que obtenga el cálculo del FOIL Gain para la condición nivel_educativo == 'terciario', con la siguiente salida:

Condición: nivel_educativo == 'terciario'

P (positivos antes) = 4

N (negativos antes) = 4

p (positivos después) = 3

n (negativos después) = 0

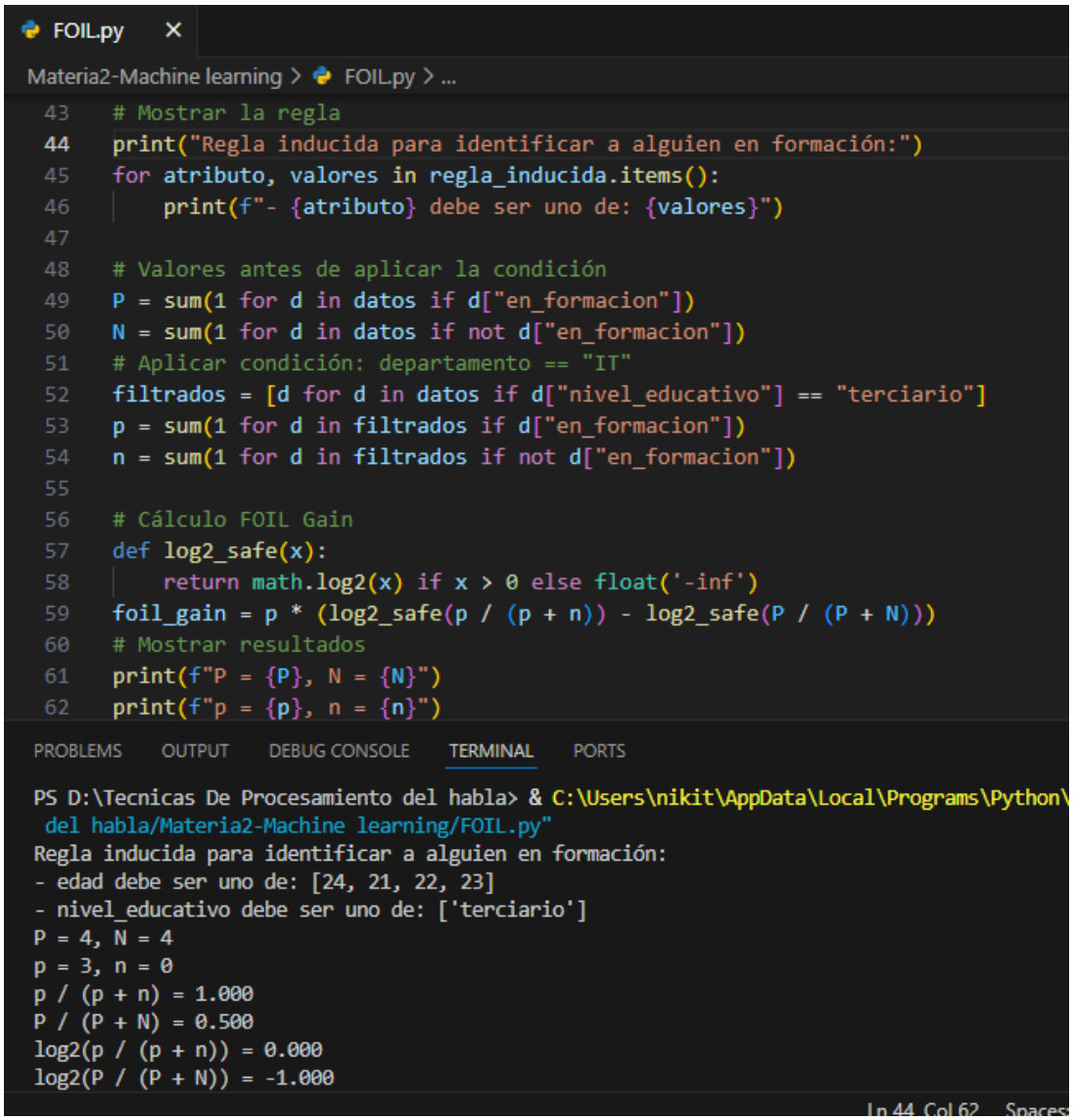
p / (p + n) = 1.000

P / (P + N) = 0.500

log2(p / (p + n)) = 0.000

log2(P / (P + N)) = -1.000

FOIL Gain = 3.000



Me dio el mismo resultado al evaluar la condición nivel_educativo == ‘terciario’

b) Realizar el calculo de FOIL Gain manual para comprobar los cálculos.
Condición: nivel_educativo == 'terciario'

Valores utilizados:

Término Valor

P (positivos antes) 4

N (negativos antes) 4

p (positivos después) 3

n (negativos después) 0

1.Fracción después de aplicar la condición:

p / (p + n) = 3/ (3+0) = 3/3 = 1

2. Fracción antes de aplicar la condición:

P / (P + N) = 4/ (4+4) = 4/8 = 0,5

3. Logaritmos:

Log2: p / (p + n) = 3/ (3+0) = 3/3 = 1 –> log2(1) = 0,000

Log2:P / (P + N) = 4/ (4+4) = 4/8 = 0,5 -> log2(0,5) = -1,000

4. FOIL Gain:

p x (Log2: p / (p + n) - Log2:P / (P + N))

