

ANÁLISIS PREDICTIVO: MINERÍA DE DATOS

La **minería de datos** (*data mining*) puede definirse como el conjunto de metodologías, procesos y tecnologías para el descubrimiento no trivial de información relevante, normalmente subyacente en grandes volúmenes de datos, y su consiguiente aplicación e integración dentro de las operaciones del negocio con el fin de mejorar el rendimiento y soportar la toma de decisiones.

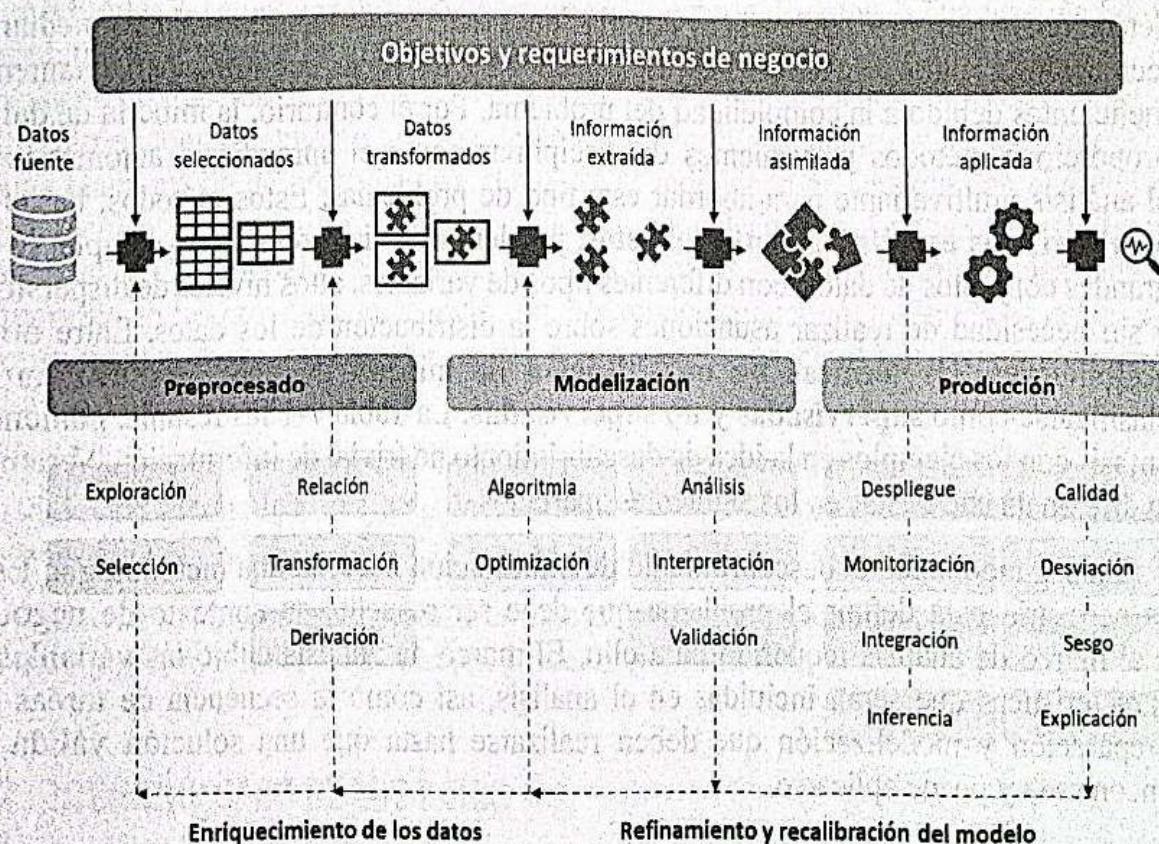


Figura 7-1. Visión general de un proceso de minería de datos.

Dentro de los sistemas de *Big Data*, la minería de datos comparte con el análisis multidimensional el grueso de la capa de acceso a la información por parte de los usuarios de negocio. Como vimos en el capítulo anterior, el análisis multidimensional se centra en la explotación descriptiva de la información. Aunque la minería de datos tiene también una importante capacidad de descripción, precisamente donde no llegan los métodos basados en consultas, podemos decir que donde aporta un mayor valor diferenciador es en el aspecto predictivo. Adicionalmente, mientras el análisis multidimensional proporciona básicamente un acceso de lectura¹⁷¹, la minería de datos genera nueva información: no solo los modelos que produce son activos que hay que gobernar, sino que el resultado de la aplicación de estos son nuevos datos que deben ser integrados y explotados.

En este capítulo veremos los fundamentos de un proceso de minería de datos, prestando especial atención a las tareas de modelización y a los distintos paradigmas de **aprendizaje automático** (*machine learning*) en los que se basan. Veremos también distintos escenarios de puesta en producción de los modelos, dejando para más adelante la gestión de su ciclo de vida.

7.1 MOTIVACIÓN Y OBJETIVOS

La definición que acabamos de dar de la minería de datos consiste en tres partes que deben ser matizadas adecuadamente.

En primer lugar, el descubrimiento no trivial de información relevante implica la detección de patrones, tendencias y correlaciones que no pueden ser reveladas mediante técnicas de consulta convencionales; estas pueden ser, de hecho, inapropiadas, o altamente inefficientes debido a la complejidad del problema. Por el contrario, la minería de datos proporciona métodos provenientes de disciplinas como el aprendizaje automático y el análisis multivariante para abordar este tipo de problemas. Estos métodos, basados en **algoritmos estadísticamente robustos**, pueden modelizar relaciones complejas en grandes conjuntos de datos, con diferentes tipos de variables, altos niveles de dispersión, y sin necesidad de realizar asunciones sobre la distribución de los datos. Entre otras posibilidades, las **técnicas de modelización** de minería de datos acostumbran a clasificarse como **supervisadas y no supervisadas**. La Tabla 7-1 las resume, poniendo énfasis con los ejemplos en la idea de descubrimiento no trivial de información. Veremos la diferencia entre estas en los siguientes apartados.

En segundo lugar, el descubrimiento de información necesita una metodología. Esta es necesaria para definir el problema que debe ser resuelto, su contexto de negocio y el marco de análisis requerido para ello. El marco de análisis cubre las variables o características que serán incluidas en el análisis, así como la secuencia de tareas de preparación y modelización que deben realizarse hasta que una solución válida es encontrada y puede aplicarse.

171 El análisis multidimensional también puede generar nuevos datos como consecuencia de procesos de simulación de escenarios.

Clase	Tipo	Ejemplos de aplicación
Supervisadas	Clasificación	<ul style="list-style-type: none"> • Propensión al abandono de clientes • Detección de fraude en tarjetas • Categorización de documentos
	Regresión	<ul style="list-style-type: none"> • Cálculo del potencial del cliente • Correlación entre el clima y las ventas • Estimación del volumen de fabricación
	Análisis de series temporales	<ul style="list-style-type: none"> • Estacionalidad en las ventas • Previsión de la demanda • Seguimiento médico
No supervisadas	Segmentación	<ul style="list-style-type: none"> • Identificación de grupos de clientes • Detección de anomalías en llamadas • <i>Marketing personalizado</i>
	Detección de reglas de asociación	<ul style="list-style-type: none"> • Venta cruzada de productos • Ubicación de productos en los lineales • Personalización de ofertas
	Descubrimiento de patrones secuenciales	<ul style="list-style-type: none"> • Diseño de promociones • Análisis de navegación en páginas web • Sendas de desvinculación de clientes

Tabla 7-1. Clasificación de técnicas de modelización en minería de datos.

La Figura 7-1 ilustra el proceso genérico de la minería de datos. A una fase inicial de definición de los requerimientos y objetivos de negocio le sigue un ciclo de desarrollo y puesta en producción de modelos. El desarrollo se debe basar en conjuntos de datos que capturen el comportamiento que sea deseado modelizar. Estos datos deben ser sometidos a una etapa previa de preprocesado, donde se acondicionan de acuerdo con el objetivo y las técnicas de modelización que se aplicaran para alcanzarlo. Una vez desarrollados y validados, los modelos se aplicarán a nuevos datos, generando inferencias que se integrarán en los procesos de negocio. En este modo productivo, los modelos deben ser monitorizados, y recalibrados en cuanto se detecten desviaciones del comportamiento inicial.

El foco en el negocio, la aproximación metodológica y la implementación basada en servicios hacen que la minería de datos sea una disciplina troncal dentro de la **analítica de negocio**, y no solo un cajón de técnicas matemáticas y algoritmos.

Vamos a ver a continuación las tres etapas en las que se basa un proceso de minería de datos: preprocesado, modelización y puesta en producción.

7.2. PREPROCESADO DE LOS DATOS

De forma equivalente a como lo planteábamos en el análisis descriptivo, el elemento base de cualquier tarea de modelización es un **conjunto de datos**, entendido este como

una colección de **observaciones**¹⁷². A su vez, cada observación está descrita por una serie de **atributos**¹⁷³. Como ya sabemos, estos atributos pueden ser de un tipo primitivo o de una índole más compleja, como imágenes, audio, texto, series, colecciones, etc.

Dependiendo de lo que queramos modelizar, las observaciones se corresponderán con clientes, cestas de la compra, llamadas telefónicas, puntos de venta, fotografías en una cadena de montaje, transcripciones de conversaciones en un centro de atención, etc. Por su parte, los atributos acostumbran a ser consistentes a lo largo de las observaciones del conjunto, especialmente en tipología, pero también en número¹⁷⁴. Este último puede oscilar entre unas pocas unidades y varios cientos o miles. Por ejemplo, en el caso de querer desarrollar un sistema que sea capaz de identificar huevos rotos en una cadena de envasado, necesitaremos un conjunto representativo de fotografías que identifiquen tanto huevos enteros como partidos, en diferentes ángulos. Con este conjunto entrenaremos un modelo de clasificación que sea capaz de discernir entre ambos; una vez entrenado, lo podremos usar para caracterizar los huevos en tiempo real, retirándolos en caso de rotura. En este caso nuestra observación estará compuesta por no más de tres atributos: un identificador de la fotografía, la imagen de esta y una etiqueta que indica si el huevo está roto o no. En el otro extremo nos podemos encontrar con agregados de clientes para modelos de segmentación, donde podemos llegar a combinar multitud de datos sociodemográficos y transaccionales.

Esta **etapa de preprocessado**¹⁷⁵ consiste en preparar los datos para las tareas de modelización posteriores. Se compone de una serie de fases de transformación cuyo objetivo es publicar, en el formato adecuado, una selección de observaciones tratadas y cuya calidad podamos asegurar. Si los datos necesarios para ello se derivan de un entorno de información gobernado, entonces esta etapa se limitará a una serie de operaciones de acondicionamiento final; de lo contrario, habrá que implementar una serie de procesos ETL *ad hoc* para cada modelización. La Tabla 7-2 resume las operaciones más habituales en esta etapa, que coinciden en gran medida con las que planteamos en la Tabla 4-2 cuando hablamos de las transformaciones en los procesos ETL. Aunque todas estas fases tienen su importancia, la de filtrado y compresión es especialmente relevante. Su principal motivación, aunque no la única, es una reducción en el tamaño de los datos de cara a su posterior modelización. Como acabamos de comentar, los conjuntos de datos que tendremos que gestionar pueden llegar a ser enormemente grandes. Estos tamaños se pueden alcanzar tanto vertical como horizontalmente. Es decir, los conjuntos de datos pueden crecer en base al número de observaciones y también de atributos. El impacto de cada uno de estos ejes es diferente, y la forma de abordarlo también.

172 También denominadas objetos, registros o entidades (*entities*).

173 También denominados variables, campos o características (*features*).

174 Desde este punto de vista, podríamos decir que en minería de datos trabajamos con datos estructurados, estando la estructura en la observación.

175 En inglés esta etapa recibe muchos nombres, como *data preprocessing*, *data wrangling*, *data mungling* o, más recientemente, *feature engineering*.

Las técnicas de **muestreo** (*sampling*) permiten disminuir el número de observaciones mediante la extracción de un subconjunto representativo de estas. Es decir, pasamos de un conjunto a otro más pequeño donde, idealmente, las propiedades y la distribución de los valores de los atributos se mantiene. Para extraer una muestra tenemos que establecer su tamaño y elegir una técnica de muestreo, habitualmente de carácter aleatorio. El **muestreo estratificado** se emplea cuando es necesario asegurar que la muestra mantendrá la misma proporción de ocurrencias respecto a uno o varios atributos de interés. Por ejemplo, nos puede interesar construir un modelo que clasifique a pacientes según la variante de la enfermedad que padecen, resultando que algunas de estas son muy poco frecuentes. Con un muestreo simple, no estratificado, existiría el riesgo de dejar fuera a estos pacientes, perdiendo la representación de sus variantes.

