Machine Learning 101

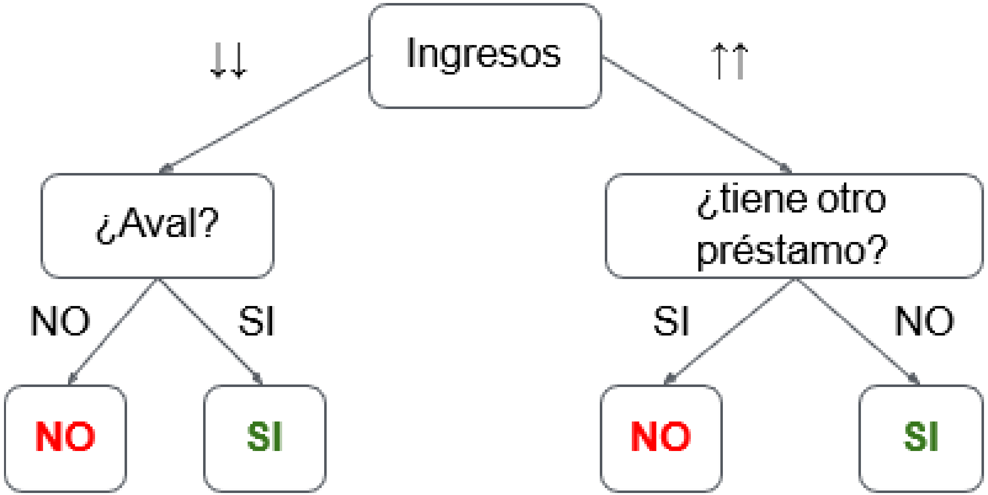
Árboles de decisión

Índice

1. **Intuición**
2. Construcción del árbol
3. Conclusiones

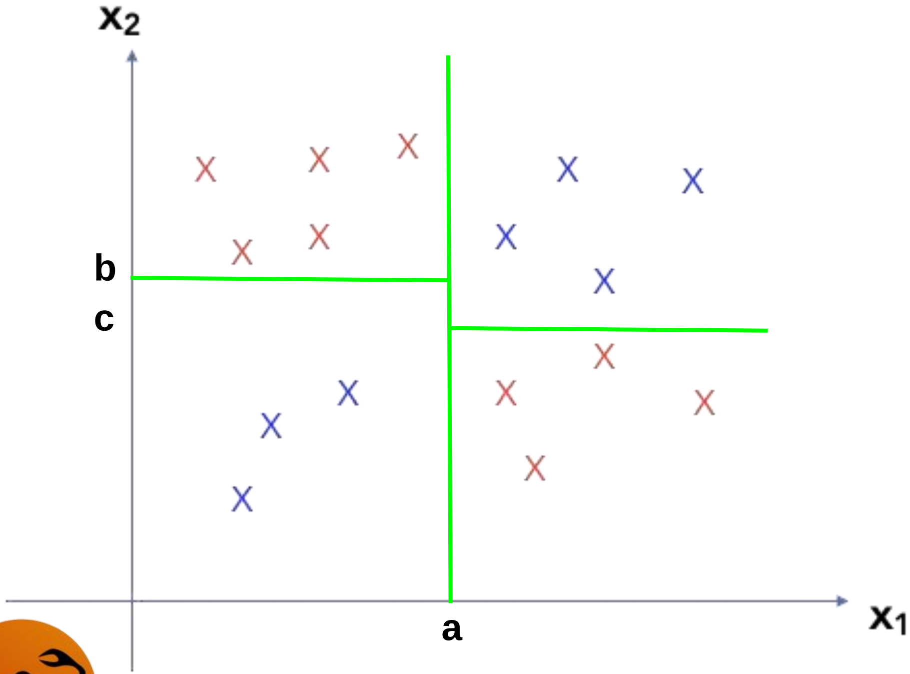
Intuición

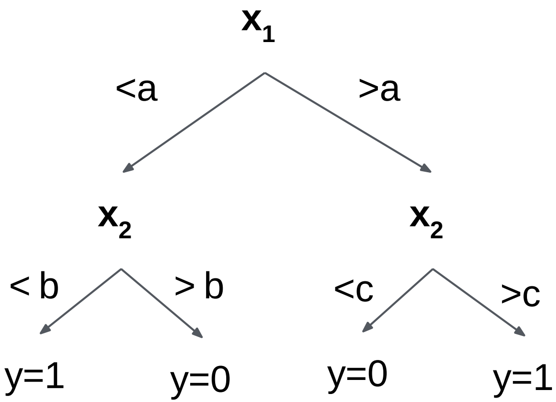
- Supongamos el problema de clasificación: concesión de un préstamo



