Introducción al aprendizaje automático

Jordi Casas Roma

1 crédito
xxx/xxxxx/xxx

Índice

Introducción

Este módulo didáctico constituye la introducción al aprendizaje automático (*machine learning*, en inglés) o minería de datos (*data mining*). Después de una breve introducción y contextualización de la temática, se presentan las etapas de un proceso de minería de datos, empleando la metodología CRISP-DM.

A continuación, se propone la clasificación básica de algoritmos y problemas relacionados, diferenciando entre las tareas de agrupación, clasificación y regresión.

En el tercer bloque de este módulo didáctico se introducen los conceptos relacionados con los conjuntos de entrenamiento (*train*), test (*test*) y validación (*validation*), aplicables a los métodos supervisados, y que serán importantes en cualquier proyecto de aprendizaje automático o minería de datos.

Finalizaremos el texto presentando las principales métricas empleadas en la evaluación de los resultados de un modelo de aprendizaje automático, que permiten comparar el rendimiento de los distintos algoritmos sobre un mismo conjunto de datos.

Objetivos

En los materiales didácticos de este módulo encontraremos las herramientas indispensables para asimilar los siguientes objetivos:

1. Conocer las etapas de un proyecto de minería de datos, según la metodología CRISP-DM.
2. Conocer la tipología básica de métodos y tareas en aprendizaje automático.
3. Comprender como se generan los conjuntos de datos de entrenamiento, test y validación.
4. Conocer las principales métricas de evaluación de resultados.

1. Contextualización

Los juegos de datos encierran estructuras, patrones y reglas de los que es posible extraer conocimiento sobre los eventos que los han generado. La física teórica más avanzada, cada vez más, habla de un Universo de eventos y no de partículas, de modo que tratar de entender o incluso predecir estos eventos se ha convertido en un reto para muchas disciplinas y la razón de ser para la minería de datos, que ha pasado de tratar de entender los datos a tratar de comprender los eventos que hay detrás.

Las cosas no son como aparentan ser, por esta razón técnicas como la visualización de datos, aunque necesarias, son del todo insuficientes para llegar hasta el conocimiento que se esconde detrás de estructuras y relaciones no triviales en los juegos de datos.

Esta nueva visión convierte necesariamente los equipos de analistas de datos en equipos interdisciplinares donde se requieren habilidades matemáticas e informáticas, pero también conocedoras del negocio y de la organización empresarial. Solo así, se podrá cubrir el proceso de extracción de conocimiento de principio a fin.

A través de una metodología adecuada y de entender tanto qué tipo de problemas trata de resolver, como con qué técnicas hace frente a estos retos, trataremos de ofrecer una visión lo más completa posible de lo que es hoy en día la minería de datos.

2. Etapas de un proyecto de minería de datos

Desde las organizaciones de hoy en día, nos enfrentamos a proyectos complejos con multitud de tareas interdisciplinares e interdependientes, que además mezclan intereses y necesidades de diferentes grupos de personas y que normalmente están condicionados por limitaciones económicas y tecnológicas.

Lo recomendable en estos casos es diseñar una hoja de ruta que nos va a permitir saber dónde estamos, dónde queremos llegar y las medidas a tomar para corregir periódicamente las desviaciones del rumbo seguido.

La metodología CRISP-DM nació en el seno de dos empresas, DaimlerChrysler y SPSS, que en su día fueron pioneras en la aplicación de técnicas de minería de datos (en inglés, *data mining*) en los procesos de negocio. CRISP-DM se ha convertido de facto en la metodología del sector. Su éxito se debe a que está basada en la práctica y experiencia real de analistas de minería de datos que han contribuido activamente al desarrollo de la misma.

Veremos que hay dos aspectos clave en esta metodología: La adopción de la estrategia de calidad total y la visión de un proyecto *data mining* como una secuencia de fases.

El compromiso con la calidad en el mundo de la gestión de proyectos pasa por seguir de forma iterativa lo que se conoce como ciclo de Deming o ciclo PDCA:

- **Planificar *(Plan)***: Establecer los objetivos y los procesos necesarios para proporcionar resultados de acuerdo con las necesidades del cliente y con las políticas de la empresa.

- **Hacer *(Do)***: Implementar los procesos.

- **Verificar *(Check)***: Monitorizar y medir los procesos y los servicios contrastándolos con las políticas, los objetivos y los requisitos, e informar sobre los resultados.

- **Actuar *(Act)***: Emprender las acciones necesarias para mejorar continuamente el rendimiento y comportamiento del proceso.

Un aspecto a destacar es que la iteración y revisión de fases y procesos se remarca como un aspecto clave si se quiere ejecutar un proyecto de calidad.

De este modo se establecen micro ciclos de planificación, ejecución y revisión, de los que solo se sale cuando el proceso de revisión es satisfactorio. Este principio está muy

presente tanto en la norma ISO 9000 como en la ISO 20000.

Todas las fases son importantes, por supuesto, pero cabe subrayar que la tendencia natural de la condición humana, por experiencia propia, es la de concentrar recursos en exceso al final del proyecto, en la fase de despliegue, por no haber hecho las cosas bien en las fases anteriores.

Merece la pena y es más eficiente y económico, no escatimar recursos en las fases iniciales de preparación, planificación, construcción e iteración.

Conviene mencionar también, que la metodología debe ser entendida siempre como una guía de trabajo que permite garantizar una calidad en la entrega del proyecto. Para conseguir que efectivamente sea una guía de trabajo útil y práctica, deberemos adaptarla a las necesidades, limitaciones y urgencias que en cada momento tengamos.

Vamos a estudiar todas las fases que nos propone la metodología CRISP-DM. Observad que en el centro del esquema que la resume se encuentra el objetivo de la misma, es decir, la conversión de los datos en conocimiento.

La Figura 2 esquematiza el ciclo de fases que propone CRISP-DM.

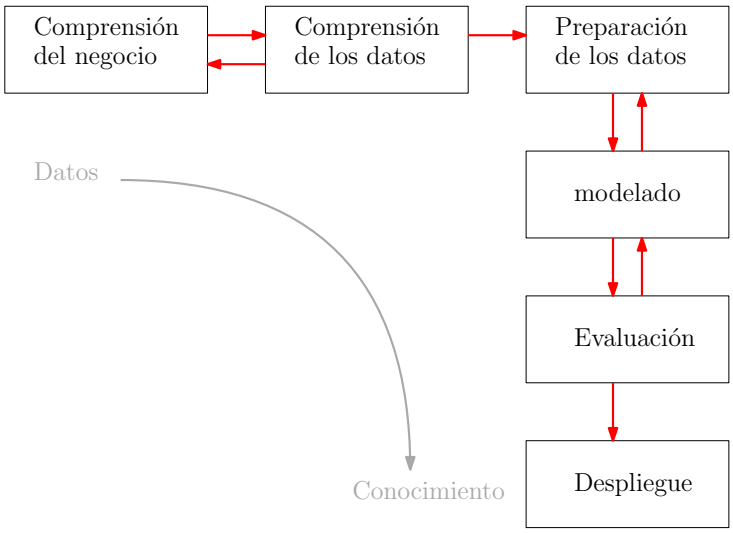


Figura 2: Fases de la metodología CRISP-DM.

