Árboles

Homero

Bart

Lisa

Edna

Moe

Milhouse

Marge

Selma

Homero

Bart

Lisa

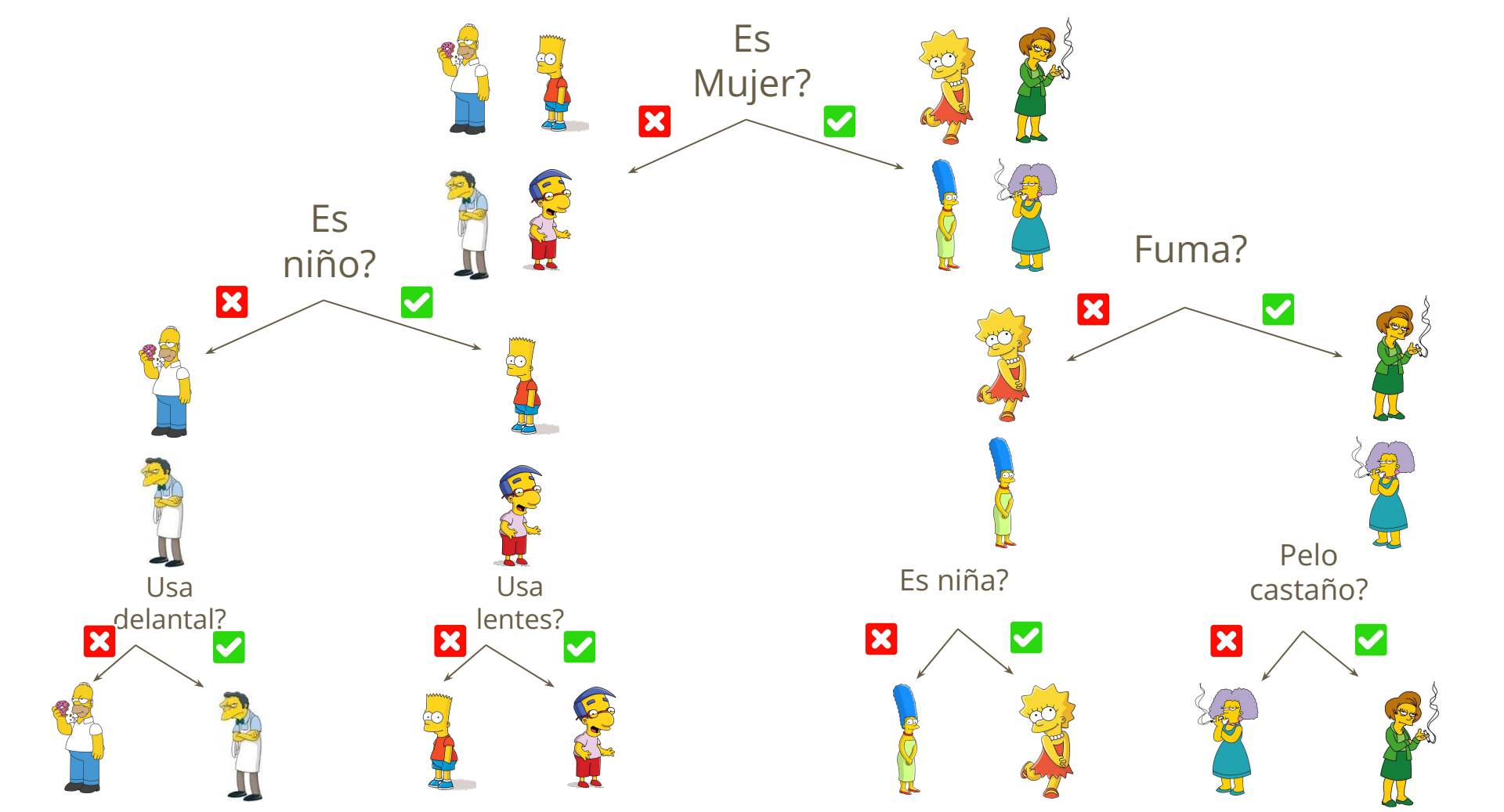
Edna

Moe

Milhouse

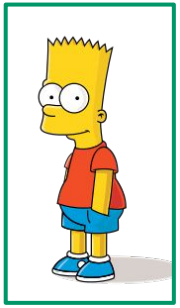
Marge

Selma


Es Mujer?
Es niño?
Fuma?
Usa delantal?
Usa lentes?
Es niña?
Pelo castaño?