Fase	Operaciones
Selección Acceso a los sistemas origen y extracción de datos	<ul style="list-style-type: none"> • Filtrado inicial de observaciones y atributos • Enmascaramiento de datos sensibles • Formateo inicial
Exploración Estudio de las características de los datos	<ul style="list-style-type: none"> • Análisis estadístico • Representación gráfica • Validación de hipótesis • Auditoría de datos
Limpieza Detección y corrección de problemas de calidad	<ul style="list-style-type: none"> • Gestión de valores erróneos • Imputación de valores omitidos • Eliminación de duplicados • Estandarización de valores
Agregación Combinación y consolidación de observaciones	<ul style="list-style-type: none"> • Cálculo de valores promedio, máximo y mínimo • Consolidación de observaciones • Unificación de valores
Filtrado y compresión Disminución del número de observaciones y/o atributos	<ul style="list-style-type: none"> • Muestreo de observaciones • Reducción de atributos • Eliminación de atributos irrelevantes • Ponderación de atributos
Enriquecimiento Creación y derivación de nuevos atributos	<ul style="list-style-type: none"> • Cálculo de métricas • Cambio del espacio de representación • Extracción de características • División de atributos
Conversión Adaptación de atributos a valores categóricos	<ul style="list-style-type: none"> • Transformación a valores categóricos • Reducción de valores • Transformación a valores binarios • Codificación
Transformación Cambio en todos los valores de un atributo	<ul style="list-style-type: none"> • Aplicación de funciones • Normalización • Estandarización • Cambio de escala
Publicación Entrega del conjunto de datos para su modelización	<ul style="list-style-type: none"> • Formateo final • Catalogación • Gestión de permisos y autorizaciones

Tabla 7-2. Operaciones habituales en el preprocesado de los datos.

Respecto a la reducción del número de atributos, la necesidad viene dada por un conjunto de fenómenos contraintuitivos, comúnmente denominados **maldición de la dimensión** (*curse of dimensionality*). Estos giran alrededor de la distorsión que produce el aumento del número de atributos en la modelización de una colección de observaciones. A medida que aumentamos el matiz de las observaciones añadiendo más atributos, aumentamos también el grado de dispersión de estas, dificultando la detección de patrones y tendencias. Esto acaba significando que, para construir modelos fiables, haya que aumentar a su vez el número de observaciones necesarias. Adicionalmente, cuantos más atributos mayor es el riesgo de **multicolinealidad**. Esta se da cuando dos o más variables presentan una alta relación lineal entre sí, provocando redundancia en la información aportada. La multicolinealidad acaba sobrevalorando unos atributos frente a otros, aumentando el riesgo de sobreajuste en los modelos, algo que veremos en los siguientes apartados.

Para reducir el número de atributos tenemos dos opciones. Una de ellas consiste en seleccionar solo un subconjunto de los existentes, empleando técnicas para identificar tanto atributos redundantes como irrelevantes. La otra pasa por construir un conjunto de nuevas variables a partir de las iniciales, que sea inferior en número, pero que conserve la mayoría de la variabilidad original, eliminando al mismo tiempo la multicolinealidad. Estas nuevas variables sustituirán a las primeras en las tareas posteriores de visualización y modelización. Entre las técnicas más empleadas para la generación de estas variables derivadas destaca el **análisis de componentes principales** (PCA, *Principal Components Analysis*).

7.3 MODELIZACIÓN DE LOS DATOS

En el contexto de la minería de datos, cuando hablamos de **modelizar** nos estamos refiriendo a la utilización de un conjunto de datos para construir una referencia con un fin principal: la inferencia. Es decir, detectar y sistematizar un patrón que sea reproducible, y que pueda ser utilizado para sacar conclusiones. Cuando estas conclusiones pueden explicar hechos que ya han sucedido hablamos de **modelos descriptivos**; cuando pueden anticipar cosas que todavía no han ocurrido hablamos de **modelos predictivos**. Los modelos que veremos exhiben ambas capacidades, aunque unos tienen más foco en una que en la otra¹⁷⁶.

¹⁷⁶ Este foco llega a ser todavía más acentuado cuando descendemos a nivel de algoritmo. Por ejemplo, las técnicas basadas en redes neuronales tienen fama (cada vez menos) de ser como una caja negra: pueden tener una gran capacidad predictiva pero su funcionamiento es muy opaco desde el punto de vista de la interpretación de los resultados.

Por ejemplo, si disponemos del historial clínico de una población de pacientes con un determinado tipo de cáncer, nos interesaría obtener el patrón por el cual un tumor se acaba agravando en unos casos pero no en otros. Este patrón, extraído quizás en forma de reglas de asignación y condiciones que cumplen unos pacientes frente a otros, nos servirá para entender y describir las causas del agravamiento, pero también para pronosticar la evolución de los enfermos.

Así definidos, los modelos están basados en un **algoritmo**: un conjunto definido de operaciones reguladas por una serie de parámetros, que hay que seguir hasta llegar a una solución que, o bien es aceptable según un criterio predefinido, o bien no es mejorable. Podemos encontrar desde algoritmos de regresión lineal, cuyo funcionamiento está controlado por un único parámetro, hasta redes neuronales artificiales, donde la cifra es del orden de decenas o miles de millones.

Dentro del ciclo de vida de un modelo hay dos etapas fundamentales:

- ▶ **Aprendizaje.** Sobre el conjunto de datos transformados, como hemos visto en el apartado anterior, definimos el modelo. Partiendo de un algoritmo concreto, el aprendizaje consiste en la búsqueda y determinación de los parámetros óptimos de este, de acuerdo con una métrica de rendimiento. Esta métrica evalúa cómo de bien representa el modelo a los datos de partida, permitiendo además la comparación de distintos algoritmos con el fin de seleccionar el que proporciona el mejor resultado. Como veremos a continuación, las diferentes fases en las que se puede dividir esta etapa dan lugar a los diferentes tipos de aprendizaje. En cualquier caso, el aprendizaje es un proceso que implica sucesivas iteraciones del algoritmo sobre el conjunto de datos en las que se van afinando los distintos parámetros.
- ▶ **Inferencia.** Una vez que el modelo está construido, podemos aplicarlo a nuevos datos con el fin de obtener una predicción en forma de respuesta. El contenido de la respuesta dependerá del tipo de modelo y algoritmo. Mientras el aprendizaje constituye el desarrollo del modelo, la inferencia se considera la puesta en producción de este, implicando tareas de despliegue e integración con los sistemas de negocio, monitorización, mantenimiento, etc.

A continuación plantearemos los diferentes modos de aprendizaje, los tipos de modelos en cada uno de ellos, así como los algoritmos más comunes.

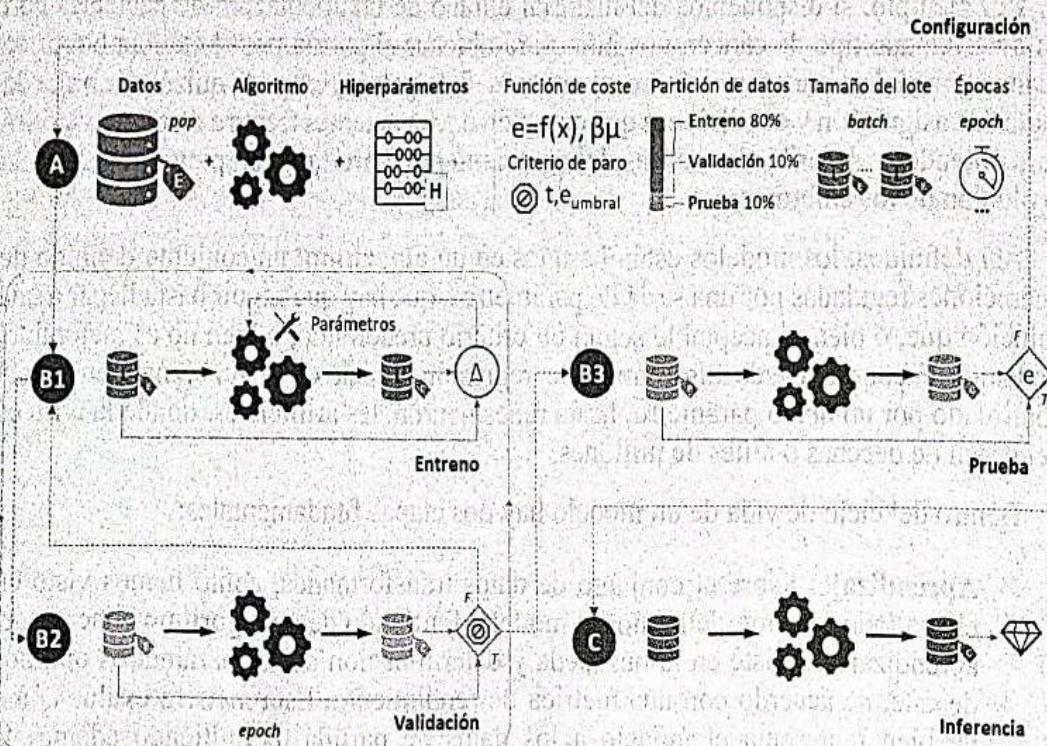


Figura 7-2. Fases en un proceso de aprendizaje supervisado.

7.3.1 Aprendizaje supervisado

En el **aprendizaje supervisado** (*supervised learning*), el desarrollo del modelo se lleva a cabo mediante un mecanismo de inspección que permite evaluar su calidad en dos aspectos: cómo de bien detecta el patrón que subyace en los datos de partida y, sobre todo, qué capacidad tiene de generalizar ese patrón sobre datos nuevos, no empleados durante el afinamiento de los parámetros. Está claro que para que se dé la segunda condición se debe dar la primera; sin embargo, podemos tener un modelo que «aprenda» excesivamente bien los datos iniciales, pero pierda esa capacidad de generalización cuando se le presentan datos nuevos. Un modelo que tiene este último comportamiento se dice que está **sobreajustado** (*overfitted*), mientras que aquel que ni siquiera tiene la capacidad de detectar patrón alguno en los datos iniciales está **subajustado** (*underfitted*).

Este mecanismo de inspección o supervisión será posible porque podemos disponer de datos etiquetados con la respuesta correcta que deberá dar el modelo. Dicho en otras palabras, el conjunto de datos de partida, denominado **conjunto de aprendizaje**, contiene para cada observación un atributo que es la variable objetivo que queremos explicar en función de las otras. Siguiendo con el ejemplo anterior, el conjunto de entrenamiento sería el conjunto de historiales de pacientes de los que ya sabemos cómo cursó la enfermedad. Por lo tanto, cada historial tiene una etiqueta que indica si el tumor evolucionó favorablemente o no, o bien en qué grado lo ha hecho. Desde el momento en que tenemos el valor esperado, podemos irlo comparando con el calculado por el

modelo durante el proceso de aprendizaje y utilizar la discrepancia entre ambos, medida mediante una **función de coste**, para guiar dicho proceso e ir adaptando los parámetros del modelo.