2.1. Comprensión del negocio

En esta fase trataremos de conseguir, desde una perspectiva de negocio, cuáles son los objetivos del mismo, tratando de evitar el gran error de dedicar el esfuerzo de todo el proyecto a proporcionar respuestas correctas a preguntas equivocadas.

Con los objetivos de negocio en mente, elaboraremos un estudio de la situación actual del negocio respecto de los objetivos planteados. En este punto, trataremos de

clarificar recursos, requerimientos y limitaciones, para así poder concretar objetivos de la minería de datos que contribuyan claramente a la consecución de los objetivos primarios.

Finalmente, elaboraremos un plan de proyecto en el que detallaremos las fases, tareas y actividades que nos deberán llevar a alcanzar los objetivos planteados.

En esta fase deberemos ser capaces de:

- Establecer los objetivos de negocio.
- Evaluar la situación actual.
- Fijar los objetivos a nivel de minería de datos.
- Obtener un plan de proyecto.

2.2. Comprensión de los datos

Comprensión se refiere a trabajar los datos con el objetivo de familiarizarse al máximo con ellos, saber de dónde provienen, en qué condiciones nos llegan, cuál es su estructura, qué propiedades tienen, qué inconvenientes presentan y cómo podemos mitigarlos o eliminarlos.

Se trata de una fase crítica puesto que es donde trabajamos de lleno con la calidad de los datos, que por otro lado debemos ver como la materia prima para la minería de datos.

Tener una buena calidad de los datos será siempre una condición necesaria aunque no suficiente para tener éxito en el proyecto.

En esta fase deberemos ser capaces de:

- Ejecutar procesos de captura de datos.
- Proporcionar una descripción del juego de datos.
- Realizar tareas de exploración de datos.
- Gestionar la calidad de los datos, identificando problemas y proporcionando soluciones.

2.3. Preparación de los datos

El objetivo de esta fase es disponer del juego de datos final sobre el que se aplicarán los modelos. Además, también se desarrollará la documentación descriptiva necesaria sobre el juego de datos.

Deberemos dar respuesta a la pregunta ¿qué datos son los más apropiados para alcanzar los objetivos marcados? Esto significa evaluar la relevancia de los datos, la calidad de los mismos y las limitaciones técnicas que se puedan derivar de aspectos como el volumen de datos. Documentaremos los motivos tanto para incluir datos, como para excluirlos.

Nos replantearemos los criterios de selección de datos basándonos, por un lado, en la experiencia adquirida en el proceso de exploración de datos, y por otro lado, en la experiencia adquirida en el proceso de modelado.

Consideraremos el uso de técnicas estadísticas de muestreo y técnicas de relevancia de atributos, que nos ayudarán, por ejemplo, a plantear la necesidad de iniciar actividades de reducción de la dimensionalidad.

Prestaremos atención a la incorporación de datos de diferentes fuentes y por supuesto a la gestión del ruido.

En esta fase deberemos ser capaces de:

- Establecer el universo de datos con los que trabajar.
- Realizar tareas de limpieza de datos.
- Construir un juego de datos apto para ser usado en modelos de minería de datos.
- Integrar datos de fuentes heterogéneas si es necesario.

2.4. Modelado

El objetivo último de esta fase será el de disponer de un modelo que nos ayude a alcanzar los objetivos de la minería de datos y los objetivos de negocio establecidos en el proyecto. Podemos entender el modelo como la habilidad de aplicar una técnica a un juego de datos con el objetivo de predecir una variable objetivo o encontrar un patrón desconocido.

El hecho de que esta fase entre en iteración tanto con su antecesora, la preparación de los datos, como con su sucesora, la evaluación del modelo, nos da una idea de la importancia de la misma en términos de la calidad del proyecto.

Dado un problema en el ámbito de la minería de datos, pueden existir una o varias técnicas que den respuesta al mismo, por ejemplo:

- Un problema de segmentación puede aceptar técnicas de *clustering*, de redes neuronales o simplemente técnicas de visualización.
- Un problema de clasificación puede aceptar técnicas de análisis discriminante, de árboles de decisión, de redes neuronales, máquinas de soporte vectorial o de k-NN.
- Un problema de regresión aceptará técnicas de análisis de regresión, de árboles de regresión, de redes neuronales o de k-NN.
- Un problema de análisis de dependencias puede afrontarse con técnicas de análisis de correlaciones, análisis de regresión, reglas de asociación, redes bayesianas o técnicas de visualización.

En definitiva, un mismo problema puede resolverse con varias técnicas y una técnica puede servir para resolver varios problemas.

En esta fase deberemos ser capaces de:

- Seleccionar las técnicas de modelado más adecuadas para nuestro juego de datos y nuestros objetivos.
- Fijar una estrategia de verificación de la calidad del modelo.
- Construir un modelo a partir de la aplicación de las técnicas seleccionadas sobre el juego de datos.
- Ajustar el modelo evaluando su fiabilidad y su impacto en los objetivos anteriormente establecidos.

2.5. Evaluación del modelo

En fases anteriores nos hemos preocupado de asegurar la fiabilidad y plausibilidad del modelo, en cambio en esta fase nos centraremos en evaluar el grado de acercamiento a los objetivos de negocio y en la búsqueda, si las hay, de razones de negocio por las cuales el modelo es ineficiente.

Una forma esquemática y gráfica de visualizar el propósito de un proyecto de minería de datos es pensar en la siguiente ecuación:

$$\text{Resultados} = \text{Modelos} + \text{Descubrimientos}$$

Es decir, el propósito de un proyecto de minería de datos no son sólo los modelos, que son por supuesto importantes, sino también los descubrimientos, que podríamos definir como cualquier cosa aparte del modelo que contribuye a alcanzar los objetivos de negocio o que contribuye a plantear nuevas preguntas, que a su vez son decisivas para alcanzar los objetivos de negocio.

Siempre y cuando sea posible probaremos el modelo en entornos de prueba para asegurarnos de que el posterior proceso de despliegue se realiza satisfactoriamente y para asegurarnos también de que el modelo obtenido es capaz de dar respuesta a los objetivos de negocio.

Estableceremos un *ranking* de resultados con respecto a los criterios de éxito con relación al grado de cumplimiento de los objetivos de negocio.

Adicionalmente, también emitiremos opinión sobre otros descubrimientos que se hayan realizado aparte del modelado, que aunque probablemente no contribuyan directamente a los objetivos planteados, quizá puedan abrir puertas a nuevos planteamientos y líneas de trabajo.

En esta fase deberemos ser capaces de:

- Evaluar el modelo o modelos generados hasta el momento.
- Revisar todo el proceso de minería de datos que nos ha llevado hasta este punto.
- Establecer los siguientes pasos a tomar, tanto si se trata de repetir fases anteriores como si se trata de abrir nuevas lineas de investigación.

2.6. Despliegue

En esta fase organizaremos y ejecutaremos tanto las tareas propias del despliegue de los resultados como del mantenimiento de las nuevas funcionalidades, una vez el despliegue haya finalizado.

El plan deberá contemplar todas las tareas a realizar en el proceso de despliegue de resultados, e incorporará medidas alternativas en forma de planes alternativos o versiones del plan inicial, que deberán permitir tener varias visiones y escoger la mejor.

Deberemos definir cómo el conocimiento obtenido en forma de resultados será propagado hacia los usuarios interesados.

En el caso de que haya que instalar o distribuir *software* por nuestros sistemas, deberemos gestionarlo para minimizar posibles efectos negativos y planificarlo para que se ejecute con suficiente antelación.