Principal

Principal



Principal

Reparto

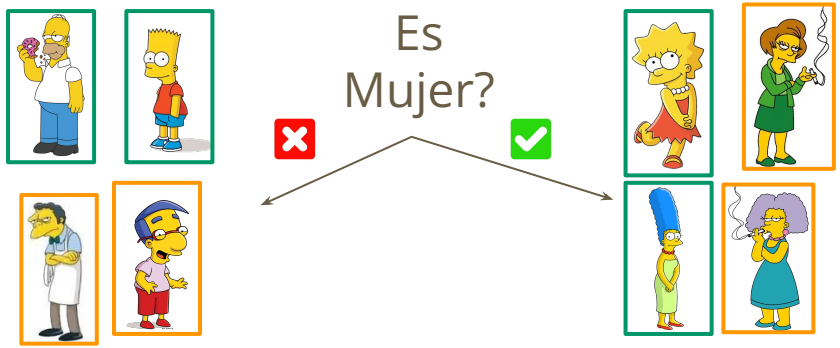
Reparto

Reparto

Principal

Reparto

2 principales
2 secundarios



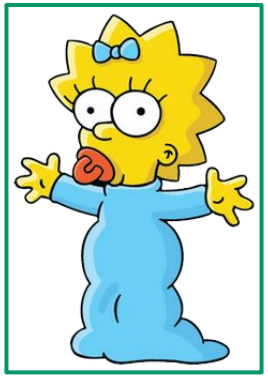
2 secundarios
2 reparto

0 principales
4 secundarios

4 principales
0 secundarios

Disco Stu

Ned Flanders



Maggie Simpson

Mona Simpson

Sobreajuste

Clasificamos perfectamente a los ejemplos que vimos cuando construimos el árbol, pero falla para ejemplos que no vimos.

El árbol clasifica correctamente 100% de las instancias usadas para construirlo, pero sólo 66% de los que aparecieron luego

Otros conceptos

El árbol -> modelo

Construir el árbol -> entrenar, ajustar, *fitear* el modelo

El conjunto de ejemplos usado -> training set, o dataset de entrenamiento

El conjunto de ejemplos en los que testeo el árbol -> test set o dataset de test

Las preguntas que hacemos en cada paso -> features o características

Principal o reparto -> etiqueta, clase, ground truth

% de clasificados correctamente -> métrica, en este caso accuracy

Otros conceptos

	Features o caracteristicas					Etiqueta
	Nombre	**Es mujer**	**Fuma**	**Usa lentes**	**Es menor**	**Principal o Reparto**
Train dataset	Homero Simpson	No	No	No	No	Principal
	Bart Simpson	No	No	No	Si	Principal
	Lisa Simpson	Si	No	No	Si	Principal
	Edna Krabappel	Si	Si	No	No	Reparto
	Moe Szyslak	No	No	No	No	Reparto
	Milhouse Van Houten	No	No	Si	Si	Reparto
	Marge Simpson	Si	No	No	No	Principal
	Selma Bouvier	Si	Si	No	No	Reparto
Test dataset	Disco Stu	No	No	Si	No	Reparto
	Maggie Simpson	Si	No	No	Si	Principal
	Mona Simpson	Si	No	No	No	Reparto

Sobre las características

El algoritmo solo ve (y puede usar) las características que estén en el dataset

La calidad de las features es **muy** determinante

En este ejemplo son todas booleanas, pero podrían ser números (edad, altura, peso, tiempo en pantalla, capítulos en los que aparece, primer capítulo en que apareció) o categóricas (color de la piel, apellido)

Puedo mejorarlas combinandolas o pre-procesandolas, eso se llama **feature engineering**: ej nombre -> se apellida simpson?

Aprendizaje Automático basado en árboles

Automatizar la construcción de estos árboles basados en datos

En cada paso, elegir “la mejor pregunta”, o sea, el atributo por el que voy a partir mi dataset, de manera automática, basada en datos.