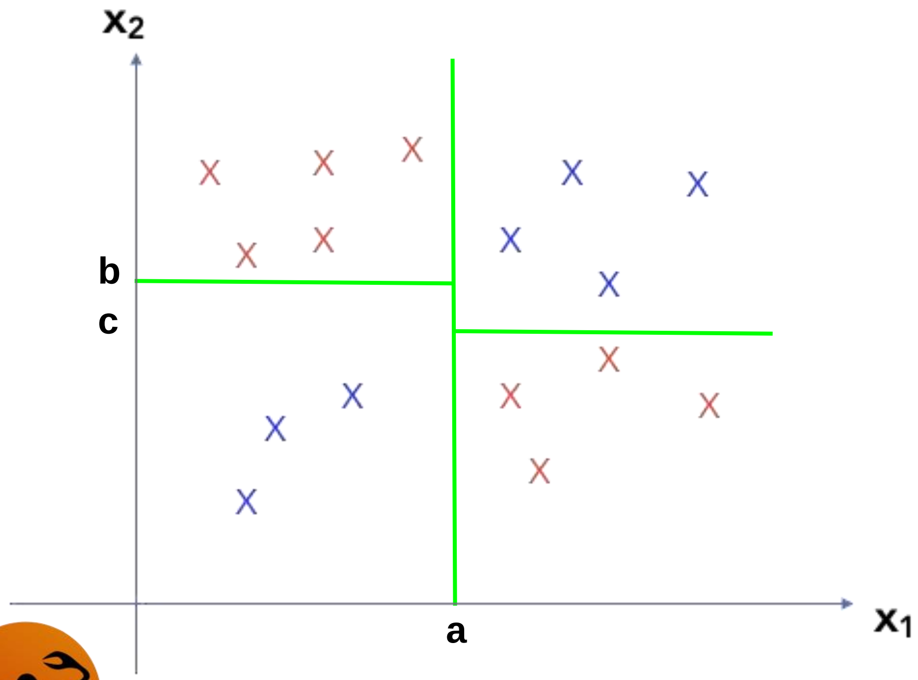
Intuición

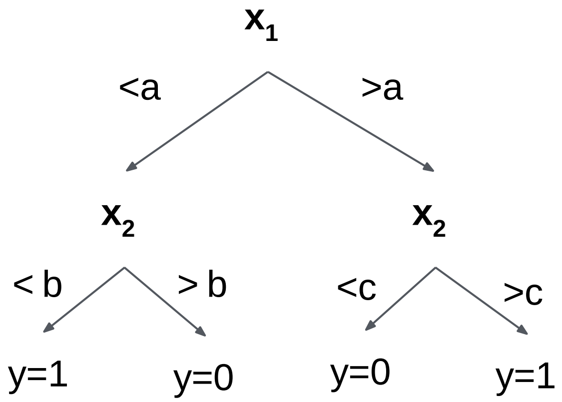
- ¿Cómo trasladamos este proceso a datos? Segmentando el espacio de características en regiones sencillas





Nomenclatura





- Hojas: región y=1
- Nodos intermedios: x_2
- Ramas: <a

Predicción

- Una vez segmentado el espacio de características, para cada nueva observación que cae en alguna de las regiones se predice:

  - Clasificación: moda de etiquetas (majority vote)
  - Regresión: media de etiquetas

- NOTA: Existen distintos algoritmos para implementar árboles de decisión: ID3, C4.5, CART... scikit-learn utiliza CART, que solo permite decisiones binarias (cada nodo tiene dos ramas).

Índice

Construcción del árbol

1. Empezamos con el árbol vacío
2. Seleccionamos la característica sobre la que particionar el espacio (*splitting*)
   a. Regresión: minimizar error cuadrático medio (MSE)
   b. Clasificación:
      i. Mínimo error de clasificación
      ii. Mínima impureza
      iii. Máxima entropía
3. Para cada región resultante repetimos el proceso (recursive splitting), hasta que se cumpla un criterio de parada:
   a. Todas las muestras con única variable target (y)
   b. Complejidad:
      i. Profundidad
      ii. Número de muestras en hoja
      iii. Mejora en el criterio de splitting

https://scikit-learn.org/stable/modules/tree.html#tips-on-practical-use

Métricas clasificación

Sea un problema de clasificación con K categorías. En el nodo m se define p_{km} como la proporción de observaciones de entrenamiento en dicho nodo para la clase k.

