

APRENDIZAJE SUPERVISADO

Guillem Sitges i Puy

Sesiones 19 a 24

Introducción al aprendizaje supervisado

Estrategias de validación

Métricas de Clasificación

Métricas de Regresión

Outliers y normalización

Correlación

Decision Trees

Ensamblés

Modelos Lineales

Remuestreo

ÍNDICE END-TO-END MACHINE LEARNING



INTRODUCCIÓN
AL APRENDIZAJE
SUPERVISADO

1

MÉTRICAS PARA
CLASIFICACIÓN

3

OUTLIERS Y
NORMAIZACIÓN

5

DECISION TREES

7

MODELOS LINEALES

9

ESTRATEGIAS DE
VALIDACIÓN

2

MÉTRICAS PARA
REGRESIÓN

4

CORRELACIÓN

6

ENSAMBLES

8

REMUESTREO

10

01. INTRODUCCIÓN AL APRENDIZAJE SUPERVISADO

1

APRENDIZAJE SUPERVISADO: DEFINICIÓN FORMAL

Objetivo del aprendizaje supervisado: a partir de un Dataset, aprender una función f de los inputs que aproxime de la mejor manera posible el Target. Esta función aprendida de los datos se conoce como modelo.

Este target puede ser discreto o continuo, y hablaremos de problemas de clasificación o regresión, respectivamente.

Predictores o Atributos Target o Clase

Predictores o Atributos			Target o Clase
Cabeza	Cuerpo	Color	Churn
Cuadrada	Redondo	Blanco	No
Redonda	Redondo	Negro	Yes
Cuadrada	Cuadrado	Blanco	Yes
Cuadrada	Cuadrado	Blanco	Yes
Cuadrada	Cuadrado	Blanco	Yes
Redonda	Cuadrado	Negro	No

$$Y = f(A_1, A_2, \dots, A_n)$$

Y : Target
 A_i : Predictores o Atributos

1

APRENDIZAJE SUPERVISADO: DEFINICIÓN PRÁCTICA

En la práctica, el aprendizaje supervisado consiste en **separar la población en grupos que difieren de otros en relación a la presencia de cierta variable de interés o target**, que será lo que intentemos predecir. Este target puede ser algo que queramos evitar, como la fuga de un cliente, o algo positivo, como que clientes van a clicar un determinado anuncio o que páginas son más adecuadas para un determinado anuncio; también podemos predecir valores continuos, como el consumo de energía, el importe de las ventas o variables con varias categorías, como una clasificación en tipos (alto/medio/bajo, por ejemplo).

El aspecto clave del aprendizaje supervisado es como localizar o seleccionar determinadas variables informativas o atributos que nos ayuden a realizar una previsión sobre el evento de interés, y por **variable informativa o explicativa entendemos la que nos ayuda a reducir la incertidumbre sobre la variable a predecir o target**. Cuanto mejor es la información proporcionada por la variable, más se reduce la incertidumbre (más homogéneo es el grupo generado). Los distintos algoritmos de clasificación supervisada implementan esta idea de uno u otro modo.

APRENDIZAJE SUPERVISADO: **EJEMPLOS**

En la práctica, el aprendizaje supervisado consiste en **separar la población en grupos que difieren de otros en relación a la presencia de cierta variable de interés o target**, que será la que intentemos predecir. Este target puede ser algo que queramos evitar, como la fuga de un cliente, o algo positivo, como que clientes van a clicar un determinado anuncio o que páginas son más adecuadas para un determinado anuncio

En este punto introduciremos una de las ideas principales del Data Science, que es **como localizar o seleccionar determinadas variables informativas o atributos** que nos ayuden a realizar una previsión sobre un evento futuro.

Diremos que **una variable es informativa o explicativa cuando nos ayuda a reducir la incertidumbre sobre la variable a predecir o target**. Cuanto mejor es la información proporcionada por la variable, más se reduce la incertidumbre. Los distintos algoritmos de clasificación supervisada implementan esta idea de uno u otro modo.

1

APRENDIZAJE SUPERVISADO: ALGORITMOS

Los algoritmos de aprendizaje son procedimientos o conjuntos de pasos que reciben un conjunto de experiencias (en forma de dataset) y devuelven una función de tipo matemático o lógico capaz de estimar el **target**. Son, por tanto, el mecanismo del que disponemos para aprender el modelo a partir de los datos.

Existen diferentes algoritmos en función del tipo de procedimiento utilizado para inferir el modelo, y diremos que cada uno define un espacio de hipótesis distinto, con diferentes características como la expresividad o la complejidad.

MÉTODOS DE
REGLAS

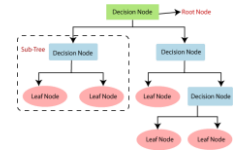
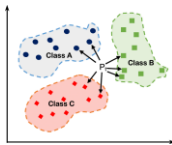
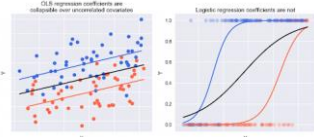
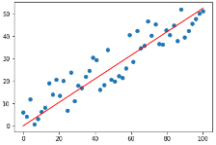
MÉTODOS DE
VECINDAD

MÉTODOS
GEOMÉTRICOS

MÉTODOS DE
PROBABILIDAD

CLASIFICACIÓN

REGRESIÓN

 <p>CLASSIFICATION AND REGRESSION TREES</p>	
 <p>K-NEAREST NEIGHBORS</p>	
 <p>LOGISTIC REGRESSION / SVM</p>	 <p>LINEAR REGRESSION</p>
$P(c x) = \frac{P(x c)P(c)}{P(x)}$ <p>Likelihood: $P(x c)$ Class Prior Probability: $P(c)$ Posterior Probability: $P(c x)$ Predictor Prior Probability: $P(x)$</p> <p>NAÏVE BAYES</p>	

02. ESTRATEGIAS DE VALIDACIÓN

2

ESTRATEGIAS DE VALIDACIÓN: EL BIAS-VARIANCE TRADEOFF

Uno de los elementos clave en la preparación de un modelo de Machine Learning es el ajuste de modelos con **buena capacidad de generalización**.

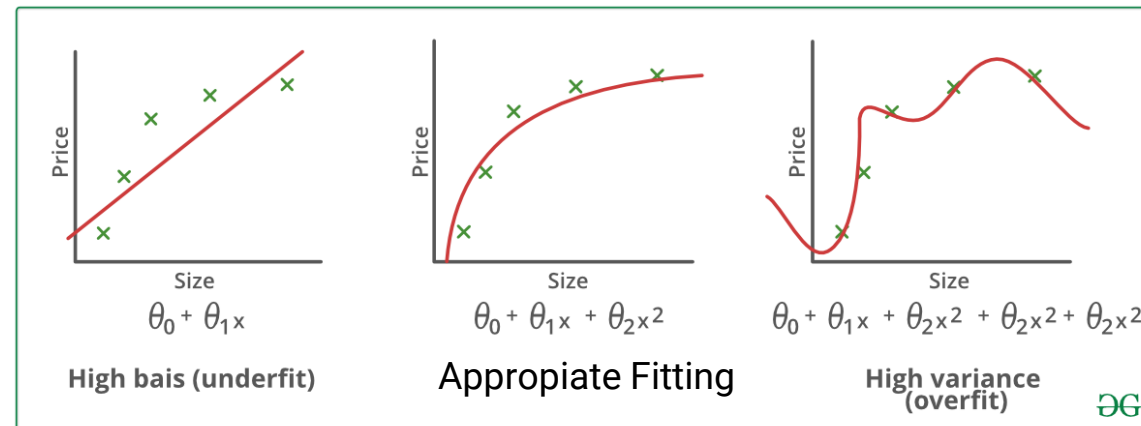
En ese sentido, **no es positivo que el modelo se ajuste en exceso al conjunto de datos utilizados para el entrenamiento**, pues se produce la **memorización** del mismo y se obtienen modelos con baja capacidad de generalización.

- Cuando esto ocurre, decimos que se ha producido un **overfitting**, pues el algoritmo se ha ajustado en exceso a los datos, y hablamos de exceso de varianza (variance).

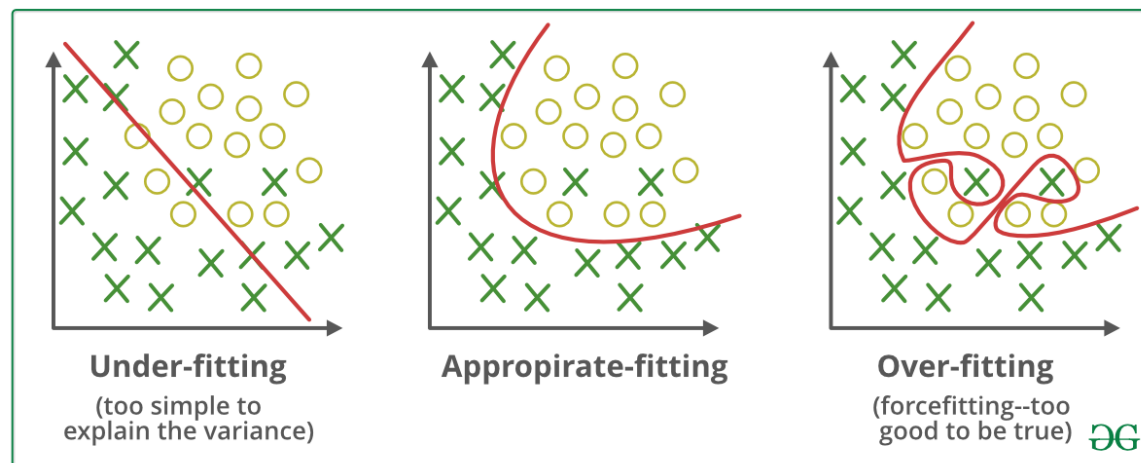
Asimismo, no es adecuado realizar modelos excesivamente simples o formales, que no ajusten bien la frontera de decisión del conjunto de datos.

- Cuando esto ocurre, decimos que se ha producido un **underfitting**, pues el algoritmo no ha sido capaz de ajustar las fronteras de decisión de los datos, y hablamos de exceso de sesgo (bias).

Este compromiso (tradeoff) entre modelos robustos (capaces de generalizar) y ajustados (precisos) se conoce como bias-variance tradeoff, y es una de las claves del éxito para generar buenos modelos de Machine Learning.



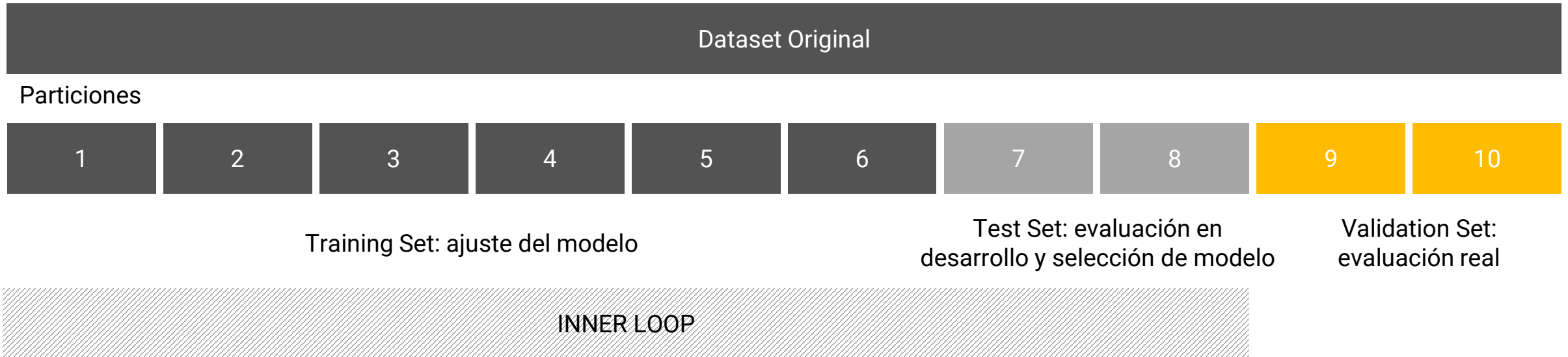
Bias-Variance tradeoff en Regresión



Bias-Variance tradeoff en Clasificación

2 ESTRATEGIAS DE VALIDACIÓN: RANDOM HOLDOUT

Se considera que un modelo de Machine Learning es preciso cuando mediante el mismo obtenemos **buenas métricas de evaluación en los conjuntos de test y validación**.

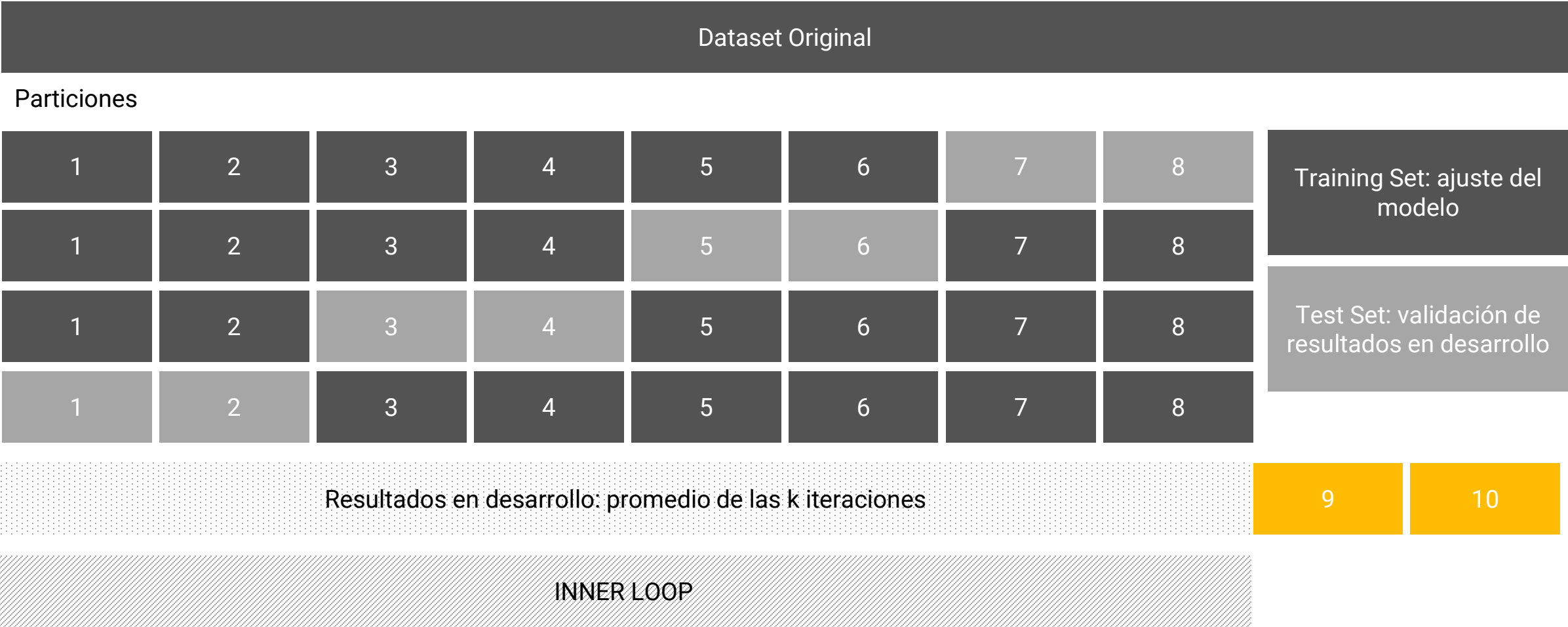


La selección de la estrategia de validación es uno de los puntos más importantes en el desarrollo de modelos de Machine Learning, y es **importante ajustarla a las características del problema concreto que estamos intentando resolver**. En ese sentido, **el tamaño de las particiones debe ajustarse al tamaño del dataset**, de manera que la partición de entrenamiento sea suficiente para ajustar el modelo y las particiones de validación sean suficientemente grandes para obtener métricas de evaluación estadísticamente significativas.