Habrá que prever cómo mediremos el beneficio producido por el despliegue y cómo monitorizaremos todo el proceso.

Identificaremos los posibles inconvenientes que pueda ocasionar nuestro despliegue.

En esta fase deberemos ser capaces de:

- Diseñar un plan de despliegue de modelos y conocimiento sobre nuestra organización.
- Realizar seguimiento y mantenimiento de la parte más operativa del despliegue.
- Revisar el proyecto en su globalidad con el objetivo de identificar lecciones aprendidas.

3. Tipología de tareas y métodos

El aprendizaje automático (más conocido por su denominación en inglés, *machine learning*, ML) es el conjunto de métodos y algoritmos que permiten a una máquina aprender de manera automática en base a experiencias pasadas.

En este capítulo veremos algunos conceptos básicos de aprendizaje automático, en general, que son especialmente relevantes para el uso de redes neuronales artificiales. En este sentido, empezaremos viendo la tipología de métodos y tareas existentes en aprendizaje automático. A continuación, revisaremos algunas medidas básicas de preprocesamiento de datos y discutiremos muy brevemente la creación del conjunto de entrenamiento y test. Finalizaremos este capítulo revisando algunas medidas de evaluación de modelos que nos resultarán de gran utilidad posteriormente.

En esta sección presentaremos, en primer lugar, las principales tareas que se pueden resolver utilizando técnicas de minería de datos. Veremos las características básicas de estas tareas y los objetivos principales que persiguen. En segundo lugar, describiremos brevemente los dos grandes grupos de métodos que existen en la minería de datos y el aprendizaje automático (*machine learning*).

3.1. Tipología de tareas

Según el objetivo de nuestro análisis, podemos distinguir entre tres grandes grupos de tareas, que revisaremos brevemente a continuación.

3.1.1. Clasificación

La clasificación (*classification*) es uno de los procesos cognitivos importantes, tanto en la vida cotidiana como en los negocios, donde podemos clasificar clientes, empleados, transacciones, tiendas, fábricas, dispositivos, documentos o cualquier otro tipo de instancias en un conjunto de clases o categorías predefinidas con anterioridad.

La tarea de clasificación consiste en asignar instancias de un dominio dado, descritas por un conjunto de atributos discretos o de valor continuo, a un conjunto de clases, que pueden ser consideradas valores de un atributo discreto seleccionado, generalmente denominado **clase**. Las etiquetas de clase correctas son, en general, desconocidas, pero se proporcionan para un subconjunto del dominio. Por lo tanto, queda claro que es necesario disponer de un subconjunto de datos correctamente etiquetado, y que se usará para la construcción del modelo.

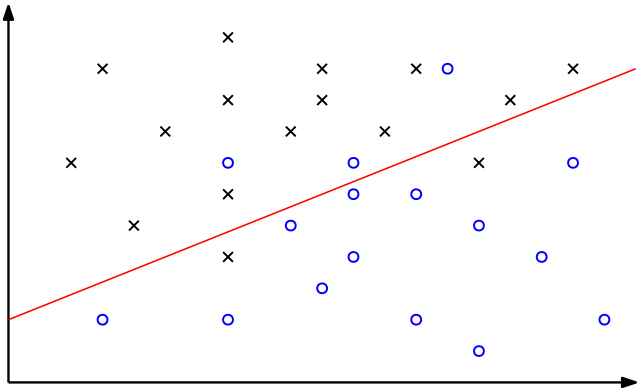
La función de clasificación puede verse como:

$$c : X \to C \tag{1}$$

donde c representa la función de clasificación, X el conjunto de atributos que forman una instancia y C la etiqueta de clase de dicha instancia.

Un tipo de clasificación particularmente simple, pero muy interesante y ampliamente estudiado, hace referencia a los problemas de clasificación binarios, es decir, problemas con un conjunto de datos pertenecientes a dos clases, i.e. $C = \{0, 1\}$. La figura 3 muestra un ejemplo de clasificación binaria, donde las cruces y los círculos representan elementos de dos clases, y se pretende dividir el espacio de tal forma que separe a la mayoría de elementos de clases diferentes.

Figura 3. Ejemplo de clasificación

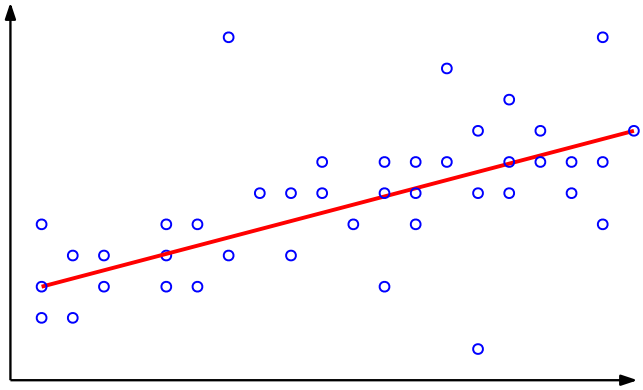


3.1.2. Regresión

Al igual que la clasificación, la regresión (*regresion*) es una tarea de aprendizaje inductivo que ha sido ampliamente estudiada y utilizada. Se puede definir, de forma informal, como un problema de "clasificación con clases continuas". Es decir, los modelos de regresión predicen valores numéricos en lugar de etiquetas de clase discretas. A veces también nos podemos referir a la regresión como "predicción numérica".

La tarea de regresión consiste en asignar valores numéricos a instancias de un dominio dado, descritos por un conjunto de atributos discretos o de valor continuo, como se muestra en la figura 4, donde los puntos representan los datos de aprendizaje y la línea representa la predicción sobre futuros eventos. Se supone que esta asignación se aproxima a alguna función objetivo, generalmente desconocida, excepto para un subconjunto del dominio. Este subconjunto se puede utilizar para crear el modelo de regresión.

Figura 4. Ejemplo de regresión lineal



En este caso, la función de regresión se puede definir como:

$$f : X \to \mathbb{R} \tag{2}$$

donde f representa la función de regresión, X el conjunto de atributos que forman una instancia y $\mathbb{R}$ un valor en el dominio de los números reales.

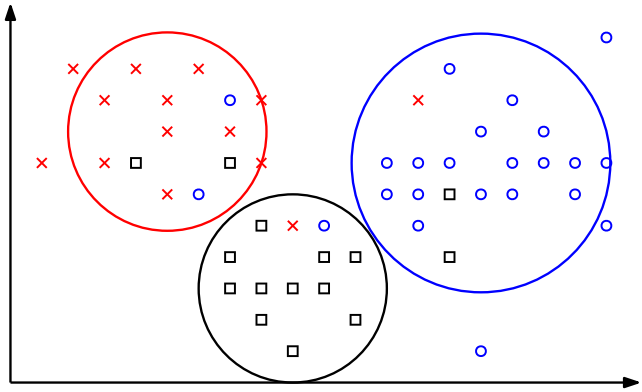
Es importante remarcar que una regresión no pretende devolver una predicción exacta sobre un evento futuro, sino una aproximación (como muestra la diferencia entre la línea y los puntos de la figura). Por lo general, datos más dispersos resultarán en predicciones menos ajustadas.