- Error de clasificación: $E(X_m) = 1 - \max\{p_{km}\}$
- Índice Gini: $G(X_m) = \sum p_{km}(1 - p_{km}) = 1 - \sum (p_{km})^2$
- Entropía: $D(X_m) = -\sum p_{km}(log(p_{km}))$

Donde X_m son los datos de entrenamiento en el nodo m.

Ejemplo sencillo

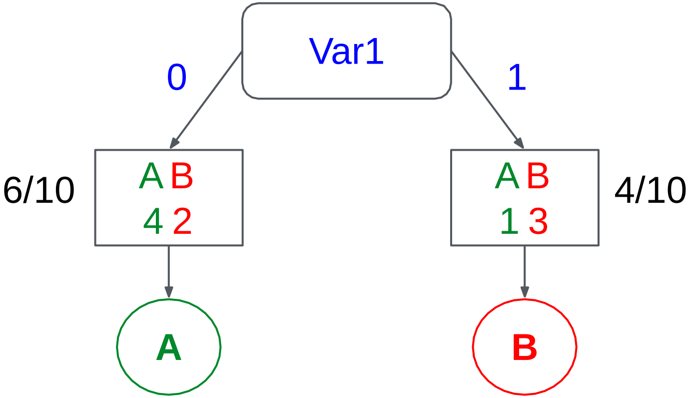
- ¿Por qué variable particionamos el árbol?

- Dos hipótesis:
  - Var1 == 1
  - Var2 >= 32

Label	Var1	Var2
A	0	33
A	0	54
A	0	56
A	0	42
A	1	50
B	1	55
B	1	31
B	0	-4
B	1	77
B	0	49

http://www.learnbymarketing.com/481/decision-tree-flavors-gini-info-gain/

Ejemplo sencillo



$G_{LEFT} = 1 - [\,(4/6)^2 + (2/6)^2\,] = 0.444$

$E_{LEFT} = 1 - \max\{4/6, 2/6\} = 2/6 = 1/3$

$G_{RIGHT} = 1 - [\,(1/4)^2 + (3/4)^2\,] = 0.375$

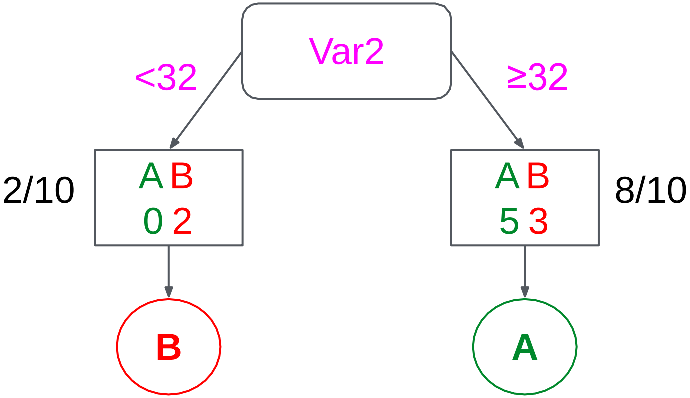
$E_{RIGHT} = 1 - \max\{1/4, 3/4\} = 1/4$

$\mathbf{G_{TOTAL}} = 6/10 \cdot 0.44 + 4/10 \cdot 0.375 = \mathbf{0.41667}$

$\mathbf{E_{TOTAL}} = 6/10 \cdot 1/3 + 4/10 \cdot 1/4 = 3/10 = \mathbf{0.3}$

Label	Var1	Var2
A	0	33
A	0	54
A	0	56
A	0	42
A	1	50
B	1	55
B	1	31
B	0	-4
B	1	77
B	0	49

Ejemplo sencillo



$$G_{LEFT} = 1 - [\,(0/2)^2 + (2/2)^2\,] = 0$$

$$G_{RIGHT} = 1 - [\,(5/8)^2 + (3/8)^2\,] = 0.469$$

$$E_{LEFT} = 1 - \max\{0/2, 2/2\} = 0$$

$$E_{RIGHT} = 1 - \max\{5/8, 3/8\} = 3/8$$

$$G_{TOTAL} = 2/10 \cdot 0 + 8/10 \cdot 0.469 = \mathbf{0.375}$$

$$E_{TOTAL} = 2/10 \cdot 0 + 8/10 \cdot 3/8 = 3/10 = \mathbf{0.3}$$

Label	Var1	Var2
A	0	33
A	0	54
A	0	56
A	0	42
A	1	50
B	1	55
B	1	31
B	0	-4
B	1	77
B	0	49

Ejemplo sencillo: resultado

- ¿Por qué variable particionamos el árbol?