La Figura 7-2 esquematiza las distintas fases que componen un proceso de aprendizaje supervisado. Son las siguientes:

- ▶ **Configuración (A).** Partimos de un conjunto de datos acondicionados y etiquetados con el valor de la variable objetivo que queremos estimar y un tipo de algoritmo seleccionado de acuerdo con la naturaleza del problema. Antes de comenzar el aprendizaje hay una serie de parámetros que se deben configurar, y cuyo valor no puede ser derivado a partir de los datos iniciales. Son los denominados **hiperparámetros del modelo**¹⁷⁷. Algunos de estos son específicos del algoritmo empleado, mientras que otros son más genéricos. Entre los segundos están los porcentajes en los que vamos a dividir de forma aleatoria los datos de entrada en tres subconjuntos (entrenamiento, validación y prueba), la función de coste que vamos a emplear, el número de iteraciones sobre los datos (número de épocas) o cada cuantas observaciones vamos a adaptar los parámetros del modelo (tamaño del lote)¹⁷⁸.
- ▶ **Entrenamiento (B1).** Aquí da comienzo el proceso de aprendizaje como tal. Durante esta fase se van afinando los parámetros del modelo, empleando para ello uno de los subconjuntos en los que dividimos el conjunto de aprendizaje. Las observaciones de esta partición, que acostumbra a representar entre el 70% y el 80% de los datos, son procesadas en lotes por el algoritmo. Después de cada lote se calcula el error según la función de coste y se adaptan los parámetros del modelo de acuerdo con la magnitud y dirección de este, intentando minimizarlo. Cuando todas las observaciones han sido procesadas, lo que constituye una **época**, pasamos a la siguiente fase.
- ▶ **Validación (B2).** A una fase de entrenamiento le sigue otra de validación y así otra vez hasta que se cumple un criterio de paro prefijado a modo de hiperparámetro. Se trata ahora de coger el **subconjunto de validación**, que puede representar el 10% de los datos iniciales, y evaluarlo con la función de coste. Si el error es aceptable, estando por debajo de un valor umbral que forma parte del **criterio de paro**, el ciclo se detiene y se pasa a la fase de prueba. De lo contrario, se vuelve a la fase de entrenamiento dando comienzo una nueva época. Para evitar que este

177 No confundir los hiperparámetros y los parámetros del modelo. Los primeros establecen en qué condiciones se va a realizar el procesos de aprendizaje y que restricciones de partida va a tener el modelo; los segundos son consecuencia del propio aprendizaje, dependiendo, por lo tanto, de los datos de partida y de los propios hiperparámetros. Dependiendo del tipo de modelo y algoritmo, nos podemos mover entre unos pocos y varios cientos o miles de hiperparámetros en un proceso de entrenamiento.

178 La selección del algoritmo y el valor que se le da a los hiperparámetros forma parte de la experimentación en un proceso de aprendizaje. Idealmente se debería plantear algún tipo de diseño factorial para evaluar el impacto de estos.

ciclo se repita indefinidamente en el caso en que la función de coste no converja por debajo del valor umbral, el criterio de paro incluye un número máximo de épocas por encima del cual el ciclo también se detendrá.

El motivo por el cual se dividen los datos iniciales, dando lugar a una fase de entrenamiento y otra de validación, es para evitar el fenómeno del sobreajuste que mencionábamos anteriormente¹⁷⁹. Al no ser empleado el subconjunto de validación para ajustar los parámetros del modelo, nos sirve como calibración de su ajuste, dándonos una buena idea de su capacidad de generalización. Ahora bien, en este ciclo que se va repitiendo entrenamos y validamos siempre con los mismos subconjuntos, lo que puede introducir un sesgo en el modelo que acabe produciendo también un sobreajuste¹⁸⁰. Es este el motivo por el cual disponemos de otro subconjunto de datos y de una fase más.

- **Prueba (B3).** Con el subconjunto que nos queda (normalmente el 10% restante) comprobamos, ahora de forma ciega, la calidad final del entrenamiento. Para ello podemos utilizar la misma función de coste que en la validación, aunque se acostumbra a usar alguna **métrica de rendimiento** que dependerá del tipo de modelo ajustado. La ventaja de estas métricas es que permiten la comparación de modelos construidos con distintos algoritmos. Si el valor que toma la métrica no es aceptable, entonces tendremos que repetir el experimento con una nueva configuración. Si lo es, entonces daríamos por finalizado el proceso de aprendizaje, dando por bueno el modelo.
- **Inferencia (C).** Aunque esta fase se considera fuera del aprendizaje, forma parte de la figura por completitud de esta. Resaltar aquí que los datos entran en el modelo sin etiquetar y salen finalmente etiquetados con un valor calculado.

En los siguientes apartados vamos a estudiar las técnicas de clasificación y predicción, ambas basadas en este proceso de aprendizaje supervisado.

7.3.1.1 CLASIFICACIÓN

Las técnicas de clasificación se encargan de **asignar objetos a categorías predefinidas**, etiquetándolos con una marca de clase. Esta marca es la variable objetivo en el aprendizaje, siendo un atributo discreto¹⁸¹ del objeto. El atributo puede ser nominal (sexo, raza de perro, color de los ojos, ...) u ordinal (riesgo crediticio bajo, medio o alto,

¹⁷⁹ Puede haber distintos motivos para que un modelo se sobreajuste a los datos de entrenamiento, y también diferentes mecanismos para evitarlo.

¹⁸⁰ Una forma de minimizar esto es haciendo variar las observaciones de cada subconjunto en cada ciclo mediante un mecanismo denominado **validación cruzada (cross-validation)**.

¹⁸¹ Como vimos en el capítulo anterior, un atributo discreto es aquel que puede tomar un número finito de posibles valores. En el caso de los modelos de clasificación es, además, un atributo categórico.

categoría oro, plata y bronce, ...), aunque a nivel matemático las técnicas no diferencian entre ambos, tratándolos como etiquetas carentes de ordenación. Si el atributo puede tomar solo dos posibles valores (verdadero/falso, abandono/no abandono, 1/0, ...) la clasificación es **binaria**.

La Tabla 7-3 resume los cuatro tipos de clasificaciones que podemos encontrar y algunos de los algoritmos más comunes. Muchos de estos disponen de adaptaciones que pueden cubrir cualquiera de los tipos de clasificación planteados, de ahí su repetición en la tabla.

Existen diferentes técnicas para la evaluación de un modelo de clasificación, pero la gran mayoría de ellas se basa en el conteo de los objetos clasificados correcta e incorrectamente, representados en forma de **matriz de confusión**. Sobre esta se definen distintas métricas de rendimiento, como la **exactitud (accuracy)**, la **precisión (precision)** o la **sensibilidad (recall)**.

Tipo	Algoritmos	Ejemplos
Binaria Asignación de un objeto a una de dos posibles clases.	<ul style="list-style-type: none"> • Regresión logística • Árboles de clasificación • Naïve Bayes • Máquinas de vector soporte (SVM, <i>Support Vector Machines</i>) 	<ul style="list-style-type: none"> • Predicción del abandono de clientes. • Clasificación de un SMS como deseado o no. • Calificación de una transacción como fraudulenta.
Multiclasa Asignación de un objeto a una de tres o más posibles clases.	<ul style="list-style-type: none"> • Bosque aleatorio (<i>Random forest</i>) • Potenciador del gradiente (<i>Gradient boosting</i>) • Redes neuronales artificiales (ANN, <i>Artificial Neural Networks</i>) 	<ul style="list-style-type: none"> • Análisis de sentimiento. • Identificación de tipos de plagas en imágenes de cultivos. • Predicción de la siguiente palabra en una traducción automática.
Multietiqueta Asignación de un objeto a varias clases, cada una con dos posibles valores.	<ul style="list-style-type: none"> • Árboles de clasificación • Bosque aleatorio • Potenciador del gradiente 	<ul style="list-style-type: none"> • Identificación de la presencia de varias personas en una imagen. • Diagnosis múltiple. • Clasificación de un texto en varias categorías
Multitarea Asignación simultánea de un objeto a varias clases, cada una con tres o más valores.		<ul style="list-style-type: none"> • Asignación de tipo y color a una imagen de una fruta. • Identificación de conceptos en artículos periodísticos. • Categorización de productos.

Tabla 7-3. Tipos de clasificación según la variable objetivo, algoritmos y ejemplos de aplicación.

Un caso particular en los problemas de clasificación se da cuando la **variable objetivo es asimétrica**. Es decir, no todas las posibles categorías aportan la misma información o son igual de relevantes para el negocio. Esto es más habitual en el caso de las clasificaciones binarias. Por ejemplo, no es lo mismo equivocarse calificando a un paciente como no enfermo cuando realmente lo está que al revés. En un sentido parecido, si comparamos clientes en base a si adquieren o no determinados productos en el supermercado, el hecho de comprarlos es más discriminante que el de no hacerlo, especialmente si el surtido es amplio.

Aunque la clasificación es una técnica básicamente predictiva, también puede tener un componente descriptivo importante, dependiendo del método empleado para construir el clasificador. En este sentido, los **árboles de clasificación** son algoritmos capaces de extraer reglas de asignación de los objetos a las categorías, proporcionando una cierta explicación de las diferencias que existen entre estas.

Básicamente, un árbol de clasificación se compone de una estructura jerárquica en forma de grafo acíclico dirigido, conteniendo tres tipos de nodos y una serie de conexiones entre ellos:

- ▀ **Nodo raíz.** Es único en cada árbol, siendo el punto de partida donde está toda la población de observaciones u objetos que se desea clasificar. No tiene, por lo tanto, ninguna conexión entrante y cero o más conexiones salientes¹⁸².
- ▀ **Nodos internos.** Dan forma al crecimiento del árbol. Cada nodo interno tiene solo una rama entrante y dos o más ramas salientes, dependiendo de si el árbol es binario o no.
- ▀ **Nodos terminales.** También denominados hojas, tienen solo una rama entrante y ninguna rama saliente. Son los extremos finales del árbol donde los objetos acaban finalmente categorizados.

Cada nodo terminal en un árbol representa una clase, pudiendo cada clase estar representada en más de un nodo terminal. El nodo raíz y cada uno de los nodos internos contiene una conjunción lógica de un subconjunto de las variables independientes que definen los objetos. La construcción del árbol se realiza partiendo del nodo raíz y determinando la condición que mejor divide a la población en dos o más subgrupos. Para seleccionar la condición se evalúa su capacidad discriminante de acuerdo con alguna métrica de homogeneidad de la variable objetivo en cada subgrupo. Esta operación se va repitiendo de forma recursiva, creándose nodos internos que hacen crecer al árbol tanto en anchura como en altura. Este crecimiento continúa hasta que en un nodo interno se alcanza una pureza mínima respecto a la variable objetivo, pasando a ser terminal.

182 Cero en el extraño caso en que no existiera ningún atributo que permitiera la división de la población de una forma lo suficientemente discriminante.

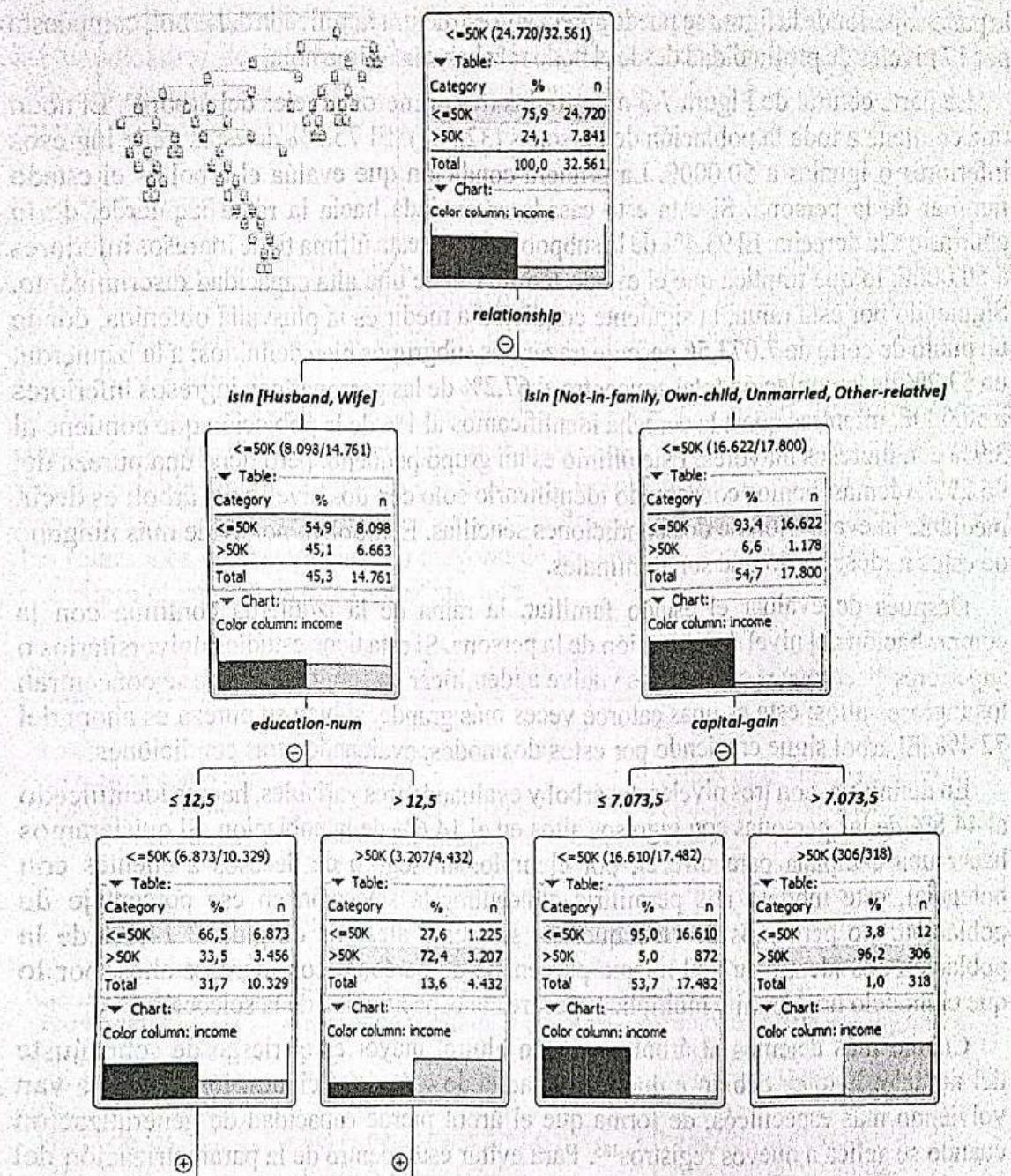


Figura 7-3. Ejemplo de árbol de clasificación binario construido con KNIME Analytics Platform para la categorización de los ingresos anuales, donde se pueden apreciar los tres primeros niveles. La figura enmarcada en la parte superior muestra la estructura completa del árbol.

