Aprendizaje automático

El *aprendizaje automático* es el campo de la inteligencia artificial cuyo objetivo es el estudio de *sistemas* que *adaptan* su *funcionamiento* a la *experiencia*. Para poder llevar a cabo ese estudio, es necesario formalizar los conceptos resaltados en la definición anterior:

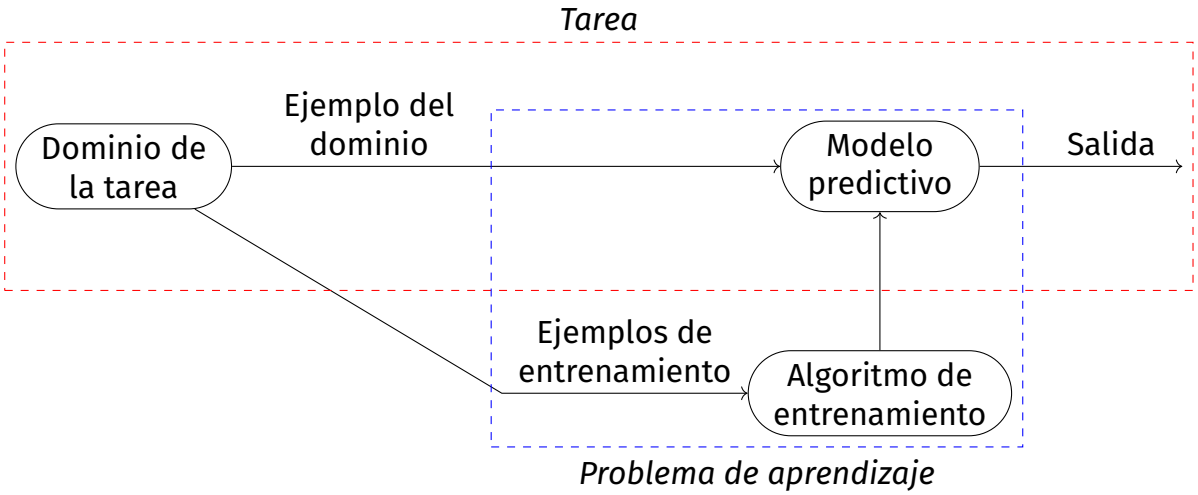
- ¿Qué es un sistema? Es un modelo matemático diseñado para realizar una determinada tarea.
- ¿Cuál es el funcionamiento de estos sistemas? Una vez construidos, estos modelos se pueden entender simplemente como funciones matemáticas que producen una salida a partir de unos datos de entrada.
- ¿Qué es la experiencia para estos sistemas? Es un conjunto de datos que proporciona ejemplos que guían la construcción del modelo.
- ¿Qué es adaptar el funcionamiento a la experiencia? Es la construcción de modelos que se ajusten lo mejor posible, según una cierta medida, a los ejemplos de partida.

Ejemplo: Filtro anti-*spam*

Un ejemplo muy conocido de modelo de aprendizaje automático son los filtros de mensajes no deseados de correo electrónico. Estos filtros reciben como dato de entrada un mensaje de correo electrónico y lo clasifican como correo legítimo o correo no deseado.

La construcción de un filtro anti-*spam* se realiza a partir de un conjunto de mensajes de ejemplo, tanto legítimos como no deseados. Cuantos más ejemplos incluya ese conjunto, más capacidad de discriminación podrá tener el filtro construido.

Un problema de aprendizaje para una determinada tarea consiste en la construcción de un modelo que la resuelva. Para ello se aplica un algoritmo de aprendizaje a partir de un conjunto de ejemplos de entrenamiento que pertenecen al dominio de la tarea. En este tema nos ceñiremos a la construcción de modelos predictivos, es decir, aquellos que abordan tareas consistentes en asociar una salida a cada ejemplo del dominio. En el siguiente gráfico se muestran de forma esquematizada los conceptos introducidos en este párrafo.



Los ejemplos del dominio de la tarea se describen por medio de atributos o características (*features*, en inglés). Estos atributos pueden ser de diversos tipos: continuo, discreto, ordinal, booleano, etc.

Ejemplo: Coches

Origen	*Cilindros*	*Potencia*	*Peso*	*Aceleración*
Norteamérica	6	105	3613	16.5
Europa	4	76	2511	18.0
Europa	5	77	3530	20.1
Japón	6	122	2807	13.5
Europa	4	90	2265	15.5
Japón	4	70	1945	16.8
Norteamérica	8	145	3880	12.5
Norteamérica	4	90	2670	16.0
Norteamérica	6	90	3211	17.0
Norteamérica	8	198	4952	11.5

Este conjunto contiene 10 ejemplos descritos por los siguientes atributos:

- *Origen*: es un atributo discreto con tres valores posibles (Norteamérica, Europa y Japón).
- *Cilindros*: a pesar de ser un atributo numérico, solo puede tomar cuatro valores distintos (4, 5, 6, 8), por lo que consideramos a este atributo como discreto. Además, al tener estos valores un orden natural (al contrario que *Origen*), se trata de un atributo ordinal.
- *Potencia*, *Peso*, *Aceleración*: son atributos continuos, ya que pueden tomar cualquier valor dentro de un intervalo de posibles valores.

La salida del modelo para cada ejemplo del dominio será un valor de un determinado atributo objetivo. Si se conoce la salida pretendida para cada ejemplo de entrenamiento, el aprendizaje realizado se dice que es supervisado. En caso contrario, se dice que es no supervisado.

Dentro del aprendizaje supervisado, si el atributo objetivo es de tipo discreto, entonces nos encontramos ante una tarea de clasificación (binaria, si solo hay dos posibles valores de salida; multiclase, si hay más de dos).

Ejemplo: Predicción del tipo de consumo

Origen	*Cilindros*	*Potencia*	*Peso*	*Aceleración*	*Tipo de consumo*
Norteamérica	6	105	3613	16.5	Bajo
Europa	4	76	2511	18.0	Bajo
Europa	5	77	3530	20.1	Alto
Japón	6	122	2807	13.5	Bajo
Europa	4	90	2265	15.5	Alto
Japón	4	70	1945	16.8	Alto
Norteamérica	8	145	3880	12.5	Bajo
Norteamérica	4	90	2670	16.0	Alto
Norteamérica	6	90	3211	17.0	Bajo
Norteamérica	8	198	4952	11.5	Bajo

Se trata de una tarea de clasificación, ya que el atributo objetivo es discreto (de hecho, es binario). El modelo a construir debe predecir, dados los datos de un coche, si ese coche tiene un consumo alto o bajo.

Si el atributo objetivo es de tipo continuo, entonces la tarea es de regresión.

Ejemplo: Predicción del nivel de consumo

Origen	*Cilindros*	*Potencia*	*Peso*	*Aceleración*	*Nivel de consumo*
Norteamérica	6	105	3613	16.5	18.0
Europa	4	76	2511	18.0	22.0
Europa	5	77	3530	20.1	25.4
Japón	6	122	2807	13.5	20.0
Europa	4	90	2265	15.5	26.0
Japón	4	70	1945	16.8	33.5
Norteamérica	8	145	3880	12.5	17.5
Norteamérica	4	90	2670	16.0	28.4
Norteamérica	6	90	3211	17.0	19.0
Norteamérica	8	198	4952	11.5	12.0

Se trata de una tarea de regresión, ya que el atributo objetivo es continuo. El modelo a construir debe predecir, dados los datos de un coche, el nivel de consumo de ese coche, siendo este un valor numérico dentro del rango de posibles valores.

Dentro del aprendizaje no supervisado, un modelo predictivo resuelve una tarea de agrupamiento (*clustering*, en inglés), es decir, de particionado del dominio de la tarea en un cierto número de grupos de ejemplos similares (será por tanto necesario establecer una medida de similitud entre ellos), de tal manera que para cada ejemplo será capaz de predecir a cuál de esos subgrupos pertenece.

Ejemplo: Predicción del origen de un coche

Cilindros	*Potencia*	*Peso*	*Aceleración*
6	105	3613	16.5
4	76	2511	18.0
5	77	3530	20.1
6	122	2807	13.5
4	90	2265	15.5
4	70	1945	16.8
8	145	3880	12.5
4	90	2670	16.0
6	90	3211	17.0
8	198	4952	11.5

Asumiendo que coches similares tienen un mismo origen, se trata de una tarea de agrupamiento en la que hay que dividir el conjunto de coches en tres subgrupos de coches similares.

Como conclusión final, podemos decir que

El aprendizaje automático consiste en usar los atributos adecuados para construir modelos apropiados que lleven a cabo la tarea correcta.

Modelos de aprendizaje automático

Modelos de aprendizaje automático los hay de muchos tipos y pueden ser descritos de muy diversas maneras, pero todos se construyen estimando (aprendiendo) los valores de ciertos parámetros, propios de cada modelo, a partir de los ejemplos de entrenamiento. Si el número de parámetros del modelo es fijo, independiente de la cantidad de ejemplos de entrenamiento, entonces se dice que es un modelo paramétrico. Si el número de parámetros del modelo depende de la cantidad de ejemplos de entrenamiento, entonces se dice que es un modelo no paramétrico. La construcción de los modelos también se puede ver influida por ciertos hiperparámetros, que son parámetros cuyos valores se prefijan de antemano, sin estimarlos a partir de los ejemplos de entrenamiento.

En este tema se estudian tres modelos de aprendizaje automático supervisado: naive (pronunciado /nɑˈiv/) Bayes, árboles de decisión y k vecinos más cercanos.

Naive Bayes

El clasificador naive Bayes (Bayes ingenuo, en español) es un modelo adecuado para abordar una tarea de clasificación a partir de atributos discretos.

Ejemplo: Discretización de atributos continuos

Una manera de transformar un atributo continuo en uno discreto es aplicarle un procedimiento de discretización. Este consiste en dividir el rango de valores del atributo en una cierta cantidad de subintervalos, que conformarán los posibles valores de la nueva variable discreta, y sustituir cada valor original del atributo por el subintervalo al que pertenece.

Para nuestra tarea de predicción del tipo de consumo de un coche, por simplicidad vamos a discretizar cada atributo continuo en únicamente dos subintervalos, determinados por la mediana de los valores del atributo. De esta forma, los atributos discretos obtenidos tendrán la misma cantidad de valores en cada subintervalo.

Origen	*Cilindros*	*Potencia*	*Peso*	*Aceleración*	*Tipo de consumo*
Norteamérica	6	$(90, 198]$	$(3009, 4952]$	$(16.25, 20.1]$	Bajo
Europa	4	$[70, 90]$	$[1945, 3009]$	$(16.25, 20.1]$	Bajo
Europa	5	$[70, 90]$	$(3009, 4952]$	$(16.25, 20.1]$	Alto
Japón	6	$(90, 198]$	$[1945, 3009]$	$[11.5, 16.25]$	Bajo
Europa	4	$(90, 198]$	$[1945, 3009]$	$[11.5, 16.25]$	Alto
Japón	4	$[70, 90]$	$[1945, 3009]$	$(16.25, 20.1]$	Alto
Norteamérica	8	$(90, 198]$	$(3009, 4952]$	$[11.5, 16.25]$	Bajo
Norteamérica	4	$[70, 90]$	$[1945, 3009]$	$[11.5, 16.25]$	Alto
Norteamérica	6	$[70, 90]$	$(3009, 4952]$	$(16.25, 20.1]$	Bajo
Norteamérica	8	$(90, 198]$	$(3009, 4952]$	$[11.5, 16.25]$	Bajo

Realización de la tarea

Dado un ejemplo del dominio de la tarea, se pretende que el modelo lo clasifique estimando cuál es la clase más probable a la que pertenece. Para ello lleva a cabo una serie de simplificaciones y asunciones con el objetivo de que los cálculos a realizar se puedan hacer más eficientemente.

Formalmente, si C es el conjunto de clases posibles y $X_1 = x_1, \dots, X_n = x_n$ son los valores de los atributos que describen al ejemplo, entonces la idea inicial es que el modelo lo clasifique en la clase $\hat{c}$ dada por

$$\hat{c} = \arg\max_{c \in C} \mathbb{P}\big(c \mid X_1 = x_1, \dots, X_n = x_n\big)$$

Dada una función que toma valores reales, el operador máx devuelve el mayor valor alcanzado por la función, mientras que el operador arg máx devuelve el elemento del dominio de la función donde esta alcanza ese valor máximo.

