Código Máquina Machine Learning

IA y Machine Learning

→ Aprendizaje → Supervisado → Clasificación previa de datos y aprendizaje guiado.
→ No supervisado → No hay clasificación previa

Supervisado ← Árboles de decisión, Redes neuronales, K vecinos más próximos } Y MÁS

- Se tiene la clase que guía el proceso de aprendizaje
- Requiere datos de cada clase
- Características / Atributos → estadísticas
- Se generan modelos lineales a partir de las clases

- Aprendizaje Supervisado → Clasificación → Detección
- Aprendizaje Supervisado → Regresión → Previsión

No Supervisado

- No se tiene una clase que guía el proceso de aprendizaje
- El algoritmo tiene que encontrar la estructura subyacente
- Se utiliza para detectar patrones / agrupamientos
  - → Conglomerados o Clusters : K-means (los + utilizados)
  - → Detección de Anomalías
  - → Redes Neuronales * → Para Supervisado y No supervisado
  - → Reglas de asociación: Apriori.

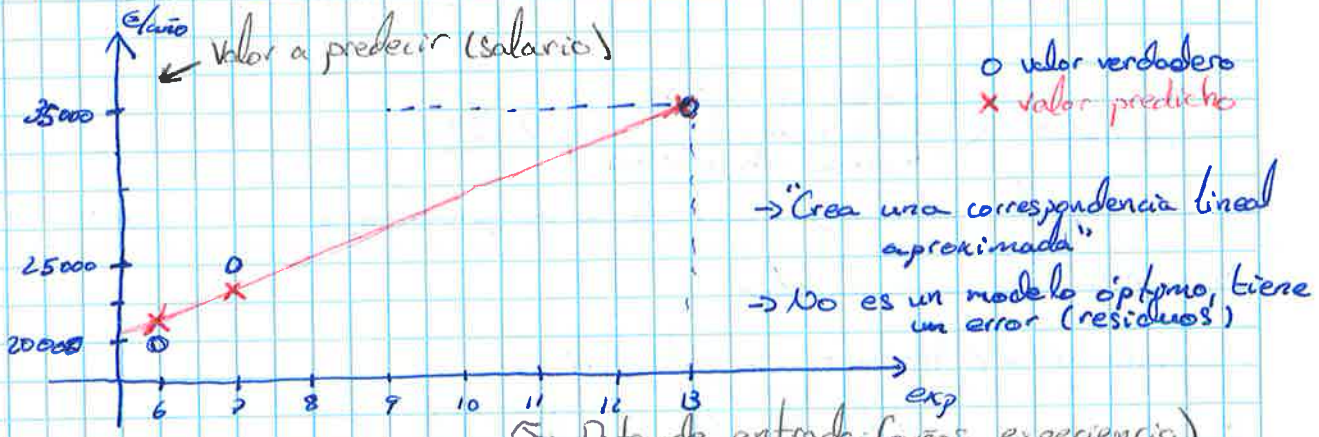
Índice

② Gradiente descendente

- **Regresión lineal** → SK learn para ver regresiones lineales de datos

Generamos un modelo lineal del que podremos sacar datos simples en el futuro

Por ejemplo: { Una persona con 6 años de exp gana 20 000 €/año
7 años 25 000 €/año
13 años 35 000 €/año



o valor verdadero
× valor predicho

→ "Crea una correspondencia lineal aproximada"

→ No es un modelo óptimo, tiene un error (residuos)

→ Se saca el error Tot se suman los errores(i) elevados al cuadrado (para eliminar signos)

→ Se busca el menor error posible

- **Función de Error**

→ Suma de los cuadrados de los residuos (RSS):

$$RSS(y, \hat{y}) = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$ ← Tamaño muestra

y_i = valor real
$\hat{y}_i$ = valor predicho

$\hat{y}_i = m * (ejeX)$
↑ pendiente

En código sería:

```
m = sym.Symbol('m')   ← crea una variable "m"
error = (20000 - m*6)**2 + (25000 - m*7)**2 + (35000 - m*13)**2
                                                  ↑ elevado
derivada = sym.diff(error, m)

# para evaluar en ≠ pendientes
for pendiente in pendientes:
    print(derivada, "Evaluada en %0.2f" % derivada.evalf(subs={m: pendiente}))
                                                  ↑ formato
```

→ Para evaluar todos los resultados posibles se debe establecer un máximo de "pasos", son las llamadas iteraciones máximas → se ajusta experimentalm.