Veamos una forma de hacerlo: **ID3**

ID3

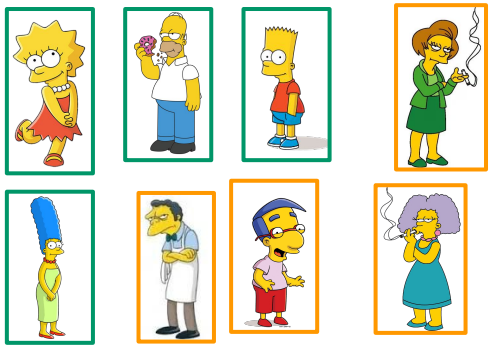
Es un algoritmo sencillo para problemas de clasificación binaria

1. Busca la mejor pregunta (atributo)
    a. La que más separa los ejemplos según la etiqueta.
    b. Mide cuánto reduce la incertidumbre (entropía).
2. Divide el dataset según las respuestas a esa pregunta.
3. Repite el proceso recursivamente con cada conjunto:
    a. Elimina la pregunta ya usada.
    b. Se detiene cuando:
        i. Todos los ejemplos tienen la misma etiqueta.
        ii. No quedan más preguntas.

Evaluar un atributo

$$\text{Entropía}(S) = -\sum_{i=1}^{c} p_i \cdot \log_2(p_i)$$

4 principales
4 reparto



$$\text{Entropía}(S) = -(p_{\text{principal}} \cdot \log_2(p_{\text{principal}}) + p_{\text{reparto}} \cdot \log_2(p_{\text{reparto}}))$$

$$P_{\text{reparto}} = \frac{4}{8} = 0.5$$

$$P_{\text{principal}} = \frac{4}{8} = 0.5$$

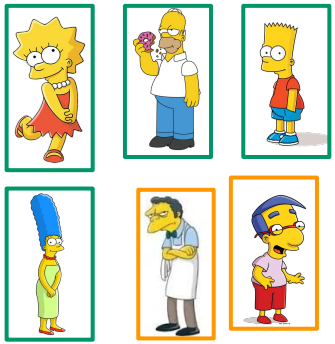
$$\text{Entropía}(S) = -(0.5 \cdot \log_2(0.5) + 0.5 \cdot \log_2(0.5))$$

$$\text{Entropía}(S) = 1$$

Evaluar un atributo: fuma

$$\text{Entropía}(S) = -\sum_{i=1}^{c} p_i \cdot \log_2(p_i)$$

4 principales
2 secundarios



Fuma?

0 principales
2 reparto

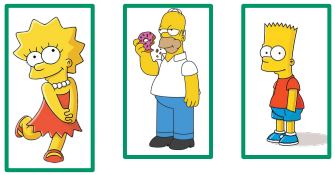
$$\begin{aligned}\text{Entropía(no fuma)} &= -\left(P_{\text{reparto}} \cdot \log_2(P_{\text{reparto}}) + P_{\text{principal}} \cdot \log_2(P_{\text{principal}})\right) \\ &= -\left(\frac{2}{6} \cdot \log_2(\frac{2}{6}) + \frac{4}{6} \cdot \log_2(\frac{2}{6})\right) \\ &\approx 0.92\end{aligned}$$

$$\begin{aligned}\text{Entropía(fuma)} &= -\left(P_{\text{reparto}} \cdot \log_2(P_{\text{reparto}}) + P_{\text{principal}} \cdot \log_2(P_{\text{principal}})\right) \\ &= -\left(1 \cdot \log_2(1) + 0 \cdot \log_2(0)\right) \\ &= 0\end{aligned}$$

Evaluar un atributo: fuma

$$\text{Entropía}(S) = -\sum_{i=1}^{c} p_i \cdot \log_2(p_i)$$

4 principales
2 secundarios



Fuma?

0 principales
2 reparto

Cuánta entropía gano si parto por fuma?

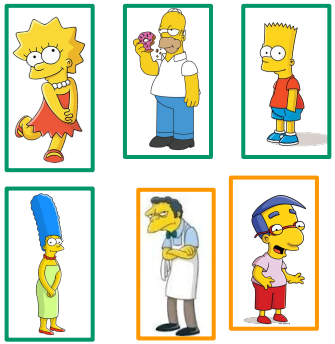
$$\text{Información ganada}(S, A) = \text{Entropía}(S) - \sum_{v \in \text{Valores}(A)} \frac{|S_v|}{|S|} \text{Entropía}(S_v)$$

$$\text{Información ganada}(S, \text{fuma?}) = 1 - \left(\frac{2}{8} \cdot 0 + \frac{6}{8} \cdot 0.92\right)$$