Para estimar las probabilidades anteriores, el modelo hace uso de la regla de Bayes

$$\mathbb{P}\big(c \mid X_1 = x_1, \dots, X_n = x_n\big) = \frac{\mathbb{P}\big(X_1 = x_1, \dots, X_n = x_n \mid c\big)\mathbb{P}(c)}{\mathbb{P}\big(X_1 = x_1, \dots, X_n = x_n\big)}$$

donde

- $\mathbb{P}\big(X_1 = x_1, \dots, X_n = x_n \mid c\big)$ es la verosimilitud del ejemplo, condicionada a la clase c. Es decir, la probabilidad de que un ejemplo de la clase c tenga exactamente esos valores de los atributos.
- $\mathbb{P}(c)$ es la probabilidad a priori de la clase c. Es decir, la probabilidad de partida de que un ejemplo pertenezca a la clase c, sin tener en cuenta los valores de los atributos.
- $\mathbb{P}\big(X_1 = x_1, \dots, X_n = x_n\big)$ es la probabilidad marginal del ejemplo. Es decir, la probabilidad de que un ejemplo tenga exactamente esos valores de los atributos, sin restringirnos a ninguna clase en particular.
- $\mathbb{P}\big(c \mid X_1 = x_1, \dots, X_n = x_n\big)$ es la probabilidad a posteriori de la clase c, condicionada al ejemplo $X_1 = x_1, \dots, X_n = x_n$. Es decir, la probabilidad final de que el ejemplo pertenezca a la clase c, tras haber tenido en cuenta los valores de los atributos.

La regla de decisión que utiliza naive Bayes para clasificar el ejemplo es, por tanto, la de máximo a posteriori (MAP)

$$\hat{c} = \operatorname*{arg\,máx}_{c \in C} \frac{\mathbb{P}\big(X_1 = x_1, \dots, X_n = x_n \mid c\big)\mathbb{P}(c)}{\mathbb{P}\big(X_1 = x_1, \dots, X_n = x_n\big)}$$

que, puesto que la probabilidad marginal del ejemplo es independiente de la clase c, puede simplificarse a

$$\hat{c} = \operatorname*{arg\,máx}_{c \in C} \mathbb{P}\big(X_1 = x_1, \dots, X_n = x_n \mid c\big)\mathbb{P}(c)$$

Nótese que si todas las clases son igualmente probables a priori ($\mathbb{P}(c) = 1/m$ para toda $c \in C$, para m clases en total), entonces la regla de decisión sería la de máxima verosimilitud (ML, del inglés *maximum likelihood*):

$$\begin{aligned}\hat{c} &= \operatorname*{arg\,máx}_{c \in C} \mathbb{P}\big(X_1 = x_1, \dots, X_n = x_n \mid c\big)\frac{1}{m} \\ &= \operatorname*{arg\,máx}_{c \in C} \mathbb{P}\big(X_1 = x_1, \dots, X_n = x_n \mid c\big)\end{aligned}$$

Para poder aplicar cualquiera de las dos reglas de decisión (MAP o ML) a un ejemplo arbitrario del dominio de la tarea, el modelo debería estimar la verosimilitud dentro de cada clase de cualquier combinación de valores de los atributos. Esto es inviable, ya que la cantidad de posibles combinaciones crece de manera exponencial (por ejemplo, para n atributos binarios habría 2^n combinaciones de valores). Para superar esta dificultad, el

modelo asume que los atributos son condicionalmente independientes dentro de cada clase, por lo que las reglas de decisión para clasificar ejemplos quedan finalmente como:

$$\hat{c} = \arg\max_{c \in C} \mathbb{P}(c) \prod_{i=1}^{n} \mathbb{P}(X_i = x_i \mid c) \qquad \text{regla MAP}$$

$$\hat{c} = \arg\max_{c \in C} \prod_{i=1}^{n} \mathbb{P}(X_i = x_i \mid c) \qquad \text{regla ML}$$

El número de probabilidades a estimar es entonces del orden de mnk, donde m es la cantidad de clases, n la cantidad de atributos y k la máxima cantidad de posibles valores de los atributos.

Considerar que los atributos son independientes, aunque sea solo dentro de cada clase, es una suposición muy simplista (de ahí el nombre ingenuo del modelo). Por ejemplo, en nuestro conjunto de datos de coches es claro que no es así: la potencia está relacionada con el número de cilindros, la aceleración depende de la potencia y el peso, etcétera. Sin embargo, aún cuando esta suposición no se cumple en general, naive Bayes es un clasificador que suele funcionar sorprendentemente bien.

Ejemplo: Predicción del tipo de consumo de un coche

Consideremos el siguiente ejemplo:

Origen	*Cilindros*	*Potencia*	*Peso*	*Aceleración*
Europa	4	85	2855	17.6

Antes de poder utilizar el modelo naive Bayes para clasificar este ejemplo como un coche de consumo alto o bajo debemos aplicar a los atributos el mismo procedimiento de discretización utilizado para los ejemplos de entrenamiento:

Origen	*Cilindros*	*Potencia*	*Peso*	*Aceleración*
Europa	4	$[70, 90]$	$[1945, 3009]$	$(16.25, 20.1]$

Utilizando las probabilidades estimadas en el ejemplo de la página 9 y usando la regla MAP, el modelo naive Bayes realiza los siguientes cálculos para clasificar el ejemplo:

- Para la clase *Tipo de consumo* = Alto,

$$\begin{aligned}
&\mathbb{P}(\textit{Tipo de consumo} = \text{Alto}) \times \\
&\mathbb{P}(\textit{Origen} = \text{Europa} \mid \textit{Tipo de consumo} = \text{Alto}) \times \\
&\mathbb{P}(\textit{Cilindros} = 4 \mid \textit{Tipo de consumo} = \text{Alto}) \times \\
&\mathbb{P}(\textit{Potencia} \in [70, 90] \mid \textit{Tipo de consumo} = \text{Alto}) \times \\
&\mathbb{P}(\textit{Peso} \in [1945, 3009] \mid \textit{Tipo de consumo} = \text{Alto}) \times \\
&\mathbb{P}(\textit{Aceleración} \in (16.25, 20.1] \mid \textit{Tipo de consumo} = \text{Alto}) = \\
&\frac{2}{5}\frac{1}{2}\frac{3}{4}\frac{3}{4}\frac{3}{4}\frac{1}{2} = \frac{27}{640}
\end{aligned}$$

- Para la clase *Tipo de consumo* = Bajo,

$$\begin{aligned}&\mathbb{P}(\textit{Tipo de consumo} = \text{Bajo})\times\\&\mathbb{P}(\textit{Origen} = \text{Europa} \mid \textit{Tipo de consumo} = \text{Bajo})\times\\&\mathbb{P}(\textit{Cilindros} = 4 \mid \textit{Tipo de consumo} = \text{Bajo})\times\\&\mathbb{P}(\textit{Potencia} \in [70, 90] \mid \textit{Tipo de consumo} = \text{Bajo})\times\\&\mathbb{P}(\textit{Peso} \in [1945, 3009] \mid \textit{Tipo de consumo} = \text{Bajo})\times\\&\mathbb{P}(\textit{Aceleración} \in (16.25, 20.1] \mid \textit{Tipo de consumo} = \text{Bajo}) =\\&\frac{3}{5}\frac{1}{6}\frac{1}{6}\frac{1}{3}\frac{1}{3}\frac{1}{2} = \frac{1}{1080}\end{aligned}$$

Como $\frac{27}{640} > \frac{1}{1080}$, el modelo clasifica el ejemplo como un coche de consumo alto.

Un detalle práctico importante que surge a la hora de aplicar un clasificador naive Bayes es que una implementación directa de las reglas de decisión puede dar lugar a que se produzca en el ordenador un desbordamiento numérico por abajo (*numeric underflow*, en inglés). Esto es debido a que la verosimilitud de un ejemplo dentro de una clase es, a menudo, un número muy pequeño, especialmente si la cantidad de atributos es alta.

Por ejemplo, supongamos que $\mathbb{P}(X_i = x_i \mid c) = 10^{-1}$, para todo $i = 1, \dots, n$. Entonces, bajo la asunción de independencia condicional del modelo,

$$\mathbb{P}(X_1 = x_1, \dots, X_n = x_n \mid c) = \prod_{i=1}^{n} \mathbb{P}(X_i = x_i \mid c) = 10^{-n}$$

Si n es suficientemente grande, este valor será tan pequeño que el ordenador no lo podrá representar y lo considerará igual a 0.

La solución es aplicar logaritmos para transformar los productos en sumas:

$$\log \prod_{i=1}^{n} \mathbb{P}(X_i = x_i \mid c) = \sum_{i=1}^{n} \log \mathbb{P}(X_i = x_i \mid c) = -n$$

(asumiendo en la última igualdad que la base del logaritmo es 10).

Esto no cambia el resultado de la regla de decisión, ya que la función logaritmo es creciente:

$$\begin{aligned}\hat{c} &= \arg\max_{c \in C} \mathbb{P}(c) \prod_{i=1}^{n} \mathbb{P}(X_i = x_i \mid c)\\&= \arg\max_{c \in C} \Big(\log \mathbb{P}(c) + \sum_{i=1}^{n} \log \mathbb{P}(X_i = x_i \mid c)\Big)\end{aligned}$$

(la base del logaritmo es irrelevante en la regla de decisión, ya que para cambiar de base basta multiplicar por una constante de proporcionalidad adecuada).

Aprendizaje del modelo

Naive Bayes es un modelo paramétrico ya que, independientemente de la cantidad de ejemplos que contenga el conjunto de entrenamiento $\mathcal{D}$, para poder realizar la tarea de clasificación únicamente debe aprender los valores de los siguientes parámetros:

- $\mathbb{P}(c)$, para cada clase c.
- $\mathbb{P}(X = x \mid c)$, para cada atributo X, cada posible valor x del atributo y cada clase c.

Los valores de estos parámetros se suelen inferir mediante una estimación de máxima verosimilitud del conjunto de entrenamiento, es decir, de tal manera que, si este último de generara aleatoriamente, entonces se maximice la probabilidad de generar exactamente el conjunto $\mathcal{D}$. En este caso, las estimaciones obtenidas son las siguientes:

$$\mathbb{P}(c) = \frac{N_c}{N} \qquad \mathbb{P}(X = x \mid c) = \frac{N_{X=x,c}}{N_c}$$

donde

- N es la cantidad total de ejemplos que hay en $\mathcal{D}$.
- N_c es la cantidad total de ejemplos de $\mathcal{D}$ que pertenecen a la clase c.
- $N_{X=x,c}$ es la cantidad total de ejemplos de $\mathcal{D}$ que pertenecen a la clase c y para los que el atributo X toma el valor x.

Ejemplo: Parámetros para la predicción del tipo de consumo de un coche

	Origen	*Cilindros*	*Potencia*	*Peso*	*Aceleración*	*Tipo de consumo*
E_1	Norteamérica	6	$(90, 198]$	$(3009, 4952]$	$(16.25, 20.1]$	Bajo
E_2	Europa	4	$[70, 90]$	$[1945, 3009]$	$(16.25, 20.1]$	Bajo
E_3	Europa	5	$[70, 90]$	$(3009, 4952]$	$(16.25, 20.1]$	Alto
E_4	Japón	6	$(90, 198]$	$[1945, 3009]$	$[11.5, 16.25]$	Bajo
E_5	Europa	4	$(90, 198]$	$[1945, 3009]$	$[11.5, 16.25]$	Alto
E_6	Japón	4	$[70, 90]$	$[1945, 3009]$	$(16.25, 20.1]$	Alto
E_7	Norteamérica	8	$(90, 198]$	$(3009, 4952]$	$[11.5, 16.25]$	Bajo
E_8	Norteamérica	4	$[70, 90]$	$[1945, 3009]$	$[11.5, 16.25]$	Alto
E_9	Norteamérica	6	$[70, 90]$	$(3009, 4952]$	$(16.25, 20.1]$	Bajo
E_{10}	Norteamérica	8	$(90, 198]$	$(3009, 4952]$	$[11.5, 16.25]$	Bajo

Para el conjunto de ejemplos de la tarea de predicción del tipo de consumo de un coche se tiene que

$$N = 10 \qquad N_{\text{Alto}} = 4 \qquad N_{\text{Bajo}} = 6$$

Por tanto, las estimaciones a priori de cada una de las clases posibles es

$$\mathbb{P}(\textit{Tipo de consumo} = \text{Alto}) = \frac{N_{\text{Alto}}}{N} = \frac{4}{10}$$
$$\mathbb{P}(\textit{Tipo de consumo} = \text{Bajo}) = \frac{N_{\text{Bajo}}}{N} = \frac{6}{10}$$

Para estimar eficientemente las probabilidades de los valores de cada atributo condicionados a cada clase es conveniente considerar una codificación *one-hot* de cada atributo: por cada posible valor del atributo se considera una variable binaria que vale 1 si para el ejemplo el atributo toma ese valor y 0 en caso contrario.