Durante el curso revisaremos métodos avanzados de validación como k-fold o bootstrap orientados a lidiar con este tipo de problemas.

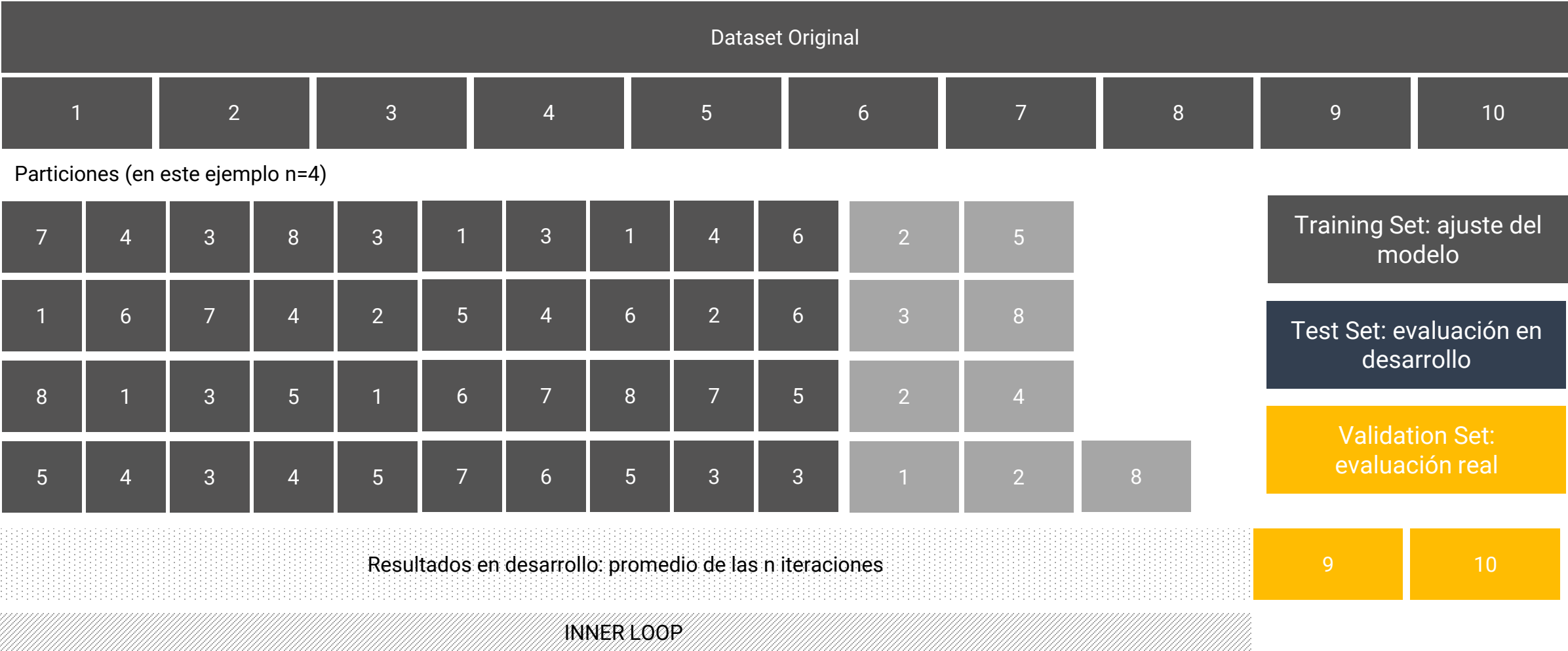
2 ESTRATEGIAS DE VALIDACIÓN: K-FOLD CROSS VALIDATION

Una alternativa para realizar la validación con mayor fiabilidad estadística es la evaluación por k-particiones (k-fold), que **consiste en repetir el proceso de modelización en train-test k veces** para obtener después la métrica de rendimiento como promedio.



2 ESTRATEGIAS DE VALIDACIÓN: BOOTSTRAP

En caso de que tengamos muy pocos datos para realizar la validación, es adecuado realizar un bootstrap (n repeticiones), y obtener un resultado promedio.

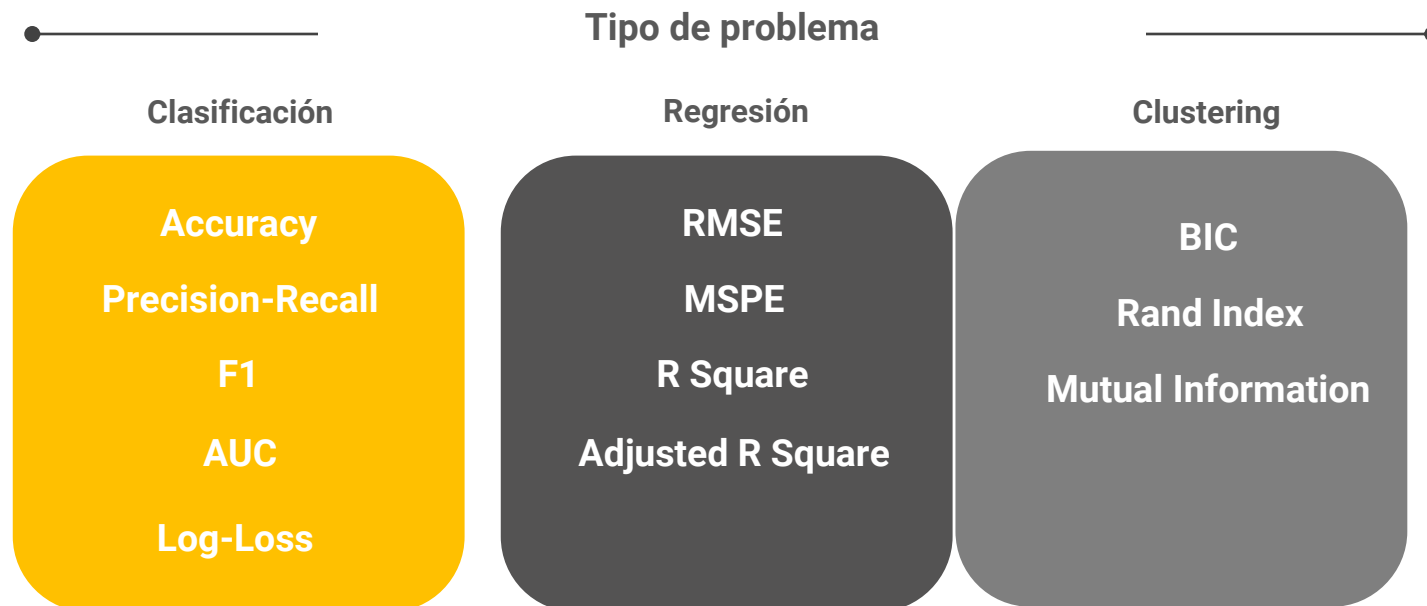


03. MÉTRICAS DE VALIDACIÓN PARA CLASIFICACIÓN

3 MÉTRICAS DE VALIDACIÓN: BASICS

Para evaluar la bondad del ajuste de un modelo de Machine Learning, es importante definir una buena **métrica de validación**. Esta métrica debe representar adecuadamente la capacidad de acierto del modelo al realizar la inferencia, y debe ser escogida adecuadamente teniendo en cuenta:

- El **tipo de problema** que estemos trabajando (existen diferentes métricas para las distintas tareas de Machine Learning)
- El volumen del dataset y, dentro de éste, el **número de ejemplos de la clase a predecir** cuando nos encontremos en problemas de clasificación



3 MÉTRICAS DE VALIDACIÓN: CONFUSION MATRIX Y ACCURACY

Para evaluar el rendimiento de los modelos de clasificación (etiqueta categórica), es habitual utilizar la **confusion matrix** o matriz de confusion, que traspone los ejemplos de test o validación en función de su clase real y la clase predicha.

La métrica más básica de validación en clasificación supervisada es el **Accuracy o Precisión**, y se calcula como el número de ejemplos clasificados de forma correcta en relación al total.

El principal problema del Accuracy es que depende de la tasa de aprendizaje (número de instancias de la clase positiva respecto al total); pensemos, por ejemplo, en un problema con una tasa de aprendizaje del 1% (1% de instancias con clase positiva): un clasificador trivial, que clasificase todos los ejemplos negativos, obtendría una precisión del 99%.

		Clase de la predicción	
		Positiva	Negativa
Clase real	Positiva	True Positive (TP)	False Negative (FN) Type II error
	Negativa	False Positive (FP) Type I error	True Negative (TN)

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

3 MÉTRICAS DE VALIDACIÓN: F1 SCORE

Una alternativa para evaluar el rendimiento de los modelos que **no depende de la tasa de aprendizaje es el par *Precision-Recall*** (precisión y alcance), que se resume en una métrica conocida como F1-Score (media armónica de *Precision* y *Recall*, muy influenciada cuando hay un valor pequeño).

		Clase de la predicción	
		Positiva	Negativa
Clase real	Positiva	True Positive (TP)	False Negative (FN) Type II error
	Negativa	False Positive (FP) Type I error	True Negative (TN)

Precision
 $\frac{TP}{TP + FP}$

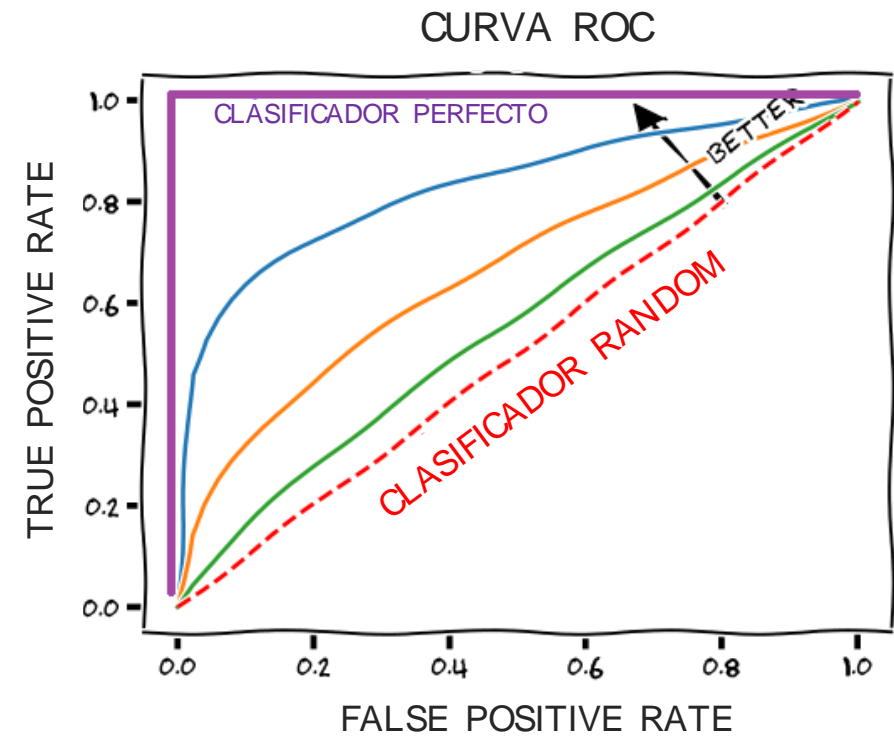
Recall
 $\frac{TP}{TP + FN}$

$$F1 = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}}$$

3 MÉTRICAS DE VALIDACIÓN: CURVA ROC Y AUC

La métrica de validación en clasificación más ampliamente utilizada es la curva ROC, que permite además evaluar el modelo cuando éste predice probabilidades, obteniendo una representación gráfica del rendimiento y una métrica agregada (el AUC o *Area Under the Curve*).

		Clase de la predicción		
		Positiva	Negativa	
Clase real	Positiva	True Positive (TP)	False Negative (FN) <i>Type II error</i>	$TPR = \frac{TP}{TP + FN}$ $FNR = \frac{FN}{TP + FN}$
	Negativa	False Positive (FP) <i>Type I error</i>	True Negative (TN)	$FPR = \frac{FP}{FP + TN}$ $TNR = \frac{TN}{FP + TN}$



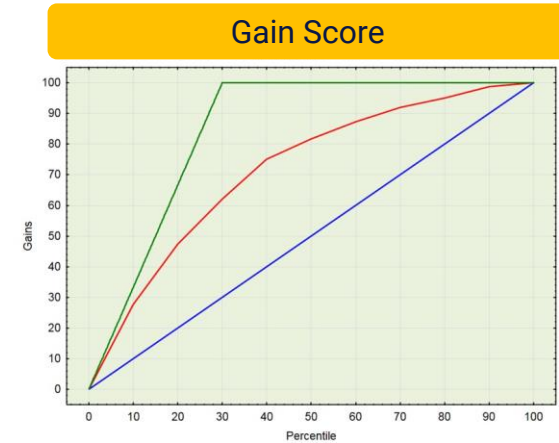
La curva de ROC es la representación gráfica de todas las combinaciones (TPR,FPR) variando el umbral de score para determinar si una predicción es P o F.

3 MÉTRICAS DE VALIDACIÓN: CURVAS DE LIFT Y GAIN

Una alternativa a la curva de ROC son las curvas de Lift y Gain, que consisten en **medir cuánto aumenta el acierto o rendimiento de una acción con una selección ordenada según el score del modelo vs la situación de base (sin modelo)**. Las curvas de Lift y Gain son muy utilizadas en el ámbito del marketing analítico, aunque se puede extender su uso a cualquier ámbito. Su principal **ventaja respecto a ROC es su interpretabilidad**, aunque tienen el **gran inconveniente de ser sensibles a la tasa de aprendizaje** (ratio de prevalencia de la clase positiva).

¿Cómo construir curvas de Lift y Gain?