$$= 0,31$$

Elijo el mejor atributo información

4 principales
2 secundarios



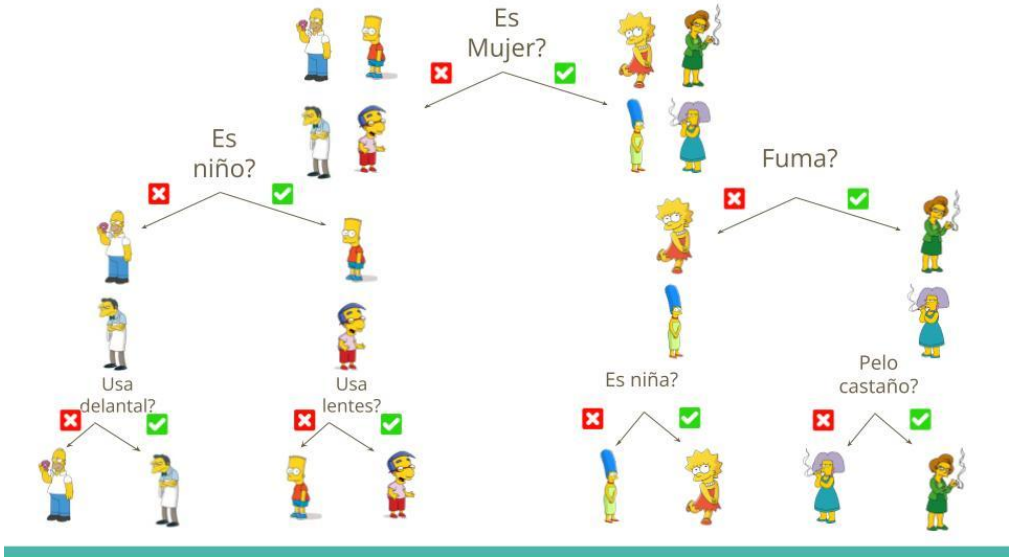
0 principales
2 reparto

- Evalúo todos los atributos
- Parto el dataset por aquel que me hace ganar más información
- Replico el procedimiento en cada uno de los nuevos subconjuntos, hasta que todos los elementos del conjunto sean de la misma clase.

Árboles: fáciles de interpretar

Lisa: es mujer y no fuma y es niña

Milhouse: no es mujer y es niño y usa lentes



Árboles

- Se pueden extender a más de 2 clases
- Se pueden extender a regresiones
- Soportan atributos numéricos (Ej: edad > 18?)
- Es muy eficiente: Si tengo n datos, con log(n) preguntas puedo definir una hoja para cada dato (asumiendo que son distinguibles).

Es mujer	Fuma	Usa lentes	Es menor	Principal o Reparto
No	No	No	No	Principal
No	No	No	No	Reparto

Árboles: sobreajustan fácilmente

Con pocas preguntas puedo poner a cada ejemplo de entrenamiento en una hoja

Cómo evitarlo?

Árboles: sobreajustan fácilmente

Navaja de Ockham: En igualdad de condiciones, la explicación más sencilla, suele ser más correcta.

Principio de simplicidad o parsimonia

A) **Tormenta**: El viento lo derribó.

B) **Mulita**: Estuvo excavando cerca, debilitó las raíces, y el árbol cedió.

C) **Tambor**: El vecino tocando el tambor en frecuencias bajas, atrajo a una mulita en celo que excavó bajo el árbol, provocando su caída.

Árboles: sobreajustan fácilmente

Principal: es mujer y no fuma y es niña y usa collar y no usa chupete y usa vestido y es amarilla y no usa lentes y ...

Secundario: no es mujer y es niño y usa lentes y no usa collar y tiene pelo azul y no tiene bigotes y ...

Árboles: sobreajustan fácilmente

- Limitar la profundidad
- Limitar la cantidad mínima de instancias para partir un conjunto
- Limitar la cantidad mínima de instancias que admito en una hoja

Limitar la capacidad del árbol hace que **generalice** más: lo que aprendió en los datos de entrenamiento sea útil en el resto del universo.

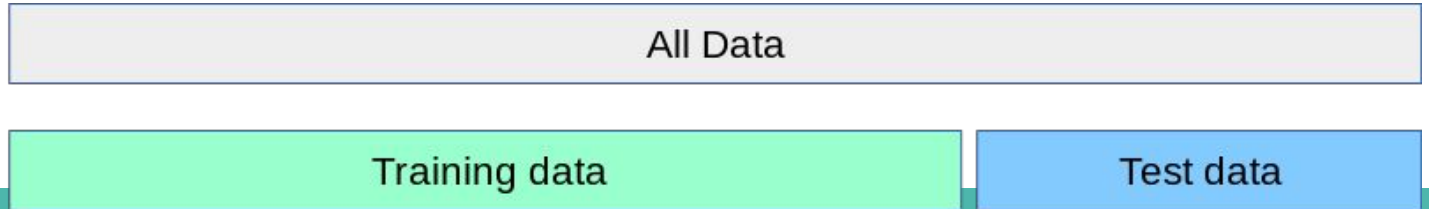
Train y test set

Queremos un modelo que funcione bien no solo en los datos que vio, sino también en personajes nuevos, incluso los que aún no existen.