	Tipo de consumo = Alto				*Tipo de consumo* = Bajo		
	Origen				*Origen*		
	Europa	Japón	Norteamérica		Europa	Japón	Norteamérica
E_3	1	0	0	E_1	0	0	1
E_5	1	0	0	E_2	1	0	0
E_6	0	1	0	E_4	0	1	0
E_8	0	0	1	E_7	0	0	1
				E_9	0	0	1
				E_{10}	0	0	1
	2	1	1		1	1	4

Por lo tanto,

$$\mathbb{P}(\textit{Origen} = \text{Europa} \mid \textit{Tipo de consumo} = \text{Alto}) = \frac{2}{4}$$
$$\mathbb{P}(\textit{Origen} = \text{Japón} \mid \textit{Tipo de consumo} = \text{Alto}) = \frac{1}{4}$$
$$\mathbb{P}(\textit{Origen} = \text{Norteamérica} \mid \textit{Tipo de consumo} = \text{Alto}) = \frac{1}{4}$$

$$\mathbb{P}(\textit{Origen} = \text{Europa} \mid \textit{Tipo de consumo} = \text{Bajo}) = \frac{1}{6}$$
$$\mathbb{P}(\textit{Origen} = \text{Japón} \mid \textit{Tipo de consumo} = \text{Bajo}) = \frac{1}{6}$$
$$\mathbb{P}(\textit{Origen} = \text{Norteamérica} \mid \textit{Tipo de consumo} = \text{Bajo}) = \frac{4}{6}$$

Los parámetros correspondientes al resto de atributos se estiman de manera análoga.

	Tipo de consumo = Alto									
	Cilindros				*Potencia*		*Peso*		*Aceleración*	
	4	5	6	8	$[70, 90]$	$(90, 198]$	$[1945, 3009]$	$(3009, 4952]$	$[11.5, 16.25]$	$(16.25, 20.1]$
E_3	0	1	0	0	1	0	0	1	0	1
E_5	1	0	0	0	0	1	1	0	1	0
E_6	1	0	0	0	1	0	1	0	0	1
E_8	1	0	0	0	1	0	1	0	1	0
	3	1	0	0	3	1	3	1	2	2
$\mathbb{P}$	$\frac{3}{4}$	$\frac{1}{4}$	$\frac{0}{4}$	$\frac{0}{4}$	$\frac{3}{4}$	$\frac{1}{4}$	$\frac{3}{4}$	$\frac{1}{4}$	$\frac{2}{4}$	$\frac{2}{4}$

Tipo de consumo = Bajo

	Cilindros				*Potencia*		*Peso*		*Aceleración*	
	4	5	6	8	$[70, 90]$	$(90, 198]$	$[1945, 3009]$	$(3009, 4952]$	$[11.5, 16.25]$	$(16.25, 20.1]$
E_1	0	0	1	0	0	1	0	1	0	1
E_2	1	0	0	0	1	0	1	0	0	1
E_4	0	0	1	0	0	1	1	0	1	0
E_7	0	0	0	1	0	1	0	1	1	0
E_9	0	0	1	0	1	0	0	1	0	1
E_{10}	0	0	0	1	0	1	0	1	1	0
	1	0	3	2	2	4	2	4	3	3
$\mathbb{P}$	$\frac{1}{6}$	$\frac{0}{6}$	$\frac{3}{6}$	$\frac{2}{6}$	$\frac{2}{6}$	$\frac{4}{6}$	$\frac{2}{6}$	$\frac{4}{6}$	$\frac{3}{6}$	$\frac{3}{6}$

La estimación de máxima verosimilitud de los parámetros sobreajusta el modelo a los datos. Esto puede dar lugar al siguiente problema: si en el conjunto de entrenamiento no hay ningún ejemplo perteneciente a la clase c para el que el atributo X tome el valor x, entonces el modelo aprende que $\mathbb{P}(X = x \mid c) = 0$ y nunca clasificaría en la clase c un ejemplo de ese tipo, aún cuando los valores de los otros atributos sí respaldaran fuertemente su pertenencia a esa clase.

Una manera de resolver este problema es llevar a cabo lo que se conoce como suavizado de Laplace: para cada combinación de clase c, atributo X y valor x del atributo, a la hora de estimar $\mathbb{P}(X = x \mid c)$ se asume que el conjunto de entrenamiento contiene adicionalmente un ejemplo virtual que pertenece a la clase c y para el que $X = x$. Así, se obtiene la estimación

$$\mathbb{P}(X = x \mid c) = \frac{N_{X=x,c} + 1}{N_c + |X|}$$

donde $|X|$ es la cantidad total de posibles valores del atributo X.

Este suavizado de Laplace se puede generalizar a un suavizado aditivo en el que se consideran k ejemplos virtuales adicionales para cada combinación, en lugar de solo uno, dando lugar a la estimación

$$\mathbb{P}(X = x \mid c) = \frac{N_{X=x,c} + k}{N_c + k|X|}$$

Ejemplo: Predicción suavizada del tipo de consumo de un coche

Consideremos el siguiente ejemplo, en el que ya se han discretizado los atributos continuos:

Origen	*Cilindros*	*Potencia*	*Peso*	*Aceleración*
Europa	6	$[70, 90]$	$[1945, 3009]$	$[11.5, 16.25]$

Usando la regla de clasificación MAP, naive Bayes clasifica el ejemplo como un coche de consumo bajo:

- Para la clase *Tipo de consumo* = Alto,

$$\begin{aligned}&\mathbb{P}(\textit{Tipo de consumo} = \text{Alto})\times\\&\mathbb{P}(\textit{Origen} = \text{Europa} \mid \textit{Tipo de consumo} = \text{Alto})\times\\&\mathbb{P}(\textit{Cilindros} = 6 \mid \textit{Tipo de consumo} = \text{Alto})\times\\&\mathbb{P}(\textit{Potencia} \in [70, 90] \mid \textit{Tipo de consumo} = \text{Alto})\times\\&\mathbb{P}(\textit{Peso} \in [1945, 3009] \mid \textit{Tipo de consumo} = \text{Alto})\times\\&\mathbb{P}(\textit{Aceleración} \in [11.5, 16.25] \mid \textit{Tipo de consumo} = \text{Alto}) =\\&\frac{2}{5}\frac{2}{4}\frac{0}{4}\frac{3}{4}\frac{3}{4}\frac{2}{4} = 0\end{aligned}$$

- Para la clase *Tipo de consumo* = Bajo,

$$\begin{aligned}&\mathbb{P}(\textit{Tipo de consumo} = \text{Bajo})\times\\&\mathbb{P}(\textit{Origen} = \text{Europa} \mid \textit{Tipo de consumo} = \text{Bajo})\times\\&\mathbb{P}(\textit{Cilindros} = 6 \mid \textit{Tipo de consumo} = \text{Bajo})\times\\&\mathbb{P}(\textit{Potencia} \in [70, 90] \mid \textit{Tipo de consumo} = \text{Bajo})\times\\&\mathbb{P}(\textit{Peso} \in [1945, 3009] \mid \textit{Tipo de consumo} = \text{Bajo})\times\\&\mathbb{P}(\textit{Aceleración} \in [11.5, 16.25] \mid \textit{Tipo de consumo} = \text{Bajo}) =\\&\frac{3}{5}\frac{1}{6}\frac{3}{6}\frac{2}{6}\frac{2}{6}\frac{3}{6} = \frac{1}{360}\end{aligned}$$

Nótese que dentro de la clase *Tipo de consumo* = Alto se descarta toda la información proporcionada por los valores de los atributos, al no haber en esa clase ningún ejemplo de entrenamiento con *Cilindros* $= 6$. Sin embargo, aplicando un suavizado de Laplace el modelo sí que es capaz de hacer uso de esa información, clasificando en ese caso el ejemplo como un coche de consumo alto:

- Para la clase *Tipo de consumo* = Alto,

$$\begin{aligned}&\mathbb{P}(\textit{Tipo de consumo} = \text{Alto})\times\\&\mathbb{P}(\textit{Origen} = \text{Europa} \mid \textit{Tipo de consumo} = \text{Alto})\times\\&\mathbb{P}(\textit{Cilindros} = 6 \mid \textit{Tipo de consumo} = \text{Alto})\times\\&\mathbb{P}(\textit{Potencia} \in [70, 90] \mid \textit{Tipo de consumo} = \text{Alto})\times\\&\mathbb{P}(\textit{Peso} \in [1945, 3009] \mid \textit{Tipo de consumo} = \text{Alto})\times\\&\mathbb{P}(\textit{Aceleración} \in [11.5, 16.25] \mid \textit{Tipo de consumo} = \text{Alto}) =\\&\frac{2}{5}\frac{2+1}{4+3}\frac{0+1}{4+4}\frac{3+1}{4+2}\frac{3+1}{4+2}\frac{2+1}{4+2} = \frac{1}{210}\end{aligned}$$

- Para la clase *Tipo de consumo* = Bajo,

$$\begin{aligned}&\mathbb{P}(\textit{Tipo de consumo} = \text{Bajo})\times\\&\mathbb{P}(\textit{Origen} = \text{Europa} \mid \textit{Tipo de consumo} = \text{Bajo})\times\\&\mathbb{P}(\textit{Cilindros} = 6 \mid \textit{Tipo de consumo} = \text{Bajo})\times\\&\mathbb{P}(\textit{Potencia} \in [70, 90] \mid \textit{Tipo de consumo} = \text{Bajo})\times\\&\mathbb{P}(\textit{Peso} \in [1945, 3009] \mid \textit{Tipo de consumo} = \text{Bajo})\times\\&\mathbb{P}(\textit{Aceleración} \in [11.5, 16.25] \mid \textit{Tipo de consumo} = \text{Bajo}) =\\&\frac{3}{5}\frac{1+1}{6+3}\frac{3+1}{6+4}\frac{2+1}{6+2}\frac{2+1}{6+2}\frac{3+1}{6+2} = \frac{3}{800}\end{aligned}$$

Árboles de clasificación y regresión

Los árboles de clasificación y regresión (CART, del inglés *classification and regresion tree*) son un tipo de modelo adecuado tanto para abordar tareas de clasificación como de regresión, a partir tanto de atributos discretos como continuos, pero necesariamente numéricos.

Ejemplo: Codificación de atributos discretos

Una manera simple de convertir en numérico un atributo discreto es codificar cada posible valor que pueda tomar mediante un número entero no negativo. Por ejemplo, en el atributo *Origen* podríamos codificar los valores Europa, Japón y Norteamérica con los números $0, 1$ y 2, respectivamente. Nótese que el atributo discreto *Cilindros* ya es numérico, por lo que no se necesita recodificarlo.