1. Realizar un score (continuo) de la partición de validación
2. Ordenar las observaciones de validación de manera descendente en función del score
3. Realizar un Split en 10 particiones de las observaciones de validación ordenadas (deciles) –también pueden realizarse 100 particiones
4. Obtener las siguientes métricas
 - Número y porcentaje de observaciones en cada decil
 - Número y porcentaje de éxitos en cada decil
 - Número y porcentaje de observaciones acumuladas en cada decil
 - Número y porcentaje de éxitos acumulados en cada decil (**Gain Score**)
 - Multiplicador del gain en el decil respecto a la tasa de base (**Lift Score**)



04. MÉTRICAS DE VALIDACIÓN PARA REGRESIÓN

4

MÉTRICAS DE REGRESIÓN: BASICS

Mean squared error

$$\text{MSE} = \frac{1}{n} \sum_{t=1}^n e_t^2$$

Root mean squared error

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{t=1}^n e_t^2}$$

Mean absolute error

$$\text{MAE} = \frac{1}{n} \sum_{t=1}^n |e_t|$$

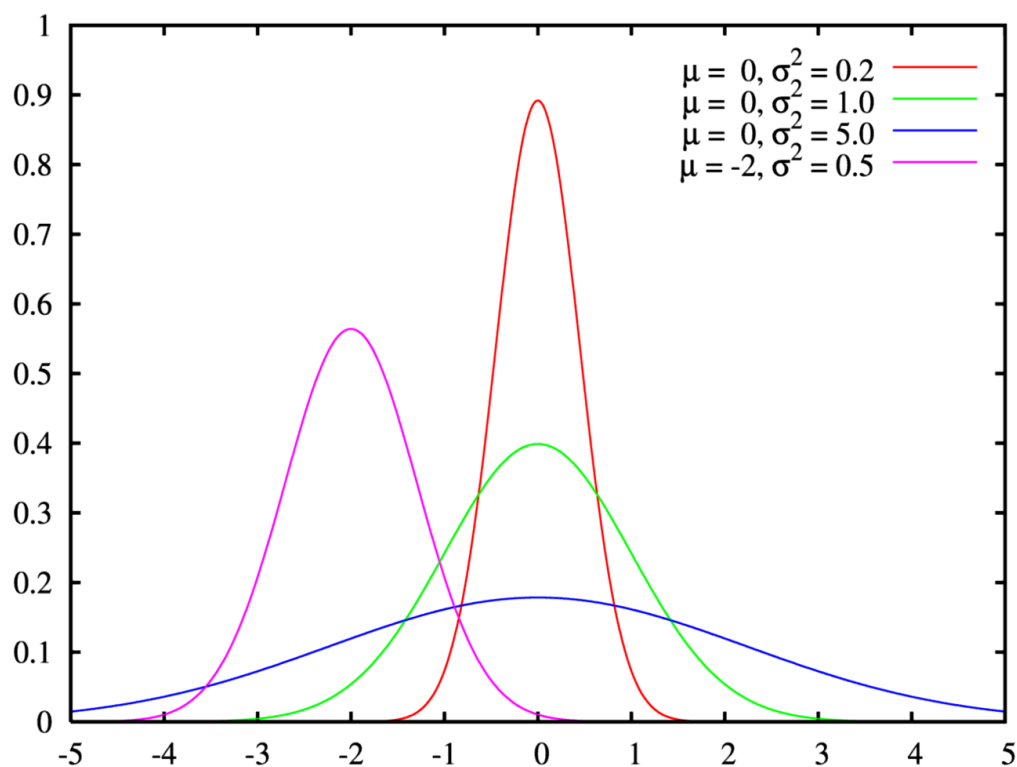
Mean absolute percentage error

$$\text{MAPE} = \frac{100\%}{n} \sum_{t=1}^n \left| \frac{e_t}{y_t} \right|$$

05. OUTLIERS Y NORMALIZACIÓN

5 LA DISTRIBUCIÓN NORMAL: INTRODUCCIÓN

Nos referimos a una **distribución de probabilidad** como la función que nos permite mapear a cada posible valor de una variable (eje de abcisas) su **probabilidad de ocurrencia**. La distribución normal o de Gauss es la distribución de probabilidad de variable continua más frecuente, y aparece frecuentemente en estadística y Machine Learning, pues permite modelar diversos fenómenos tanto naturales como artificiales. La distribución normal queda caracterizada por dos valores: la media y la desviación tipo.



- **Media o promedio:** número que representa la esperanza (valor esperado) de la distribución, obtenido sumando todos los valores y dividiendo entre el número total de elementos de la muestra. Para la distribución normal coincide con la mediana.

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i$$

- **Desviación tipo:** número que representa la dispersión de la distribución, obtenido promediando la desviación de los datos respecto a la media. Al cuadrado de la desviación tipo se le denomina varianza.

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}$$

5 LA DISTRIBUCIÓN NORMAL: INTERVALOS DE CONFIANZA

$$\mu \pm \sigma$$

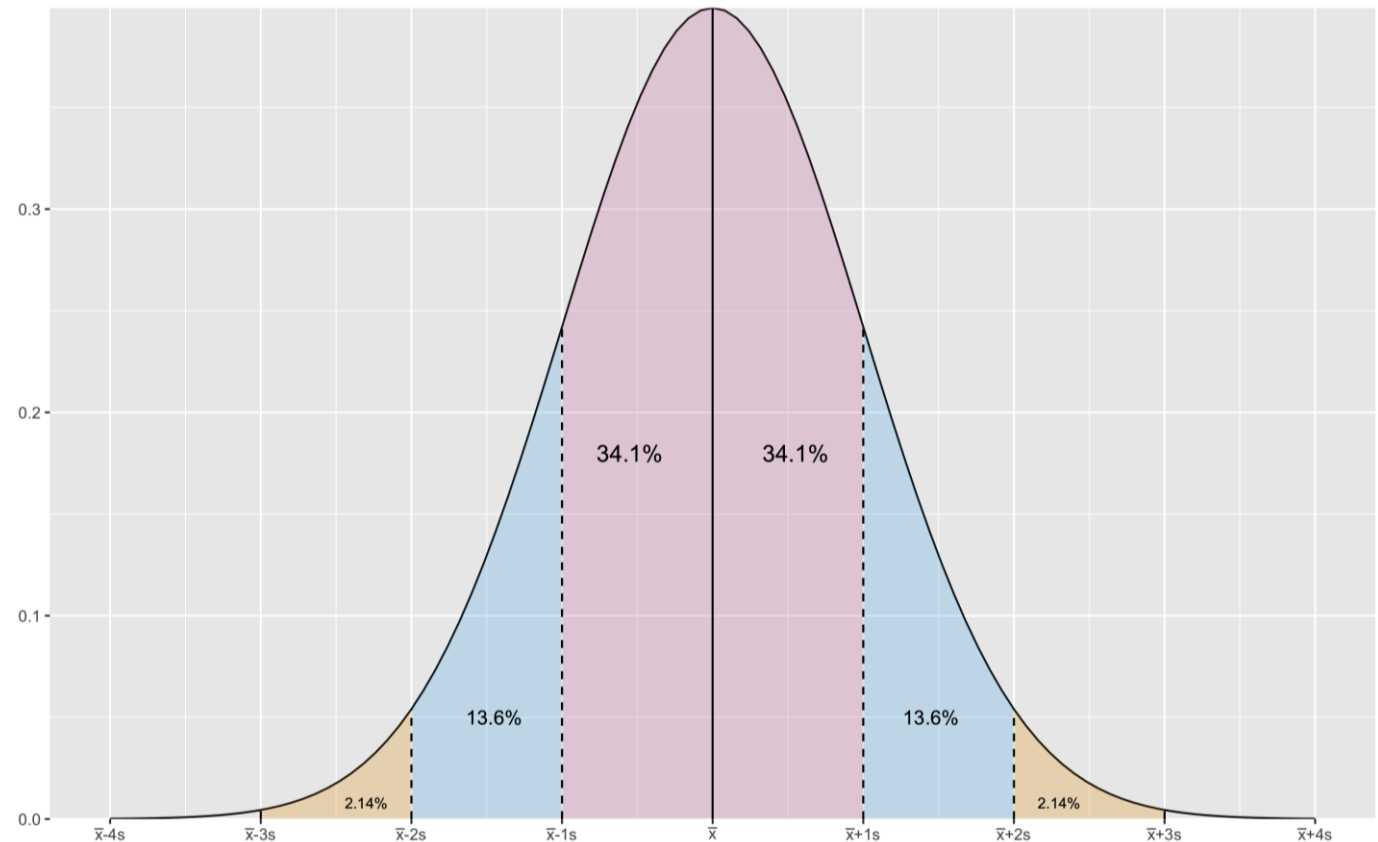
- Incluirán el 68% de los datos

$$\mu \pm 2\sigma$$

- Incluirán el 95% de los datos

$$\mu \pm 3\sigma$$

- Incluirán prácticamente todas las observaciones.

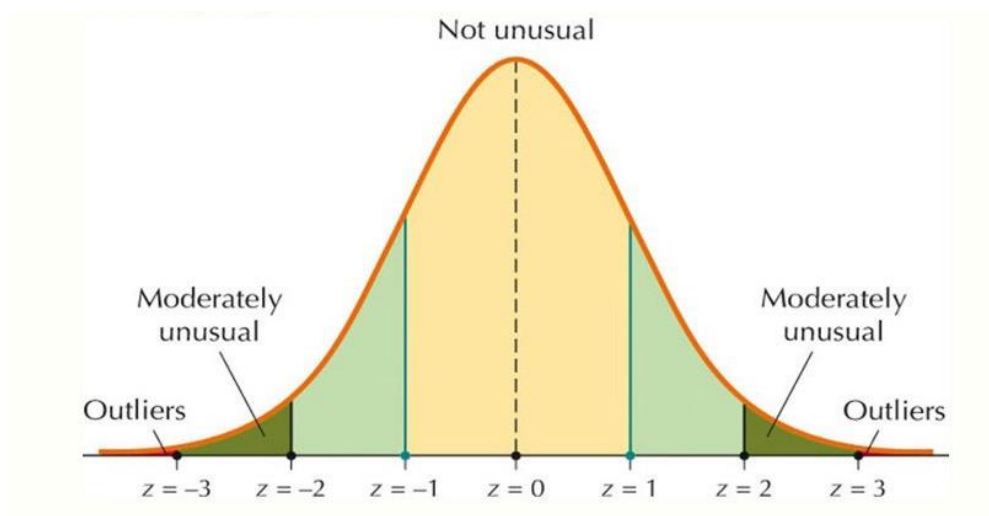


5 LA DISTRIBUCIÓN NORMAL: OUTLIERS

La distribución normal se utiliza frecuentemente para detectar valores atípicos (outliers), mediante los intervalos de confianza del 95 o el 99%. No es el único método para la detección de éstos, y también puede utilizarse la regla del 1,5IQR (media y 1,5 veces la diferencia entre los cuartiles 1 y 3).

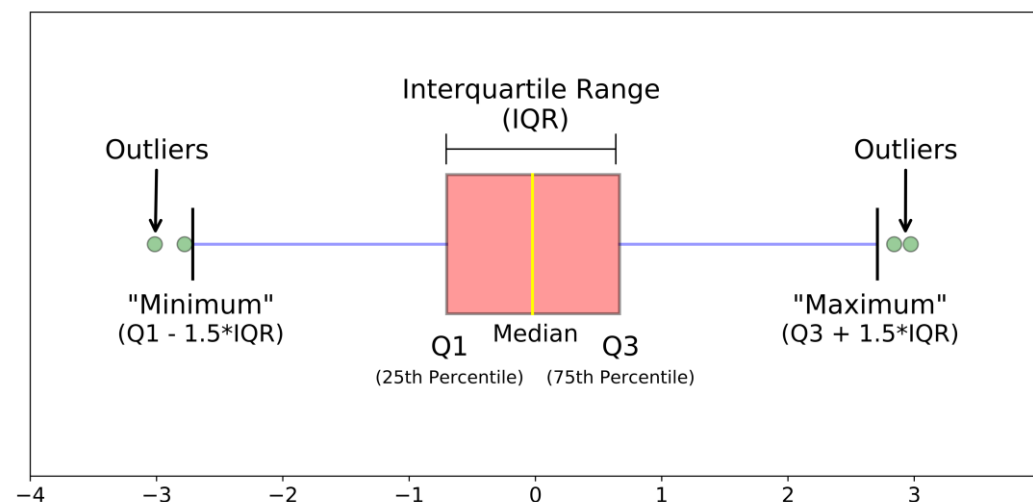
Mediante la desviación tipo

- Upper bound: $\mu + 3\sigma$
- Lower bound: $\mu - 3\sigma$



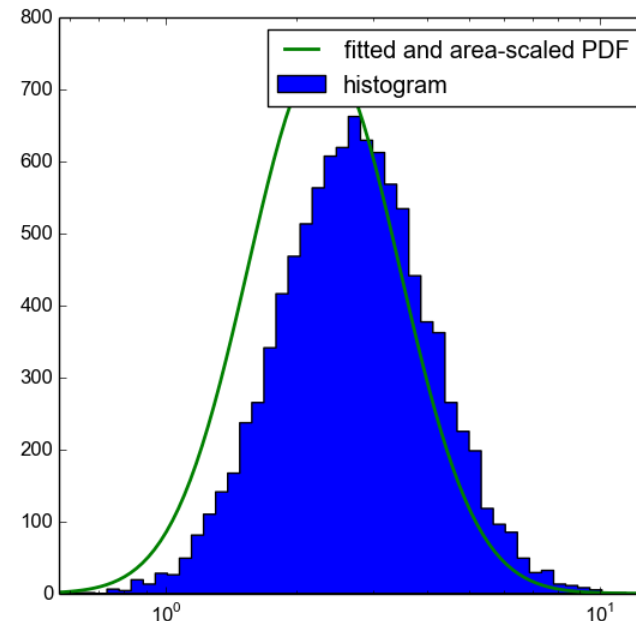
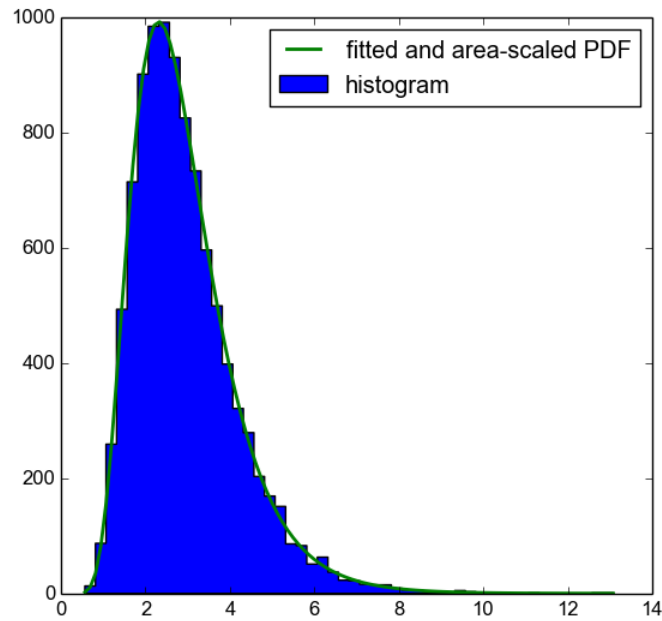
Mediante el rango intercuartílico

- Upper bound: $Q3 + 1,5IQR$
- Lower bound: $Q1 - 1,5IQR$



5 LA DISTRIBUCIÓN NORMAL: DISTRIBUCIONES NO NORMALES

Un error común es inferir propiedades de la distribución normal cuando la distribución de la variables es claramente no normal. Para solventarlo, existen diversos métodos que nos permiten transformar la distribución original, como la utilización de escalas alternativas (transformación logarítmica, por ejemplo). Esto es útil, por ejemplo, cuando tenemos variables con una gran acumulación de datos alrededor del 0, y menos casos en valores elevados.