La Figura 7-3 muestra un ejemplo de **árbol de clasificación binario**. Su objetivo es categorizar personas en bandas de ingresos anuales, teniendo en cuenta una serie de atributos (14 en total) como el país de nacimiento, la educación, el estado civil, la edad, el sexo, la plusvalía obtenida mediante la venta de activos, etc. El árbol es binario por partida doble, ya que no sólo la variable objetivo es binaria (se trata de discernir entre ingresos inferiores o iguales a 50.000€, y superiores), sino que se ha restringido su crecimiento, mediante la parametrización del algoritmo, a hacerse de forma dicotómica en cada nodo. En

la parte superior de la figura se puede apreciar la estructura ramificada del árbol, compuesta por 17 niveles de profundidad desde el nodo raíz hasta la última hoja.

La parte central de Figura 7-3 muestra los tres primeros niveles del árbol¹⁸³. El nodo raíz contiene a toda la población de personas (32.561). El 75,9% de estos tiene ingresos inferiores o iguales a 50.000€. La primera condición que evalúa el árbol es el estado familiar de la persona. Si esta está casada es enviada hacia la rama izquierda, de lo contrario a la derecha. El 93,4% de la subpoblación en esta última tiene ingresos inferiores a 50.000€, lo que implica que el estado familiar tiene una alta capacidad discriminante. Siguiendo por esta rama, la siguiente condición a medir es la plusvalía obtenida, donde un punto de corte de 7.073,5€ permite trazar dos subgrupos bien definidos: a la izquierda, un 53,7% de la población total concentra el 67,2% de las personas con ingresos inferiores a 50.000€, mientras que a la derecha identificamos al 1% de la población que contiene al 3,9% con ingresos mayores. Este último es un grupo pequeño, pero tiene una pureza del 96,2%. Además, hemos conseguido identificarlo solo con dos niveles del árbol; es decir, mediante la evaluación de dos condiciones sencillas. El árbol ya no divide más ninguno de estos nodos, por lo que son terminales.

Después de evaluar el estado familiar, la rama de la izquierda continúa con la comprobación del nivel de educación de la persona. Si esta tiene estudios universitarios o superiores¹⁸⁴, entonces el árbol nos vuelve a identificar un subgrupo donde se concentran los ingresos altos; este es unas catorce veces más grande, si bien su pureza es ahora del 72,4%. El árbol sigue creciendo por estos dos nodos, evaluando otras condiciones.

En definitiva, con tres niveles del árbol y evaluando tres variables, hemos identificado al 44,8% de las personas con ingresos altos en el 14,6% de la población. Si quisieramos hacer una campaña para ofrecer, por ejemplo, un seguro de decesos a clientes con potencial, este modelo nos permitiría concentrar la selección en ese porcentaje de población. No perdamos de vista que una selección aleatoria dirigida al 14,6% de la población nos identificaría el mismo porcentaje de personas con ingresos altos, por lo que el modelo nos permite multiplicar por tres la especificidad de la selección.

Cuanto más dejemos al árbol crecer en altura, mayor es el riesgo de sobreajuste del modelo. Esto es debido a que al ir añadiendo más condiciones, los nodos se van volviendo más específicos, de forma que el árbol pierde capacidad de generalización cuando se aplica a nuevos registros¹⁸⁵. Para evitar esto, dentro de la parametrización del algoritmo se establece algún criterio de paro para detener el crecimiento del árbol; por ejemplo, uno nodo se convierte en terminal cuando su pureza alcanza un determinado umbral. Adicionalmente, es frecuente colapsar determinadas ramas del árbol de forma manual, convirtiendo nodos en hojas, con el fin de evitar estas especificidades.

183 A la operación de colapsar uno o más nodos del árbol se le denomina poda.

184 En este conjunto de datos, el nivel de estudios se representa como un valor entero, ya que nos interesa darle carácter ordinal. El punto de corte en la condición de 12,5 no existe en los datos, pero es la forma en que el árbol diferencia entre valores menores o iguales a 12 y valores superiores (universitarios).

185 Si dejáramos crecer el árbol sin límite podríamos llegar a tener tantas hojas como personas. Es decir, el árbol sería capaz de distinguir cada caso particular a base de establecer condiciones.

En modo inferencia, la aplicación del árbol a nuevos registros es directa, aplicándose las condiciones de forma secuencial desde el nodo raíz hasta que el registro queda ubicado en una hoja. La clase mayoritaria en esta establecerá su etiqueta, siendo la pureza el grado de propensión en la asignación.

7.3.1.2 REGRESIÓN

Al igual que la clasificación, la regresión es una técnica predictiva, pero con la particularidad de que la variable objetivo es continua. Sus aplicaciones cubren todos los sectores, centrándose en tareas de correlación, cuantificación de la causalidad o detección de tendencias.

El método de regresión más universal es la **regresión lineal**. Esta nos permite expresar la variable dependiente que queremos explicar como una combinación lineal de m variables independientes o predictores. El problema de la regresión lineal, y en general el de los modelos lineales, es su baja flexibilidad para identificar y cuantificar las relaciones entre variables. La mayoría de los sistemas son no lineales, y esto hace necesario el empleo de otras aproximaciones¹⁸⁶.

Tipo	Algoritmos	Ejemplos
Lineal La variable objetivo se estima como una combinación lineal de los predictores	<ul style="list-style-type: none"> • Regresión lineal • Regresión contraída (<i>Ridge regression</i>) • Regresión LASSO (<i>Least Absolute Shrinkage and Selection Operator</i>) • Red elástica (<i>Elastic Net</i>) • Regresión parcial en mínimos cuadrados (PLSR, <i>Partial Least Square Regression</i>) 	<ul style="list-style-type: none"> • Calibración de modelos en análisis químico. • Análisis de la eficacia del precio, campañas y promociones en las ventas. • Estimación del número de partes en seguros del automóvil. • Predicción del rendimiento de un cultivo en base a las precipitaciones.
No lineal La variable objetivo no se estima como una combinación lineal de los predictores	<ul style="list-style-type: none"> • Regresión polinómica • Regresión logística¹⁸⁷ • Regresión gaussiana • Árboles de regresión • Redes neuronales artificiales (ANN, <i>Artificial Neural Networks</i>) • Bosque aleatorio (<i>Random forest</i>) 	<ul style="list-style-type: none"> • Predicción financiera (ingresos, margen, gastos). • Modelos biológicos de crecimiento de plantas. • Predicción meteorológica. • Proyección demográfica. • Mantenimiento predictivo en fábricas. • Modelos de supervivencia en física de partículas.

Tabla 7-4. Tipos de regresión según el tipo de modelización, algoritmos y ejemplos de aplicación.

186 Dicho esto, los métodos lineales gozan de una gran aplicabilidad. Además, su simplicidad y elegancia han hecho que la aproximación generalizada a los sistemas no lineales haya consistido, precisamente, en la aplicación de distintas técnicas de linealización.

187 Aunque la regression logística se emplea principalmente en problemas de clasificación, formalmente es un método de regresión, ya que estima probabilidades.

La Tabla 7-4 contiene los principales algoritmos de regresión, agrupados en lineales y no lineales. Respecto a las métricas de rendimiento, las más utilizadas se calculan a partir del error cometido en el ajuste entre el valor observado y el calculado, como el **error absoluto promedio** o el **error cuadrático promedio**.

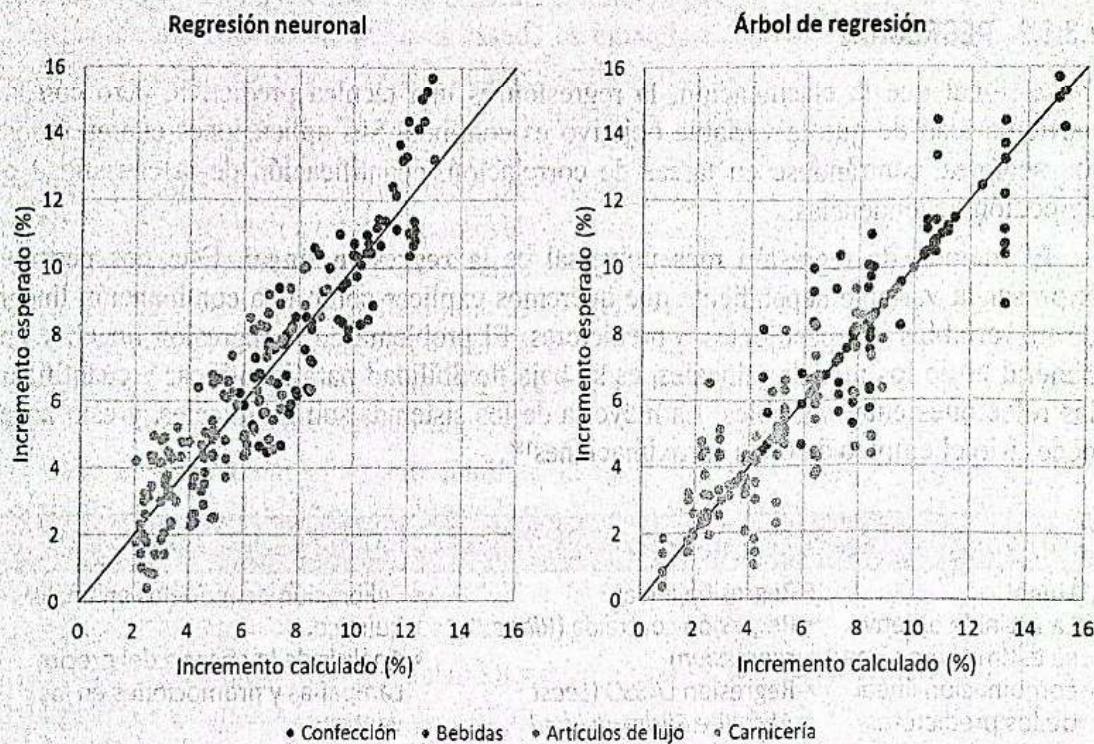


Figura 7-4. Aplicación de modelos de regresión en la estimación de las ventas.

En relación a los **métodos no lineales**, nos encontramos aquí con variantes y adaptaciones de algoritmos empleados también en problemas de clasificación. Como ya avanzamos, los **árboles de decisión** cuentan con versiones para la estimación de variables continuas, proporcionando también reglas de asignación que combinan variables categóricas y numéricas. Otra familia importante de métodos son las **redes neuronales artificiales**. Aunque abordaremos estas en otro capítulo, avanzar aquí que se componen de múltiples elementos de procesado básico, organizados en capas e interconectados, que en conjunto son capaces de aproximar funciones muy complejas, ofreciendo una gran versatilidad en cuanto a sus posibles aplicaciones.