Para lograr eso, tomamos una muestra representativa del universo y la dividimos en dos partes:

- Conjunto de **entrenamiento**: lo usamos para que el modelo aprenda.
- Conjunto de **test**: lo guardamos aparte, sin mostrarlo durante el entrenamiento, para evaluar si el modelo realmente generaliza.

Este conjunto de test debe ser independiente y se usa recién al final, porque es lo más parecido a lo que el modelo verá en producción.



Train - Val - Test set

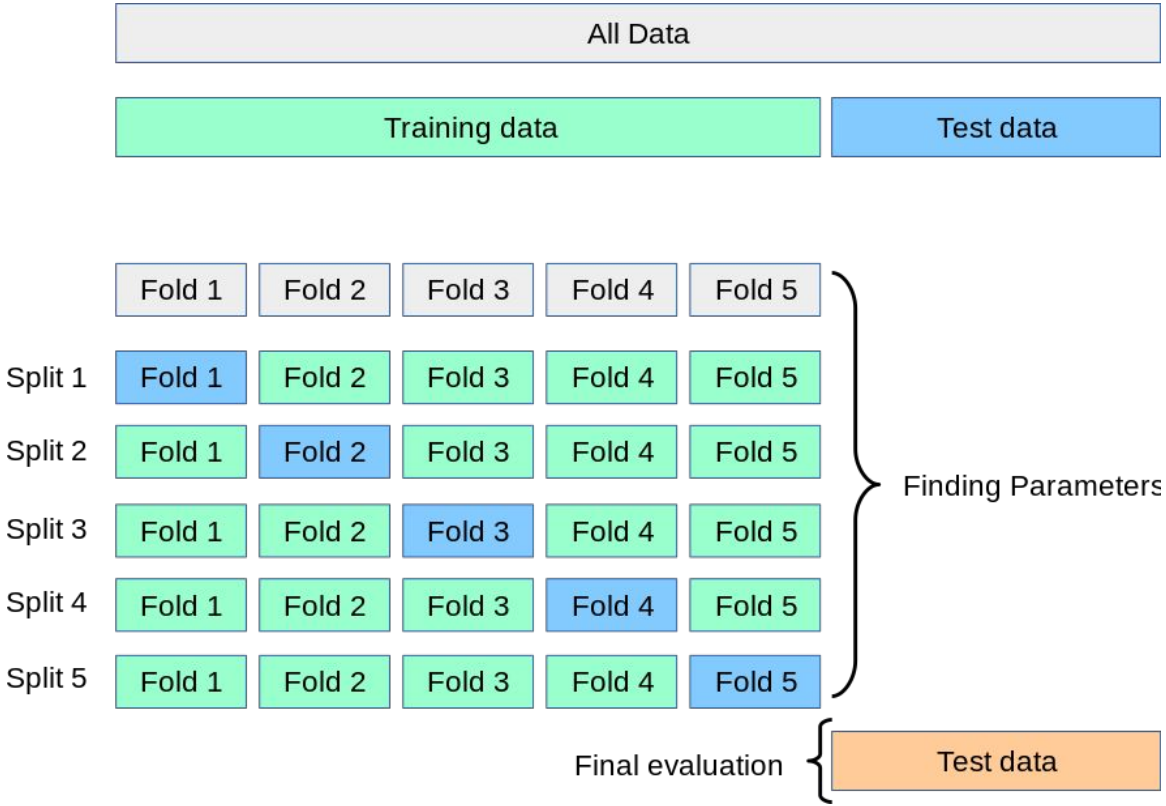
scikit: `sklearn.model_selection.train_test_split`

Si usamos el conjunto de test para tomar decisiones intermedias, corro riesgo de sobreajustarme a los datos de test!