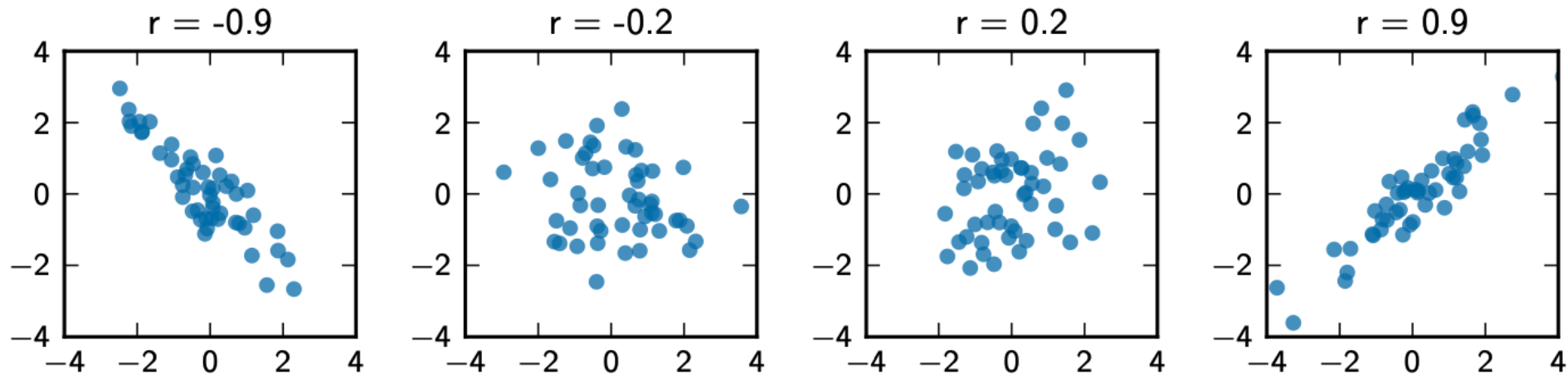


06. CORRELACIÓN

6 CORRELACIÓN LINEAL ENTRE VARIABLES: BASICS

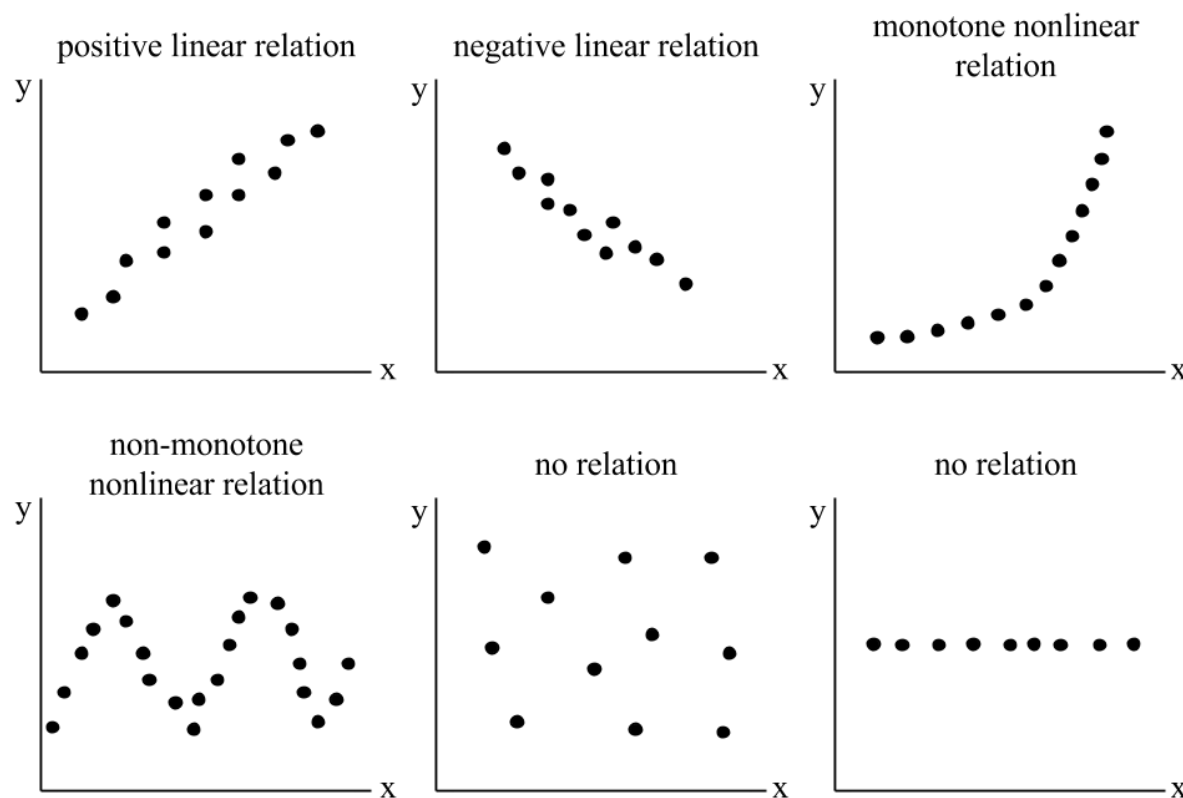
La correlación lineal o de Pearson entre dos variables se define como una **medida de la fuerza y sentido de la relación lineal entre dos variables**. Se considera que dos variables cuantitativas están correlacionadas cuando los valores de una de ellas varían sistemáticamente con respecto a los valores homónimos de la otra: si tenemos dos variables (A y B) existe correlación entre ellas si al disminuir los valores de A lo hacen también los de B y viceversa. Geométricamente, puede interpretarse como el coseno del ángulo que forman las variables una vez centradas. El coeficiente de correlación de Pearson toma valores de -1 a 1, reservándose el valor 0 para la ausencia de correlación lineal, tal y como se muestra en los ejemplos inferiores.

$$r = \frac{\sigma_{xy}}{\sigma_x \sigma_y}$$



6 CORRELACIÓN LINEAL ENTRE VARIABLES: RELACIONES NO LINEALES

Es importante notar que el coeficiente de correlación lineal sólo mide la relación lineal entre variables, y no otros tipos de relación. En estos casos, es posible trabajar transformaciones sobre éstas que nos permitan ajustar la escala de una o ambas variables para linearizar su relación, o utilizar alternativas al coeficiente de correlación de Pearson como el coeficiente de correlación de Spearman.



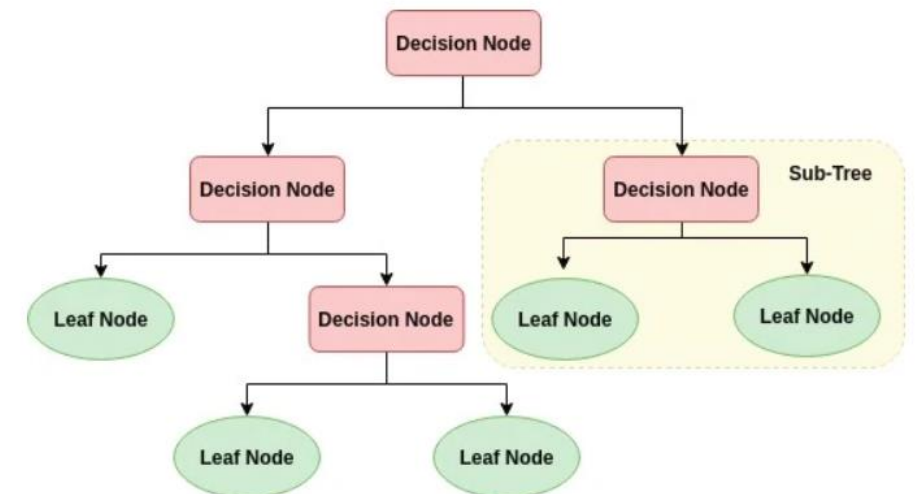
07. DECISION TREES

7 DECISION TREES: BASICS

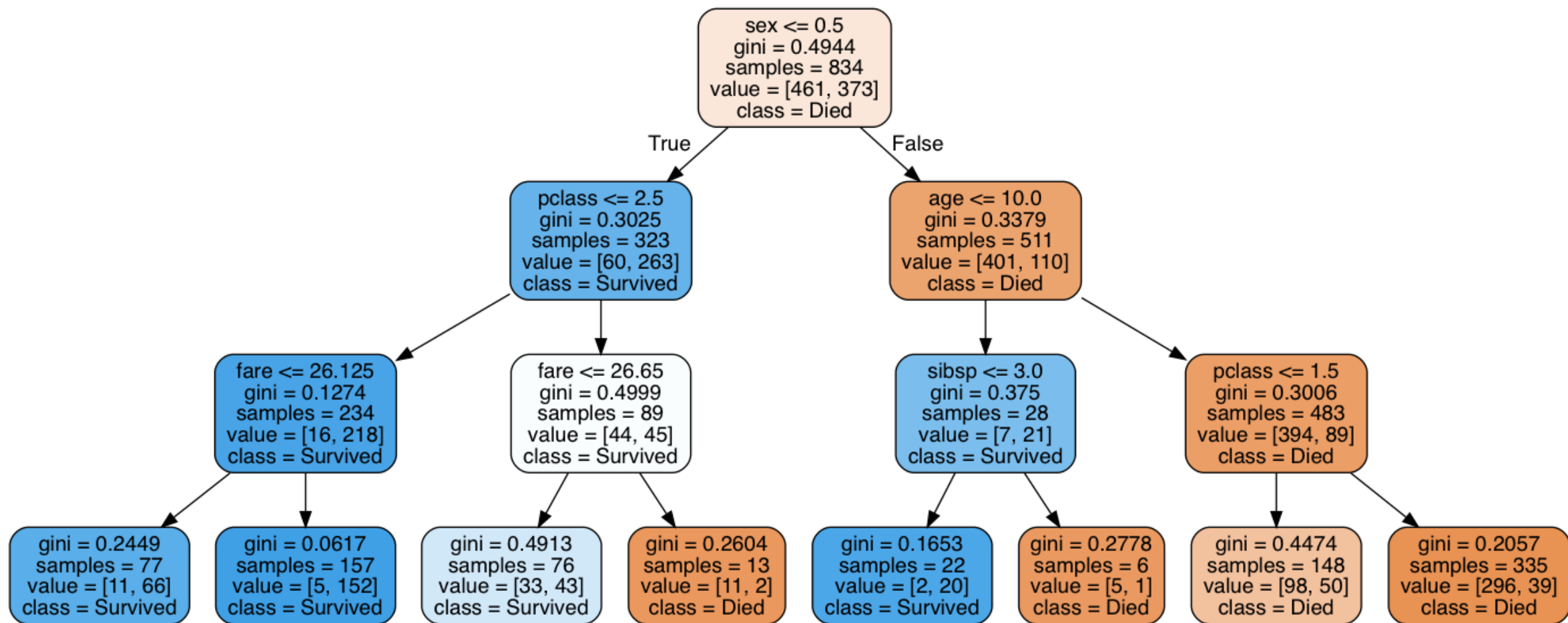
Los árboles de decisión son **algoritmos que construyen el modelo a partir de la observación de los datos utilizando una estrategia de selección iterativa de atributos** que nos proporcionen nueva información sobre el target, creando en **cada iteración una partición del conjunto de datos que haga que los subconjuntos generados sean más puros que el original en cuanto al valor del target**. Los árboles de decisión pueden utilizarse para **clasificación (Classification Trees)** o para **Regresión (Regression Trees)**, y pueden gestionar atributos categóricos o numéricos. En la práctica, los árboles de decisión se asimilan a la representación en forma de reglas del criterio experto.

En los árboles de decisión, las reglas generadas por el algoritmo dan lugar a una representación en forma de árbol, de manera que **cada nodo interior (nodo de decisión) evalúa una regla concreta para un atributo informativo**, generando dos o más particiones con una ganancia de información (cada partición queda representada por una rama).

Los nodos finales o terminales de los árboles de decisión contienen el etiquetado de la clase, que se obtiene a través del promedio del valor del target en dicho nodo terminal (se asocia a las instancias que cumplan las reglas que llevan a ese nodo terminal el valor promedio del target – regresión – o el valor de la clase mayoritaria – clasificación).



7 DECISION TREES: EJEMPLO



7 DECISION TREES: EJEMPLO

Para construir árboles de decisión, por tanto, partiremos de la premisa de que podemos encontrar un atributo en la población tal que si separamos la población según un valor concreto de dicho atributo los grupos generados tendrán un valor más cercano en relación al target (los miembros de un grupo serán más parecidos entre sí y más distintos a los miembros del resto de grupos). Buscamos, por tanto, variables que nos ayuden a reducir la incertidumbre en relación al target.

¿Cómo podemos encontrar estas particiones?

Como se ha indicado, el objetivo al construir un árbol de decisión es particionar la población generando grupos lo más puros u homogéneos posible en relación al valor del target, y diremos por tanto:

- Cuando todos los miembros del grupo tienen el mismo valor de target, el grupo es puro
- Cuando existan miembros del grupo con un valor distinto de target, el grupo no es puro, y es más impuro cuanto más miembros haya con un valor distinto o cuanto más distintos sean los valores

La generación de árboles de decisión, por tanto, se basará en la construcción de métricas que permitan medir el desorden (como de mezclados o impuros son los segmentos en relación al target), y en el particionado iterativo del dataset guiado por estas métricas. Una vez dispongamos de estas métricas, podemos recurrir a la capacidad computacional para evaluar los distintos atributos y puntos de corte a fin de seleccionar el más explicativo.

7 **DECISION TREES:** SELECCIÓN DE ATRIBUTOS

Según lo visto, por tanto, la construcción de árboles de decisión se basa en una estrategia de divide-y-vencerás (divide-and-conquer) con una aproximación top-down. En concreto, el algoritmo se materializa de la siguiente forma:

1. Selecciona el mejor atributo y un punto de corte en cuanto a reducción de la impureza (el que genera grupos más homogéneos)
 - ✓ Para decidir cuál es el mejor atributo, nos basamos en la medición de una métrica de desorden
2. Parte el dataset en función de dicho atributo en dicho punto de corte
3. Procede iterativamente hasta que se obtiene una configuración de árbol óptima en cuanto a medidas del rendimiento
4. Asigna a cada nuevo ejemplo no etiquetado el valor del nodo terminal al que corresponda (clasificación) o un valor promedio (regresión).



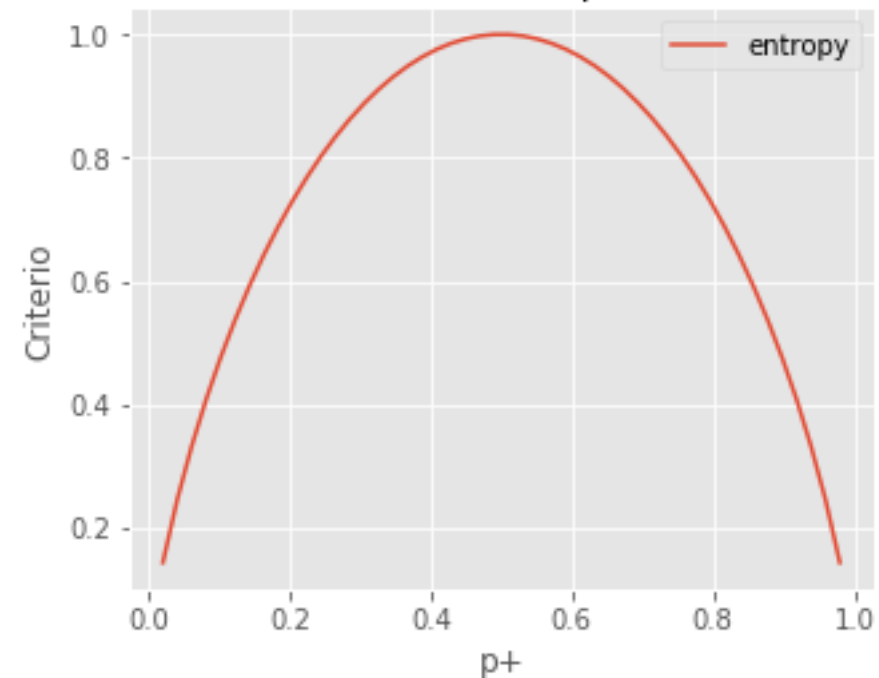
7 DECISION TREES: SELECCIÓN DE ATRIBUTOS

Como se ha indicado, la construcción de árboles de decisión se basa en **encontrar una métrica capaz de medir el desorden en un conjunto de datos en relación al valor del atributo, a la que denominaremos entropía**. Se espera, por tanto, que el valor de la métrica sea 0 cuando todos los individuos tengan el mismo valor de target (sea éste 0 o 1) y mayor de 0 cuanto mayor sea el desorden en el sistema (más individuos con un valor de target distinto), hasta un punto máximo en el cuál el grupo se distribuya en dos mitades idénticas.