3.1.3. Agrupamiento

El agrupamiento (*clustering*) es una tarea de aprendizaje inductiva que, a diferencia de las tareas de clasificación y regresión, no dispone de una etiqueta de clase a predecir. Puede considerarse como un problema de clasificación, pero donde no existen un conjunto de clases predefinidas, y éstas se "descubren" de forma autónoma por el método o algoritmo de agrupamiento, basándose en patrones de similitud identificados en los datos.

La tarea de agrupamiento consiste en dividir un conjunto de instancias de un dominio dado, descrito por un número de atributos discretos o de valor continuo, en un conjunto de grupos (*clusters*) basándose en la similitud entre las instancias, y crear un modelo que puede asignar nuevas instancias a uno de estos grupos o *clusters*. La figura 5 muestra un ejemplo de agrupamiento, donde las cruces, círculos y cuadros pertenecen a tres clases de elementos distintos que pretendemos agrupar.

Figura 5. Ejemplo de agrupamiento



Un proceso de agrupación puede proporcionar información útil sobre los patrones de similitud presentes en los datos, como por ejemplo, segmentación de clientes o creación de catálogos de documentos. Otra de las principales utilidades del agrupamiento es la detección de anomalías. En este caso, el método de agrupamiento permite distinguir instancias con un patrón distinto a las demás instancias "normales" del conjunto de datos, facilitando la detección de anomalías y posibilitando la generación de alertas automáticas para nuevas instancias que no corresponden a ningún *cluster* existente.

La función de agrupamiento o *clustering* se puede modelar mediante:

$$h : X \rightarrow C_h \tag{3}$$

donde h representa la función de agrupamiento, X el conjunto de atributos que forman una instancia y C_h un conjunto de grupos o *clusters*. Aunque esta definición se parece mucho a la tarea de clasificación, una diferencia aparentemente pequeña, pero muy importante, consistente en que el conjunto de "clases" no está predeterminado ni es conocido *a priori*, sino que se identifica como parte de la creación del modelo.

3.2. Tipología de métodos

Generalmente, un algoritmo de aprendizaje automático debe construir un modelo en base a un conjunto de datos de entrada que representan el conjunto de aprendizaje, lo que se conoce como *conjunto de entrenamiento*. Durante esta fase de aprendizaje, el algoritmo va comparando la salida de los modelos en construcción con la salida ideal que deberían tener estos modelos, para ir ajustándolos y aumentando la precisión. Esta comparación forma la base del aprendizaje en sí, y puede ser **supervisado** o **no supervisado**.

Los algoritmos de minería de datos se dividen en dos grandes grupos:

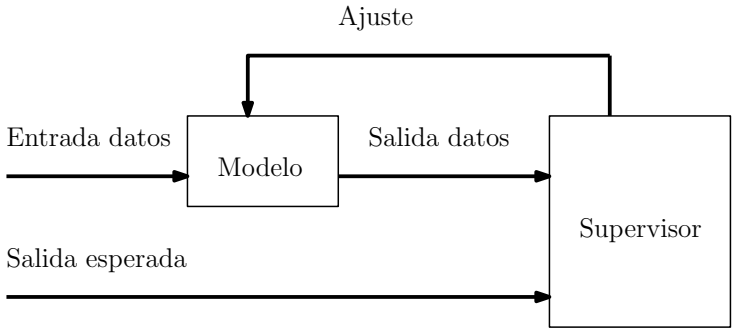
- Los **métodos supervisados**, que requieren de un conjunto de datos previamente etiquetado con el conjunto de clases.

- Los **métodos no supervisados**, donde los datos no tienen ningún etiqueta o clasificación previa.

3.2.1. Métodos supervisados

En el aprendizaje supervisado (figura 6), hay un componente externo que compara los resultados obtenidos por el modelo con los resultados esperados por éste, y proporciona retroalimentación al modelo para que vaya ajustándose. Para ello, pues, será necesario proporcionar al modelo con un conjunto de datos de entrenamiento que contenga tanto los datos de entrada como la salida esperada para cada uno de esos datos.

Figura 6. Aprendizaje supervisado



Todas ellas se basan en el paradigma del aprendizaje inductivo. La esencia de cada una de ellas es derivar inductivamente a partir de los **datos** (que representan la información del entrenamiento), un **modelo** (que representa el conocimiento) que tiene utilidad predictiva, es decir, que puede aplicarse a nuevos datos.

Las grandes familias de algoritmos de aprendizaje supervisado son:

- Algoritmos de **clasificación**, indicados cuando el atributo objetivo es categórico.

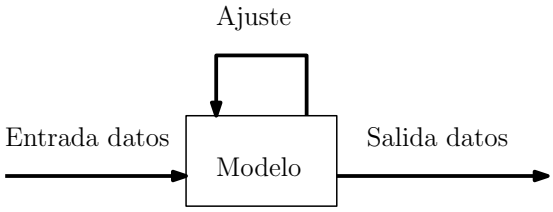
- Algoritmos de **regresión**, indicados cuando el atributo objetivo es numérico.

Estos dos grandes grupos corresponden, en esencia, con la tipología de problemas descrita en la sección anterior.

3.2.2. Métodos no supervisados

En el aprendizaje no supervisado (figura 7), el algoritmo de entrenamiento aprende sobre los propios datos de entrada, descubriendo y agrupando patrones, características, correlaciones, etc.

Figura 7. Aprendizaje no supervisado



Los métodos no supervisados (*unsupervised methods*) son algoritmos que basan su proceso de entrenamiento en un juego de datos sin etiquetas o clases previamente definidas. Es decir, *a priori* no se conoce ningún valor objetivo o de clase, ya sea categórico o numérico.

El aprendizaje no supervisado está dedicado a las tareas de agrupamiento, también llamadas *clustering* o segmentación, donde su objetivo es encontrar grupos similares en los juegos de datos.

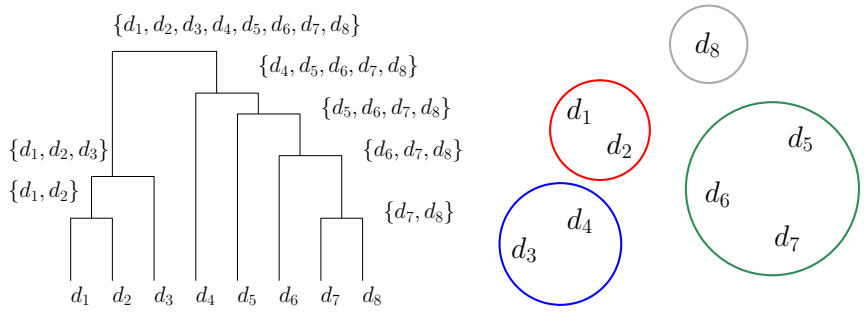
Existen dos grupos principales de métodos o algoritmos de agrupamiento:

1) Los **métodos jerárquicos**, que producen una organización jerárquica de las instancias que forman el conjunto de datos, posibilitando de esta forma distintos niveles de agrupación.

2) Los **métodos particionales** o no jerárquicos, que generan grupos de instancias que no responden a ningún tipo de organización jerárquica.

En la Figura 8 podemos distinguir de forma visual la diferencia entre los métodos de agrupamiento jerárquico y los métodos no jerárquicos o particionales. La principal diferencia es que los métodos jerárquicos permiten escoger el nivel de granularidad, es decir, dependiendo de la "altura" se obtiene un distinto número de conjuntos con distinto número de elementos.