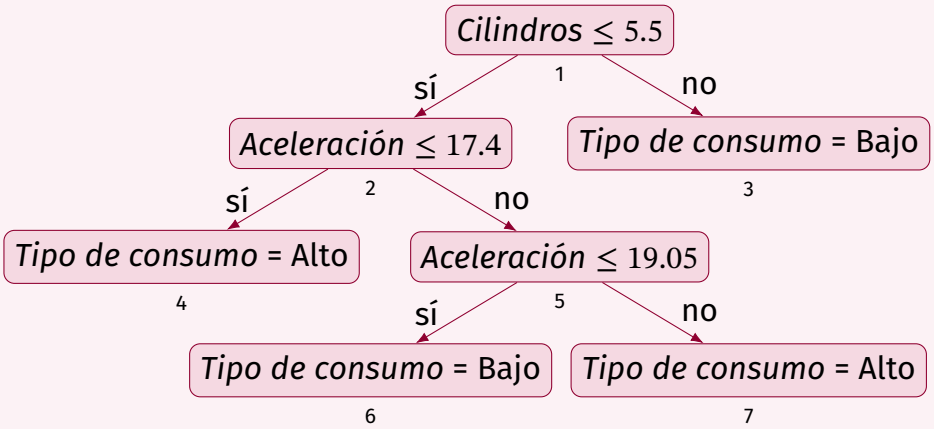
Origen	*Cilindros*	*Potencia*	*Peso*	*Aceleración*	*Tipo de consumo*
2	6	105	3613	16.5	Bajo
0	4	76	2511	18.0	Bajo
0	5	77	3530	20.1	Alto
1	6	122	2807	13.5	Bajo
0	4	90	2265	15.5	Alto
1	4	70	1945	16.8	Alto
2	8	145	3880	12.5	Bajo
2	4	90	2670	16.0	Alto
2	6	90	3211	17.0	Bajo
2	8	198	4952	11.5	Bajo

Realización de la tarea

Los CART son árboles de decisión binarios en los que cada nodo interno está etiquetado con un atributo y un valor umbral para ese atributo y cada nodo hoja está etiquetado con una clase, en el caso de una tarea de clasificación, o un valor numérico, en el caso de una tarea de regresión.

Dado un ejemplo del dominio de la tarea, el modelo le asocia una salida recorriendo el árbol desde la raíz hasta una de las hojas: en cada nodo interno con atributo X y umbral u elige la rama de la izquierda si en el ejemplo el valor de X no supera u y la rama de la derecha en caso contrario; la respuesta para el ejemplo es la clase o valor numérico asociado a la hoja a la que se llegue.

Ejemplo: Predicción del tipo de consumo de un coche



Dados los ejemplos

Origen	*Cilindros*	*Potencia*	*Peso*	*Aceleración*
2	6	108	2930	15.5
2	4	90	2711	15.5
0	4	85	2855	17.6

el árbol de decisión mostrado los clasificaría como sigue:

- Para el primer ejemplo, el valor del atributo *Cilindros* supera el umbral indicado por el nodo 1. Descendemos por la rama derecha, lo que nos lleva a la hoja 3, que clasifica el ejemplo como un coche de consumo bajo.
- Para el segundo ejemplo, el valor del atributo *Cilindros* es inferior al umbral indicado por el nodo 1. Descendemos por la rama izquierda hasta el nodo 2, que indica un umbral para el atributo *Aceleración* que no se supera en el ejemplo. Descendemos de nuevo por la rama izquierda, lo que nos lleva a la hoja 4, que clasifica el ejemplo como un coche de consumo alto.
- Para el tercer ejemplo, el valor del atributo *Cilindros* es inferior al umbral indicado por el nodo 1. Descendemos por la rama izquierda hasta el nodo 2, que indica un umbral para el atributo *Aceleración* que se supera en el ejemplo. Descendemos entonces por la rama derecha, donde ahora el umbral para el atributo *Aceleración* no se supera. Descendemos finalmente por la rama izquierda, lo que nos lleva a la hoja 6, que clasifica el ejemplo como un coche de consumo bajo.

Un CART puede entenderse como una colección de reglas de tipo condicional: cada

hoja proporciona una regla en la que el antecedente es la conjunción de todas las condiciones de acotación asociadas a los nodos internos de la rama que va desde la raíz hasta la hoja y el consecuente es la salida asociada a la hoja.

Ejemplo: Predicción del tipo de consumo de un coche mediante reglas

A partir del CART usado en el ejemplo previo se obtendría el siguiente conjunto de reglas:

- Si *Cilindros* ≤ 5.5 y *Aceleración* ≤ 17.4, entonces *Tipo de consumo* es Alto.
- Si *Cilindros* ≤ 5.5 y *Aceleración* > 17.4 y *Aceleración* ≤ 19.05, entonces *Tipo de consumo* es Bajo.
- Si *Cilindros* ≤ 5.5 y *Aceleración* > 17.4 y *Aceleración* > 19.05, entonces *Tipo de consumo* es Alto.
- Si *Cilindros* > 5.5, entonces *Tipo de consumo* es Bajo.

Aprendizaje del modelo

Dados un conjunto de ejemplos de entrenamiento $\mathcal{D}$ y un nodo n de un CART, denotemos por $\mathcal{D}_n$ el subconjunto de ejemplos asociado a n, es decir, el subconjunto de ejemplos que satisfacen todas las condiciones de los nodos desde la raíz hasta n. La construcción de un CART a partir del conjunto $\mathcal{D}$ se realiza extendiendo el árbol nodo a nodo. Para extender un CART parcialmente construido a partir de un nodo n se busca la condición que proporcione la mejor partición posible de $\mathcal{D}_n$ en dos subconjuntos, se asocia esa condición a n y se extiende el árbol con dos nuevos nodos hijos de n. Este proceso se reitera hasta obtener subconjuntos $\mathcal{D}_n$ no divisibles.

El siguiente algoritmo formaliza el proceso anterior:

CART($\mathcal{D}$)
1 **Si** NODIVISIBLE($\mathcal{D}$) **entonces**
2 **Devolver** un nodo etiquetado con ETIQUETA($\mathcal{D}$)
3 **Si no entonces**
4 Elegir el par (X, u) que proporcione la mejor partición $(\mathcal{D}^{\text{Izq}}, \mathcal{D}^{\text{Der}})$ de $\mathcal{D}$
5 $T_1 \leftarrow$ CART($\mathcal{D}^{\text{Izq}}$)
6 $T_2 \leftarrow$ CART($\mathcal{D}^{\text{Der}}$)
7 **Devolver** un nodo etiquetado con (X, u) y cuyos hijos sean T_1 y T_2

El funcionamiento del algoritmo consiste entonces en realizar un particionado recursivo del conjunto $\mathcal{D}$ de ejemplos de entrenamiento, de tal manera que se obtengan conjuntos cada vez más puros (en el sentido que se definirá más adelante).

Dados un nodo n y un atributo X, sea $x_1, \ldots, x_k$ la secuencia de valores distintos que toma X en $\mathcal{D}_n$, ordenados de menor a mayor. Los umbrales candidatos $u_1, \ldots, u_{k-1}$ para X son los puntos medios de los elementos de esa secuencia: $u_i = (x_i + x_{i+1})/2$. Dado un umbral candidato u, se divide $\mathcal{D}_n$ en el subconjunto $\mathcal{D}_n^{\text{Izq}}$ de ejemplos para los que X toma un valor menor o igual que u y el subconjunto $\mathcal{D}_n^{\text{Der}}$ de ejemplos para los que X toma un valor mayor que u.

Para poder elegir el par (X, u) que proporcione la mejor partición de $\mathcal{D}_n$ es necesario disponer de una función I que permita medir la impureza de un conjunto de ejemplos. Entonces basta seleccionar

$$\underset{\substack{X \text{ atributo} \\ u \text{ umbral candidato}}}{\arg\min} \frac{|\mathcal{D}_n^{\text{Izq}}|}{|\mathcal{D}|} I(\mathcal{D}_n^{\text{Izq}}) + \frac{|\mathcal{D}_n^{\text{Der}}|}{|\mathcal{D}|} I(\mathcal{D}_n^{\text{Der}})$$

Es decir, se eligen el atributo X y el umbral u que minimizan la impureza promedio de la partición.

Para una tarea de clasificación, la función de impureza que se suele utilizar es el índice de Gini, que estima la probabilidad de clasificar incorrectamente un ejemplo elegido aleatoriamente del conjunto de ejemplos y al que se le asigna una clase aleatoria elegida según la distribución de clases en el conjunto de ejemplos:

$$G(\mathcal{D}) = \sum_{c \in C} \hat{\pi}_c (1 - \hat{\pi}_c) = 1 - \sum_{c \in C} \hat{\pi}_c^2$$

donde C es el conjunto de clases posibles y $\hat{\pi}_c$ es la proporción de ejemplos de $\mathcal{D}$ etiquetados con la clase c (y que, por tanto, estima la probabilidad de que un ejemplo pertenezca a esa clase).

El índice de Gini de un conjunto de ejemplos $\mathcal{D}$ toma siempre un valor entre 0 y 1 y toma el valor nulo únicamente para los conjuntos puros, es decir, aquellos cuyos ejemplos pertenecen todos a la misma clase.

Para una tarea de regresión, la función de impureza que se suele utilizar es la varianza:

$$\text{Var}(\mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{(\boldsymbol{x}, y) \in \mathcal{D}} \left(y - \bar{y}\right)^2$$

donde $\bar{y}$ es la media de los valores del atributo objetivo para los ejemplos del conjunto, $\bar{y} = \frac{1}{|\mathcal{D}|} \sum_{(\boldsymbol{x}, y) \in \mathcal{D}} y$.

La varianza de un conjunto de ejemplos $\mathcal{D}$ toma siempre un valor mayor o igual que 0 y toma el valor nulo únicamente para los conjuntos puros, es decir, aquellos cuyos ejemplos tienen asociados todos el mismo valor.

En principio, el particionado solo deja de realizarse cuando se llega a un conjunto de ejemplos totalmente puro. Es decir, la definición de NODIVISIBLE($\mathcal{D}$) es: para una tarea de clasificación, que todos los ejemplos contenidos en $\mathcal{D}$ tengan asociada la misma clase; para una tarea de regresión, que el conjunto de valores del atributo objetivo asociados a los ejemplos contenidos en $\mathcal{D}$ tenga varianza nula. Sin embargo, esto suele dar lugar a árboles de decisión que han «memorizado» el conjunto de entrenamiento (por ejemplo, en un CART para una tarea de regresión, al exigirles que tengan varianza nula los conjuntos de ejemplos asociados a las hojas del árbol acabarán conteniendo, en general, un único ejemplo). Es por ello habitual añadir a NODIVISIBLE($\mathcal{D}$) condiciones de parada adicionales (lo que se conoce como realizar una poda a priori del árbol), como que se haya alcanzado una determinada profundidad prefijada de antemano o que $\mathcal{D}$ no contenga una cantidad mínima de ejemplos.

Finalmente, para una tarea de clasificación, ETIQUETA($\mathcal{D}$) es la clase mayoritaria de los ejemplos de $\mathcal{D}$, es decir, cada hoja del árbol se etiqueta con la clase mayoritaria de los

ejemplos asociados a esa hoja. Para una tarea de regresión, $\text{ETIQUETA}(\mathcal{D}) = \frac{1}{|\mathcal{D}|}\sum_{(x,y)\in\mathcal{D}} y$, es decir, cada hoja del árbol se etiqueta con la media de los valores del atributo objetivo para los ejemplos asociados a esa hoja.

Ejemplo: Construcción de un CART predictor del tipo de consumo de un coche

	Origen	*Cilindros*	*Potencia*	*Peso*	*Aceleración*	*Tipo de consumo*
E_1	2	6	105	3613	16.5	Bajo
E_2	0	4	76	2511	18.0	Bajo
E_3	0	5	77	3530	20.1	Alto
E_4	1	6	122	2807	13.5	Bajo
E_5	0	4	90	2265	15.5	Alto
E_6	1	4	70	1945	16.8	Alto
E_7	2	8	145	3880	12.5	Bajo
E_8	2	4	90	2670	16.0	Alto
E_9	2	6	90	3211	17.0	Bajo
E_{10}	2	8	198	4952	11.5	Bajo

La construcción de un CART a partir de los ejemplos anteriores comienza con un árbol con un solo nodo, la raíz del árbol.

El conjunto de ejemplos asociado al nodo 1 es el conjunto completo

$$\mathcal{D}_1 = \{E_1, E_2, \dots, E_{10}\}$$

Por lo tanto, $\hat{\pi}_{\text{Alto}} = \frac{4}{10}, \hat{\pi}_{\text{Bajo}} = \frac{6}{10}$ y $G(\mathcal{D}_1) = 1 - \left(\frac{4}{10}\right)^2 - \left(\frac{6}{10}\right)^2 = 0.48$.