Con esta base, Shannon expresó una expresión para la entropía basado en el valor medio de la clase en la muestra (p_i) y utilizando como multiplicador el $\log_2(p_i)$ para conseguir que la expresión diese el resultado deseado (0 cuando p_i sea 0 o 1 y distinto de 0 en otro caso).

$$S = - \sum_{i=1}^N p_i \log_2(p_i) = -p_+ \log_2(p_+) - p_- \log_2(p_-)$$

Nivel de desorden en función de p_+ (clasificación binaria)













7 DECISION TREES: EJEMPLO DE ENTROPÍA DE SHANNON (1)

Análisis de base o *a priori*:

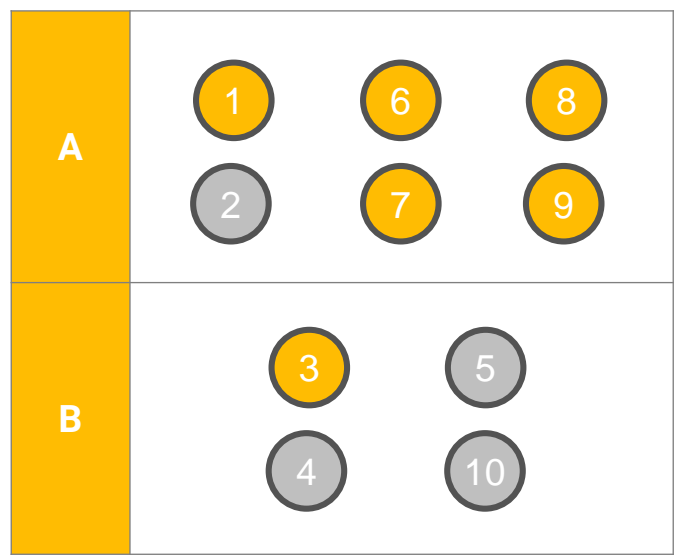
$$p_1(\text{AMARILLO}) = \frac{6}{10} = 0.6 \quad p_2(\text{GRIS}) = \frac{4}{10} = 0.4$$

$$S_0 = -0.6 \cdot \log_2(0.6) - 0.4 \cdot \log_2(0.4) = \mathbf{0.97}$$

Índice	Atributo 1	Atributo 2	Clase
1	A	6	
2	A	1	
3	B	7	
4	B	7	
5	B	8	
6	A	5	
7	A	2	
8	A	4	
9	A	3	
10	B	8	

7 DECISION TREES: EJEMPLO DE ENTROPÍA DE SHANNON (2)

Análisis cortando en el **Atributo 1**:



$$p_1(\text{AMARILLO}) = \frac{5}{6} = 0.83$$

$$p_2(\text{GRIS}) = \frac{1}{6} = 0.17$$

$$p_1(\text{AMARILLO}) = \frac{1}{4} = 0.25$$

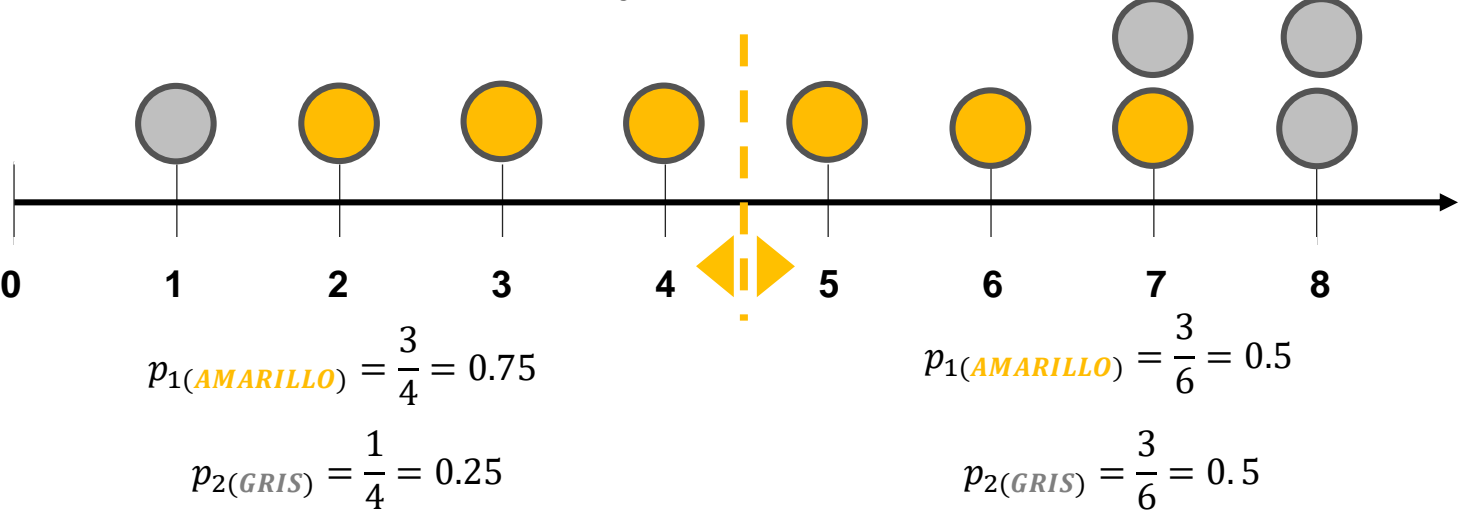
$$p_2(\text{GRIS}) = \frac{3}{4} = 0.75$$

$$S_{1-A} = -0.83 \cdot \log_2(0.83) - 0.17 \cdot \log_2(0.17) = \mathbf{0.66}$$
$$S_{1-B} = -0.25 \cdot \log_2(0.25) - 0.75 \cdot \log_2(0.75) = \mathbf{0.81}$$

Índice	Atributo 1	Atributo 2	Clase
1	A	6	
2	A	1	
3	B	7	
4	B	7	
5	B	8	
6	A	5	
7	A	2	
8	A	4	
9	A	3	
10	B	8	

7 DECISION TREES: EJEMPLO DE ENTROPÍA DE SHANNON (3)

Análisis cortando entre 4 y 5 en el **Atributo 2**:

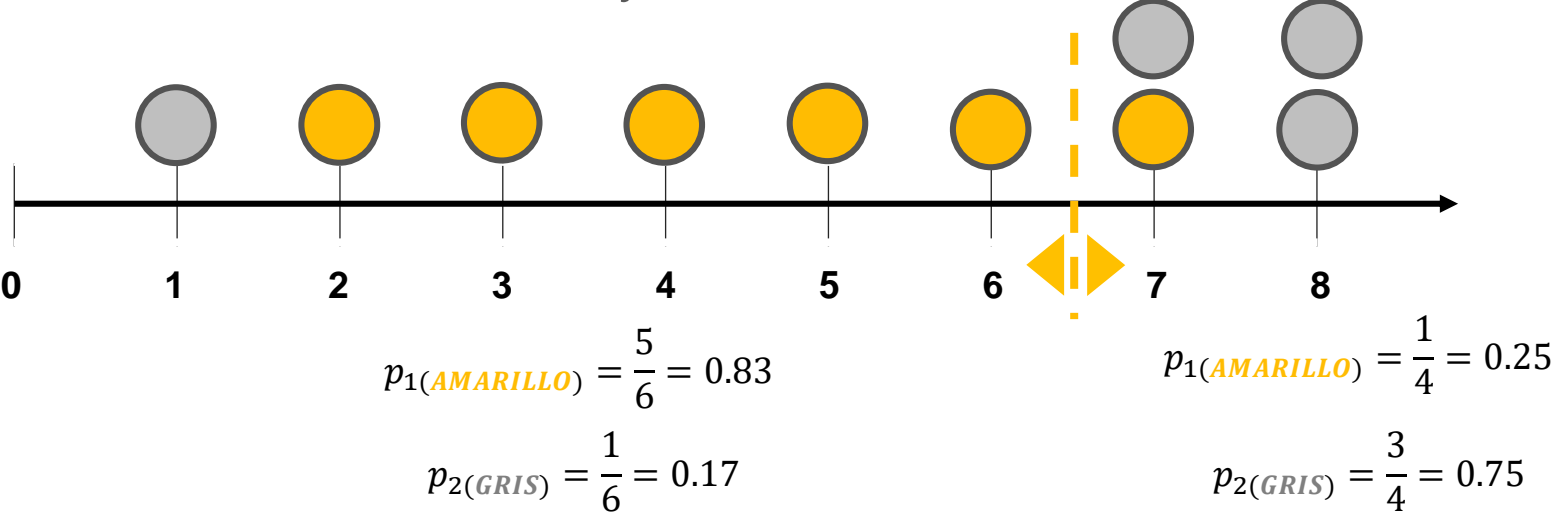


$$S_{1-A} = -0.75 \cdot \log_2(0.75) - 0.25 \cdot \log_2(0.25) = \mathbf{0.81}$$
$$S_{1-B} = -0.5 \cdot \log_2(0.5) - 0.5 \cdot \log_2(0.5) = \mathbf{1}$$

Índice	Atributo 1	Clase
1	6	
2	1	
3	7	
4	7	
5	8	
6	5	
7	2	
8	4	
9	3	
10	8	

7 DECISION TREES: EJEMPLO DE ENTROPÍA DE SHANNON (4)

Análisis cortando entre 6 y 7 en el **Atributo 2**:



$$S_{1-A} = -0.83 \cdot \log_2(0.83) - 0.17 \cdot \log_2(0.17) = 0.66$$
$$S_{1-B} = -0.25 \cdot \log_2(0.25) - 0.75 \cdot \log_2(0.75) = 0.81$$

Índice	Atributo 1	Clase
1	6	
2	1	
3	7	
4	7	
5	8	
6	5	
7	2	
8	4	
9	3	
10	8	

7 DECISION TREES: INFORMATION GAIN

Dado que la entropía es una medida de la cantidad de desorden o incertidumbre en el sistema, a la **reducción de la entropía se la conoce como Ganancia de Información** (Information Gain). Formalmente, definimos la Ganancia de Información IG para un corte basado en el valor de atributo Q como:

$$IG(Q) = S_0 - \sum_{i=1}^q \frac{N_i}{N} S_i$$

Dónde q es el número de grupos después del corte y N es el número de ejemplos en ese corte. Por ejemplo, para el caso del corte en el Atributo 1:

A	<div>1</div> <div>6</div> <div>8</div> <div>2</div> <div>7</div> <div>9</div>
B	<div>3</div> <div>5</div> <div>4</div> <div>10</div>