3 x (0,000 - (-1,000))

3x 1,000 = **3.000**

5.Interpretación:

Una regla con 3 positivos y 0 negativos, significa una regla bastante precisa y la ganancia de información es bastante significativa (3,000)

c) Desarrollar un programa en Python que obtenga el cálculo del FOIL Gain para la condición edad ≤ 23, con la siguiente salida:

P (positivos antes) = 4

N (negativos antes) = 4

p (positivos después) = 3

n (negativos después) = 0

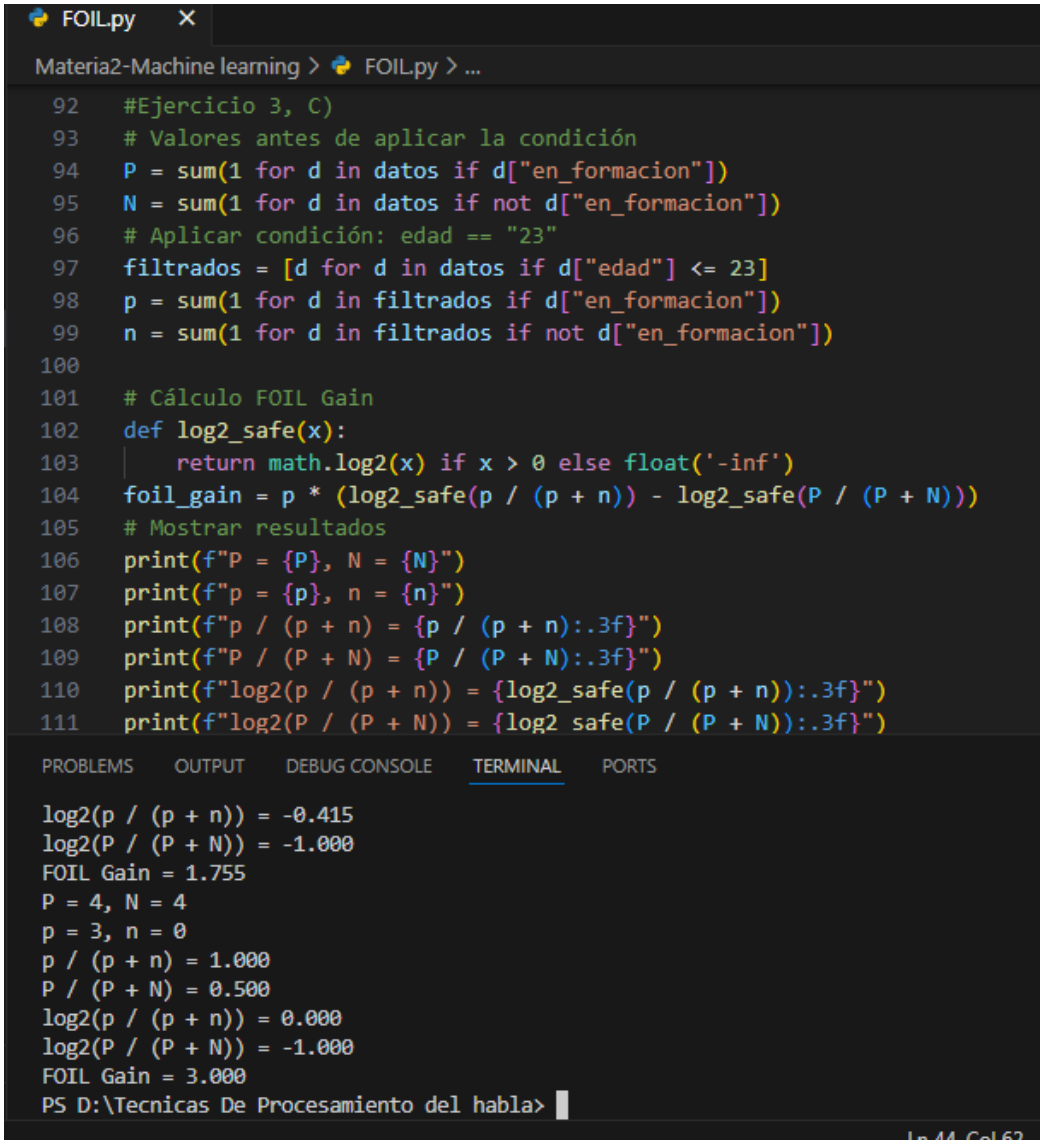
p / (p + n) = 1.000

P / (P + N) = 0.500

log2(p / (p + n)) = 0.000

log2(P / (P + N)) = -1.000

FOIL Gain = 3.000



d) Realizar el cálculo de FOIL Gain manual para comprobar los cálculos. Condición: edad ≤ 23 Valores utilizados: Término