Figura 8. Tipologías de métodos de agrupamiento



(a) Agrupamiento jerárquico (b) Agrupamiento particional

Ambos métodos proporcionan una forma directa de descubrir y representar la similitud en los datos, formando conjuntos de datos "similares" entre ellos. El concepto de "similitud" deberá ser escogido para cada problema y representado según una métrica

concreta. Una vez creados los grupos, estos algoritmos también pueden predecir a qué conjunto corresponde una nueva instancia, lo que les permite implementar análisis predictivos (*predictive analysis*). Para este tipo de tarea se suele emplear la misma medida de similitud entre la nueva instancia y los conjuntos definidos.

3.2.3. Resumen de métodos y tareas

Finalmente, la Tabla 1 identifica los principales métodos de minería de datos, indicando en cada caso la tipología de algoritmo y de tarea o problema que puede resolver. En los siguientes capítulos veremos en detalle cada uno de estos métodos.

Tabla 1. Tipología de los algoritmos de minería de datos

Métodos	Supervisado		No supervisado
	Clasificación	Regresión	Agrupamiento
Agrupamiento jerárquico			X
k-means y derivados			X
k-NN	X		
SVM	X	X	
Redes neuronales	X	X	
Árboles de decisión	X	X	
Métodos probabilísticos	X	X	

4. Datos de entrenamiento y test

En este capítulo se describe el procedimiento previo a la construcción de modelos una vez los datos ya han sido preparados y se consideran adecuados. El objetivo es asegurar que los modelos construidos a partir de los datos disponibles funcionen correctamente para nuevos datos que haya que procesar (clasificar, agrupar, etc.) en un futuro, es decir, asegurar que el modelo es válido y capaz de ser usado en producción.

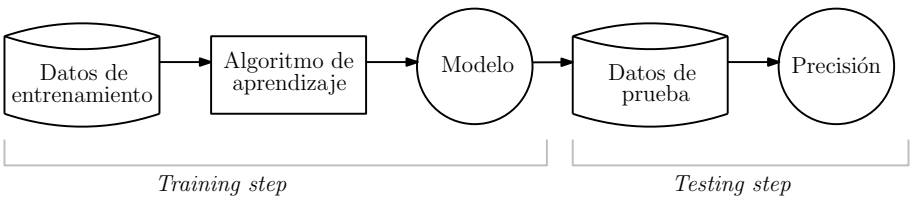
4.1. Conjuntos de entrenamiento y test

Para validar un algoritmo de aprendizaje o modelo es necesario asegurar que éste funcionará correctamente para los datos de prueba o test futuros, de forma que capture la esencia del problema a resolver y generalice correctamente. En esencia se trata de evitar que sea dependiente de los datos utilizados durante su entrenamiento, evitando el problema conocido como "sobreentrenamiento" (en inglés, *overfitting*).

El **sobreentrenamiento** se puede definir, informalmente, como el peligro que corremos al sobreentrenar un modelo es que éste acabe respondiendo estrictamente a las propiedades del juego de datos de entrenamiento y que sea incapaz de extrapolarse con niveles de acierto adecuados a otros juegos de datos que puedan aparecer en un futuro.

En la figura 9, queremos subrayar que cuando hablamos de datos de entrenamiento y de test, estamos refiriéndonos en exclusiva a los algoritmos de aprendizaje supervisado, ya que es necesario evaluar los resultados obtenidos sobre datos etiquetados nunca vistos anteriormente y se pueden comparar los errores cometidos para cada conjunto.

Figura 9. Proceso de creación y validación de un modelo basado en aprendizaje supervisado



Así, usaremos el conjunto de datos de entrenamiento para crear el modelo supervisado, mientras que el conjunto de datos de test se usará para medir la precisión alcanzada por el modelo. Formará parte del proceso de construcción del modelo la repetición iterativa de entrenamiento y verificación hasta conseguir unos niveles de precisión y de capacidad de predicción aceptables.

Habitualmente los juegos de datos de entrenamiento y de test suelen ser extracciones aleatorias del juego de datos inicial. En función del número de datos disponibles, existen diferentes técnicas para la construcción de los conjuntos de entrenamiento y de prueba. Se trata de un compromiso entre la robustez del modelo construido (a mayor número de datos usados para el entrenamiento, más robusto será el modelo) y su capacidad de generalización (a mayor número de datos usados para la validación, más fiable será la estimación del error cometido).

4.1.1. Leave-One-Out

Tal y como su nombre indica, la técnica *Leave-One-Out* consiste en utilizar todos los datos menos uno para construir el modelo y utilizar el dato no usado para evaluarlo. Obviamente, si el número de datos es n, este proceso puede repetirse n veces, siendo posible calcular el error promedio de los n modelos construidos.

Este método suele usarse cuando n es pequeño, del orden de cientos de datos o menos, dada la necesidad de construir tantos modelos como datos, lo cual puede ser computacionalmente muy costoso.

4.1.2. Leave-p-Out

Una generalización del método anterior es utilizar un subconjunto de $p \ll n$, de forma que se utilizan $n - p$ datos para entrenar el modelo y p para validarlo, siguiendo el mismo proceso. El problema es que el número de subconjuntos posibles de p elementos tomados de un conjunto de n crece exponencialmente, con lo cual para n y/o p grandes el problema puede ser intratable computacionalmente, por lo que p suele ser muy pequeño. Por ejemplo, si $n = 100$ y $p = 10$, el número de modelos a construir sería de:

$$\binom{100}{10} = \frac{100!}{90!\,10!} = 17310309456440 \tag{4}$$

Obviamente, dicha cantidad está fuera de toda consideración. Incluso con valores más pequeños de p el número de modelos a construir es enorme si n es grande, por lo que este método queda restringido a conjuntos realmente pequeños donde no pueden usarse las técnicas descritas a continuación.

4.1.3. k-fold cross validation

Para conjuntos del orden de cientos o miles de muestras, una solución de compromiso es realizar una partición aleatoria del conjunto de datos en k conjuntos del mismo tamaño, usando $k-1$ conjuntos para entrenar el modelo y el conjunto restante para evaluarlo, repitiendo el proceso k veces y promediando el error estimado. Nótese que el método *Leave-One-Out* descrito anteriormente es el caso particular $k = n$. Este método permite trabajar con conjuntos más grandes, incluso hasta decenas de miles de elementos.

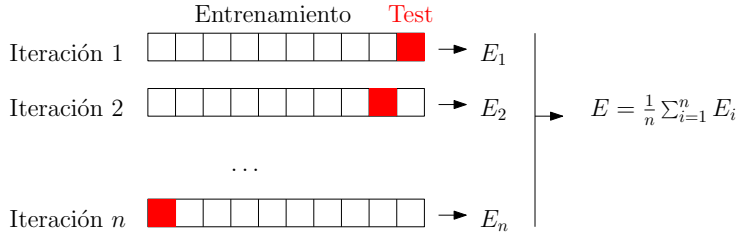
Habitualmente $k = 5$ o $k = 10$, aunque no existe ninguna base teórica que sustente dichos valores, sino que es resultado de la experimentación (Kohavi, 1995). También es típico usar $k = 3$, dando lugar a la que se conoce como regla de los dos tercios, debido al tamaño relativo del conjunto de entrenamiento con respecto al conjunto de datos original.

Lectura complementaria

R. Kohavi. "A study of cross-validation and bootstrap for accuracy estimation and model selection". In Proc. of IJCAI, pp. 1137-1143, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

La figura 10 muestra el esquema de validación cruzada con $k = 10$.