→ La tasa de aprendizaje → Define el "tamaño" de esos pasos
Se ajusta experimentalmente y no conviene que sea un valor alto del orden de 0'001)

→ Se debe indicar en qué valor se inicializan los parámetros (debería llevar al mismo resultado independientemente del valor, siempre que las iteraciones y la tasa de aprendizaje sean los idóneos)

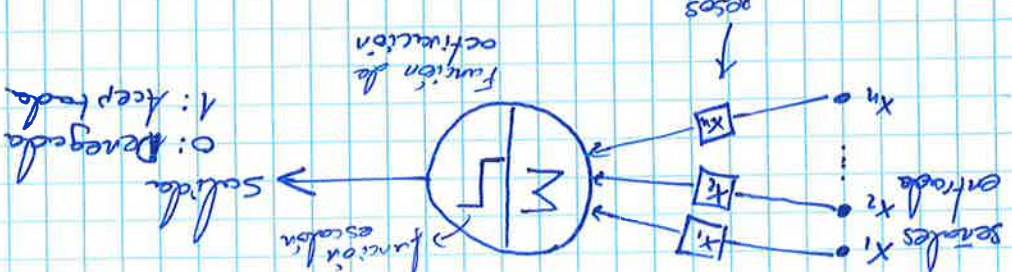
00:30:18
(5)

→

3 Perceptrón

→ Inspirado en neuronas biológicas

↳ Interpreta las entradas abren dando del "peso"
↳ Si: la entrada tiene el "peso" requerido ⇒ la función de activación activa la salida



→ El perceptrón es un clasificador binario ⇒ solo puede clasificar instancias entre 0 y 1

→ El Perceptrón es un modelo de aprendizaje de máquina supervisado, necesita unos datos etiquetados previamente

PSEUDOCÓDIGO PERCEPTRÓN

→ El Perceptrón se entrena por un proceso interactivo

0. Inicializar los pesos y el umbral
1. Repetir mientras ...
2. Para cada ...
3. Para cada instancia de entrenamiento hacer:
4. Calcular salida del Perceptrón para esa instancia
5. Calcular el error
6. Actualizar pesos y umbral usando la regla de aprendizaje del Perceptrón
7. Repetir hasta que ...

→ Existen librerías python que integran el funcionamiento del perceptrón, como puede ser SciKit-learn (sklearn)

* Se pasa como una muestra de datos para que el algoritmo entrene, cada instancia empezando en 0 hasta cierto número de épocas

@author: Octavio Gutiérrez de Código Máquina

URL del canal: https://www.youtube.com/CodigoMaquina

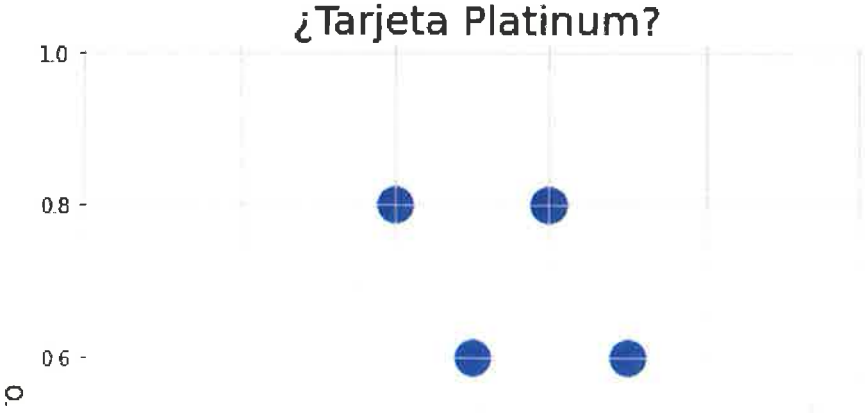
URL del video: https://youtu.be/dkhXGTersP0