La Figura 7-4 muestra el resultado de la aplicación de dos modelos de regresión a un conjunto de prueba para la estimación de las ventas. En base a resultados históricos, el objetivo es predecir el efecto de las promociones publicitarias en el incremento de la facturación para determinadas secciones de productos. Es un caso sencillo que tiene en cuenta la sección del producto, el tipo de promoción, el coste asociado de la misma, así como el volumen de facturación anterior a la acción comercial. La resolución de un modelo predictivo acostumbra a representarse enfrentando el valor esperado frente al calculado por el modelo. Si este proporcionara un ajuste perfecto, entonces todos los puntos caerían sobre la diagonal y el valor de los diferentes errores sería cero. Como este

no es el caso, vemos que los puntos se apartan de la recta, siendo las distancias a esta el valor de las llamadas **residuales**¹⁸⁸.

En el caso del árbol de regresión se puede apreciar que los puntos tienden a alinearse verticalmente. Esto es debido a que los árboles dan un valor promedio de la predicción para todos los registros que cumplen las mismas condiciones, es decir, que pertenecen a un mismo nodo¹⁸⁹. Esta dispersión no existe en el caso de la regresión empleando redes neuronales, donde los puntos están más apretados alrededor de la diagonal, con la excepción de las promociones más exitosas en la sección de bebidas, donde el modelo es más conservador¹⁹⁰.

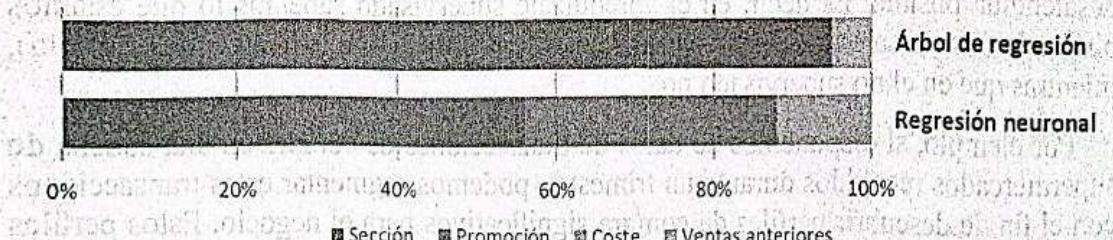


Figura 7-5. Importancia relativa de cada predictor en la estimación de las ventas.

La Figura 7-5 muestra el peso relativo de cada variable a la hora explicar la variable objetivo, información proporcionada por el propio algoritmo. Aunque la sección del producto domina en ambos modelos, en el neuronal el resto de las variables tiene mayor presencia (el árbol no llega a considerar siquiera el tipo de promoción). En este sentido, el árbol consigue explicar prácticamente la misma variabilidad que la red neuronal teniendo en cuenta menos factores, proporcionando, además, reglas de asignación que nos pueden ayudar en el diseño de la promoción. Como en el caso de los modelos de clasificación, el foco en la capacidad descriptiva o predictiva nos indicará cuál de los dos modelos es el más interesante para el negocio, teniendo en cuenta a su vez las distintas métricas de rendimiento que ofrece cada uno.

Dentro de los modelos de regresión cabe destacar el **análisis de series temporales**, donde además de entender la composición, tendencia y estacionalidad de una sucesión de datos fechados, se busca estimar la evolución futura de la serie según su comportamiento histórico (*forecasting*).

188 Los puntos están coloreados según la sección a la que pertenece el producto con fines descriptivos, siendo el modelo único para todas ellas.

189 Es el mismo comportamiento que veíamos en el caso de la clasificación, donde la propensión de asignación a una clase venía dada por la pureza del nodo.

190 Esto puede sugerir la conveniencia de realizar modelos predictivos individuales para cada sección de productos.

7.3.2 Aprendizaje no supervisado

Cuando las observaciones carecen de etiqueta, perdemos entonces la referencia que nos permite calibrar en qué medida un modelo está produciendo resultados correctos. Esto no quiere decir que no podamos evaluar ni inspeccionar la evolución del proceso de aprendizaje, pero sí que tenemos que hacerlo de una forma diferente.

El **aprendizaje no supervisado** (*unsupervised learning*) viene a dar una vía para estos casos. Aquí el objetivo no está en encontrar el patrón que soporta una relación determinada entre los atributos de una observación, sino en descubrir esas relaciones mediante la identificación de agrupaciones y coocurrencias en los datos de la forma más desatendida posible. Es decir, en el aprendizaje supervisado sabemos lo que estamos buscando (otra cosa es que seamos capaz de encontrarlo o, simplemente, que no exista), mientras que en el no supervisado no.

Por ejemplo, si disponemos de datos de transacciones de compra en una cadena de supermercados recogidos durante un trimestre, podemos segmentar estas transacciones con el fin de descubrir perfiles de compra significativos para el negocio. Estos perfiles pueden identificar a un grupo de clientes orientados a la compra de comida, con visitas centradas en los fines de semana y gasto medio; un segundo grupo, interesados en la compra de vinos de calidad alta, con una frecuencia de visita más esporádica pero de gasto elevado, y que viven en zonas de alto poder adquisitivo; un tercero podría estar formado por buscadores de ofertas, con tamaños de cesta pequeños y de baja rentabilidad, etc. Podemos ir un paso más allá e intentar determinar si existen productos o familias de productos que los clientes de cada uno de estos perfiles compran de forma conjunta en cada visita al supermercado. Un modelo de reglas de asociación nos puede decir que, entre otras combinaciones, el segmento de compradores de vino acompaña en el 43% de los casos sus caldos con un tipo de queso muy específico. Es más, el modelo nos puede indicar que el 74% de las cestas que contenían queso también tenían vino, pero que ese porcentaje desciende cuando contabilizamos aquellas que teniendo vino también tenían queso, lo que indica que en la mayoría de los casos el queso es el detonante de la compra del vino. Para cerrar el ejemplo, resulta que la cadena de supermercados está pensando en discontinuar la venta de ese queso, ya que su margen es muy estrecho y además es un producto muy perecedero. Conclusión: si eliminamos el queso de la lista de productos provocaremos que el segmento de compradores de vino se vaya a hacer sus compras a la competencia, allí donde encuentre el queso. Es decir, perderemos un nicho de clientes de alta rentabilidad y muy fidelizado.

Este relato del queso y el vino está basado en un caso real (hay muchos como este), y ejemplifica muy bien lo que la minería de datos, y concretamente la combinación de dos modelos de aprendizaje no supervisado, puede aportar al negocio; llegar a estas conclusiones a través solo de herramientas de consulta y visualización es tarea imposible. También ilustra la envergadura de un problema de *Big Data* en cuanto a la posible volumetría de los datos, y a la necesidad de procesarlos todos para poder extraer información relevante.

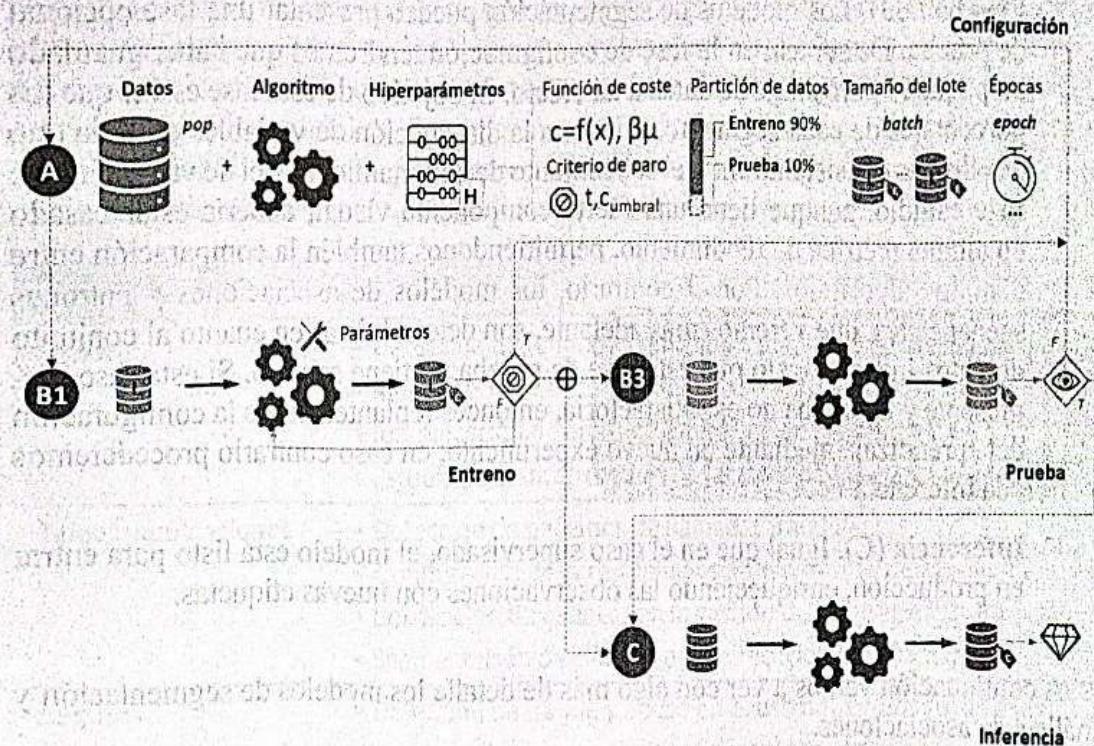


Figura 7-6. Fases en un proceso de aprendizaje no supervisado.

Al no disponer de observaciones etiquetadas, el proceso de aprendizaje no supervisado es diferente al del caso supervisado. La Figura 7-6 muestra sus fases, que matizamos a continuación:

- ▶ **Configuración (A).** El punto de partida es un conjunto de observaciones, en este caso sin etiquetar, un tipo de algoritmo y también un conjunto de hiperparámetros del modelo. Entre estos nos encontramos una función de coste y, como antes, una serie de criterios de paro del aprendizaje. La función de coste no puede estar basada ahora en métricas de discrepancia entre valores esperados y calculados, sino que dependerá más del tipo de modelo¹⁹¹. Por el mismo motivo, la división de datos en subconjuntos de entrenamiento, validación y prueba pierde sentido, aunque comentaremos enseguida este aspecto.
- ▶ **Entrenamiento (B1).** Conceptualmente esta fase es equivalente a la del aprendizaje supervisado. El objetivo es ir afinando los parámetros del modelo mediante sucesivos pases sobre los datos¹⁹². La principal diferencia es que ahora no tenemos una fase de validación. Esto implica entrenar con todos los datos disponibles, aunque hay excepciones.

¹⁹¹ Por ejemplo, en el caso de la segmentación de transacciones que veímos anteriormente, la función de coste tendrá en cuenta alguna medida de similitud u homogeneidad entre las cestas que componen cada segmento.

¹⁹² El empleo de lotes para la modificación de los parámetros dependerá del tipo de modelo y algoritmo, pero no es tan frecuente en este caso.

- ▀ **Prueba (B3).** Los modelos de segmentación pueden presentar una fase opcional de prueba. De ser así, en la fase de configuración tendremos que haber guardado un pequeño porcentaje de datos a tal efecto. El objetivo de esta fase es ver que los porcentajes de cada segmento, así como la distribución de variables en cada uno de ellos, es consistente entre el subconjunto de entrenamiento y el de validación¹⁹³. Este estudio, aunque tiene una fuerte componente visual, debería estar basado en alguna métrica de rendimiento, permitiéndonos también la comparación entre distintos algoritmos. Por el contrario, los modelos de asociaciones y patrones secuenciales, que veremos más adelante, son deterministas en cuanto al conjunto de datos inicial, por lo que esta fase de prueba no tiene sentido. Si esta fase está presente y la prueba no es satisfactoria, entonces replantearemos la configuración del aprendizaje mediante un nuevo experimento; en caso contrario procederemos a la inferencia¹⁹⁴.
- ▀ **Inferencia (C).** Igual que en el caso supervisado, el modelo está listo para entrar en producción, enriqueciendo las observaciones con nuevas etiquetas.

A continuación vamos a ver con algo más de detalle los modelos de segmentación y análisis de asociaciones.