Figura 10. Esquema de división de conjuntos de entrenamiento y test según el método *k-fold cross validation*



Por otra parte, dado que la partición en k conjuntos es aleatoria, este proceso podría repetirse un cierto número de ocasiones, promediando todos los errores cometidos (que, de hecho, ya eran promedios del resultado de aplicar *k-fold cross validation*), siempre y cuando se utilice el mismo valor de k.

Como siempre que hay una partición aleatoria, es importante comprobar que la distribución de los valores de la variable objetivo sea similar, para evitar la creación de modelos muy sesgados. De hecho, la partición aleatoria se debe realizar teniendo en cuenta dicha distribución, para evitar el posible sesgo.

4.1.4. Otras consideraciones

En el caso de que el valor de n sea muy grande, del orden de cientos de miles o millones de datos, pueden usarse otros criterios para la partición de los datos originales en un conjunto de entrenamiento y otro de prueba, intentando reducir el coste computacional de estimar el error cuando se deben generar diferentes modelos. Así, una opción habitual es utilizar la regla de los dos tercios en una única partición, entrenando una única vez con dos tercios de los datos originales y validando con el tercio restante. Además, conforme aumenta el número de datos n, se puede reducir el

tamaño del conjunto de test, utilizando una fracción más pequeña (p.e. un quinto), incrementando así el número de datos usados como entrenamiento para construir el modelo y mejorando su robustez.

Por otra parte, la dimensionalidad m del conjunto de datos también juega un papel relevante. Es aconsejable que el numero de datos n satisfaga $n \gg m$, recomendando habitualmente que $n > 10m$, o incluso más (20 o 30 veces) en función del tipo de problema a resolver (clasificación, regresión, etc.). Por lo tanto, para valores de n pequeños la única opción posible es utilizar técnicas para reducir la dimensionalidad m del conjunto de entrada, mejorando así el ratio n/m.

Finalmente, la complejidad intrínseca del modelo construido también determina el número de datos necesarios para asegurar la robustez del modelo construido. Mientras más complejo sea el modelo y más parámetros necesite para su construcción, mayor será el número necesario de elementos del conjunto de entrenamiento.

4.2. Conjunto de validación

En algunos casos es posible afinar ciertos parámetros del modelo construido, para intentar mejorar su eficiencia. Por ejemplo, en el caso del modelo k-NN es posible probar diferentes valores de k y también diferentes métricas usadas como distancia, en función de la naturaleza de los datos.

En este caso, es necesario disponer de tres conjuntos diferentes, los dos ya comentados conjuntos de entrenamiento (para construir el modelo) y de test (usado únicamente para estimar el error cometido por el modelo ante datos nuevos) y un tercero llamado de validación, con el cual el mejor modelo (con respecto al error estimado mediante el conjunto de test) es afinado antes de ser validado mediante el conjunto de test.

Desafortunadamente, en la literatura del tema es habitual usar indistintamente conjunto de test o conjunto de validación dado el uso de cada conjunto. No obstante, siguiendo el esquema representado por la figura 9, la idea es que cada etapa donde se toma una decisión con respecto al modelo construido utilice un conjunto de datos diferente, para asegurar su capacidad de generalización.

Como resumen, el lector debe tener una idea clara: la evaluación final del modelo construido debe hacerse con un conjunto de datos que no haya sido usado de ninguna manera durante su proceso de construcción.

5. Evaluación de modelos

En esta sección revisaremos los principales indicadores o métricas para evaluar los resultados de un proceso de minería de datos, en general, y de un modelo basado en redes neuronales, en particular. El objetivo es cuantificar el grado o valor de "bondad" de la solución encontrada, permitiendo la comparación entre distintos métodos sobre los mismos conjuntos de datos.

Las métricas para realizar este tipo de evaluación dependen, principalmente, del tipo de problema con el que se está lidiando. En este sentido, veremos métricas específicas para problemas de clasificación y regresión.

5.1. Modelos de clasificación

Las medidas de calidad de modelos de clasificación se calculan comparando las predicciones generadas por el modelo en un conjunto de datos D con las etiquetas de clase verdaderas de las instancias de este conjunto de datos.

5.1.1. Matriz de confusión

La matriz de confusión (*confusion matrix*, CM) presenta en una tabla una visión gráfica de los errores cometidos por el modelo de clasificación. Se trata de un modelo gráfico para visualizar el nivel de acierto de un modelo de predicción. También es conocido en la literatura como tabla de contingencia o matriz de errores.

Figura 11. Matriz de confusión binaria

		Clase predicha	
		P	N
Clase verdadera	P	TP	FN
	N	FP	TN

La figura 11 presenta la matriz de confusión para el caso básico de clasificación binaria. En esencia, esta matriz indica el número de instancias correcta e incorrectamente clasificadas. Los parámetros que nos indica son:

- Verdadero positivo (*true positive*, TP): número de clasificaciones correctas en la

clase positiva (P).

- Verdadero negativo (*true negative*, TN): número de clasificaciones correctas en la clase negativa (N).

- Falso negativo (*false negative*, FN): número de clasificaciones incorrectas de clase positiva clasificada como negativa.

- Falso positivo (*false positive*, FP): número de clasificaciones incorrectas de clase negativa clasificada como positiva.

Como se puede observar, en la diagonal de la matriz aparecen los aciertos del modelo, ya sean positivos o negativos.

5.1.2. Matriz de confusión para k clases

La matriz de confusión se puede extender a más de dos clases de forma natural, tal y como podemos ver en la Figura 12. Esta representación nos permite identificar de forma rápida el número de instancias correctamente clasificadas, que se corresponde con la diagonal de la tabla.

Figura 12. Matriz de confusión para k clases

		Clase predicha		
		C_1	...	C_k
Clase verdadera	C_1			
	...			
	C_k			

Por otro lado, en el caso de problemas de clasificación que impliquen más de dos clases, se puede realizar la evaluación de dos formas distintas:

- *1-vs-1* (OvO): Midiendo la capacidad de discriminar entre instancias de una clase, considerada la clase positiva, frente a las instancias de otra clase, consideradas negativas.

- *1-vs-All* (OvA): Midiendo la capacidad de discriminar entre instancias de una clase, considerada la clase positiva, frente a las instancias de las demás clases, consideradas negativas.

La aproximación de *1-vs-1* nos conduce a una matriz de confusión, con formato 2×2, para cada par de clases existentes. Por otra parte, la aproximación *1-vs-All* produce

una matriz de confusión 2×2 para cada clase.

5.1.3. Métricas derivadas de la matriz de confusión

A partir de la matriz de confusión, definimos un conjunto de métricas que permiten cuantificar la bondad de un modelo de clasificación.