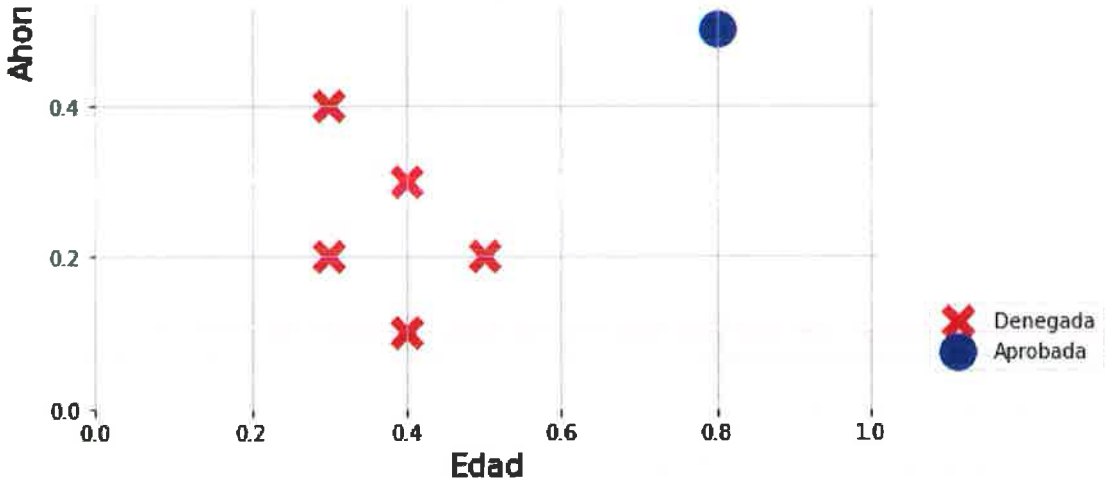
Perceptrón

Datos linealmente separables

In [1]:

```
import numpy as np
import matplotlib.pyplot as plt

# Datos de 10 personas -> [edad, ahorro]

personas = np.array([[0.3, 0.4], [0.4, 0.3],
                     [0.3, 0.2], [0.4, 0.1],
                     [0.5, 0.2], [0.4, 0.8],
                     [0.6, 0.8], [0.5, 0.6],
                     [0.7, 0.6], [0.8, 0.5]])

# 1 : aprobrada    0 : denegada

clases = np.array([0, 0, 0, 0, 0, 1, 1, 1, 1, 1])

# Gráfica de dispersión (edad, ahorro)
plt.figure(figsize=(7, 7))
plt.title("¿Tarjeta Platinum?", fontsize=20)
plt.scatter(personas[clases == 0].T[0],
            personas[clases == 0].T[1],
            marker="x", s=180, color="red",
            linewidths=5, label="Denegada")
plt.scatter(personas[clases == 1].T[0],
            personas[clases == 1].T[1],
            marker="o", s=180, color="blue",
            linewidths=5, label="Aprobada")
plt.xlabel("Edad", fontsize=15)
plt.ylabel("Ahorro", fontsize=15)
plt.legend(bbox_to_anchor=(1.3, 0.15))
plt.box(False)
plt.xlim((0, 1.01))
plt.ylim((0, 1.01))
plt.grid()
plt.show()
```





Función de Activación (Escalón)

In [2]:

```
# w1*x1 + w2*x2 + ⋯ + wn*xn

def activacion(pesos, x, b):
    z = pesos * x
    if z.sum() + b > 0:
        return 1
    else:
        return 0

pesos = np.random.uniform(-1, 1, size=2)
b = np.random.uniform(-1, 1)
```

```
pesos, b, activacion(pesos, [0.5, 0.4], b)
```

Out[2]: (array([-0.14362188, -0.21542272]), 0.5691661131756678, 1)

Pseudocódigo del Perceptrón

0. Inicializar los pesos y el umbral
1. epocas máximas = 100
2. epoca = 0
3. tasa de aprendizaje = 0.01
4. Mientras (epoca < epocas máximas) hacer:
5. Para cada instancia de entrenamiento hacer:
6. Calcula salida del perceptrón para esa estancia
7. Calcula el error
8. Actualiza pesos y umbral usando la tasa de aprendizaje, la instancia y el error
9. epoca += 1

Entrenamiento del Perceptrón

In [ ]:
```
pesos = np.random.uniform(-1, 1, size=2)
```

```
b = np.random.uniform(-1, 1)
tasa_de_aprendizaje = 0.01
epocas = 100

for epoca in range(epocas):
    error_total = 0
    for i in range(len(personas)):
        prediccion = activacion(pesos, personas[i], b)
        error = clases[i] - prediccion
        error_total += error**2
        pesos[0] += tasa_de_aprendizaje * personas[i][0] * error
        pesos[1] += tasa_de_aprendizaje * personas[i][1] * error
        b += tasa_de_aprendizaje * error
    print(error_total, end=" ")

activacion(pesos, [0.5, 1], b)
```

```
2 2 1 1 1 1 2 2 2 1 2 2 2 2 1 2 2 2 1 2 2 2 1 2 2 2 1 2 2 2 1 2 2 2 1 2 2 2 1
2 2 2 1 2 2 2 1 2 2 2 1 2 2 2 2 1 2 2 2 1 2 2 2 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Out[3]: 1

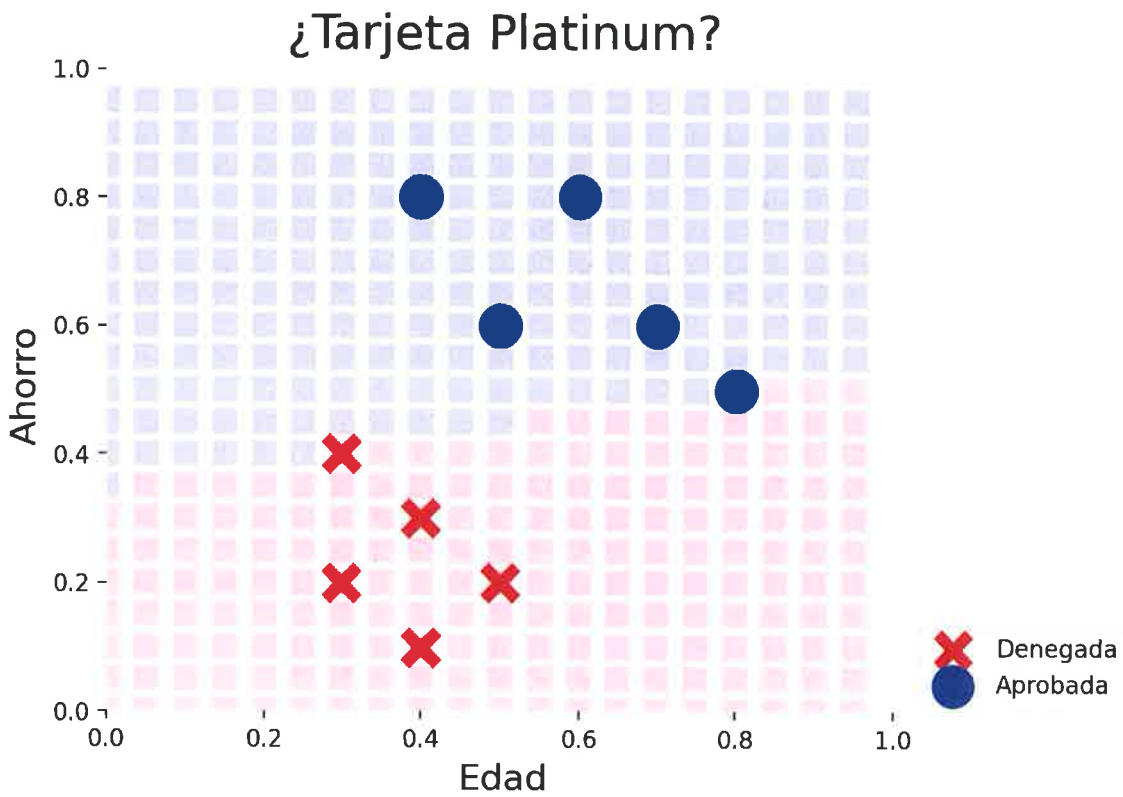
Zonas de aprobación

Gráfica de dispersión [edad, ahorro]

In [4]:

```
plt.figure(figsize=(6, 5), dpi=200)
plt.title("¿Tarjeta Platinum?", fontsize=20)
```

```
plt.scatter(personas[clases == 0].T[0],
            personas[clases == 0].T[1],
            marker="x", s=180, color="red",
            linewidths=5, label="Denegada")

plt.scatter(personas[clases == 1].T[0],
            personas[clases == 1].T[1],
            marker="o", s=180, color="blue",
            linewidths=5, label="Aprobada")

for edad in np.arange(0, 1, 0.05):
    for ahorro in np.arange(0, 1, 0.05):
        color = activacion(pesos, [edad, ahorro], b)
        if color == 1:
            plt.scatter(edad, ahorro, marker="s", s=110,
                        color="blue", alpha=0.2, linewidths=0)
        else:
            plt.scatter(edad, ahorro, marker="s", s=110,
                        color="red", alpha=0.2, linewidths=0)

plt.xlabel("Edad", fontsize=15)
plt.ylabel("Ahorro", fontsize=15)
plt.legend(bbox_to_anchor=(1.3, 0.15))
plt.box(False)
plt.xlim((0, 1.01))
plt.ylim((0, 1.01))
plt.show()
```

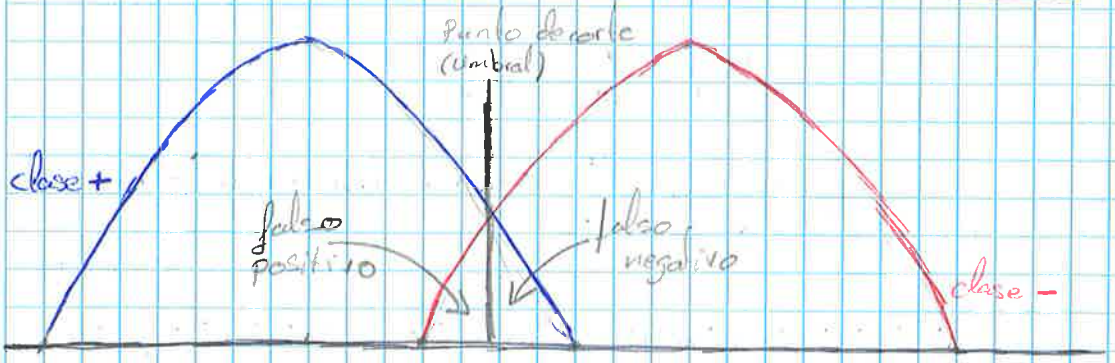


Perceptrón con Scikit-learn

In [5]:
```
from sklearn.linear_model import Perceptron
perceptron = Perceptron().fit(personas, clases)
perceptron.predict([[0.2, 0.2], [0.8, 0.8]])
```

Out[5]:
```
array([0, 1])
```

④ Área bajo la curva ROC (Receiver Operating Characteristic)