$$p_{1(\text{AMARILLO})} = \frac{5}{6} = 0.83$$

$$p_{2(\text{GRIS})} = \frac{1}{6} = 0.17$$

$$p_{1(\text{AMARILLO})} = \frac{1}{4} = 0.25$$

$$p_{2(\text{GRIS})} = \frac{3}{4} = 0.75$$

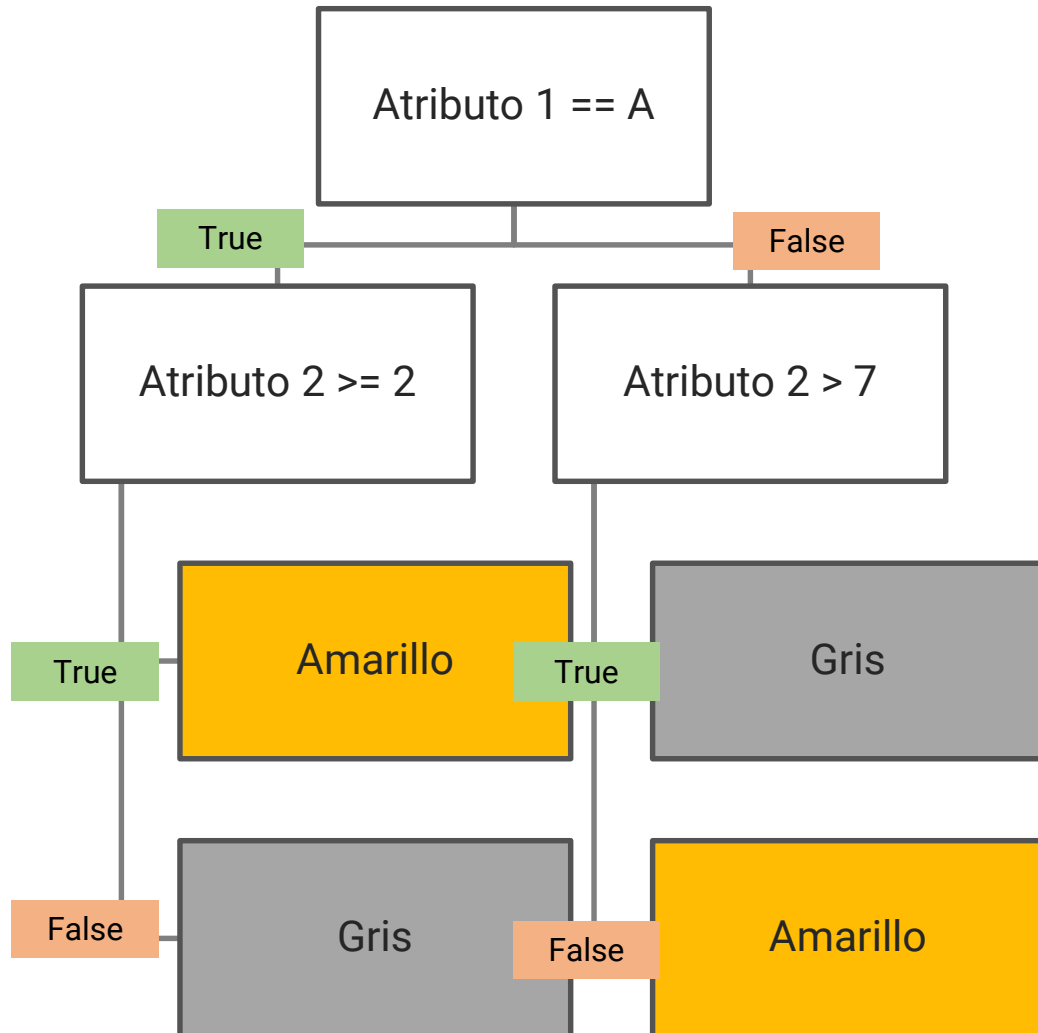
$$S_0 = 0.97$$

$$S_{1-A} = 0.66$$

$$S_{1-B} = 0.81$$

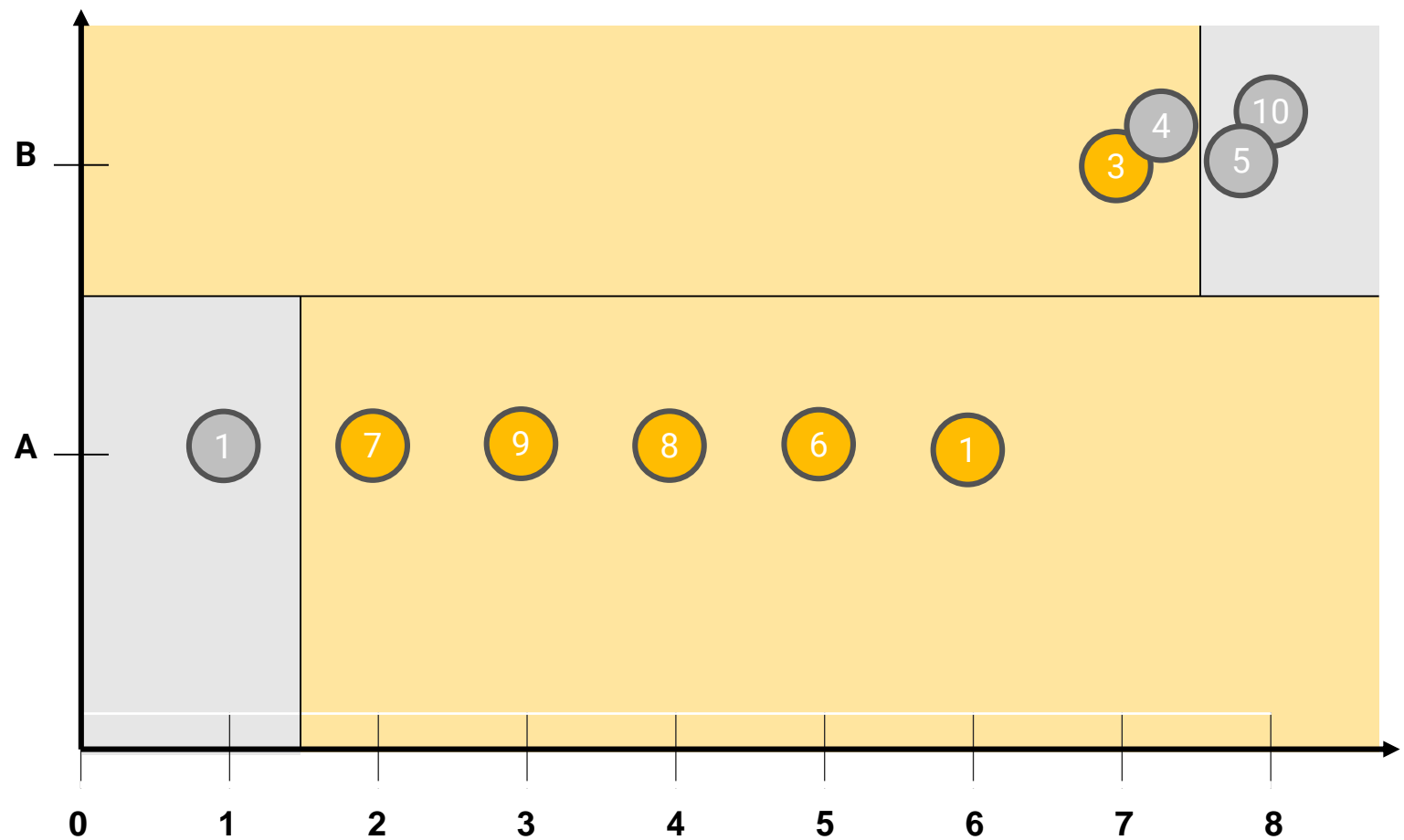
$$IG(A_1) = 0.97 - \frac{6}{10} 0.66 - \frac{4}{10} 0.81 = 0.25$$

7 DECISION TREES: VISUALIZACIÓN DE RESULTADOS



Índice	Atributo 1	Atributo 2	Clase
1	A	6	
2	A	1	
3	B	7	
4	B	7	
5	B	8	
6	A	5	
7	A	2	
8	A	4	
9	A	3	
10	B	8	

7 DECISION TREES: VISUALIZACIÓN DE RESULTADOS



Índice	Atributo 1	Atributo 2	Clase
1	A	6	
2	A	1	
3	B	7	
4	B	7	
5	B	8	
6	A	5	
7	A	2	
8	A	4	
9	A	3	
10	B	8	

7 DECISION TREES: OTRAS MEDIDAS DE DESORDEN

Además de la Entropía de Shannon, **existen otras medidas de desorden que se pueden utilizar para construir árboles de decisión**. Una de las más populares es el **criterio de impureza de Gini** G , que se define como el número de pares de la misma clase que hay en un corte:

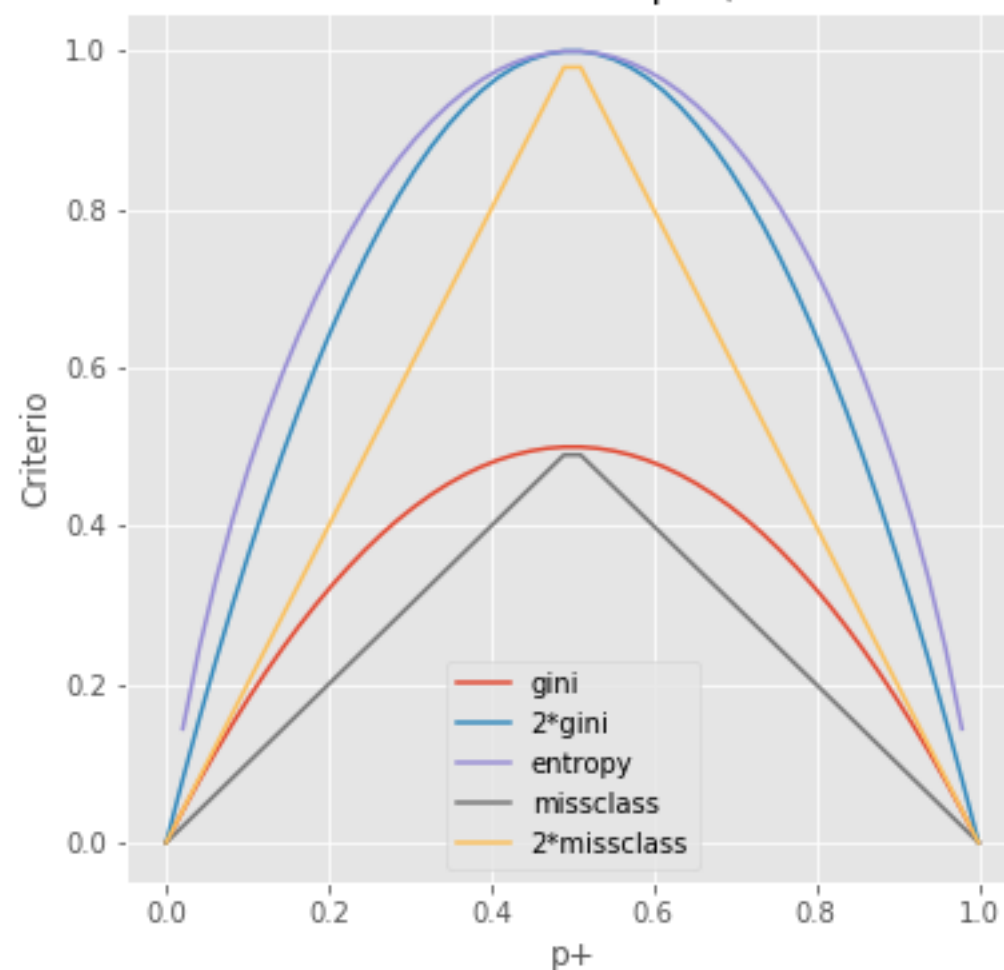
$$G = 1 - \sum_{i=1}^N p_i^2 = 1 - p_+^2 - p_-^2$$

Otra medida del nivel de desorden es el Missclasification Error E , que se define como el complementario de la clase más probable:

$$E = 1 - \max_k p_k$$

En la práctica, **la impureza de Gini y la entropía de Shannon dan resultados muy similares**, y pueden utilizarse indistintamente, mientras que el error de clasificación casi nunca se utiliza.

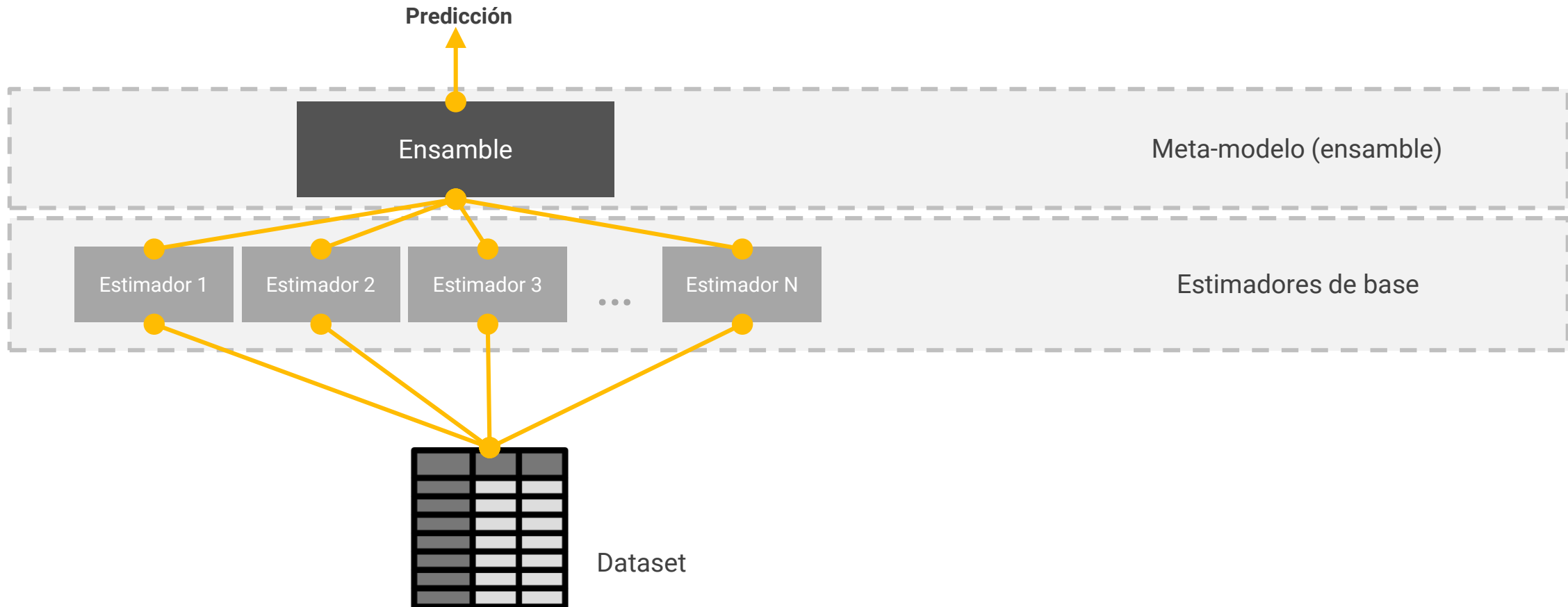
Nivel de desorden en función de p_+ (clasificación binaria)



08. ENSEMBLE METHODS

8 ENSEMBLE METHODS: BASICS

Los métodos de ensemble consisten en **combinar distintos modelos de base (estimadores base o *base learners*) en un nuevo modelo (meta-modelo o ensemble) que considera el resultado de todos éstos para dar una predicción**, con la esperanza de que la predicción combinada mejore la predicción de cada uno de los modelos individuales.



8 ENSEMBLE METHODS: TÉCNICAS

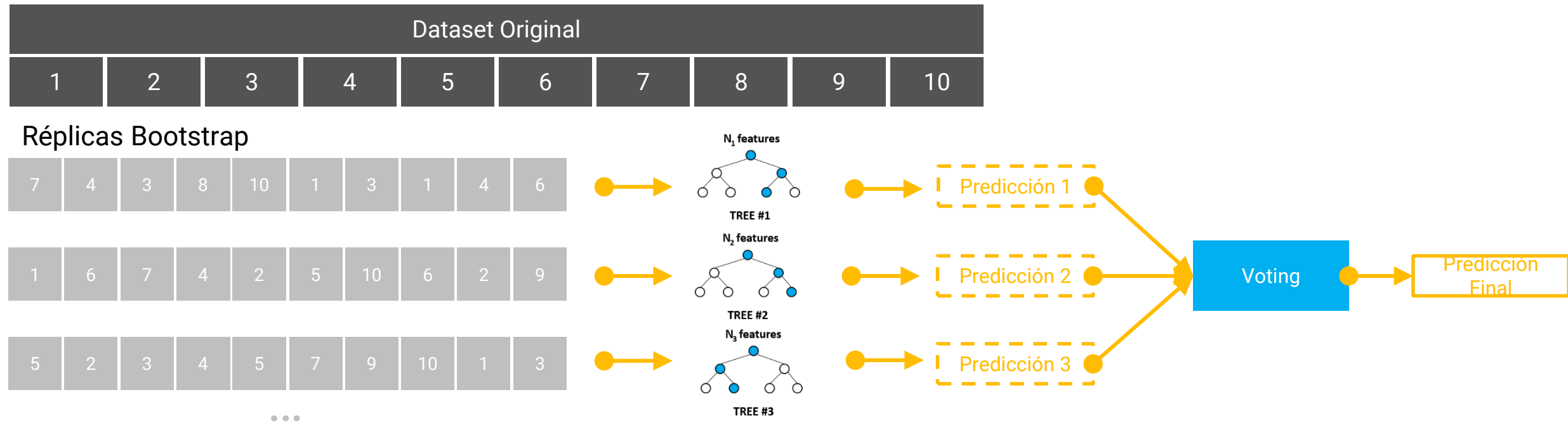
Existen distintos métodos de ensamble en función del racional utilizado para construir el meta-modelo:

A	VOTING	En los ensambles por votación, se generan distintos clasificadores (o regresores) y la predicción se genera por Hard Voting (voto mayoritario, en clasificación con etiqueta simple) o Average Voting (promedio, en clasificación probabilística o regresión). Adicionalmente, puede calcular el Weighted Average, asignando pesos a las predicciones de los distintos estimadores. Para que sea efectivo, deben cumplirse los principios de Variedad (modelos suficientemente distintos) y Precisión (modelos suficientemente precisos) .
B	STACKING	El Stacking o Stacked Generalization se basa en la idea de entrenar un modelo basado en los resultados de N modelos de base o estimadores. En concreto, los resultados de los estimadores de base se utilizan como predictores para el nuevo modelo , al que llamamos meta-modelo o ensamble. De este modo, el meta-modelo genera una nueva predicción (el resultado final del ensamble) basada en estos resultados previos.
C	BAGGING	El Bagging (Bootstrap Aggregation) es un algoritmo de Voting que se basa en la idea de generar N estimadores utilizando el mismo algoritmo para todas las predicciones (por ejemplo un <i>Decision Tree</i> en el algoritmo Random Forest) pero usando diferentes subconjuntos del dataset original (con muestreo de observaciones o predictores) para entrenar un modelo final con mayor capacidad de generalización.
D	BOOSTING	El Boosting se basa en la idea de generar distintos clasificadores débiles (<i>weak learners</i>) en un nuevo clasificador (<i>strong classifier</i>) en un proceso secuencial en el que cada nuevo modelo intenta fijar el error de los modelos precedentes . El algoritmo se basa en un modelo de optimización que procura mejorar el rendimiento global en cada nueva iteración. Los más conocidos son los <i>Gradient Boosting</i> .

8 ENSEMBLE METHODS: BAGGING Y RANDOM FOREST

El Random Forest es una combinación de árboles de decisión tal que cada árbol se entrena en una réplica bootstrap del dataset original y, opcionalmente, con un subconjunto aleatorio de atributos (*random subspace*), generando diferentes estimadores que después se combinan mediante políticas de votación (promediando los resultados o asignando la clase mayoritaria). La eficacia del algoritmo se basa en construir árboles no correlacionados (suficientemente distintos). Este algoritmo ha sido ampliamente utilizado por su elevada precisión y simplicidad, y tanto para hacer modelos predictivos como modelos descriptivos (ver *feature importances*).