7.3.2.1 SEGMENTACIÓN

Dada una colección de objetos caracterizada por una serie de atributos, los métodos de **segmentación** (*clustering*) persiguen la separación de dichos objetos en distintas agrupaciones (clústeres), de forma que intentan maximizar dos criterios simultáneamente: la homogeneidad intrasegmento y la heterogeneidad intersegmento. Estos criterios acostumbran a actuar como función de coste.

Es importante no confundir la segmentación con la clasificación. En esta última disponemos ya de los grupos y lo que queremos es desarrollar un sistema, mediante aprendizaje supervisado, que sea capaz de asignar objetos a esos grupos. Por el contrario, en la segmentación no disponemos de los grupos, siendo nuestro objetivo descubrirlos mediante aprendizaje no supervisado.

Esta búsqueda de la separabilidad llevada al extremo, produciría finalmente tantas agrupaciones como observaciones en la colección. Sin embargo, el propósito de la segmentación es obtener un número de agrupaciones que sea manejable para el objetivo de negocio marcado. Por ejemplo, si estamos hablando de trabajar sobre una base de datos de 10 millones de clientes para lanzar una campaña comercial, la segmentación nos permite situarnos en un punto intermedio entre gestionar a todos los clientes por igual y diferenciarlos individualmente. La primera opción es

193 Un modelo de segmentación puede también presentar sobreajuste.

excesivamente grosera, mientras que la segunda puede resultar impracticable. Por el contrario, una segmentación que compartimentara a los clientes en, por ejemplo, 15 agrupaciones, nos permitiría detectar perfiles homogéneos en cuanto a rasgos sociodemográficos, hábitos de consumo y uso de los servicios. De esta manera se puede personalizar la campaña para cada agrupación, llegando a derivar después acciones individuales sobre cada cliente¹⁹⁵.

Sector	Ejemplos de aplicación
Salud	<ul style="list-style-type: none"> Detección de fenotipos clínicos en nuevas enfermedades Identificación de perfiles de pacientes en ensayos clínicos Identificación de áreas de interés en imágenes médicas Estudio de la interrelación de factores de riesgo para la salud
Telecomunicaciones	<ul style="list-style-type: none"> Detección de patrones de llamadas fraudulentas Optimización de la red celular según patrones de movilidad Correlación de eventos en la gestión de la infraestructura Segmentación de clientes e identificación del abandono
Seguros	<ul style="list-style-type: none"> Detección de fraude y abuso en el uso de seguros médicos Perfilado de partes para identificar extensiones de pólizas Modelización actuarial de pólizas y diseño de precios Caracterización de rasgos comunes en partes de accidente
Finanzas	<ul style="list-style-type: none"> Diseño y diversificación de carteras de inversión Auditoría y detección de anomalías en datos contables Análisis de rentabilidad y riesgo de entidades bancarias Evaluación de la volatilidad en series temporales financieras
Distribución	<ul style="list-style-type: none"> Identificación de similitudes y diferencias entre tiendas Caracterización de clientes para sistemas de recomendación Generación de grupos de productos para optimizar el surtido Segmentación de cestas para el diseño de promociones
Agricultura	<ul style="list-style-type: none"> Estudio de la productividad de distintas zonas de cultivo Evaluación de la sostenibilidad ecológica de granjas Comparación de tipos de suelo para cultivo Ánálisis del rendimiento de diferentes genotipos de semillas

Tabla 7-5. Ejemplos de aplicación de las técnicas de segmentación en distintos sectores.

195 Un caso en el que tiene sentido manejar un mayor número de segmentos es en la **detección de anomalías**. Aquí el objetivo es detectar grupos que pueden ser muy pequeños en tamaño (nichos), pero con un comportamiento muy marcado y diferenciado.

Hay que dejar claro que no existe una segmentación única que se pueda construir sobre un conjunto de datos. Dependiendo del objetivo de negocio, unos atributos tendrán más sentido y peso que otros, ocurriendo lo mismo con el número óptimo de agrupaciones.

La Tabla 7-5 contiene una relación de aplicaciones de las técnicas de segmentación para diferentes sectores. Todas aquellas relacionadas con la gestión de clientes tienen una gran transversalidad. En cualquier caso, la segmentación es una técnica muy común como base a la hora de realizar la modelización de un conjunto de datos, actuando como técnica exploratoria inicial, método de muestreo o punto de partida para modelos de clasificación y predicción.

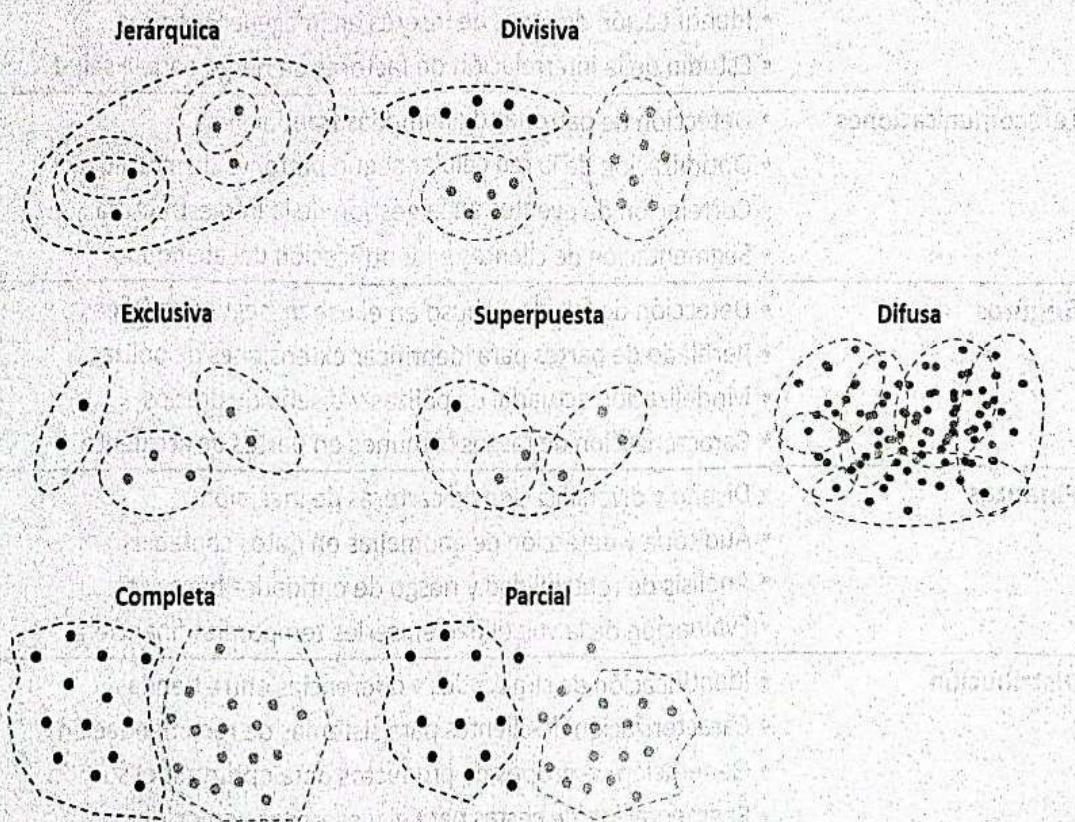


Figura 7-7. Tipos de segmentaciones.

Existen una gran variedad de métodos de segmentación, así como múltiples alternativas a la hora de categorizarlos. La Figura 7-7 muestra tres criterios de clasificación (cada uno en una fila) que no son excluyentes. El objetivo y los requerimientos de la segmentación guiarán la elección de un método u otro. Por ejemplo, una segmentación de especies botánicas requerirá un método exclusivo (*hard*), y probablemente jerárquico y completo, ya que nos interesa que cada especie esté en un solo segmento. Por el contrario, una tipificación de pacientes se efectuará mejor con un método superpuesto (*soft*), debido a que los perfiles exhibirán un cierto grado de solapamiento. En el extremo de los métodos superpuestos están los **métodos difusos** (*fuzzy*), donde todas las observaciones están asignadas a todos los segmentos mediante un grado de pertenencia.

Familia	Características	Algoritmos
Basados en la conectividad Métricas de distancia	Métodos de segmentación jerárquica donde los segmentos están anidados. La generación puede partir de las observaciones que se van agregando, formando los segmentos (aglomerante), o divisiva, donde la población global se va repartiendo en distintos grupos	<ul style="list-style-type: none"> Segmentación jerárquica aglomerante (AGNES, <i>AGglomerative NESting</i>) Segmentación jerárquica divisiva (DIANA, <i>Dlvisive ANALysis</i>) BIRCH (<i>Balanced Iterative Reducing and Clustering using Hierarchies</i>)
Basados en el prototipo Métricas de distancia	Métodos divisivos donde cada segmento está caracterizado por un centro de gravedad que se calcula minimizando la distancia entre sus observaciones	<ul style="list-style-type: none"> K-medias K-prototipos Mapas autorganizativos (SOM, <i>Self-Organizing Maps</i>)
Basados en la distribución Métricas de probabilidad	Las observaciones son agrupadas de acuerdo con su afinidad a seguir una misma distribución de probabilidad estadística	<ul style="list-style-type: none"> Mezcla gaussiana (GMM, <i>Gaussian Mixture Modeling</i>) Mixtura de Dirichlet (DMM, <i>Dirichlet Mixture Models</i>) EM (<i>Expectation-Maximization</i>)
Basados en la concentración Métricas de densidad	Los segmentos se crean considerando zonas de alta concentración de observaciones que a su vez están rodeadas por zonas poco densas	<ul style="list-style-type: none"> DBSCAN (<i>Density-Based Spatial Clustering of Applications with Noise</i>) OPTICS (<i>Ordering Points to Identify the Clustering Structure</i>) DENCLUE (<i>DENSity-based CLUstering</i>)

Tabla 7-6. Clasificación de las técnicas de segmentación

La Tabla 7-6 contiene los principales algoritmos clasificados en 4 familias. Los métodos basados en la conectividad y en prototipos son los más convencionales. En ambos casos se utilizan criterios de proximidad o similitud para la asignación de los segmentos. En estos algoritmos es necesario especificar el número de segmentos o la distancia umbral a modo de hiperparámetros.

En modo inferencia, la aplicación de un modelo de segmentación a un conjunto de datos nos dará como resultado la asignación de un identificador de segmento a cada observación, normalmente acompañado por alguna métrica de similitud.

7.3.2.2 REGLAS DE ASOCIACIÓN Y PATRONES SECUENCIALES

El análisis de asociaciones persigue el descubrimiento y la cuantificación de relaciones relevantes en grandes volúmenes de datos. Estas relaciones toman la forma de **reglas de implicación**, donde la presencia (o ausencia) de una serie de ítems en un conjunto de datos conlleva la de otros.

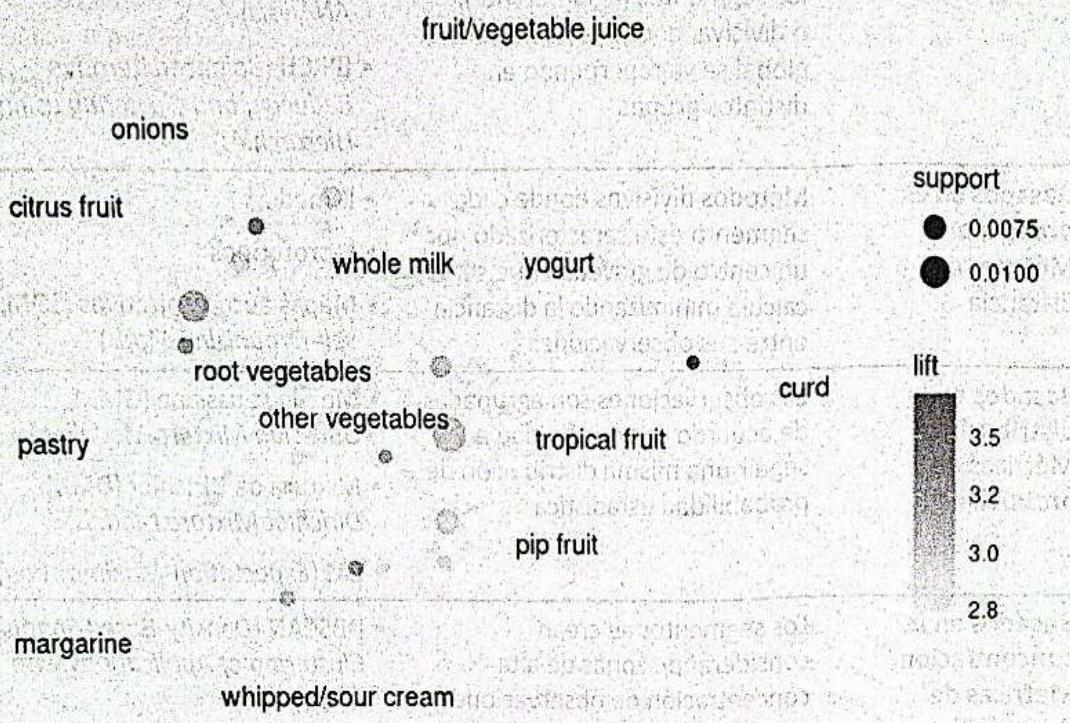


Figura 7-8. Visualización de reglas de asociación en un análisis de cestas de la compra realizada con la librería de R arulesViz.