Como su índice de Gini no es nulo, hay que buscar el par atributo-umbral que proporcione la mejor partición de $\mathcal{D}_1$.

- Para el atributo *Origen* se ordenan de menor a mayor los valores distintos que toma en $\mathcal{D}_1$: $0, 1, 2$. Los umbrales candidatos son los puntos medios:
  - Para el umbral 0.5 se tiene que

    $$\mathcal{D}_1^{\text{Izq}} = \{E_2, E_3, E_5\} \qquad \mathcal{D}_1^{\text{Der}} = \{E_1, E_4, E_6, E_7, E_8, E_9, E_{10}\}$$

    con índices de Gini

    $$G(\mathcal{D}_1^{\text{Izq}}) = 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2 \cong 0.4444 \qquad G(\mathcal{D}_1^{\text{Der}}) = 1 - \left(\frac{2}{7}\right)^2 - \left(\frac{5}{7}\right)^2 \cong 0.4082$$

    Luego la impureza promedio de la partición es

    $$\frac{3}{10}0.4444 + \frac{7}{10}0.4082 \cong 0.4191$$

- Para el umbral 1.5 se tiene que

$$\mathcal{D}_1^{\text{Izq}} = \{E_2, E_3, E_4, E_5, E_6\} \qquad \mathcal{D}_1^{\text{Der}} = \{E_1, E_7, E_8, E_9, E_{10}\}$$

con índices de Gini

$$G(\mathcal{D}_1^{\text{Izq}}) = 1 - \left(\frac{3}{5}\right)^2 - \left(\frac{2}{5}\right)^2 = 0.48 \qquad G(\mathcal{D}_1^{\text{Der}}) = 1 - \left(\frac{1}{5}\right)^2 - \left(\frac{4}{5}\right)^2 = 0.32$$

Luego la impureza promedio de la partición es

$$\frac{5}{10}0.48 + \frac{5}{10}0.32 = 0.4$$

- Para el atributo *Cilindros* se ordenan de menor a mayor los valores distintos que toma en $\mathcal{D}_1$: $4, 5, 6, 8$. Los umbrales candidatos son los puntos medios:
  - Para el umbral 4.5 se tiene que

$$\mathcal{D}_1^{\text{Izq}} = \{E_2, E_5, E_6, E_8\} \qquad \mathcal{D}_1^{\text{Der}} = \{E_1, E_3, E_4, E_7, E_9, E_{10}\}$$

con índices de Gini

$$G(\mathcal{D}_1^{\text{Izq}}) = 1 - \left(\frac{3}{4}\right)^2 - \left(\frac{1}{4}\right)^2 \cong 0.375 \qquad G(\mathcal{D}_1^{\text{Der}}) = 1 - \left(\frac{1}{6}\right)^2 - \left(\frac{5}{6}\right)^2 \cong 0.2778$$

Luego la impureza promedio de la partición es

$$\frac{4}{10}0.375 + \frac{6}{10}0.2778 \cong 0.3167$$

  - Para el umbral 5.5 se tiene que

$$\mathcal{D}_1^{\text{Izq}} = \{E_2, E_3, E_5, E_6, E_8\} \qquad \mathcal{D}_1^{\text{Der}} = \{E_1, E_4, E_7, E_9, E_{10}\}$$

con índices de Gini

$$G(\mathcal{D}_1^{\text{Izq}}) = 1 - \left(\frac{4}{5}\right)^2 - \left(\frac{1}{5}\right)^2 = 0.32 \qquad G(\mathcal{D}_1^{\text{Der}}) = 1 - \left(\frac{0}{5}\right)^2 - \left(\frac{5}{5}\right)^2 = 0$$

Luego la impureza promedio de la partición es

$$\frac{5}{10}0.32 + \frac{5}{10}0 = 0.16$$

  - Para el umbral 7 se tiene que

$$\mathcal{D}_1^{\text{Izq}} = \{E_1, E_2, E_3, E_4, E_5, E_6, E_8, E_9\} \qquad \mathcal{D}_1^{\text{Der}} = \{E_7, E_{10}\}$$

con índices de Gini

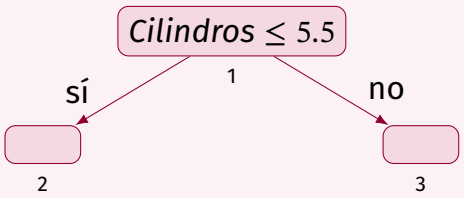
$$G(\mathcal{D}_1^{\text{Izq}}) = 1 - \left(\frac{4}{8}\right)^2 - \left(\frac{4}{8}\right)^2 = 0.5 \qquad G(\mathcal{D}_1^{\text{Der}}) = 1 - \left(\frac{0}{2}\right)^2 - \left(\frac{2}{2}\right)^2 = 0$$

Luego la impureza promedio de la partición es

$$\frac{8}{10}0.5 + \frac{2}{10}0 = 0.4$$

- Para el atributo *Potencia* se ordenan de menor a mayor los valores distintos que toma en $\mathscr{D}_1$: $70, 76, 77, 90, 105, 122, 145, 198$. Los umbrales candidatos son los puntos medios:
  - Para el umbral 73 la impureza promedio de la partición es 0.4.
  - Para el umbral 76.5 la impureza promedio de la partición es 0.475.
  - Para el umbral 83.5 la impureza promedio de la partición es 0.4191.
  - Para el umbral 97.5 la impureza promedio de la partición es 0.2666.
  - Para el umbral 113.5 la impureza promedio de la partición es 0.3429.
  - Para el umbral 133.5 la impureza promedio de la partición es 0.4.
  - Para el umbral 171.5 la impureza promedio de la partición es 0.4444.
- Para el atributo *Peso* se ordenan de menor a mayor los valores distintos que toma en $\mathscr{D}_1$: $1945, 2265, 2511, 2670, 2807, 3211, 3530, 3613, 3880, 4952$. Los umbrales candidatos son los puntos medios:
  - Para el umbral 2105 la impureza promedio de la partición es 0.4.
  - Para el umbral 2388 la impureza promedio de la partición es 0.3.
  - Para el umbral 2590.5 la impureza promedio de la partición es 0.4191.
  - Para el umbral 2738.5 la impureza promedio de la partición es 0.3167.
  - Para el umbral 3009 la impureza promedio de la partición es 0.4.
  - Para el umbral 3370.5 la impureza promedio de la partición es 0.45.
  - Para el umbral 3571.5 la impureza promedio de la partición es 0.3429.
  - Para el umbral 3746.5 la impureza promedio de la partición es 0.4.
  - Para el umbral 4416 la impureza promedio de la partición es 0.4444.
- Para el atributo *Aceleración* se ordenan de menor a mayor los valores distintos que toma en $\mathscr{D}_1$: $11.5, 12.5, 13.5, 15.5, 16.0, 16.5, 16.8, 17.0, 18.0, 20.1$. Los umbrales candidatos son los puntos medios:
  - Para el umbral 12 la impureza promedio de la partición es 0.4444.
  - Para el umbral 13 la impureza promedio de la partición es 0.4.
  - Para el umbral 14.5 la impureza promedio de la partición es 0.3429.
  - Para el umbral 15.75 la impureza promedio de la partición es 0.45.
  - Para el umbral 16.25 la impureza promedio de la partición es 0.48.
  - Para el umbral 16.65 la impureza promedio de la partición es 0.4666.
  - Para el umbral 16.9 la impureza promedio de la partición es 0.4762.
  - Para el umbral 17.5 la impureza promedio de la partición es 0.475.
  - Para el umbral 19.05 la impureza promedio de la partición es 0.4.

Por lo tanto, el atributo *Cilindros* y el umbral 5.5 es el par que proporciona la partición con menor impureza promedio, por lo que es el que asociamos al nodo 1.

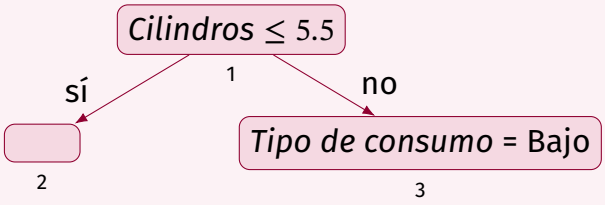


El conjunto de ejemplos asociado al nodo 3 es el conjunto

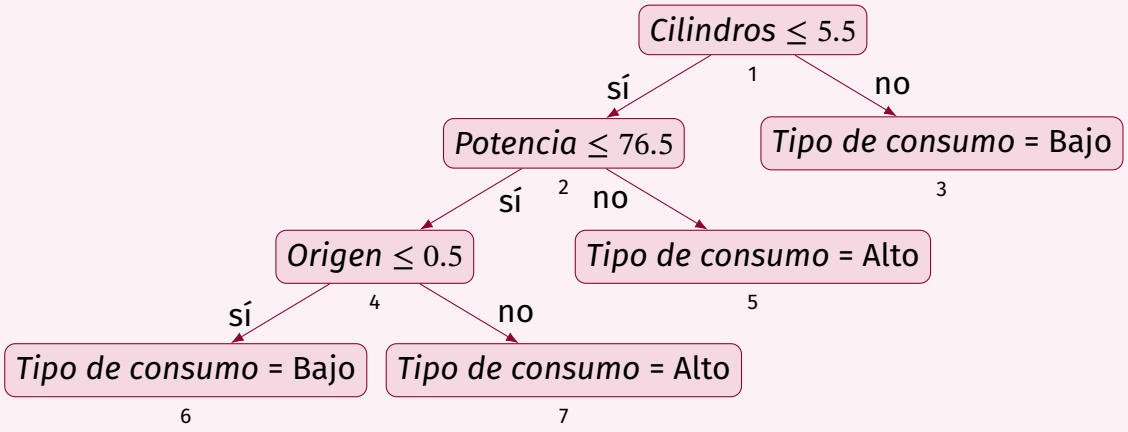
$$\mathcal{D}_3 = \{E_1, E_4, E_7, E_9, E_{10}\}$$

Por lo tanto, $\hat{\pi}_{\text{Alto}} = \frac{0}{5}, \hat{\pi}_{\text{Bajo}} = \frac{5}{5}$ y $G(\mathcal{D}_3) = 1 - \left(\frac{0}{5}\right)^2 - \left(\frac{5}{5}\right)^2 = 0$.

Como su índice de Gini es nulo, el nodo 3 será una hoja del árbol etiquetada con la clase mayoritaria de los ejemplos de $\mathcal{D}_3$.



Reiterando el proceso (eligiendo, para cada nodo, de todos los pares atributo-umbral que proporcionan la partición con menor impureza promedio el considerado en primer lugar) se obtiene finalmente el CART



CART es un modelo no paramétrico, ya que la cantidad de parámetros (nodos y etiquetas de los nodos) que debe aprender depende del conjunto de entrenamiento. De hecho, la simple adición de un nuevo ejemplo de entrenamiento puede dar lugar a un árbol de decisión radicalmente distinto.

kNN

El modelo k vecinos más cercanos (kNN, del inglés k *nearest neighbours*) es un modelo de aprendizaje supervisado que es adecuado para abordar tanto tareas de clasificación como de regresión.

Realización de la tarea

Dado un ejemplo del dominio de la tarea, el modelo le asocia una clase, en el caso de una tarea de clasificación, o un valor numérico, en el caso de una tarea de regresión, en función de las clases o valores asociados a ejemplos similares del conjunto de entrenamiento. Para ello es necesario fijar previamente el valor de los siguientes hiperparámetros:

- La distancia o métrica d a utilizar: el modelo kNN define el concepto de similitud entre ejemplos a partir del concepto de cercanía (dos ejemplos son similares si son cercanos), lo que a su vez implica la necesidad de definir un concepto de distancia.

  Para ejemplos descritos mediante atributos numéricos es habitual utilizar la distancia Manhattan
  $$d(\boldsymbol{x}, \boldsymbol{x}') = \sum_{i=1}^{n} |x_i - x'_i|$$
  o la distancia euclídea
  $$d(\boldsymbol{x}, \boldsymbol{x}') = \sqrt{\sum_{i=1}^{n} \left(x_i - x'_i\right)^2}$$

  Para ejemplos descritos mediante atributos discretos la distancia más utilizada es la distancia de Hamming, que cuenta la cantidad de atributos en los que los ejemplos difieren entre sí
  $$d(\boldsymbol{x}, \boldsymbol{x}') = \sum_{i=1}^{n} \mathbb{1}\left(x_i \neq x'_i\right)$$
  donde $\mathbb{1}$ es la función indicador, que vale 1 si su argumento es verdadero y 0 en caso contrario.