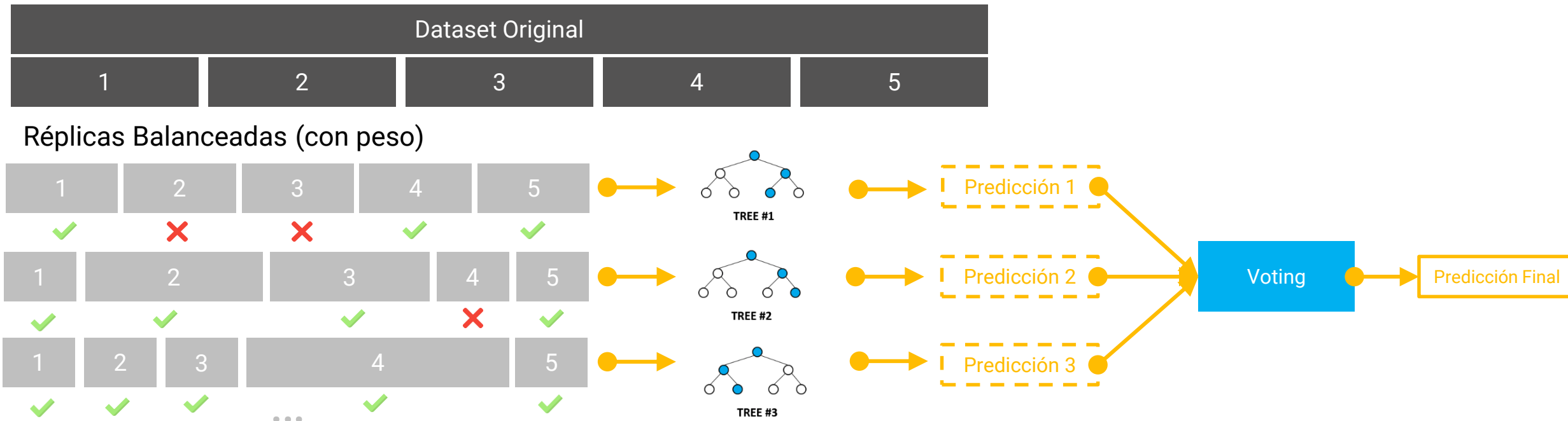
¿Cómo funciona el Random Forest?



8 ENSEMBLE METHODS: BOOSTING Y GRADIENT BOOSTING (I)

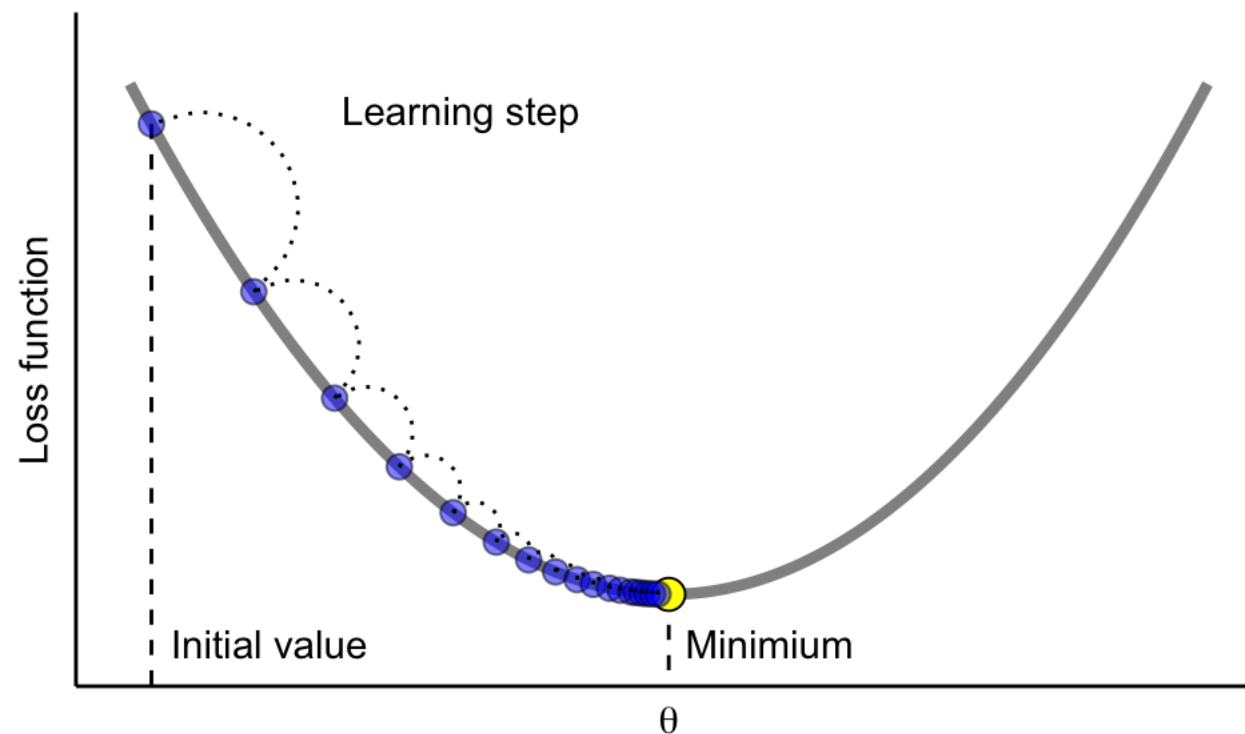
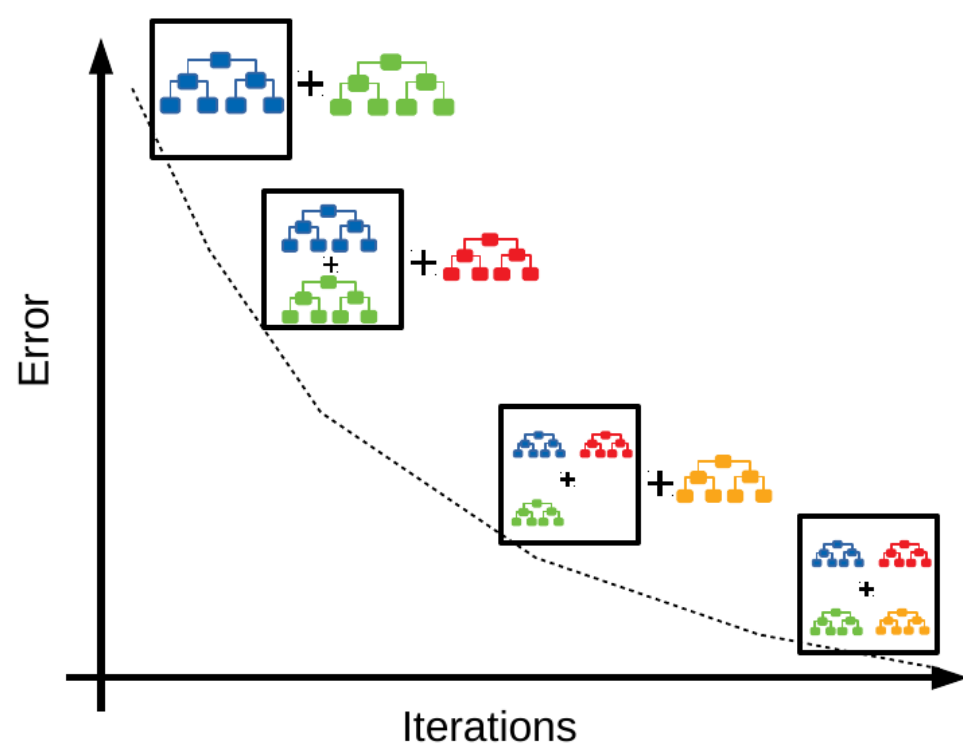
El Gradient Boosting es algoritmo de construcción progresiva de modelos basada en el **ensamble de modelos débiles para obtener un modelo robusto**, de manera que **cada modelo se genera sobre el anterior con la intención de corregir el error producido por éste**. En concreto, se utiliza un vector de pesos asociados a las distintas observaciones del dataset, inicializado con pesos uniformes. A medida que se avanza, se incrementa el peso de las observaciones mal clasificadas, de manera que éstas tienen mayor relevancia en la construcción de los nuevos modelos. Finalmente, se combinan todos los modelos con políticas de Voting. Habitualmente se utilizan como estimadores base para el GBM árboles de decisión y como algoritmo de optimización (mejora de la precisión) el método de descenso de gradiente (Gradient Descent). Las implementaciones más populares son el xGBoost (eXtreme Gradient Boostin) y LightGBM.

¿Cómo funciona el Gradient Boosting Machines?



8 ENSEMBLE METHODS: BOOSTING Y GRADIENT BOOSTING (II)

El punto más relevante en la construcción del Gradient Boosting es **cómo ajustar los pesos de una manera óptima**. Para ello, fijamos una **función a optimizar denominada función de coste (Loss Function)** como el logloss, y utilizamos un algoritmo de optimización denominado **método de descenso del gradiente (Gradient Descent)**, que consiste en obtener las derivadas parciales de la función de coste para decidir la dirección de avance.



09. MODELOS LINEALES

9 MODELOS LINEALES: BASICS

Los modelos lineales, al igual que el coeficiente de correlación, buscan **cuantificar la correlación lineal entre dos variables**, pero en lugar de tratarlas de forma simétrica hacemos una distinción entre ambas. En consecuencia, los objetivos que pueden resolverse con un análisis de correlación y un análisis de regresión son distintos.

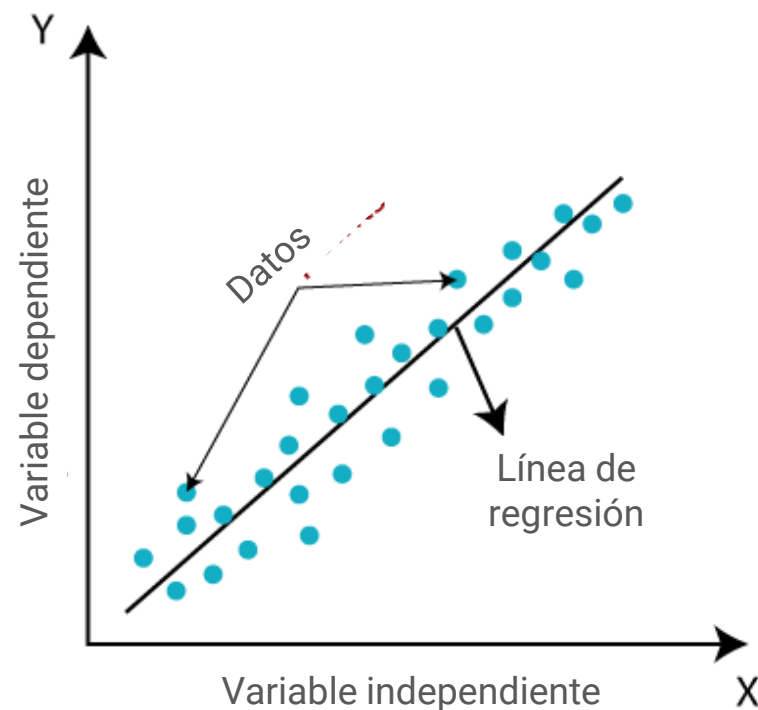
Variables Independientes

- Predictores, variables explicativas.

Variable Dependiente

- Target, variable objetivo.

- **Correlación:**
 - ¿Existe algún tipo de relación lineal entre las variables X e Y?
 - ¿Qué sentido y nivel de fuerza tiene la relación entre las variables X e Y?
- **Regresión:**
 - ¿Cuánto cambia Y si cambia X (o el vector de X)?
 - ¿Dado un valor de X, cuánto vale Y?



MODELOS LINEALES: ECUACIÓN DE LA RECTA (I)

$$Y = a + bX$$

Donde:

- Y : Valor estimado de la variable Y para un valor de X .
- a : Intercepto. Valor de Y cuando $X = 0$.
- b : Pendiente de la recta.
- X : Variable independiente

9 MODELOS LINEALES: ECUACIÓN DE LA RECTA (II)

Pendiente

$$b = r \frac{\sigma_Y}{\sigma_X}$$

Donde:

- r : coeficiente de correlación
- σ_Y : desviación estándar en Y
- σ_X : desviación estándar en X

Intercepto

$$a = Y - bX$$

Donde:

- Y : Media de Y
- X : Media de X

- **El término de pendiente:**

Al igual que en la correlación, el signo del término de pendiente nos sirve para medir el sentido de relación entre el predictor y el target, de manera que si es positivo el target aumenta cuando el valor de la variable aumenta y si es negativa el target se reduce cuando el valor de la variable aumenta. Además, el valor del término indica el grado de la relación (cuánto aumenta el target con un aumento unitario de la variable).

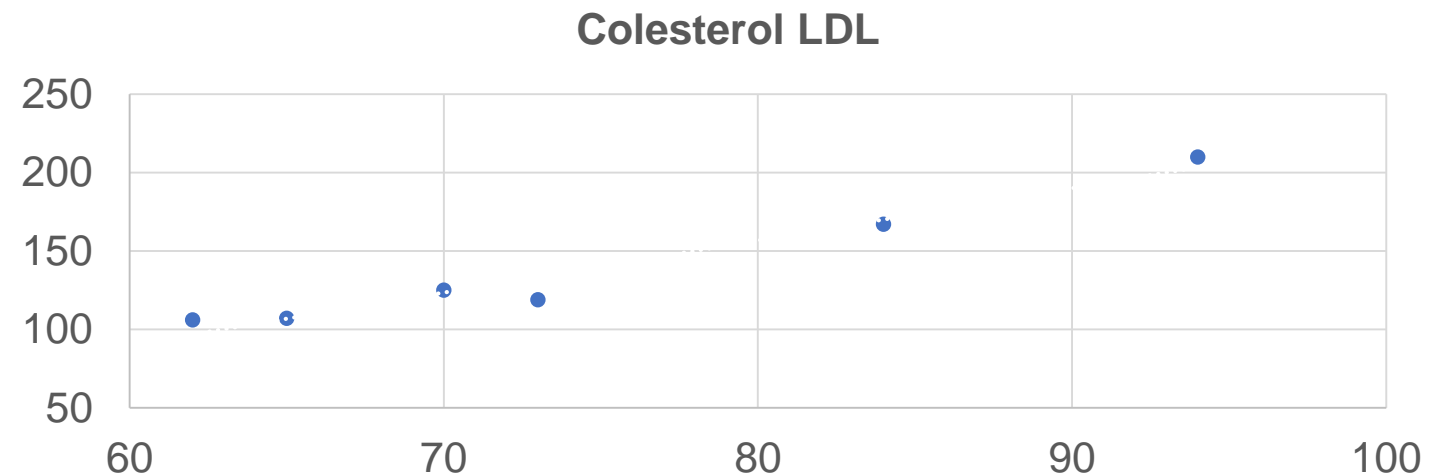
9 MODELOS LINEALES: EJEMPLO (I)

Imaginemos que tenemos que generar una aplicación capaz de estimar el nivel de colesterol LDL en sangre de un paciente a partir del peso de éste. Una opción posible es **recoger conocimiento experto** (médico) y **programar las reglas que éste establezca en un ordenador**, de manera que dado un input (peso) el programa ejecute un algoritmo (conjunto de reglas) y obtenga una estimación del colesterol LDL.

Este procedimiento, sin embargo, **tiene varias carencias**: (1) debemos asumir que existe un conocimiento experto completo y ser capaces de recogerlo y programarlo y (2) este conocimiento será estático o requerirá de actualización manual.