La detección de estas relaciones se realiza dentro del contexto de una **transacción**, entendida de forma general como una colección de atributos que forman parte de un evento. Uno de los campos donde más se aplica el análisis de asociaciones es en el **análisis de cestas de la compra** (MBA, *Market Basket Analysis*). En este caso, cada transacción es una cesta de la compra, y sus atributos son los distintos productos que la forman¹⁹⁶. Partiendo de un conjunto de cestas registradas durante un periodo de tiempo, el objetivo es identificar qué combinaciones de productos son adquiridos de forma conjunta dentro de una misma compra. Si en el mismo conjunto de cestas disponemos del identificador del cliente que ha realizado cada compra, entonces podemos hacer que la transacción sea igual al cliente; de esta manera, el análisis nos dirá qué productos combinan los clientes en sus compras, con independencia de que lo hagan en una o varias visitas.

196 Es decir, en el contexto del análisis de asociaciones, la transacción es también comercial.

En el caso anterior, los ítems o atributos de la transacción son homogéneos, ya que se trata de identificadores de productos¹⁹⁷. Esto no tiene por qué ser siempre así. Si estamos interesados en detectar relaciones entre los síntomas, la medicación y los efectos secundarios que puede sufrir un paciente, entonces el paciente constituye la transacción y los ítems son, por ejemplo, el nivel de colesterol, la estatina recetada y la aparición de dolores musculares en la zona de las rodillas. Este caso muestra, además, que los atributos pueden ser binarios, indicando la presencia o ausencia de dolor, y también continuos, como el nivel de glucosa¹⁹⁸.

Sin entrar en mucho formalismo, podemos decir que una regla de asociación es una vinculación de la forma

$$A \rightarrow B, A \cap B = \emptyset$$

donde A y B son conjuntos disjuntos de ítems, denominados *itemsets*. A recibe el nombre de **antecedente** (o cuerpo) de la regla, mientras que B es el **consecuente** (o cabeza). Para un conjunto de p transacciones $T = \{t_1, \dots, t_p\}$, una transacción t_i contendrá el *itemset* A si A es un subconjunto de sus ítems. Volviendo al ejemplo del análisis de cestas de la compra, la detección de reglas de asociación en una cadena de electrodomésticos podría dar resultados como

[video cámara]+[pantalla de proyección] → [teléfono móvil]

[pantalla de plasma]+[teléfono móvil] → [consola]+[tostadora]

En este caso, el cliente es la transacción, estando sus compras recogidas durante un periodo de seis meses. En este punto, y de forma intuitiva, podemos decir que cuando un cliente compra una video cámara y una pantalla de proyección entonces compra también un teléfono móvil¹⁹⁹.

El problema del análisis de asociaciones radica en que el número posible de combinaciones para formar reglas crece de forma exponencial con el número de ítems. Es necesario, por tanto, disponer de algoritmos que sean capaces de explorar de forma eficiente el espacio de todas las combinaciones de productos, extrayendo aquellas más relevantes. La cuestión, por lo tanto, está en determinar cuándo una regla es relevante. Esta evaluación puede ser subjetiva, teniendo en cuenta el conocimiento del negocio, o objetiva, basada en una serie de métricas estadísticas como el **soporte** (*support*), la **confianza** (*confidence*) o el **interés** de la regla (*lift*), que cuantifican su frecuencia,

¹⁹⁷ La estructura de datos típica para la realización de un análisis de asociaciones se compone de una tabla con al menos dos columnas: un identificador de la transacción y un identificador del ítem, siendo la clave primaria la combinación de ambas. Otra posibilidad es esta misma estructura pivotada, donde cada registro es una transacción y cada ítem una columna. El problema de esta última es que no es escala bien con el número de ítems.

¹⁹⁸ Estos casos requieren un preprocesado adecuado de los datos.

¹⁹⁹ El análisis de asociaciones no nos informa de la secuencia en que estos productos son comprados, sino que se limita a informarnos de una concurrencia entre los ítems en el antecedente y en el consecuente de la regla.

relevancia y capacidad predictiva. El resultado de un modelo de análisis de asociaciones se puede visualizar de forma gráfica, como en la Figura 7-8. La representación en forma de grafo dirigido permite asignar las distintas métricas de interés que acabamos de comentar al tamaño y color de los nodos, y también al trazo o grosor de los arcos.

Cuando añadimos la componente temporal al análisis de asociaciones llegamos a los modelos de **patrones secuenciales**. Se trata también de encontrar relaciones entre ítems, pero ahora entre múltiples transacciones que están relacionadas entre sí, perteneciendo todas a lo que llamamos un **grupo de transacciones**. Además, nos puede interesar cuantificar la cadencia en que se producen esas relaciones. Por ejemplo, las expresiones

[colesterol alto]»[estatina+ejercicio físico]→[colesterol correcto]

[pan]»[huevos+queso]→[leche]

[congelados]»[carro de la compra]»[confirmación]→[abandono]

son ejemplos de reglas donde el antecedente está formado por una secuencia temporal de *itemsets* separados por una doble flecha.

La cuantificación de una regla de secuencias se realiza también mediante el soporte, la confianza o la elevación. A la hora de detectarlas, además de establecer umbrales para las métricas anteriores, nos puede interesar que estas cumplan una serie de restricciones temporales, estableciendo intervalos y lapsos de tiempo mínimos y máximos entre los *itemsets* que forman la secuencia.

Tipo	Algoritmos	Ejemplos
Reglas de asociación	<ul style="list-style-type: none"> • Apriori • FP-Growth • PrefixSpan • SIDE (<i>Simultaneous Depth-first Expansion</i>) • ECLAT (<i>Equivalence CLass Transformation</i>) 	<ul style="list-style-type: none"> • Ubicación de productos en el punto de venta • Diseño de promociones y venta cruzada • Optimización de almacenes • Diagnosis médica • Secuenciación de proteínas • Detección de relaciones entre palabras dentro de un texto • Análisis del fracaso escolar
Patrones secuenciales	<ul style="list-style-type: none"> • Apriori • PrefixSpan • SIDE (<i>Simultaneous Depth-first Expansion</i>) • GSP (<i>Generalized Sequential Pattern</i>) • SPADE (<i>Sequential Pattern Discovery using Equivalence classes</i>) 	<ul style="list-style-type: none"> • Incremento de ventas (<i>upselling</i>) • Análisis de navegación en páginas web (<i>clickstream analysis</i>) • Detección de fraude en partes de seguro médico • Análisis de sendas de abandono de clientes • Estudio de la evolución de pacientes • Identificación de intrusiones en redes informáticas

Tabla 7-7. Algoritmos y ejemplos para el análisis de asociaciones y patrones secuenciales.

Desde el punto de vista de la inferencia, los modelos de reglas pueden ser aplicados a nuevas transacciones, obteniendo los productos que mejor se asocian con el contenido de estas según las reglas existentes en el modelo. Esto permite, por ejemplo, la implementación de sistemas de personalización de ofertas. La Tabla 7-7 reúne los principales algoritmos y aplicaciones para el análisis de asociaciones.

Además del supervisado y no supervisado, existen otros paradigmas de aprendizaje, como el **aprendizaje semisupervisado**, que combina los dos, el **aprendizaje por refuerzo** (*reinforcement learning*) o el **aprendizaje profundo** (*deep learning*). Este último no hace referencia tanto a un nuevo concepto de aprendizaje, sino a una tipología de algoritmos alrededor de las redes neuronales artificiales. Veremos estos dos últimos en un capítulo posterior.

7.4 PUESTA EN PRODUCCIÓN E INFERENCIA DE MODELOS

Una vez que un modelo de minería de datos ha sido desarrollado, el siguiente y más importante paso es su **despliegue a un entorno productivo**, donde puede ser aplicado a nuevos datos. Es importante resaltar que este **entorno de inferencia** (*scoring*) será, en principio, diferente al de modelización. Puede tratarse de un sistema transaccional, analítico o con una carga de trabajo mixta. Por ejemplo, un modelo predictivo se presta a ser integrado dentro de una aplicación OLTP, etiquetando transacciones a medida que estas se producen. Adicionalmente, una solución OLAP también se beneficiaría de dicha integración, publicando los resultados de la inferencia en informes y cuadros de mando.

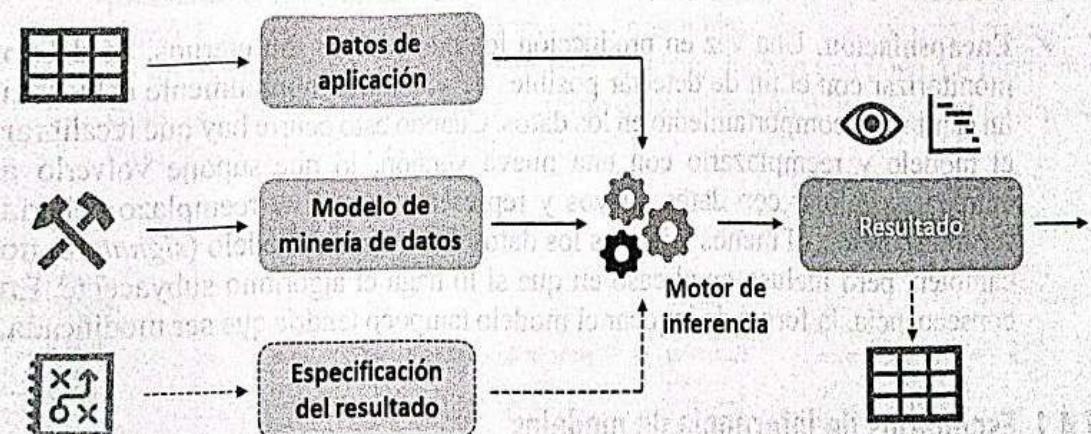


Figura 7-9. Inferencia de un modelo de minería de datos.

La Figura 7-9 resume el proceso de inferencia. Además del modelo en sí y de los datos sobre los que debe ser aplicado, nos hace falta un **motor de inferencia** (*scoring engine*). Esta es la pieza que tiene la capacidad de entender el modelo (tipo, parámetros, atributos de entrada) y utilizarlo para inferir un resultado de acuerdo con una especificación. La especificación dicta en qué contenido de la inferencia estamos interesados y dependerá del tipo de modelo. Finalmente, el resultado puede ser almacenado, visualizado o utilizado dentro de un proceso y una lógica de negocio.

La puesta en producción de un modelo implica una serie de consideraciones que se deben tener en cuenta:

- ▶ **Estandarización de la definición del modelo.** El entorno de producción debe soportar diferentes tipos de modelos, desarrollados mediante herramientas de distintos proveedores. Es decir, el mecanismo de inferencia debe ser agnóstico en cuanto al origen del modelo. Existen diferentes estándares de representación de modelos, destacando el formato **PMML** (*Predictive Model Markup Language*) y el **PFA** (*Portable Format for Analytics*)²⁰⁰. Un gran número de soluciones del mercado pueden exportar y/o importar modelos en estos formatos.
- ▶ **Desacoplamiento del preprocesado.** Los datos sobre los que se aplica el modelo pueden ser diferentes en el entorno de desarrollo y en el de producción. Esto no solo es cierto para la ubicación y formato de los atributos, sino también para las transformaciones requeridas sobre estos antes de alimentar el modelo. Incluso dentro del mismo entorno productivo podrá ser necesario soportar diferentes formatos de registro.
- ▶ **Variación en el contenido de la inferencia.** El resultado de aplicar un modelo puede variar de un escenario a otro. Por ejemplo, en la aplicación de un modelo de segmentación en un entorno analítico nos puede interesar la obtención de las métricas de calidad y confianza en la asignación de cada observación a cada segmento obtenido. Sin embargo, en una aplicación de CRM operacional solo necesitaremos el identificador del segmento al que mejor se ajusta dicha observación. Los requerimientos de persistencia de los resultados también podrán variar de un caso a otro.
- ▶ **Encapsulación.** Una vez en producción los modelos no son eternos. Se deben monitorizar con el fin de detectar posibles desviaciones, normalmente debidas a un cambio de comportamiento en los datos. Cuando esto ocurre hay que recalibrar el modelo y reemplazarlo con una nueva versión, lo que supone volverlo a entrenar y validar con datos nuevos y representativos. Este reemplazo debería ser transparente, al menos mientras los datos de entrada al modelo (*signature*) no cambien, pero incluso en el caso en que sí lo haga el algoritmo subyacente. En consecuencia, la forma de invocar el modelo tampoco tendría que ser modificada.