- La cantidad k de vecinos a considerar: como veremos, en el caso de una tarea de clasificación binaria es conveniente que este parámetro tome un valor impar, para de esta forma evitar que se produzcan empates.

Cuando el modelo recibe un ejemplo $\boldsymbol{x}$ como entrada actúa de la siguiente manera:

1. Determina los k ejemplos de entrenamiento, $(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_k, y_k)$, más cercanos a $\boldsymbol{x}$.

   Para ello debe calcular todas las distancias entre $\boldsymbol{x}$ y cada ejemplo de entrenamiento y seleccionar de estos últimos los k ejemplos que se encuentren a menor distancia.

2. Para una tarea de clasificación, asocia a $\boldsymbol{x}$ la clase mayoritaria (es decir, la que más se repite) de $y_1, \dots, y_k$ (hay que tener establecida alguna regla para deshacer los empates, para el caso en que estos se pudieran producir).

   Para una tarea de regresión, asocia a $\boldsymbol{x}$ el valor medio de $y_1, \dots, y_k$.

Ejemplo: Predicción del tipo de consumo de un coche

	Origen	*Cilindros*	*Potencia*	*Peso*	*Aceleración*	*Tipo de consumo*
E_1	2	6	105	3613	16.5	Bajo
E_2	0	4	76	2511	18.0	Bajo
E_3	0	5	77	3530	20.1	Alto
E_4	1	6	122	2807	13.5	Bajo
E_5	0	4	90	2265	15.5	Alto
E_6	1	4	70	1945	16.8	Alto
E_7	2	8	145	3880	12.5	Bajo
E_8	2	4	90	2670	16.0	Alto
E_9	2	6	90	3211	17.0	Bajo
E_{10}	2	8	198	4952	11.5	Bajo

Consideremos el siguiente ejemplo E:

Origen	*Cilindros*	*Potencia*	*Peso*	*Aceleración*
Europa	4	75	2542	17

que, recodificando numéricamente el atributo *Origen*, se convierte en

Origen	*Cilindros*	*Potencia*	*Peso*	*Aceleración*
0	4	75	2542	17

La distancia euclídea del ejemplo E a cada uno de los ejemplos de entrenamiento son las siguientes:

$$d(E,E_1) = \sqrt{(2-0)^2+(6-4)^2+(105-75)^2+(3613-2542)^2+(16.5-17)^2} \cong 1071.42$$
$$d(E,E_2) = \sqrt{(0-0)^2+(4-4)^2+(76-75)^2+(2511-2542)^2+(18.0-17)^2} \cong 31.03$$
$$d(E,E_3) = \sqrt{(0-0)^2+(5-4)^2+(77-75)^2+(3530-2542)^2+(20.1-17)^2} \cong 988.01$$
$$d(E,E_4) = \sqrt{(1-0)^2+(6-4)^2+(122-75)^2+(2807-2542)^2+(13.5-17)^2} \cong 269.17$$
$$d(E,E_5) = \sqrt{(0-0)^2+(4-4)^2+(90-75)^2+(2265-2542)^2+(15.5-17)^2} \cong 277.41$$
$$d(E,E_6) = \sqrt{(1-0)^2+(4-4)^2+(70-75)^2+(1945-2542)^2+(16.8-17)^2} \cong 597.02$$
$$d(E,E_7) = \sqrt{(2-0)^2+(8-4)^2+(145-75)^2+(3880-2542)^2+(12.5-17)^2} \cong 1339.84$$
$$d(E,E_8) = \sqrt{(2-0)^2+(4-4)^2+(90-75)^2+(2670-2542)^2+(16.0-17)^2} \cong 128.9$$
$$d(E,E_9) = \sqrt{(2-0)^2+(6-4)^2+(90-75)^2+(3211-2542)^2+(17.0-17)^2} \cong 669.17$$
$$d(E,E_{10}) = \sqrt{(2-0)^2+(8-4)^2+(198-75)^2+(4952-2542)^2+(11.5-17)^2} \cong 2413.15$$

Por lo tanto, un modelo kNN que use la distancia euclídea como métrica y 3 como valor de k clasificaría el ejemplo E como un coche de consumo bajo, ya que los tres ejemplos de entrenamiento más cercanos a E son E_2, E_8 y E_4, que son, respectivamente, coches de consumo bajo, alto y bajo.

Por el contrario, un modelo kNN que use la distancia euclídea como métrica y 5 como valor de k clasificaría el ejemplo E como un coche de consumo alto, ya que los cinco ejemplos de entrenamiento más cercanos a E son E_2, E_8, E_4, E_5 y E_6, que son, respectivamente, coches de consumo bajo, alto, bajo, alto y alto.

En el caso de atributos numéricos es conveniente en general aplicar, previamente a la construcción de un modelo kNN, un procedimiento de normalización de los valores de los atributos. De esta forma se evita que en el cálculo de las distancias entre ejemplos uno de los atributos domine sobre los demás, al tomar valores en un rango mayor que el resto de atributos.

Existe una gran variedad de maneras de normalizar un atributo numérico, siendo dos de las más habituales las siguientes:

- Restar m y dividir por $M-m$, siendo m y M, respectivamente, el mínimo y el máximo de los valores del atributo (normalización mín-máx). De esta forma se consigue que el rango de valores sea el intervalo $[0, 1]$.

- Restar μ y dividir por σ, siendo μ y σ la media y la desviación típica de los valores del atributo (tipificación). De esta forma se consigue que los valores tengan media 0 y desviación típica 1.

Hay que tener en cuenta que a la hora de aplicar el modelo a la realización de la tarea en cuestión, debe aplicarse a los ejemplos nuevos el mismo procedimiento de normalización que se ha utilizado en cada atributo y usando para ello los mismos parámetros obtenidos a partir de los ejemplos de entrenamiento (el mínimo y el máximo del atributo para la normalización mín-máx, la media y la desviación típica del atributo para la tipificación).

Ejemplo: Predicción del tipo de consumo de un coche con atributos normalizados

Puesto que el rango de valores del atributo *Peso* es mucho mayor que el del resto de atributos, el primero domina en el cálculo de la distancia euclídea, haciendo irrelevantes los valores de los últimos. En consecuencia, a la hora de construir un modelo kNN es conveniente normalizar los atributos.

Para aplicar una normalización mín-máx es necesario obtener el mínimo y el máximo de cada atributo:

	Origen	*Cilindros*	*Potencia*	*Peso*	*Aceleración*
Mínimo	0	4	70	1945	11.5
Máximo	2	8	198	4952	20.1

Una vez aplicada la normalización, se tiene el siguiente conjunto de ejemplos de entrenamiento (redondeando a dos decimales):

	Origen	*Cilindros*	*Potencia*	*Peso*	*Aceleración*	*Tipo de consumo*
E_1	1	0.5	0.27	0.55	0.58	Bajo
E_2	0	0	0.05	0.19	0.76	Bajo
E_3	0	0.25	0.05	0.53	1	Alto
E_4	0.5	0.5	0.41	0.29	0.23	Bajo
E_5	0	0	0.16	0.11	0.47	Alto
E_6	0.5	0	0	0	0.62	Alto
E_7	1	1	0.59	0.64	0.12	Bajo
E_8	1	0	0.16	0.24	0.52	Alto
E_9	1	0.5	0.16	0.42	0.64	Bajo
E_{10}	1	1	1	1	0	Bajo

Consideremos el siguiente ejemplo E:

Origen	*Cilindros*	*Potencia*	*Peso*	*Aceleración*
0	4	75	2542	17

Antes de proceder a clasificarlo, hay que normalizar los valores de sus atributos, usando para ello los valores mínimo y máximo aprendidos de los ejemplos de entrenamiento:

Origen	*Cilindros*	*Potencia*	*Peso*	*Aceleración*
$\frac{0-0}{2-0} = 0$	$\frac{4-4}{8-4} = 0$	$\frac{75-70}{198-70} = 0.04$	$\frac{2542-1945}{4952-1945} = 0.2$	$\frac{17-11.5}{20.1-11.5} = 0.64$

Las distancias euclídeas del ejemplo E a cada uno de los ejemplos de entrenamiento son entonces las siguientes:

$d(E, E_1) \cong 1.198$ $d(E, E_2) \cong 0.117$ $d(E, E_3) \cong 0.5483$ $d(E, E_4) \cong 0.899$
$d(E, E_5) \cong 0.2294$ $d(E, E_6) \cong 0.5399$ $d(E, E_7) \cong 1.6646$ $d(E, E_8) \cong 1.0144$
$d(E, E_9) \cong 1.146$ $d(E, E_{10}) \cong 1.9937$

Por lo tanto, un modelo kNN que use la distancia euclídea como métrica y 3 como valor de k clasificaría el ejemplo E como un coche de consumo alto, ya que los tres ejemplos de entrenamiento más cercanos a E son E_2, E_5 y E_6, que son, respectivamente, coches de consumo bajo, alto y alto. Nótese como ha cambiado la clasificación con respecto al modelo que usaba los ejemplos de entrenamiento sin normalizar, debido a que ahora sí se está teniendo en cuenta la información proporcionada por todos los atributos.

Aprendizaje del modelo

Para construir un modelo kNN basta memorizar los ejemplos del conjunto de entrenamiento. Se trata, pues, de un modelo no paramétrico, ya que sus parámetros son precisamente estos ejemplos, por lo que la cantidad de parámetros del modelo coincide exactamente con la cantidad de ejemplos de entrenamiento.

Evaluación y selección de modelos

El objetivo a la hora de construir un modelo de aprendizaje automático es que la tarea que resuelva nos facilite la resolución de un determinado problema inicial. Es por ello que, una vez construido el modelo, se debe evaluar su rendimiento en el desempeño de la tarea. Esto implica ineludiblemente definir una cierta medida de ese rendimiento, adecuada para el problema que se pretenda resolver.

Para un modelo f que realiza una tarea de clasificación multiclase, con m clases posibles, y un conjunto de ejemplos $\mathscr{D}$ (con $|\mathscr{D}|$ ejemplos en total), se puede definir una plétora de medidas a partir de la matriz de confusión $\boldsymbol{C} = \left(c_{ij}\right)_{i,j=1,\dots,m}$, que es una matriz de contingencias donde

$$c_{ij} = \text{cantidad de ejemplos de la clase } i \text{ clasificados en la clase } j \text{ por el modelo } f$$

De esta forma,

- El elemento c_{ii} de la diagonal denota la cantidad de ejemplos de la clase i clasificados correctamente por f. La tasa de acierto o exactitud (*accuracy*, en inglés) del modelo es la proporción de ejemplos clasificados correctamente, es decir,

$$acc = \frac{1}{|\mathscr{D}|} \sum_{i=1}^{m} c_{ii}$$

- El elemento c_{ij}, con $i \neq j$, denota la cantidad de ejemplos de la clase i clasificados incorrectamente en la clase j por f. La tasa de error (*error rate*, en inglés) del modelo es la proporción de ejemplos clasificados incorrectamente, es decir,

$$err = \frac{1}{|\mathscr{D}|} \sum_{\substack{i,j=1 \\ i \neq j}}^{m} c_{ij}$$

En el caso particular de una tarea de clasificación binaria, una de las clases se suele llamar positiva y la otra negativa, adoptando la matriz de confusión la siguiente forma:

		Clase predicha	
		Positiva	*Negativa*
Clase correcta	*Positiva*	Verdaderos positivos (*VP*)	Falsos negativos (*FN*)
	Negativa	Falsos positivos (*FP*)	Verdaderos negativos (*VN*)

De esta forma,

- La tasa de verdaderos positivos, sensibilidad o recuerdo (*true positive rate*, *sensitivity* o *recall*, en inglés) del modelo es la proporción de ejemplos positivos clasificados correctamente, es decir,

$$tpr = \frac{VP}{VP + FN}$$

- La tasa de verdaderos negativos, especificidad o recuerdo negativo (*true negative rate*, *specifity* o *negative recall*, en inglés) del modelo es la proporción de ejemplos negativos clasificados correctamente, es decir,

$$tnr = \frac{VN}{FP + VN}$$

- La tasa de falsos negativos del modelo es la proporción de ejemplos positivos clasificados incorrectamente, es decir,

$$fnr = \frac{FN}{VP + FN}$$

- La tasa de falsos positivos del modelo es la proporción de ejemplos negativos clasificados incorrectamente, es decir,

$$fpr = \frac{FP}{FP + VN}$$

- La precisión (*precision*, en inglés) del modelo es la proporción de ejemplos realmente positivos entre los clasificados como positivos, es decir,

$$prec = \frac{VP}{VP + FP}$$

Todas estas medidas están interrelacionadas y la elección de una u otra es dependiente del problema que se esté tratando de resolver.