- El **error de clasificación** (*misclassification error*, ERR) y la **exactitud** (*accuracy*, ACC) proporcionan información general sobre el número de instancias incorrectamente clasificadas. El error (ecuación 5) es la suma de las predicciones incorrectas sobre el número total de predicciones. Por el contrario, la exactitud (*accuracy*) es el número de predicciones correctas sobre el número total de predicciones, como se puede ver en la ecuación 6.

$$ERR = \frac{FP + FN}{FP + FN + TP + TN} \tag{5}$$

$$ACC = \frac{TP + TN}{FP + FN + TP + TN} = 1 - ERR \tag{6}$$

- En algunos problemas nos puede interesar medir el error en los falsos positivos o negativos. Por ejemplo, en un sistema de diagnosis de tumores, nos interesa centrarnos en los casos de tumores malignos que han sido clasificados incorrectamente como tumores benignos. En estos casos, la **tasa de verdaderos positivos** (*True positive rate*, TPR) y la **tasa de falsos positivos** (*False positive rate*, FPR), que definimos a continuación, pueden ser muy útiles:

$$TPR = \frac{TP}{FN + TP} \tag{7}$$

$$FPR = \frac{FP}{FP + TN} \tag{8}$$

- La **precisión** (*precision*, PRE) mide el rendimiento relacionado con las tasas de

verdaderos positivos y negativos, tal y como podemos ver a continuación.

$$PRE = \frac{TP}{TP + FP} \tag{9}$$

- El **recall** (*recall*, REC) y la **sensibilidad** (*sensitivity*, SEN) se corresponden con la tasa de verdaderos positivos (TPR), mientras que la **especificidad** (*specificity*, SPE) se define como la tasa de instancias correctamente clasificadas como negativas respecto a todas las instancias negativas.

$$REC = SEN = TPR = \frac{TP}{FN + TP} \tag{10}$$

$$SPE = \frac{TN}{TN + FP} = 1 - FPR \tag{11}$$

- Finalmente, en la práctica es habitual combinar la precisión y el recall en una métrica llamada **F1** (*F1 score*), que se define de la siguiente forma:

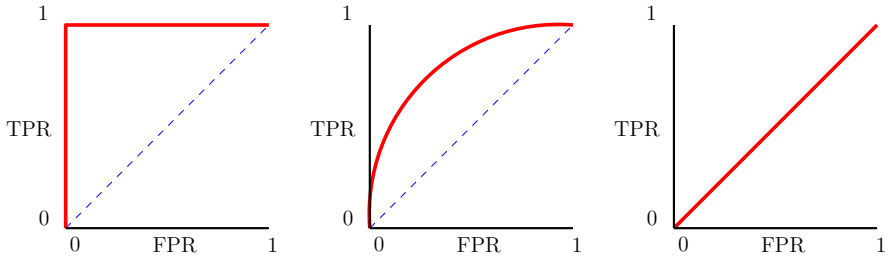
$$F1 = 2 \times \frac{PRE \times REC}{PRE + REC} \tag{12}$$

5.1.4. Curvas ROC

Una curva ROC (acrónimo de *Receiver Operating Characteristic*) mide el rendimiento respecto a los falsos positivos (FP) y verdaderos positivos (TP). La diagonal de la curva ROC se interpreta como un modelo generado aleatoriamente, mientras que valores inferiores se consideran peores que una estimación aleatoria de los nuevos datos.

En esta métrica, un clasificador perfecto ocuparía la posición superior izquierda de la gráfica, con una tasa de verdaderos positivos (TPR) igual a 1 y una tasa de falsos positivos (FPR) igual a 0. A partir de la curva ROC se calcula el **área bajo la curva** (*Area Under the Curve*, AUC) que permite caracterizar el rendimiento de un

Figura 13. Ejemplo de curvas ROC



(a) AUC = 1 (b) AUC ≈ 0.8 (c) AUC = 0.5

modelo de clasificación. La Figura 13 ejemplifica un rendimiento excelente, bueno y malo (aleatorio) de una curva ROC.

A modo de guía para interpretar las curvas ROC se han establecido los siguientes intervalos para los valores de AUC:

- [0,5, 0,6): Test malo
- [0,6, 0,75): Test regular
- [0,75, 0,9): Test bueno
- [0,9, 0,97): Test muy bueno
- [0,97, 1): Test excelente

5.2. Modelos de regresión

La evaluación de modelos de regresión comparte el mismo principio que la evaluación de modelos de clasificación, en el sentido de que se compara los valores predichos con los valores reales de las instancias del conjunto de datos. En esta sección veremos algunas de las métricas más empleadas para evaluar el rendimiento de un modelo de regresión.

- El **error absoluto medio** (*mean absolute error*, MAE) es la métrica más simple y directa para la evaluación del grado de divergencia entre dos conjuntos de valores, representado por la ecuación:

$$MAE = \frac{1}{|D|} \sum_{d \in D} |f(d) - h(d)| \tag{13}$$

donde D es el conjunto de instancias, $h : X \to \mathbb{R}$ es la función del modelo y

$f : X \to \mathbb{R}$ es la función objetivo con las etiquetas correctas de las instancias.

En este caso, todos los residuos tienen la misma contribución al error absoluto final.

- El **error cuadrático medio** (*mean square error*, MSE) es, probablemente, la métrica más empleada para la evaluación de modelos de regresión. Se calcula de la siguiente forma:

$$MSE = \frac{1}{|D|} \sum_{d \in D} (f(d) - h(d))^2 \tag{14}$$

Utilizando esta métrica se penaliza los residuos grandes. En este sentido, si el modelo aproxima correctamente a gran parte de las instancias del conjunto de datos, pero comete importantes errores en unos pocos, la penalización en esta métrica será muy superior a la indicada si se emplea la métrica anterior (MAE).

- La **raíz cuadrada del error cuadrático medio** (*root mean square error*, RMSE) se define aplicando la raíz cuadrada al error cuadrático medio (MSE). Tiene las mismas características que éste, pero aporta la ventaja adicional de que el error se expresa en la misma escala que los datos originales. En el caso del MSE no era así, ya que la diferencia de valores son elevados al cuadrado antes de realizar la suma ponderada. En algunos casos esta diferencia puede ser importante, mientras que irrelevante en otros.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{|D|} \sum_{d \in D} (f(d) - h(d))^2} \tag{15}$$

- Finalmente, en algunos modelos de regresión se pueden tolerar diferencias importantes entre los valores predichos y verdaderos, siempre que el modelo tenga un comportamiento general similar a los valores verdaderos, prestando especial atención a la monotonicidad. En estos casos, las métricas vistas anteriormente pueden no ser representativas del rendimiento de un modelo de regresión. Por el contrario, los **índices de correlación** lineal o de rango, como por ejemplo Pearson o Spearman, pueden ser una buena métrica para medir la bondad del modelo generado.

5.3. Modelos de agrupamiento

En el caso de los modelos de agrupamiento, de forma contraria a las dos secciones vistas anteriormente, no se dispone de "etiquetas" que indiquen *a priori* la clase a la que pertenecen las instancias de entrenamiento o test. Por lo tanto, las métricas que veremos en esta sección intentan capturar el grado de similitud o disimilitud de los patrones detectados en el conjunto de datos. Generalmente, este tipo de métricas son referenciadas como métricas de calidad.

5.3.1. Métricas de calidad por partición

Este conjunto de métricas de calidad tienen como objetivo medir el nivel de cohesión de una partición (o *cluster*) y/o el nivel de separación de ésta con las demás particiones obtenidas en el proceso de agrupamiento. Aunque estas métricas no proporcionan información directa sobre la calidad del modelo completo, sí que aportan información relativa a la propiedades de cada partición y las diferencias entre ellas.

El **diámetro** (*diameter*) de una partición es la máxima disimilitud entre cualquier par de instancias de la partición. Podemos expresar el diámetro de la siguiente forma:

$$DIAM_\delta(p) = max\ \delta(x_1, x_2) \mid x_1, x_2 \in p \tag{16}$$

donde δ se refiere a la métrica de distancia o similitud empleada y p es la partición que evaluamos.