$$\text{Tasa verdaderos positivos} = \frac{\text{Verdaderos Positivos}}{\text{Verdaderos Positivos} + \text{Falsos Negativos}}$$

$$\text{Tasa falsos positivos} = \frac{\text{Falsos Positivos}}{\text{Falsos Positivos} + \text{Verdaderos Negativos}}$$

(5) Árboles de decisión usando Entropía con Python

→ CAOS

→ Técnicas caja negra → p. ej.: Redes Neuronales → No permiten ver en qué, o cómo, se toman las decisiones. (base a)

→ Técnicas de caja blanca → Sí que lo permiten → p. ej: árboles de decisiones

- Contamos con 2 tipos de Nodos ↔ de decisión / de clase
- Los árboles de decisión son técnicas NO PARAMÉTRICAS → No hacen suposiciones con respecto a la disposición de los datos
- Para ver la capacidad de almacenamiento de un valor de entropía (en bits) hay que aplicar:

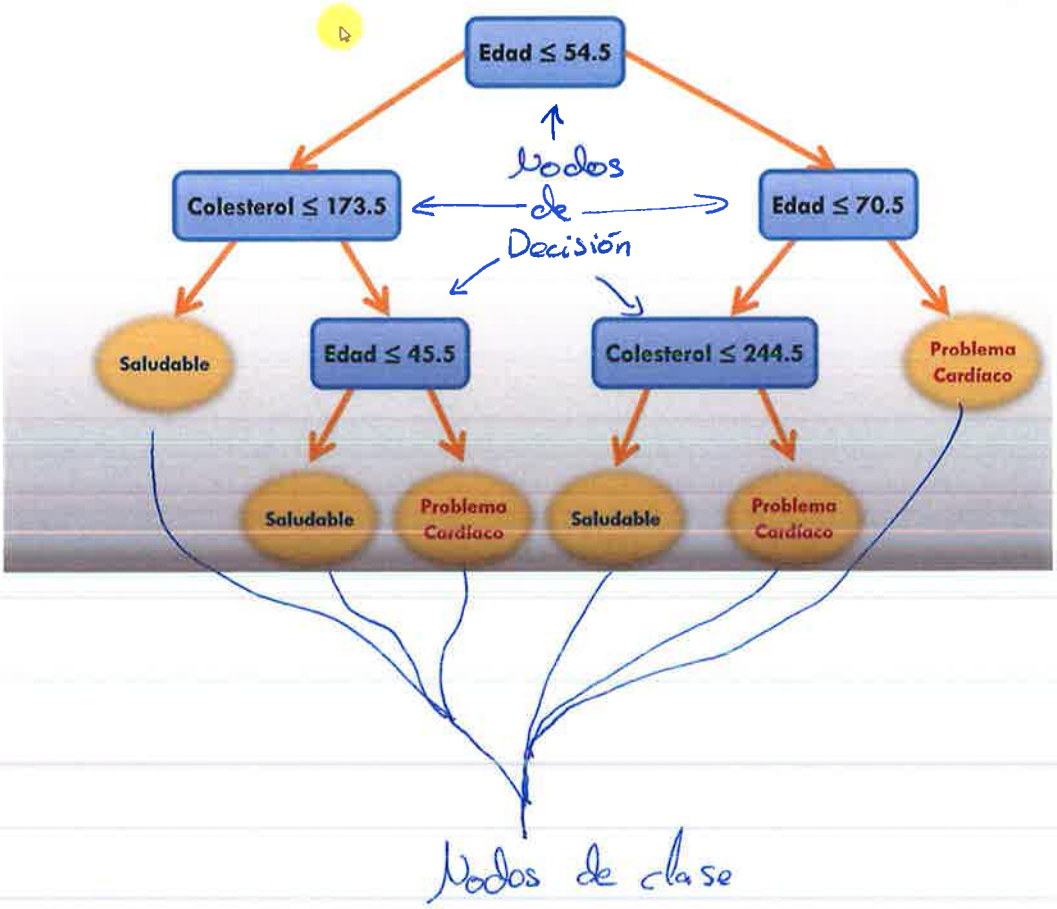
[redacted] ∴ [redacted] = [redacted] $n^{\circ}\,dc = 2^{\overbrace{n+1}^{n^{\circ}\,bits}}$

Árboles de Decisión

Padecimientos cardíacos

Árboles de Decisión

Padecimientos cardíacos

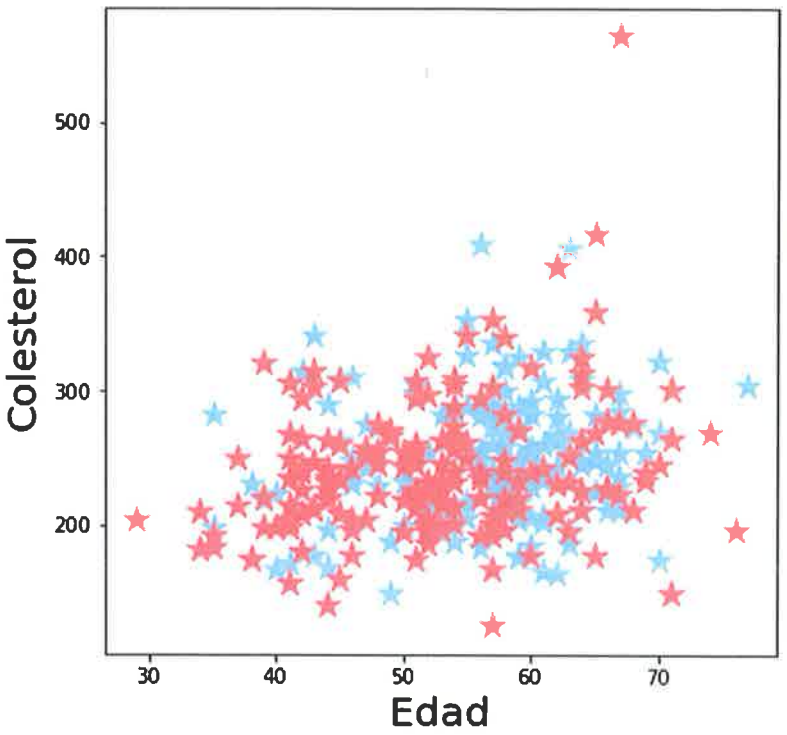


Gráfica dispersión: Edad y Colesterol

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt

pacientes =

pd.read_csv("pacientes.csv")

saludables = pacientes[pacientes["problema_cardiaco"]==0]
cardiacos = pacientes[pacientes["problema cardiaco"]==1]
```