- Dos hipótesis:
  - Var1 == 1
  - **Var2 >= 32**

- De este modo continuaríamos construyendo el árbol hasta cumplir criterio de parada

Label	Var1	Var2
A	0	33
A	0	54
A	0	56
A	0	42
A	1	50
B	1	55
B	1	31
B	0	-4
B	1	77
B	0	49

http://www.learnbymarketing.com/481/decision-tree-flavors-gini-info-gain/

Gini vs Error clasificación

- Preferible Gini (medida de impureza, ejemplo anterior)

- Valores pequeños significan que un nodo contiene predominantemente muestras de una única clase

- Entropía es similar a Gini.

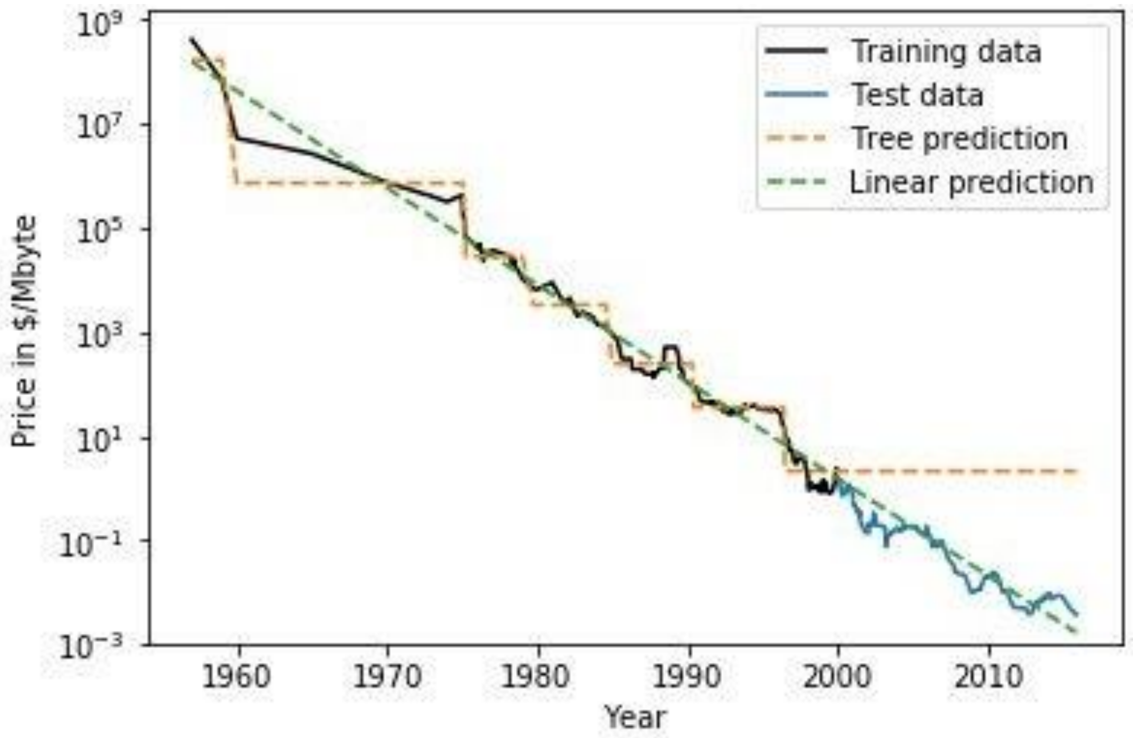
Índice

1. Intuición
2. Construcción del árbol
3. **Conclusiones**

Conclusiones

- Sencillos e interpretables
- Clasificación binaria o multiclase
- Variables numéricas y categóricas
- No necesidad de normalización
- Estimación de la probabilidad
- Útiles cuando se utilizan en combinación
    - Random Forest
    - Boosted Trees
- Cuando hay muchas variables, riesgo de overfitting: control de la complejidad
- Prestaciones peores que las de otros algoritmos
- No miran al futuro

Sobre series temporales



Sobre estimación de la probabilidad

- Se calcula como % de la clase mayoritaria en una hoja: P(y=k|x)

- Si el árbol no se poda, P(y=k|x) = 1!
  - No hay métodos de poda en sklearn
  - Es necesario, por tanto, controlar la complejidad

Referencias

- Introduction to Statistical Learning
  - Capítulo 8
- Hands On Machine Learning.
  - Capítulo 6

Let’s code!