Ejemplo: Modelo kNN que predice el tipo de consumo de un coche

Un modelo kNN que use la distancia euclídea como métrica y 5 como valor de k clasifica los ejemplos de entrenamiento (normalizados) como sigue:

- E_1 lo clasifica como un coche de consumo bajo, lo que es correcto.
- E_2 lo clasifica como un coche de consumo alto, lo que es incorrecto.
- E_3 lo clasifica como un coche de consumo alto, lo que es correcto.
- E_4 lo clasifica como un coche de consumo bajo, lo que es correcto.
- E_5 lo clasifica como un coche de consumo alto, lo que es correcto.
- E_6 lo clasifica como un coche de consumo alto, lo que es correcto.
- E_7 lo clasifica como un coche de consumo bajo, lo que es correcto.
- E_8 lo clasifica como un coche de consumo bajo, lo que es incorrecto.
- E_9 lo clasifica como un coche de consumo bajo, lo que es correcto.
- E_{10} lo clasifica como un coche de consumo bajo, lo que es correcto.

Considerando consumo bajo como la clase positiva, se tiene entonces la siguiente matriz de confusión:

		Clase predicha			
		Consumo bajo		*Consumo alto*	
Clase correcta	*Consumo bajo*	5	(*VP*)	1	(*FN*)
	Consumo alto	1	(*FP*)	3	(*VN*)

De donde se obtienen las siguientes medidas del rendimiento del modelo:

$$acc = \frac{VP + VN}{|\mathcal{D}|} = \frac{8}{10} \qquad err = \frac{FP + FN}{|\mathcal{D}|} = \frac{2}{10}$$
$$tpr = \frac{VP}{VP + FN} = \frac{5}{6} \qquad fnr = \frac{FN}{VP + FN} = \frac{1}{6}$$
$$tnr = \frac{VN}{FP + VN} = \frac{3}{4} \qquad fpr = \frac{FP}{FP + VN} = \frac{1}{4}$$
$$prec = \frac{VP}{VP + FP} = \frac{5}{6}$$

Para un modelo f que realiza una tarea de regresión, su rendimiento sobre un conjunto de ejemplos $\mathcal{D}$ (con $|\mathcal{D}|$ ejemplos en total) se mide mediante alguna función que compare los valores predichos con los valores correctos. Algunas de las más utilizadas son:

- El error absoluto medio (MAE, *mean absolute error*, en inglés) calcula el promedio del error cometido por el valor predicho con respecto al valor correcto para cada ejemplo. El error se calcula como el valor absoluto de la diferencia, ya que no nos interesa si es por exceso o por defecto.

$$\text{MAE} = \frac{1}{|\mathcal{D}|} \sum_{(\boldsymbol{x},y)\in\mathcal{D}} |\hat{y} - y| \qquad \text{donde } \hat{y} = f(\boldsymbol{x})$$

- El error cuadrático medio (MSE, *mean squared error*, en inglés) calcula el promedio del error cometido por el valor predicho con respecto al valor correcto para cada ejemplo. El error se calcula como la diferencia al cuadrado, obteniéndose por tanto una función diferenciable más fácil de optimizar matemáticamente que el MAE. Por contra, y al contrario de lo que ocurre con el MAE, esta función penaliza los errores grandes mucho más que los errores pequeños. Esto quiere decir que basta con que en un ejemplo el modelo proporcione una respuesta lejos de la correcta para que el error se incremente en exceso.

$$\text{MSE} = \frac{1}{|\mathcal{D}|} \sum_{(\boldsymbol{x},y)\in\mathcal{D}} (\hat{y} - y)^2 \qquad \text{donde } \hat{y} = f(\boldsymbol{x})$$

- La unidad de medida del MSE es la unidad del atributo objetivo, al cuadrado. Es por ello que se suele usar la raíz del error cuadrático medio (RMSE, *root mean squared error*, en inglés), definida como la raíz cuadrada del MSE, para que el error esté

medido en la misma unidad que el atributo objetivo.

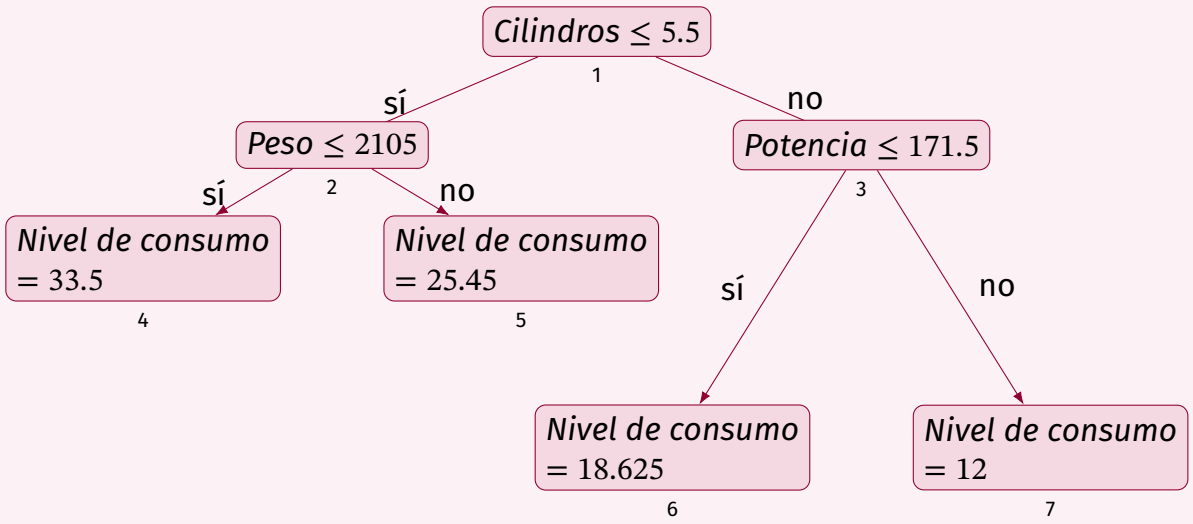
$$\text{RMSE} = \sqrt{\frac{1}{|\mathcal{D}|}\sum_{(\boldsymbol{x},y)\in\mathcal{D}}(\hat{y}-y)^2} \qquad \text{donde } \hat{y} = f(\boldsymbol{x})$$

- El coeficiente de determinación R^2 compara el error cuadrático medio con la varianza de los valores correctos, que sería el error que cometería un modelo que siempre proporcionara como salida la media de los valores correctos.

$$R^2 = 1 - \frac{\text{MSE}}{\text{Var}(\mathcal{D})} \qquad \text{donde } \text{Var}(\mathcal{D}) = \frac{1}{|\mathcal{D}|}\sum_{(\boldsymbol{x},y)\in\mathcal{D}}(y-\bar{y})^2, \quad \bar{y} = \frac{1}{|\mathcal{D}|}\sum_{(\boldsymbol{x},y)\in\mathcal{D}} y$$

El coeficiente de determinación toma valores en el intervalo $(-\infty, 1)$, donde el valor 1 correspondería a un modelo que hiciera predicciones correctas para todos los ejemplos y el valor 0 a un modelo que siempre predijera la media de los valores correctos. Un modelo con un coeficiente de determinación negativo tendría un rendimiento peor que predecir siempre esa media.

Ejemplo: CART que predice el consumo de un coche



Este CART asocia a cada ejemplo el siguiente nivel de consumo:

- A E_1 un nivel de consumo de 18.625.
- A E_2 un nivel de consumo de 25.45.
- A E_3 un nivel de consumo de 25.45.
- A E_4 un nivel de consumo de 18.625.
- A E_5 un nivel de consumo de 25.45.
- A E_6 un nivel de consumo de 33.5.

- A E_7 un nivel de consumo de 18.625.
- A E_8 un nivel de consumo de 25.45.
- A E_9 un nivel de consumo de 18.625.
- A E_{10} un nivel de consumo de 12.

Se tienen entonces las siguientes medidas del rendimiento del modelo:

$$\begin{aligned}
\mathrm{MAE} &= \frac{1}{10}\big(|18.625-18| + |25.45-22| + |25.45-25.4| + |18.625-20| + |25.45-26| + \\
&\qquad |33.5-33.5| + |18.625-17.5| + |25.45-28.4| + |18.625-19| + |12-12|\big) \\
&= 1.05 \\
\mathrm{MSE} &= \frac{1}{10}\big((18.625-18)^2 + (25.45-22)^2 + (25.45-25.4)^2 + (18.625-20)^2 + \\
&\qquad (25.45-26)^2 + (33.5-33.5)^2 + (18.625-17.5)^2 + (25.45-28.4) + \\
&\qquad (18.625-19)^2 + (12-12)^2\big) \\
&= 2.45975 \\
\mathrm{RMSE} &= \sqrt{\mathrm{MSE}} = 1.568359 \\
R^2 &= 1 - \frac{\mathrm{MSE}}{\mathrm{Var}(\{E_1, \ldots, E_{10}\})} = 1 - \frac{2.45975}{38.85511} = 0.9366943
\end{aligned}$$

Aunque en los ejemplos anteriores se ha calculado el rendimiento del modelo sobre el conjunto de entrenamiento, lo que realmente nos interesa es calcular su rendimiento sobre ejemplos nuevos. Este último es lo que se conoce como capacidad de generalización del modelo: su aptitud para proporcionar la respuesta adecuada para ejemplos que no ha conocido previamente. Un modelo que responda correctamente para los ejemplos de entrenamiento, pero incorrectamente para ejemplos nuevos, diremos que se ha sobreajustado al conjunto de entrenamiento.

A la hora de calcular la capacidad de generalización de un modelo de aprendizaje automático surgen dos dificultades: la primera de ellas es que se desconoce la distribución del conjunto de posibles nuevos ejemplos que, además, habitualmente será infinito; la segunda es que, por definición, de un ejemplo nuevo no se conoce la respuesta correcta, algo que se necesita para calcular la medida de rendimiento seleccionada.

Una primera metodología que permite abordar estas dificultades es realizar lo que se conoce como validación por retención (*holdout validation*, en inglés). Esta consiste en dividir el conjunto de ejemplos conocidos en dos subconjuntos: un subconjunto de entrenamiento a partir del cual se construirá el modelo y un subconjunto de prueba del que sí que se podrá calcular el rendimiento del modelo, a través de la medida elegida, ya que de sus ejemplos sí que se conoce la respuesta correcta (esto explica el nombre de la metodología, ya que los ejemplos de este subconjunto de prueba quedan «retenidos» hasta que se construye el modelo).

La separación del conjunto de ejemplos en los subconjuntos de entrenamiento y prueba se realiza, en general, de manera aleatoria (siendo habitual reservar para este último entre un $20\,\%$ y un $30\,\%$ de los ejemplos). Además, las proporciones de ejemplos que

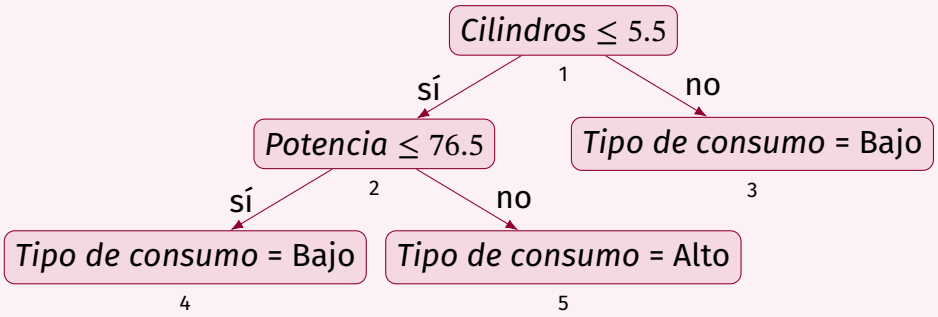
pertenecen a las distintas clases en el conjunto inicial deberían mantenerse en los subconjuntos, para lo que es conveniente que el reparto aleatorio se realice mediante un procedimiento de muestreo estratificado.