7.4.1 Escenarios de inferencia de modelos

Hay diferentes escenarios en los que se puede invocar un modelo. La selección entre uno u otro dependerá de los requerimientos de latencia de la aplicación, pero también de como de volátiles son los datos y, consecuentemente, de con qué frecuencia hay que recalcular los resultados y persistirlos.

- Inferencia por lotes con persistencia.** El modelo es aplicado a un conjunto de observaciones en bloque. Los datos de entrada residen típicamente en un fichero o en una tabla de una base de datos, a donde van a parar también los resultados de la inferencia. Este escenario es interesante cuando es necesario procesar un gran número de observaciones y guardar los resultados para su posterior consumo. La inferencia por lotes puede ejecutarse bajo demanda o bien de forma planificada mediante un proceso periódico. La latencia en la aplicación del modelo no es importante.
- Inferencia por lotes sin persistencia.** Este caso es muy similar al anterior, con la salvedad de que no hay el requerimiento de persistir los resultados. Lo que se persigue es obtener siempre, y de forma ágil, el último resultado de la inferencia en el momento de su consulta. Las bases de datos relacionales se prestan a este escenario, ya que la invocación del modelo puede ser embebida dentro de la definición SQL de una vista. De esta manera, cada vez que se accede a esta los resultados son actualizados.
- Inferencia en cascada.** Aquí la inferencia se produce como reacción a un evento. Aunque no es exclusivo, este escenario vuelve a ajustarse muy bien a los gestores de datos. En estos, un **disparador** (*trigger*) es un objeto de la base de datos que define un conjunto de acciones que se ejecutarán en respuesta a una operación SQL sobre una tabla o una vista, incluyendo inserciones, actualizaciones y borrado de registros. Adicionalmente, el disparador tiene en su definición el momento de su activación, que puede ser antes o después de la operación. Las acciones consisten en elementos de lenguaje procedimental, incluyendo la invocación de procedimientos almacenados, así como operaciones laterales en la base de datos, como inserciones o actualizaciones en otras tablas. Por ejemplo, la invocación de un modelo puede ubicarse dentro de un disparador en una tabla, de forma que cada vez que se produzca una inserción o actualización de un registro en ella, el modelo es aplicado y el resultado almacenado.

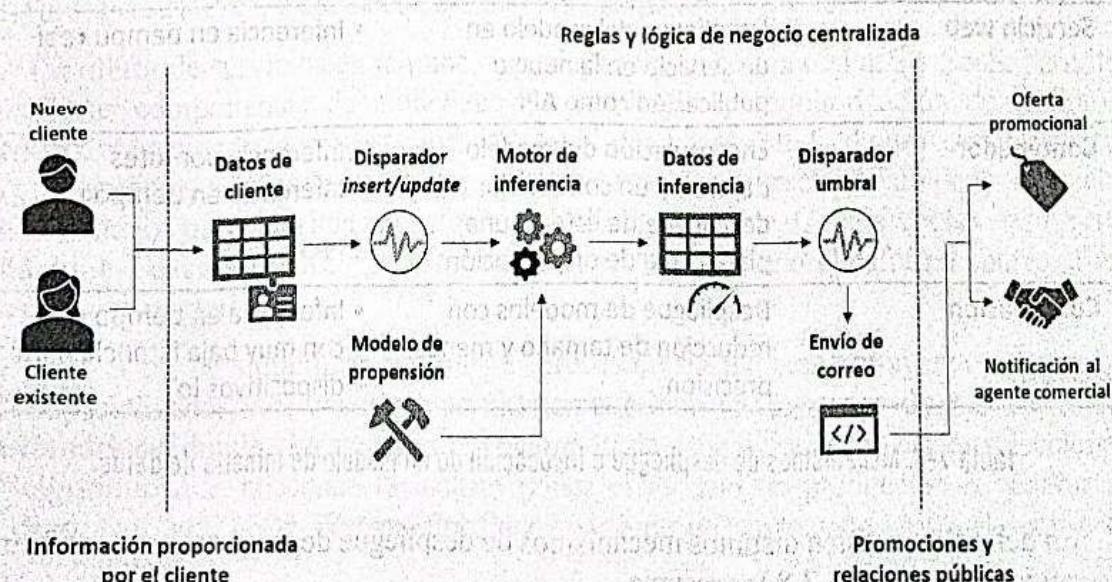


Figura 7-10. Activación de un modelo de propensión mediante disparadores en una base de datos.

4. **Inferencia en tiempo real.** Este escenario se da principalmente cuando se desea aplicar, con una latencia mínima, un modelo a observaciones individuales que todavía no tienen persistencia. Estas acaban de ser generadas por una aplicación a la que hay que devolver los resultados en tiempo real para soportar una acción de negocio. Un ejemplo sería la decisión de generar un cupón de descuento en la caja de un supermercado, teniendo en cuenta el resultado de aplicar un modelo de propensión al contenido de la compra.

El escenario de inferencia en cascada es interesante desde el momento en que concentra la aplicación de los modelos en un único punto, allí donde residen los datos. De esta manera, no es necesario programar esta operación en cada proceso de negocio, quedando ligada a la misma transacción en la base de datos. Así, es posible implementar, por ejemplo, campañas de fidelización de clientes sin necesidad de utilizar un *software* especializado. La Figura 7-10 representa un ejemplo de esto. Cada vez que se registra un nuevo cliente, o que cambia la información de uno existente, se activa un disparador en la tabla de clientes que puntuá el perfil aplicando un modelo de propensión de compra. El resultado se almacena en otra tabla, donde otro disparador comprobará si el valor es mayor que un determinado umbral, fijo o dinámico. En caso afirmativo, un procedimiento almacenado en la base de datos enviará un correo al cliente con una promoción especial, notificando al mismo tiempo a un agente comercial.

Forma de despliegue	Características	Escenarios
Código propio	El modelo se invoca con las funciones de inferencia de la propia librería con la que fue generado u otra compatible	<ul style="list-style-type: none"> • Inferencia por lotes
Base de datos	El modelo es generado en la base de datos o importado	<ul style="list-style-type: none"> • Inferencia por lotes • Inferencia en cascada
Servicio web	Despliegue del modelo en un servicio en la nube o publicación como API	<ul style="list-style-type: none"> • Inferencia en tiempo real
Contenedor	Encapsulación del modelo dentro de un contenedor y despliegue de este en una plataforma de orquestación	<ul style="list-style-type: none"> • Inferencia por lotes • Inferencia en tiempo real
Cuantización	Despliegue de modelos con reducción de tamaño y menos precisión	<ul style="list-style-type: none"> • Inferencia en tiempo real con muy baja latencia para dispositivos IoT

Tabla 7-8. Mecanismos de despliegue e invocación de un modelo de minería de datos.

En definitiva, existen distintos mecanismos de despliegue dependiendo del escenario de inferencia. La Tabla 7-8 los resume.

7.5 HERRAMIENTAS Y SOLUCIONES PARA MINERÍA DE DATOS

Desde una perspectiva de herramientas, hasta la mitad de los años noventa del siglo pasado, la modelización se realizaba caso por caso a través de lenguajes de programación, empleando en ocasiones paquetes estadísticos especializados. Surgirían entonces una serie de entornos de desarrollo integrados, equipados con sofisticadas interfaces gráficas y librerías de algoritmos, donde la programación era sustituida por la parametrización. Soluciones de aquella época como **SAS Enterprise Miner** o **IBM SPSS Modeler** siguen presentes en el mercado, siendo **KNIME Analytics Platform** una opción más reciente en esta línea. Todas ellas proporcionan un entorno de desarrollo visual de modelos, con extensiones para la conexión a repositorios, preparación de datos y representación gráfica.

Otra área importante de herramientas está dentro de los gestores de bases de datos, normalmente relacionales, en forma de librerías programables mediante SQL y lenguajes procedimentales. Al evitar el movimiento de grandes volúmenes de datos a un motor analítico externo, esta aproximación permite un gran escalado y una gestión integrada de datos y modelos. Proveedores de **plataformas de data warehouse**, como Oracle, Microsoft, Vertica, Teradata o IBM tienen extensiones para minería de datos dentro de sus soluciones.

Las arquitecturas actuales basadas en microservicios desacoplados, estándares abiertos y entornos en la nube híbridos demandan una gran flexibilidad a la hora de integrar los diferentes componentes que constituyen una solución. Bajo estos requerimientos, la minería de datos ha experimentado una vuelta a los orígenes de la mano de un amplio ecosistema de marcos de desarrollo alrededor de lenguajes de programación como **Python**, **Julia** o **R**. Cabe destacar aquí librerías como **scikit-learn**, **TensorFlow**, **Keras** o **PyTorch**, estas tres últimas más enfocadas al aprendizaje profundo, sistemas como **Spark MLlib**, que proporciona un alto rendimiento para el entrenamiento de modelos de forma distribuida, o entornos de desarrollo, como **JupyterLab**.

La oferta de servicios en la nube, especialmente en forma de PaaS y SaaS, permite combinar componentes de modelización, despliegue y monitorización de distintos proveedores, siendo la orquestación de las API una necesidad. Cada proveedor tiene su propio entorno, cubriendo las tres fases del proceso y las operaciones alrededor del ciclo de vida de los modelos. Podemos destacar **AWS SageMaker**, **Google Vertex AI**, **Azure Machine Learning** o **Kaggle**, esta última consistente en una plataforma colaborativa basada en JupyterLab.

Con el fin de simplificar aún más el proceso, especialmente atrayendo a usuarios no especializados, nos encontramos últimamente con los **entornos de modelización automáticos**, donde el sistema es el encargado de acondicionar los datos, seleccionar el algoritmo más adecuado, e incluso poner el modelo en producción o generar el código asociado. **IBM Watson Studio** dispone de una extensión (AutoAI) con esta funcionalidad.

7.6 RESUMEN DEL CAPÍTULO

La minería de datos proporciona un conjunto de técnicas y metodologías para el descubrimiento de patrones, relaciones y tendencias difíciles de detectar mediante métodos de consulta convencionales.

- Un proceso de minería de datos se compone de tres etapas: **preprocesado y acondicionamiento** de los datos, **modelización** y **puesta en producción**.
- Las **técnicas** de modelización pueden clasificarse en **supervisadas** y **no supervisadas**. En el primer caso se parte de observaciones con uno de sus atributos identificado como objetivo. Trabajando sobre datos históricos, el propósito del modelo es estimar ese atributo a partir de los otros, de forma que pueda ser aplicado a nuevas observaciones y obtener predicciones.
- Las principales técnicas supervisadas son la **clasificación**, donde el atributo objetivo es categórico, y la **regresión**, donde es numérico.
- En la modelización no supervisada no hay un atributo objetivo; de lo que se trata es de descubrir relaciones entre los atributos mediante la identificación de agrupaciones y coocurrencias en los datos. Los modelos de **segmentación**, la **detección de asociaciones** y de **patrones secuenciales** son ejemplos de técnicas no supervisadas.
- Una vez obtenidos y validados, los modelos son **puestos en producción** para ser aplicados a nuevos datos, integrándose con el resto de los sistemas transaccionales y analíticos de la organización.

Más adelante estudiaremos el **aprendizaje profundo**, y como su reciente desarrollo está permitiendo nuevos tipos de análisis que tienen que ver con la percepción, el razonamiento y otros procesos cognitivos.