p [p [p _]]

```
plt.figure(figsize=(6, 6))
plt.xlabel('Edad', fontsize = 20.0)
plt.ylabel('Colesterol', fontsize = 20.0)
plt.scatter(saludables["edad"], saludables["colesterol"],
            label="Saludable (Clase: 0)", marker="*", c="skyblue", s=200)
plt.scatter(cardiacos["edad"], cardiacos["colesterol"],
            label="Cardíaco (Clase: 1)", marker="*", c="lightcoral", s=200)
plt.legend(bbox_to_anchor=(1, 0.15))
plt.show()
```



No se puede trazar una línea que separe los datos

||

Árboles de decisión

↳ Capaces de clasificar un conjunto de datos NO LINEALES

Entropía:

Promedio de información almacenada en una variable aleatoria

In [2]:
```
from scipy.stats import entropy
from math import log

edades = pd.Series([40, 30, 20, 50])
colesterol = pd.Series([100, 110, 100, 110])

print(edades.value_counts()/edades.size)
print(colesterol.value_counts()/colesterol.size)
print(entropy(edades.value_counts()/edades.size, base=2))
print(entropy(colesterol.value_counts()/colesterol.size, base=2))
```

```
30    0.25
20    0.25
50    0.25
40    0.25
dtype: float64
110    0.5
100    0.5
dtype: float64
2.0
1.0
```

```
from sklearn.model_selection import train_test_split

datos_entrena, datos_prueba, clase_entrena, clase_prueba = train_test_split(
pacientes[["edad", "colesterol"]],     pacientes["problema_cardiaco"],
test_size=0.30)
```

Datos de Entrenamiento y Prueba

In [3]:

Creación del Árbol de Decisión

In [4]:

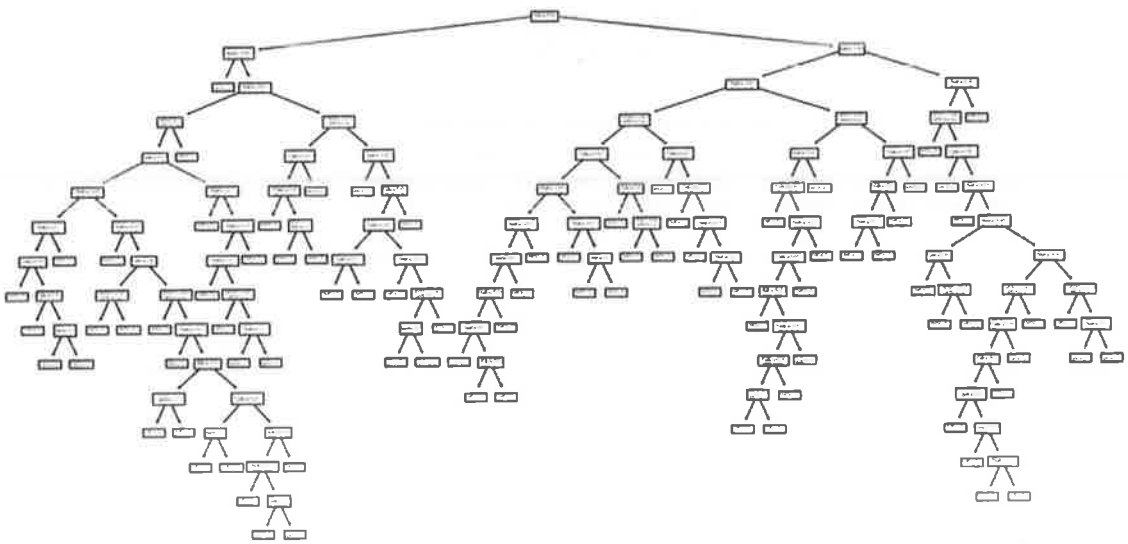
```
from sklearn import tree arbol_decision =
tree.DecisionTreeClassifier(criterion="entropy") arbol = arbol
decision.fit(datos entrena, clase entrena)
             _           (     _       ,      _       )



accuracy = arbol_decision.score(datos_prueba, clase_prueba) print(accuracy)

print(tree.export_text(arbol,
                       feature_names=["Edad", "Colesterol"]))
plt.figure(figsize=(12, 6)) tree.plot_tree(arbol,
              feature_names=["Edad", "Colesterol"]) plt.show()
```

```
).4111111111111111 |--
  Edad <= 54.50
     |--- Colesterol <= 173.50
     |   |--- class: 0
     |--- Colesterol >  173.50
     |   |--- Colesterol <= 271.50
     |   |   |--- Edad <= 52.50
     |   |   |   |--- Edad <= 51.50
     |   |   |   |   |--- Colesterol <= 198.50
     |   |   |   |   |   |--- Colesterol <= 194.50
     |   |   |   |   |   |   |--- Edad <= 42.50
     |   |   |   |   |   |   |   |--- class: 1
     |   |   |   |   |   |   |--- Edad >  42.50
     |   |   |   |   |   |   |   |--- Edad <= 44.50
     |   |   |   |   |   |   |   |   |--- class: 0
     |   |   |   |   |   |   |   |--- Edad >  44.50
     |   |   |   |   |   |   |   |   |--- Edad <= 47.50
     |   |   |   |   |   |   |   |   |   |--- class: 1
     |   |   |   |   |   |   |   |   |--- Edad >  47.50
     |   |   |   |   |   |   |   |   |   |--- class: 0
     |   |   |   |   |   |--- Colesterol >  194.50
     |   |   |   |   |   |   |--- class: 0
     |   |   |   |   |--- Colesterol >  198.50
     |   |   |   |   |   |--- Colesterol <= 217.00 |
     |   |   |   |   |   |--- class: 1
     |   |   |   |   |   |--- Colesterol >  217.00
     |   |   |   |   |   |   |--- Edad <= 44.50
     |   |   |   |   |   |   |   |--- Colesterol <= 219.50 |
     |   |   |   |   |   |   |   |--- class: 0
     |   |   |   |   |   |   |   |--- Colesterol >  219.50
     |   |   |   |   |   |   |   |   |--- class: 1
     |   |   |   |   |   |   |--- Edad >  44.50
     |   |   |   |   |   |   |   |--- Colesterol <= 229.00
     |   |   |   |   |   |   |   |   |--- class: 1
     |   |   |   |   |   |   |   |--- Colesterol >  229.00
     |   |   |   |   |   |   |   |   |--- Colesterol <= 233.50 |
     |   |   |   |   |   |   |   |   |--- class: 0
     |   |   |   |   |   |   |   |   |--- Colesterol >  233.50
     |   |   |   |   |   |   |   |   |   |--- Edad <= 45.50
     |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2 |
     |   |   |   |   |   |   |   |   |--- Edad >  45.50
     |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 5
     |   |   |   |--- Edad >  51.50
     |   |   |   |   |--- Colesterol <= 202.50 |
     |   |   |   |   |--- class: 1
     |   |   |   |   |--- Colesterol >  202.50
     |   |   |   |   |   |--- Colesterol <= 226.50
     |   |   |   |   |   |   |--- Colesterol <= 204.50 |
     |   |   |   |   |   |   |--- class: 0
     |   |   |   |   |   |   |--- Colesterol >  204.50
     |   |   |   |   |   |   |   |--- Colesterol <= 208.50
     |   |   |   |   |   |   |   |   |--- class: 1

|    |   |   |   |   |   |   |   |--- Colesterol >  208.50
|    |   |   |   |   |   |   |   |   |--- Colesterol <= 217.50
|    |   |   |   |   |   |   |   |   |   |--- class: 0
|    |   |   |   |   |   |   |   |   |--- Colesterol >  217.50
|    |   |   |   |   |   |   |   |   |   |--- class: 1
|    |   |   |   |   |   |   |--- Colesterol >  226.50
|    |   |   |   |   |   |   |   |--- class: 0
|    |   |   |   |--- Edad >  52.50
|    |   |   |   |   |--- class: 1
|    |   |   |--- Colesterol >  271.50
|    |   |   |   |--- Colesterol <= 292.50
```

```
|   |   |   |   |--- Colesterol <= 278.50
|   |   |   |   |   |--- Colesterol <= 274.50
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- Colesterol >  274.50
|   |   |   |   |   |   |--- Edad <= 47.50 |
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- Edad >  47.50
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- Colesterol >  278.50
|   |   |   |   |   |   |--- class: 0
|   |   |   |--- Colesterol >  292.50
|   |   |   |   |--- Colesterol <= 304.50
|   |   |   |   |   |--- class: 1
|   |   |   |   |--- Colesterol >  304.50
|   |   |   |   |   |--- Edad <= 51.50
|   |   |   |   |   |   |--- Colesterol <= 308.50
|   |   |   |   |   |   |   |--- Colesterol <= 305.50 |
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- Colesterol >  305.50
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- Colesterol >  308.50
|   |   |   |   |   |   |   |--- Colesterol <= 313.00
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- Colesterol >  313.00
|   |   |   |   |   |   |   |   |--- Colesterol <= 328.00
|   |   |   |   |   |   |   |   |   |--- Edad <= 42.50
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- Edad >  42.50
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- Colesterol >  328.00
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- Edad >  51.50
|   |   |   |   |   |   |--- class: 1
|   |--- Edad >  54.50
|   |   |--- Edad <= 63.50
|   |   |   |--- Colesterol <= 251.50
|   |   |   |   |--- Colesterol <= 235.00
|   |   |   |   |   |--- Colesterol <= 222.50
|   |   |   |   |   |   |--- Colesterol <= 206.50
|   |   |   |   |   |   |   |--- Colesterol <= 202.50