Ejemplo: Capacidad de generalización de un CART predictor del tipo de consumo de un coche por validación por retención

	Origen	*Cilindros*	*Potencia*	*Peso*	*Aceleración*	*Tipo de consumo*
E_1	2	6	105	3613	16.5	Bajo
E_2	0	4	76	2511	18.0	Bajo
E_3	0	5	77	3530	20.1	Alto
E_4	1	6	122	2807	13.5	Bajo
E_5	0	4	90	2265	15.5	Alto
E_6	1	4	70	1945	16.8	Alto
E_7	2	8	145	3880	12.5	Bajo
E_8	2	4	90	2670	16.0	Alto
E_9	2	6	90	3211	17.0	Bajo
E_{10}	2	8	198	4952	11.5	Bajo

Mediante muestreo estratificado se obtiene $\{E_1, E_2, E_3, E_4, E_5, E_7\}$ como conjunto de entrenamiento y $\{E_6, E_8, E_9, E_{10}\}$ como conjunto de prueba. Nótese que en ambos se mantiene la proporción (aproximada) de $60\,\%$ de los coches de consumo bajo y $40\,\%$ de los coches de consumo alto.

Entrenando un árbol de decisión a partir de los seis ejemplos de entrenamiento se obtiene el siguiente CART:



A partir de las predicciones que este árbol de decisión realiza de cada uno de los cuatro ejemplos de prueba se obtiene, considerando consumo bajo como la clase positiva, la siguiente matriz de confusión:

		Clase predicha			
		Consumo bajo		*Consumo alto*	
Clase correcta	*Consumo bajo*	2	(*VP*)	0	(*FN*)
	Consumo alto	1	(*FP*)	1	(*VN*)

Cuando el conjunto de ejemplos de que se dispone es pequeño, el método de vali-

dación por retención no es adecuado. Un método que se suele utilizar en su lugar es el de validación cruzada con k pliegues (k-*fold cross validation*, en inglés). Este consiste en subdividir el conjunto de ejemplos en k subconjuntos (que se denominan pliegues, de ahí el nombre del método). Esta subdivisión se realiza de manera aleatoria y es conveniente utilizar muestreo estratificado para ello. Para cada uno de estos pliegues se realiza lo siguiente:

- Se separan por un lado los ejemplos que no pertenecen y los que sí pertenecen al pliegue.
- Se entrena un modelo a partir de los ejemplos que no pertenecen al pliegue.
- Se calcula el rendimiento del modelo sobre los ejemplos que sí pertenecen al pliegue.

Es decir, se realizan k procesos de validación por retención, cada uno de ellos utilizando un pliegue distinto como subconjunto de prueba. Finalmente, el método devuelve la media de las k estimaciones obtenidas.

Ejemplo: Capacidad de generalización de un CART predictor del tipo de consumo de un coche por validación cruzada

	Origen	*Cilindros*	*Potencia*	*Peso*	*Aceleración*	*Tipo de consumo*
E_1	2	6	105	3613	16.5	Bajo
E_2	0	4	76	2511	18.0	Bajo
E_3	0	5	77	3530	20.1	Alto
E_4	1	6	122	2807	13.5	Bajo
E_5	0	4	90	2265	15.5	Alto
E_6	1	4	70	1945	16.8	Alto
E_7	2	8	145	3880	12.5	Bajo
E_8	2	4	90	2670	16.0	Alto
E_9	2	6	90	3211	17.0	Bajo
E_{10}	2	8	198	4952	11.5	Bajo

Consideramos como pliegues los subconjuntos de ejemplos

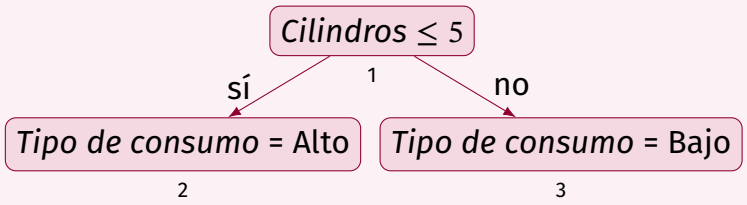
$$\{E_1, E_2, E_3\} \qquad \{E_4, E_5, E_6, E_7\} \qquad \{E_8, E_9, E_{10}\}$$

y como medida de generalización la tasa de aciertos.

- Cuando se selecciona el primer pliegue como conjunto de prueba, el conjunto de entrenamiento es la unión del segundo y tercer pliegues:

$$\{E_4, E_5, E_6, E_7, E_8, E_9, E_{10}\}$$

Entrenando un árbol de decisión a partir de estos ejemplos se obtiene el CART

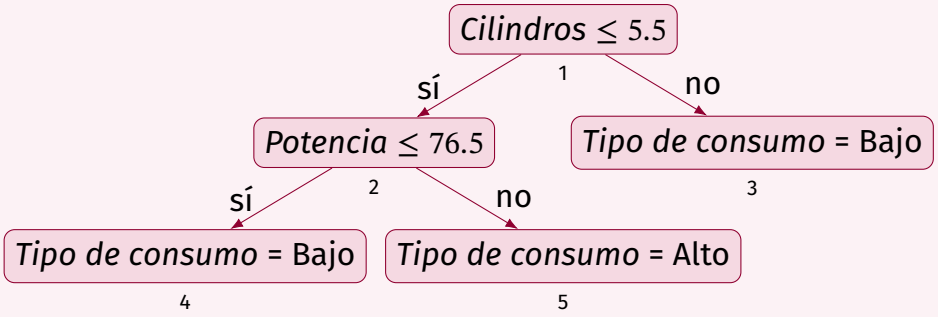


que predice que los coches E_1, E_2 y E_3 del primer pliegue son de consumo bajo, alto y alto, respectivamente. La tasa de acierto es, por tanto, de $2/3$.

- Cuando se selecciona el segundo pliegue como conjunto de prueba, el conjunto de entrenamiento es la unión del primer y tercer pliegues:

$$\{E_1, E_2, E_3, E_8, E_9, E_{10}\}$$

Entrenando un árbol de decisión a partir de estos ejemplos se obtiene el CART

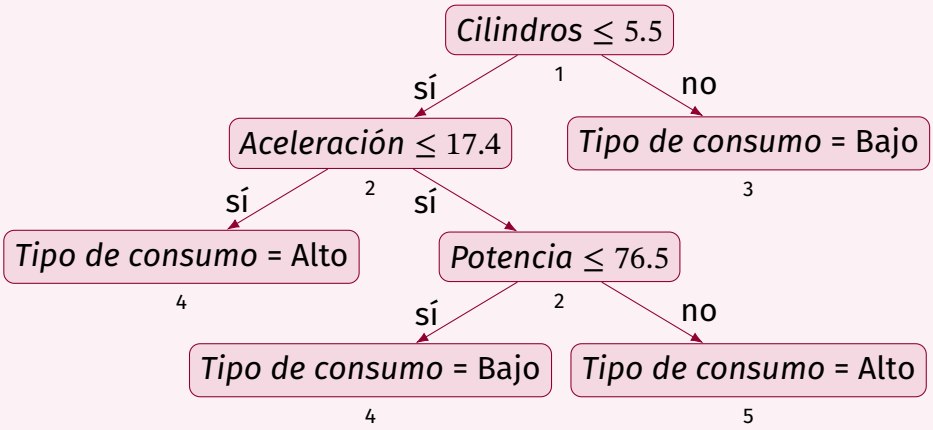


que predice que los coches E_4, E_5, E_6 y E_7 del segundo pliegue son de consumo bajo, alto, bajo y bajo, respectivamente. La tasa de acierto es, por tanto, de $3/4$.

- Cuando se selecciona el tercer pliegue como conjunto de prueba, el conjunto de entrenamiento es la unión del primer y segundo pliegues:

$$\{E_1, E_2, E_3, E_4, E_5, E_6, E_7\}$$

Entrenando un árbol de decisión a partir de estos ejemplos se obtiene el CART



que predice que los coches E_8, E_9 y E_{10} del tercer pliegue son de consumo alto, bajo y bajo, respectivamente. La tasa de acierto es, por tanto, de $3/3$.

La tasa de acierto promedio proporcionada por el método de validación cruzada es, entonces,

$$\left(\frac{2/3 + 3/4 + 3/3}{3}\right) = \frac{29}{36} \cong 0.81$$

Nótese que el método de validación por retención realiza una estimación a posteriori de la capacidad de generalización de un modelo, es decir, una vez que se ha construido este. El método de validación cruzada realiza una estimación a priori de esa capacidad de generalización. Posteriormente se utilizan todos los ejemplos disponibles para construir el modelo, cuya capacidad de generalización será aproximadamente la estimada por el método.

Normalmente no se conoce cuál es el modelo que se ajusta mejor al problema que se pretende resolver, por lo que debe realizarse un proceso de selección de modelos. La metodología que se sigue habitualmente es: considerar un conjunto de modelos candidatos, pudiendo ser modelos de distinto tipo o modelos del mismo tipo diferenciados por distintos valores de sus hiperparámetros; para cada uno de ellos estimar las características relevantes, entre ellas su capacidad de generalización, aunque quizás deban tenerse en cuenta otras consideraciones, como pueden ser el coste computacional de entrenar el modelo, su capacidad explicativa, etc.; seleccionar uno de los modelos, en base a las características estimadas de los mismos.

Cuando lo que se busca es seleccionar los valores más adecuados de los hiperparámetros de un modelo de un cierto tipo, una manera sencilla de hacerlo es efectuar una búsqueda en rejilla: primero se fijan para cada uno de los hiperparámetros los valores a considerar; después, para cada combinación de esos valores se estiman las características del modelo correspondiente; una vez seleccionado un valor para cada hiperparámetro, se entrena el modelo correspondiente con el conjunto completo de ejemplos.

Ejemplo: Búsqueda en rejilla de los hiperparámetros de un modelo kNN predictor del tipo de consumo de un coche

Una búsqueda en rejilla de los mejores valores del número de vecinos y de la métrica a utilizar en un modelo kNN que prediga el tipo de consumo de un coche puede consistir en los siguiente:

1. Considerar la misma partición del conjunto de ejemplos en subconjunto de entrenamiento y de prueba realizada en el ejemplo de la página 30.
2. Llevar a cabo una normalización mín-máx de los valores de los atributos de los ejemplos de entrenamiento.
3. Para cada combinación entre los valores $1, 3, 5$ para el número de vecinos y los valores distancia Manhattan y distancia euclídea para la métrica de cercanía, entrenar un modelo kNN y determinar su tasa de acierto sobre los ejemplos

de prueba (normalizando previamente a estos con la normalización aprendida en el punto 2).

Las tasas de acierto obtenidas para cada combinación son

		Número de vecinos		
		1	3	5
Métrica	Distancia Manhattan	3/4	2/4	2/4
	Distancia euclídea	3/4	2/4	2/4

Por lo tanto, en este caso a la hora de seleccionar un modelo kNN de entre todos los seleccionados es recomendable establecer $k = 1$, siendo indiferente usar la distancia Manhattan o la distancia euclídea.

En general no es necesario conseguir que el modelo final seleccionado tenga una capacidad de generalización máxima, sino que basta que sea suficientemente bueno, en relación a los requisitos y restricciones del problema objetivo. Es habitual también seguir el principio de la navaja de Occam, que establece que entre modelos con poder predictivo similar debe elegirse el más simple (esto último suele significar el que tenga menor número de parámetros).

También es importante tener presente que si, una vez seleccionado y construido un modelo, se pretende comunicar sus características (como por ejemplo su capacidad de generalización), estas deben estimarse a partir de un subconjunto de ejemplos nuevos reservado previamente a la ejecución del proceso de selección de modelos.