Para evitar esto las opciones más frecuentes son volver a partir en entrenamiento y validación o utilizar **validación cruzada.**

Va a depender de la cantidad de datos disponibles

Validación cruzada



Métricas

Es importante definir qué métrica(s) usar antes de empezar

La métrica a usar depende del problema

Elegir una métrica a optimizar, pero igualmente monitorear las demás

Métricas: clasificación

- En problemas de clasificación, casi todas las métricas parten de la **matriz de confusión**
- Permite comparar el valor real, con el resultado de la clasificación.

positive := principal

Métricas: clasificación

La primera medida que se viene a la mente es el acierto:

$$Acierto = \frac{TP + TN}{TP + FP + TN + FN}$$

¿Qué problemas pueden haber?

Métricas: clasificación

Estas medidas sí tienen en cuenta qué pasa en las distintas clases de nuestro dataset.

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN} \qquad F1 = 2\frac{P * R}{P + R}$$

- Miden cosas distintas.
- Dependiendo del sistema, conviene priorizar una o la otra.
- El f1-score pondera ambas.

Métricas: regresión

Regresión: el modelo predice una **cantidad**

¿Por qué no probar con el acierto? (Accuracy)

Real (ground truth)	Predicho	Acierto
20000	20000	1
35000	35001	0
40000	20000	0

Métricas: regresión - MAE

$$\mathrm{MAE}(y, \hat{y}) = \frac{1}{n_{\text{ejemplos}}} \sum_{i=0}^{n_{\text{ejemplos}}-1} |y_i - \hat{y}_i| .$$

Error absoluto medio (Mean Absolute Error)

- Todos los errores pesan lo mismo
- Mantiene las unidades
- Variante sencilla: tomar la mediana en vez del promedio

Métricas: regresión - MSE

$$\mathrm{MSE}(y, \hat{y}) = \frac{1}{n_{\text{ejemplos}}} \sum_{i=0}^{n_{\text{ejemplos}}-1} (y_i - \hat{y}_i)^2.$$

Error cuadrático medio (Mean Squared Error)

- NO todos los errores pesan lo mismo
- NO Mantiene las unidades
- Variante sencilla: tomar la raíz cuadrada, RMSE

Métricas: regresión - Error Máximo

$$\text{Max Error}(y, \hat{y}) = max(|y_i - \hat{y}_i|)$$

- **MUY** sensible a outliers
- Mantiene las unidades

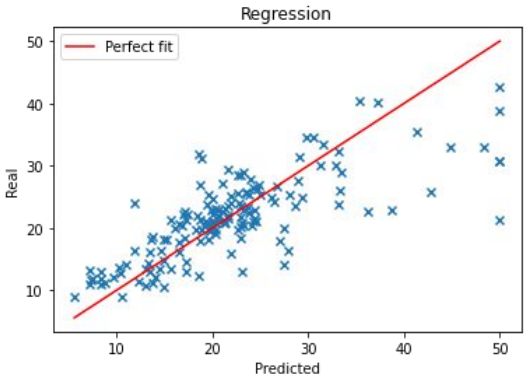
Métricas: regresión - R^2

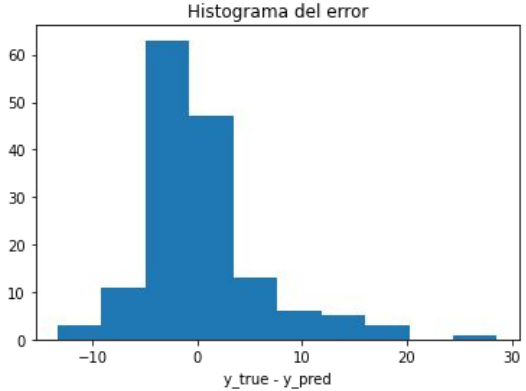
$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

Coeficiente de determinación R^2

- Modelo predice siempre promedio, da 0
- El máximo es 1, y es un ajuste perfecto
- No hay unidades

Métricas: regresión





Es útil ver el error gráficamente como un scatter entre (`y_true, y_pred`)
Podemos hacer un histograma, debería quedar centrado en cero y con desviación estándar pequeña.

¿Es Aprendizaje Automático?

Métodos que permiten a las computadoras “aprender”: lograr mejor desempeño en determinada tarea a partir de la experiencia.

Accuracy

*«Un programa de computadora **aprende** de la experiencia E con respecto a alguna clase de **tareas T** y de una medida de **rendimiento P**, si su rendimiento en las tareas de **T, medida por P, mejora** con la experiencia **E**»*

Principal o Secundario?

Tom Mitchell, Machine Learning, 1997

Datos de entrenamiento