Una alternativa es utilizar un **conjunto de datos etiquetados previos y utilizarlos para obtener un patrón**, a partir del cuál podamos inferir, a partir del valor del peso, el nivel de colesterol LDL para ejemplos futuros no etiquetados (en los que no dispongamos del valor del colesterol).

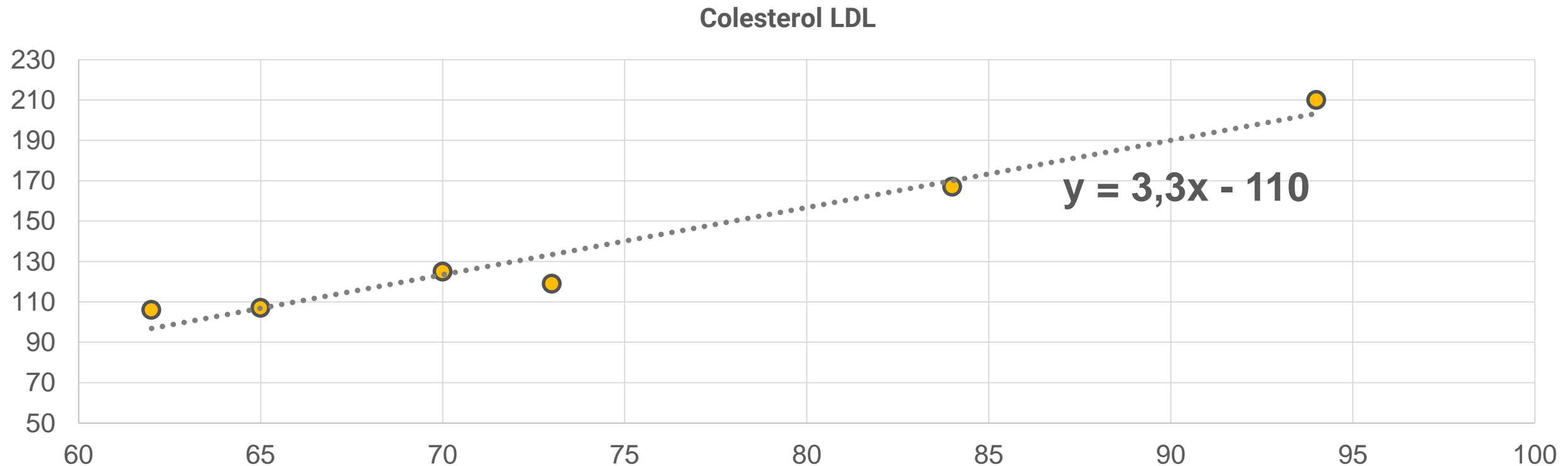
Peso	Colesterol LDL
84	167
73	119
65	107
70	125
62	106
94	210



9

MODELOS LINEALES: EJEMPLO (II)

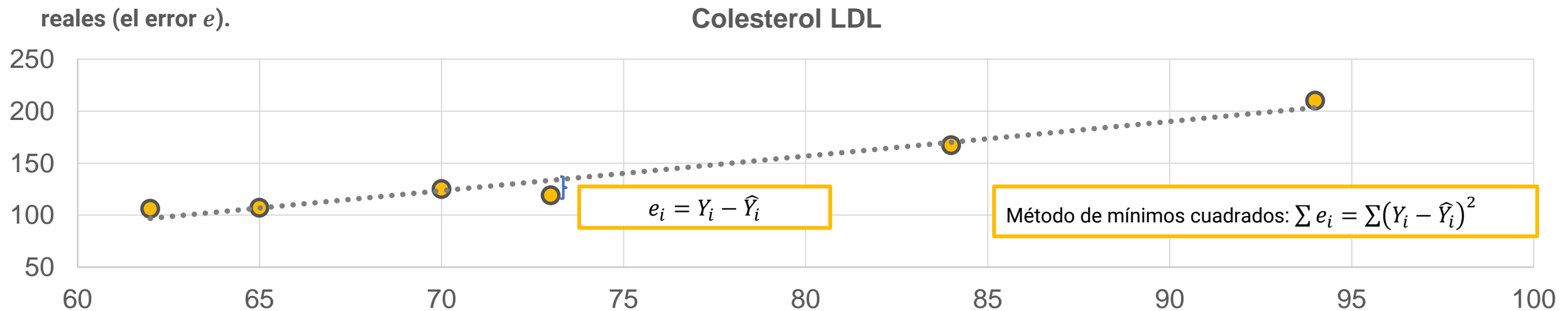
La recta de mejor ajuste para este problema es $y=3,3x-110$, dónde 3,3 es el pendiente de la recta (por cada aumento de 1 en X, Y aumenta 3) y 110 es el intercepto (valor de partida).



9 MODELOS LINEALES: AJUSTE DE LA RECTA

¿Cómo podemos encontrar la línea que mejor ajusta la distribución del Scatter Plot? Es decir, **la línea que mejor describe la relación entre las variables X e Y .**

El método de ajuste parte del racional que la mejor recta es aquella que **minimiza la distancia entre los valores de la predicción y los valores reales (el error e).**



Para evitar el problema de las desviaciones positivas y negativas, miramos las desviaciones cuadráticas. El método de mínimos cuadrados se basa en la minimización del error cuadrático, y por tanto en buscar la recta cuyos valores minimizan la suma de los cuadrados del error vertical entre la predicción y el valor real. Este método de estimación se conoce como *Ordinary Least Squares* (OLS).

9 MODELOS LOGÍSTICOS: INTRODUCCIÓN

Como alternativa a los modelos de regresión lineal, para problemas de clasificación binaria se dispone de los modelos de regresión logística. **En los modelos de regresión logística se utiliza la función logística (la sigmoide) para ajustar los valores del target.** Todos los racionales del modelo lineal son válidos para el modelo logístico, que además no necesita que se establezca una relación lineal entre las variables independientes y la dependiente.

La función logística permite mapear los predictores a un valor continuo entre 0 y 1, que se demuestra que **puede interpretarse como la probabilidad de membresía a cada una de las clases.** Dicha probabilidad puede interpretarse directamente o convertirse a un valor categórico a través de la definición de un umbral de probabilidad.

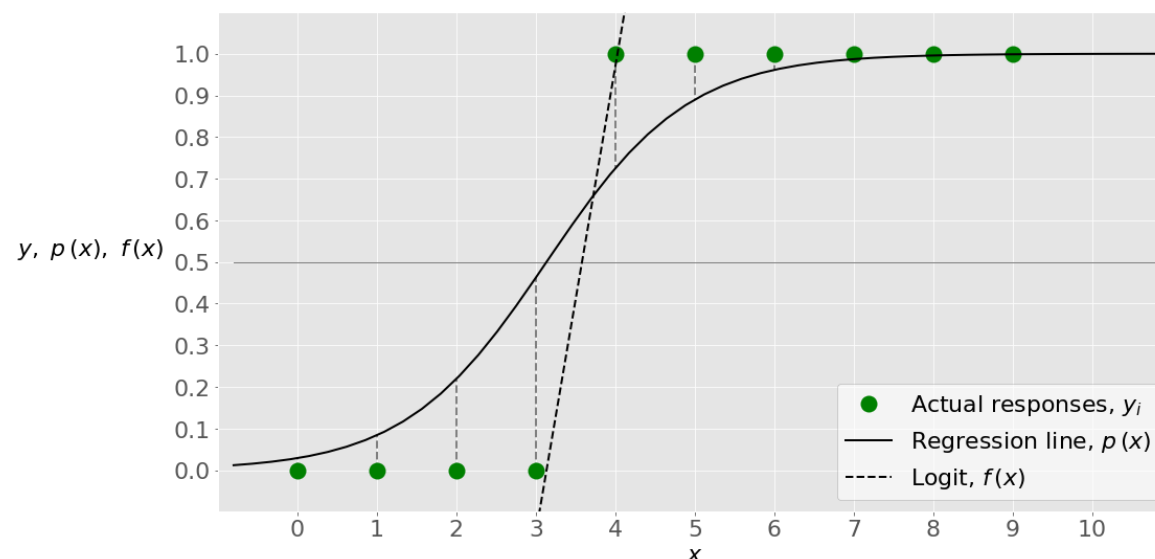
A diferencia de los árboles de decisión, la regresión logística es un modelo paramétrico (con alto bias), por lo que es muy robusto frente a overfitting (sacrificando el variance). Al ser un modelo robusto y fácilmente explicable (los coeficientes son interpretables), por lo que es muy utilizado en scoring on-line y modelos auditados.

Variables Independientes

- Predictores, variables explicativas.

Variable Dependiente

- Target, variable objetivo.



MODELOS LOGÍSTICOS: LA ECUACIÓN DE LA SIGMOIDE

Regresión lineal: $Y = a + bX$ + Ecuación de la sigmoide: $p = \frac{1}{1+e^{-y}}$

$$p = \frac{1}{1 + e^{-(a+bX)}}$$

Donde:

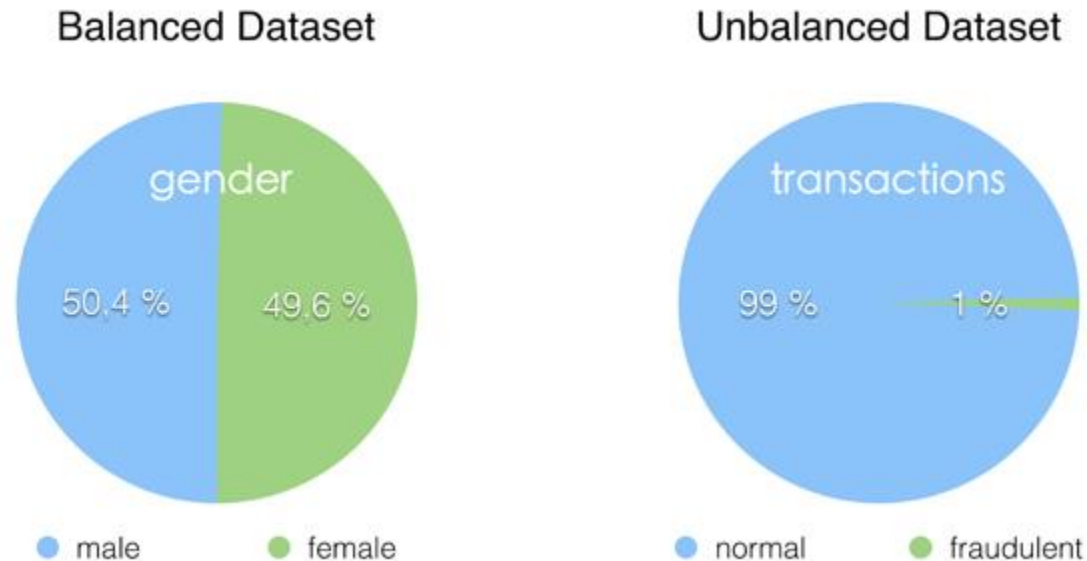
- Y : Valor estimado de la variable Y para un valor de X .
- a : Intercepto. Valor de Y cuando $X = 0$.
- b : Pendiente de la recta.
- X : Variable independiente

10. ESTRATEGIAS DE REMUESTREO

10 ESTRATEGIAS DE REMUESTREO : EL ENGAÑO DE LA MÉTRICA

Como ya se indicó, uno de los aspectos clave en la resolución de un problema de Machine Learning es la correcta selección de la métrica de validación.

Como ejemplo, en los problemas de clasificación supervisada, hablamos de la Precisión (Accuracy) como una de las métricas más intuitivas y fáciles de entender. Como ya indicamos, esta métrica tiene una elevada dependencia de la ratio de prevalencia de la clase positiva (número de instancias de la clase positiva sb. total), y no es adecuada para casos en los que esta ratio se aleje de valores cercanos al 50%. Por este motivo, **al trabajar con conjuntos de datos no balanceados, es adecuado seleccionar métricas de validación distintas al Accuracy, como el AUC o el F1.**

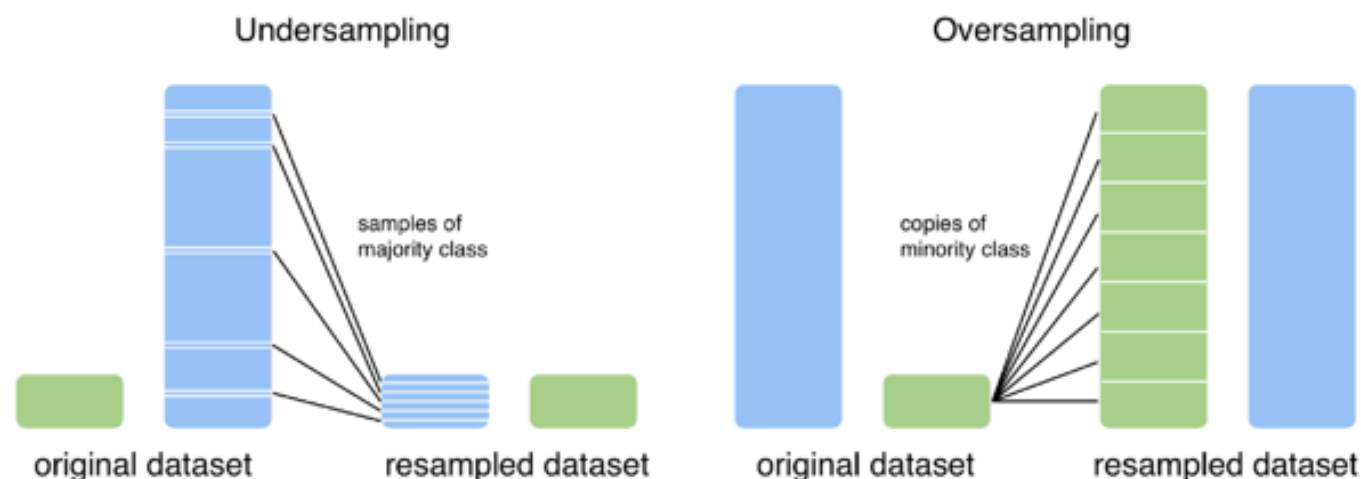


10 ESTRATEGIAS DE REMUESTREO : ESTRATEGIAS DE REMUESTREO

Ocurre, sin embargo, que un gran número de algoritmos funciona utilizando métricas internas de precisión (como la maximización de la entropía en los árboles de decisión o el RMSE en los modelos lineales), lo que hace que en el desarrollo del modelo se premie la reducción del error sobre el reconocimiento de casos positivos, llevando a soluciones sub-óptimas en el proceso que conocemos como sub-ajuste.

Por este motivo, **es importante que al trabajar con datos no balanceados utilicemos un segundo mecanismo de control consistente en rebalancear el conjunto de datos**, haciendo que la relación de presencia de ambas clases sea suficiente como para evitar los problemas de subajuste.

Los dos métodos más sencillos y eficientes de remuestreo son el Oversampling (generar elementos de la clase minoritaria por copia) y el Undersampling (eliminar elementos de la clase mayoritaria). Al trabajar con Oversampling es importante tener en cuenta que la replicación de instancias puede llevar a over-fitting y que el consumo de memoria será mayor, mientras que al trabajar con Undersampling hay que tener en cuenta que puede producirse pérdida de información o quedarnos con un Dataset demasiado pequeño que no nos permita un buen ajuste. Por este motivo, **es importante controlar la ratio de rebalanceo evitando dichos inconvenientes.**



¡THAT'S ALL FOLKS!

Guillem Sitges i Puy

Sesiones 19 a 24