P (positivos antes) Valor 4

N (negativos antes) 4

p (positivos después) 3

n (negativos después) 0

1. Fracción después de aplicar la condición:

p / (p + n) = 3/ (3+0) = 3/3 = 1

2. Fracción antes de aplicar la condición:

P / (P + N) = 4/ (4+4) = 4/8 = 0,5

3. Logaritmos:

Log2: p / (p + n) = 3/ (3+0) = 3/3 = 1 –> log2(1) = 0,000

Log2:P / (P + N) = 4/ (4+4) = 4/8 = 0,5 -> log2(0,5) = -1,000

4. FOIL Gain:

p x (Log2: p / (p + n) - Log2:P / (P + N))

3 x (0,000 - (-1,000))

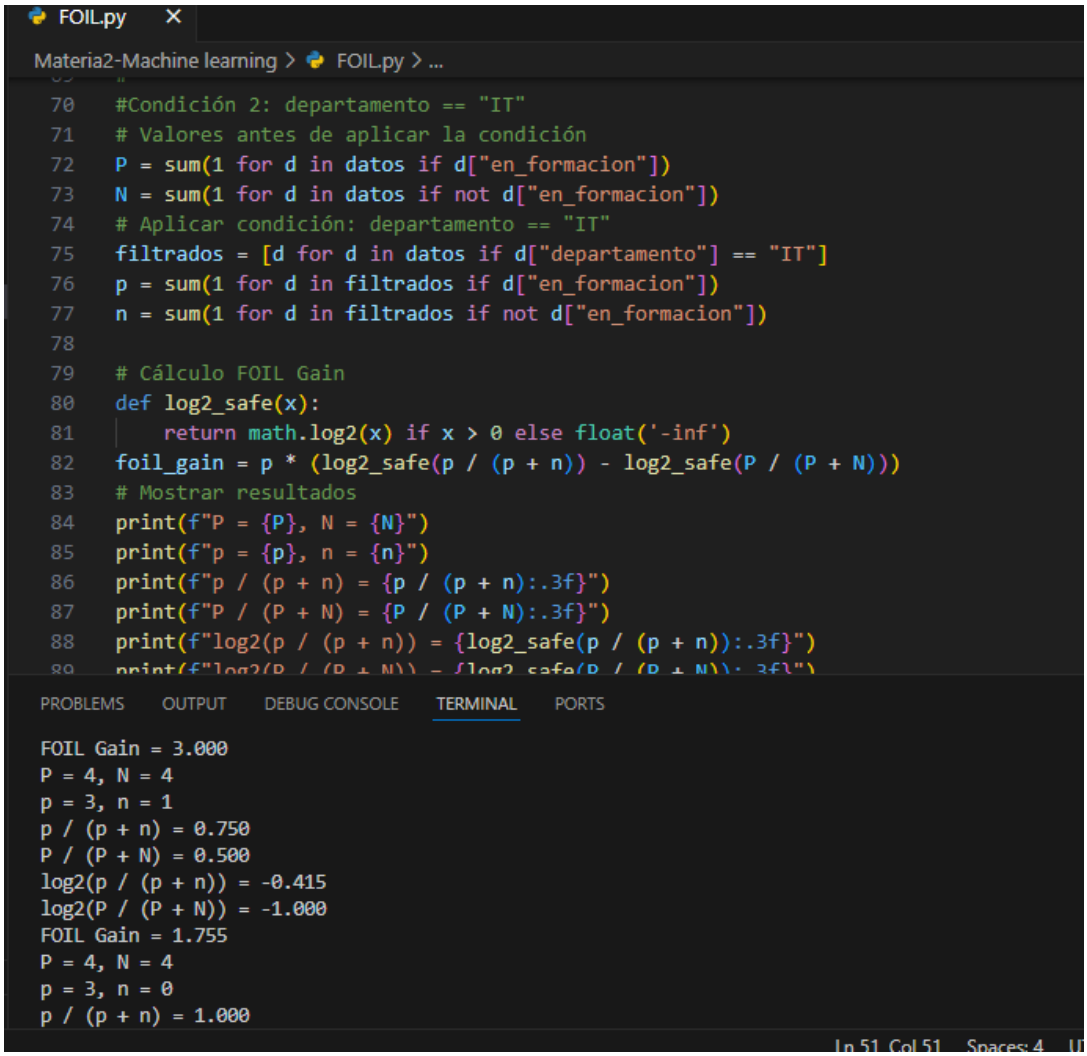
3x 1,000 = **3.000**

5. Interpretación

Como con la regla anterior también hay 3 positivos y 0 negativos, es eficaz y precisa, con buena ganancia de información.

Condición extra

Departamento = “IT”



Conclusión

Al aplicar esta regla me da una ganancia distinta a los dos anteriores

Ya que ahora tengo 3 p y 1 n después de aplicar la condición Gracias a eso el FOIL gain es de 1,755, Como la mitad entre las otras dos condiciones, esto quiere decir que es mucho mejor aplicar las otras dos condiciones antes que esta ya que ambas eliminan los negativos y eso genera una mayor ganancia de información