|   |   |   |   |   |   |   |   |--- Colesterol <= 190.00
|   |   |   |   |   |   |   |   |   |--- Edad <= 60.50
|   |   |   |   |   |   |   |   |   |   |--- Colesterol <= 172.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- Colesterol >  172.00
|   |   |   |   |   |   |   |   |   |   |   |--- Edad <= 59.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |   |--- Edad >  59.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- Edad >  60.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- Colesterol >  190.00
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |    |--- Colesterol > 202.50
|   |   |   |   |   |   |   |     Colesterol >  202.50
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- Colesterol >  206.50
|   |   |   |   |   |   |   |--- Colesterol <= 213.50 |
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- Colesterol >  213.50
|   |   |   |   |   |   |   |   |--- Edad <= 58.50
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- Edad >  58.50
|   |   |   |   |   |   |   |   |   |--- class: 1
```

```
|   |   |   |   |--- Colesterol >  222.50
|   |   |   |   |   |--- Edad <= 58.50 |
|   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- Edad >  58.50
|   |   |   |   |   |   |--- Edad <= 59.50
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- Edad >  59.50
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |--- Colesterol >  235.00
|   |   |   |   |--- Colesterol <= 240.50 |
|   |   |   |   |--- class: 1
|   |   |   |   |--- Colesterol >  240.50
|   |   |   |   |   |--- Colesterol <= 246.00
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- Colesterol >  246.00
|   |   |   |   |   |   |--- Colesterol <= 248.50
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- Colesterol >  248.50
|   |   |   |   |   |   |   |--- Colesterol <= 249.50
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- Colesterol >  249.50
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |--- Colesterol >  251.50
|   |   |   |--- Colesterol <= 337.50
|   |   |   |   |--- Edad <= 59.50
|   |   |   |   |   |--- Colesterol <= 266.00 |
|   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- Colesterol >  266.00
|   |   |   |   |   |   |--- Colesterol <= 311.00
|   |   |   |   |   |   |   |--- Colesterol <= 301.50
|   |   |   |   |   |   |   |   |--- Edad <= 57.50
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- Edad >  57.50
|   |   |   |   |   |   |   |   |   |--- Colesterol <= 283.50
|   |   |   |   |   |   |   |   |   |   |--- Colesterol <= 276.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |   |   |   |--- Colesterol >  276.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- Colesterol >  283.50
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- Colesterol >  301.50
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- Colesterol >  311.00
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |--- Edad >  59.50
|   |   |   |   |   |--- class: 0
|   |   |   |--- Colesterol >  337.50
|   |   |   |   |--- Colesterol <= 400.50
|   |   |   |   |   |--- Edad <= 56.00
|   |   |   |   |   |   |--- Colesterol <= 347.50 |
|   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- Colesterol >  347.50
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- Edad >  56.00
|   |   |   |   |   |   |--- class: 1
```

```
|   |   |   |   |--- Colesterol >  400.50
|   |   |   |   |   |--- class: 0
|   |--- Edad >  63.50
|   |   |--- Colesterol <= 341.00
|   |   |   |--- Colesterol <= 204.50
|   |   |   |   |--- class: 1
|   |   |   |--- Colesterol >  204.50
|   |   |   |   |--- Colesterol <= 217.50
|   |   |   |   |   |--- class: 0
|   |   |   |   |--- Colesterol >  217.50
|   |   |   |   |   |--- Colesterol <= 228.50
|   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- Colesterol >  228.50
|   |   |   |   |   |   |--- Colesterol <= 259.50
|   |   |   |   |   |   |   |--- Edad <= 68.00
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- Edad >  68.00
|   |   |   |   |   |   |   |   |--- Colesterol <= 246.50
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- Colesterol >  246.50
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- Colesterol >  259.50
|   |   |   |   |   |   |   |   |--- Colesterol <= 303.50
|   |   |   |   |   |   |   |   |   |--- Colesterol <= 292.00
|   |   |   |   |   |   |   |   |   |   |--- Edad <= 70.50
|   |   |   |   |   |   |   |   |   |   |   |--- Edad <= 69.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 4
|   |   |   |   |   |   |   |   |   |   |   |--- Edad >  69.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- Edad >  70.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- Colesterol >  292.00
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- Colesterol >  303.50
|   |   |   |   |   |   |   |   |   |--- Colesterol <= 311.00
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- Colesterol >  311.00
|   |   |   |   |   |   |   |   |   |   |--- Colesterol <= 317.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- Colesterol >  317.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |--- Colesterol >  341.00
|   |   |   |--- class: 1
```



(6) Random Forest (Bosque aleatorio)

- Formados por múltiples árboles de decisiones
- Aumentan la precisión de las decisiones