Claramente, las particiones compactas, es decir, con un diámetro pequeño, suelen ser las más deseadas. Cuando aparecen particiones con diámetros muy dispares, puede ser un signo de que debemos ampliar (o reducir) el número de particiones.

La **separación** (*separation*) indica el grado de disimilitud entre una partición y las demás. Se define, generalmente, como la mínima disimilitud entre cualquier instancia de una partición y cualquier otra instancia de las demás particiones. Matemáticamente se puede representar de la siguiente forma:

$$SEP_\delta(p) = min\ \delta(x_1, x_2) \mid x_1 \in D^p, x_2 \in D - D^p \tag{17}$$

donde D^p referencia al conjunto de instancias de la partición p y $D - D^p$ indica el

conjunto de instancias excepto las incluidas en la partición p.

Idealmente, sería preferible que las distancias entre todas las particiones fueran similares, pero es difícil en problemas y conjuntos de datos reales. En todo caso, que exista diferencia entre el valor de separación de las distintas particiones no se debe entender como una debilidad del modelo.

Para finalizar, definiremos el **aislamiento** (*isolation*) como una propiedad que dice que el diámetro de una partición debería ser inferior al valor de separación del mismo. Debemos remarcar que el aislamiento no es una métrica en el sentido estricto, si no más bien una propiedad deseable en el resultado de un proceso de agrupamiento; aunque no siempre es posible que se cumpla dicha propiedad, y su ausencia no indica, necesariamente, que el modelo generado no sea válido.

5.3.2. Métricas de calidad general

Aunque las métricas de calidad asociadas a cada partición proporcionan información muy relevante, a veces es preferible un indicador simple que pueda ser utilizado para comparar directamente modelos alternativos sobre un mismo conjunto de datos. Éste es el objetivo de las métricas de calidad general, que combinan distintos indicadores sobre la cohesión y separación de todas las particiones en una única métrica.

Aunque es posible realizar la media de los valores de calidad por partición obtenidos en las distintas particiones, existen métricas específicas para evaluar la calidad del modelo de forma global, obteniendo una única puntuación o valor para el modelo de agrupamiento analizado.

El **índice Dunn** (*Dunn index*) mide la calidad del modelo utilizando el valor mínimo de separación y el valor máximo de diámetro de las particiones, tal y como se describe a continuación:

$$DUNN_\delta = \frac{min\ sep_\delta(p)}{max\ diam_\delta(p)} \mid p \in P \tag{18}$$

donde p representa una partición y P es el conjunto de todas la particiones que ha generado el modelo de agrupamiento.

Claramente, valores altos de esta métrica indican que el grado de disimilitud entre particiones es alto, mientras que el grado de disimilitud dentro de las particiones es bajo.

Otra métrica de calidad general interesante es conocida como el **índice C** (*C index*).

Esta métrica compara la disimilitud de los pares de instancias de una misma partición con los valores observados en el conjunto de instancias sin considerar las particiones. Se expresa de la siguiente forma:

$$CIND_\delta = \frac{\sum_{p \in P} \sum_{\substack{x_1, x_2 \in D^p \\ x_1 \neq x_2}} \delta(x_1, x_2) - \sum_{x_1, x_2 \in \Gamma_\delta^{min}} \delta(x_1, x_2))}{\sum_{\langle x_1, x_2 \rangle \in \Gamma_\delta^{max}} \delta(x_1, x_2) - \sum_{\langle x_1, x_2 \rangle \in \Gamma_\delta^{min}} \delta(x_1, x_2)} \tag{19}$$

donde Γ_δ^{min} y Γ_δ^{max} son los conjuntos de mínima y máxima disimilitud, respectivamente, entre pares de instancias de D.

El valor de esta métrica se presenta en el rango $[0, 1]$, donde valores próximos a 0 indican particiones más cohesionadas en el modelo generado. Esta métrica permite comparar resultados de distintos algoritmos sobre un mismo conjunto de datos, proporcionando una métrica única para la comparación. Por contra, el valor obtenido no puede ser utilizado directamente para estimar si el número de particiones es adecuado para el problema a resolver.

5.3.3. Métricas de calidad externas

Las medidas de calidad de agrupamiento presentadas anteriormente asumen que no existe ninguna información externa disponible sobre la forma "correcta" o "deseada" de agrupar los datos. En cambio, adoptan varios enfoques para evaluar la cohesión y la separación de las particiones o su grado de coincidencia con los datos. Por el contrario, las medidas externas asumen la existencia de etiquetas de clase proporcionadas externamente, como en la tarea de clasificación, que -aunque no se utilizan para la creación de modelos- pueden utilizarse para la evaluación de modelos. Estas etiquetas de clase representan una partición de los datos en subconjuntos contra los cuales se comparan las asignaciones de particiones generadas por el modelo evaluado. La aplicación principal de las medidas externas de calidad es la comparación entre distintos algoritmos sobre un mismo conjunto de datos, conocido habitualmente como *benchmark*, que es una tarea común de investigación.

El enfoque más directo para la evaluación del modelo de agrupamiento con medidas externas de calidad es adoptar medidas de calidad para los problemas de clasificación, vistas en la Sección 5.1, como por ejemplo el error de clasificación (*Misclassification error*, ERR). Esto puede hacerse tratando el modelo de agrupación evaluado como un modelo de clasificación que asocia la clase mayoritaria con cada grupo basado en el conjunto de entrenamiento y luego para cada instancia predice la clase asociada con su grupo.

Adicionalmente, podemos definir el **índice Rand** (*Rand index*) a partir de un conjunto de objetos o instancias y dos posibles particiones de estos objetos, la partición externa que identificamos como correcta y la partición que indica el modelo de agrupamiento evaluado. Formalmente, definimos un conjunto de n objetos o instancias $D = \{x_1, \ldots, x_n\}$ y dos particiones de D para comparar, una partición de D en r subconjuntos que indica la clase externa $c : D \to 1, \ldots, r$, y una partición de D en s subconjuntos que indica la clase predicha por el modelo de agrupamiento $h : D \to 1, \ldots, s$.

Cada par de objetos $\langle x_1, x_2 \rangle \mid x_1, x_2 \in D$ es asignado a una de las siguientes cuatro categorías, de forma similar a la matrix de confusión vista anteriormente:

- Verdaderos positivos (*True positives*, TP), como el número de pares de elementos en D que están en el mismo subconjunto externo y en el mismo subconjunto predicho, i.e. $c(x_1) = c(x_2) \wedge h(x_1) = h(x_2)$.

- Verdaderos negativos (*True negatives*, TN), como el número de pares de elementos en D que están en el distintos subconjuntos externos y en distintos subconjuntos predichos, i.e. $c(x_1) \neq c(x_2) \wedge h(x_1) \neq h(x_2)$.

- Falsos positivos (*False positives*, FP), como el número de pares de elementos en D que están en distintos subconjuntos externos y en el mismo subconjunto predicho, i.e. $c(x_1) \neq c(x_2) \wedge h(x_1) = h(x_2)$.

- Falsos negativos (*False negatives*, FN), como el número de pares de elementos en D que están en el mismo subconjunto externo y en distintos subconjuntos predichos, i.e. $c(x_1) = c(x_2) \wedge h(x_1) \neq h(x_2)$.

El índice Rand se calcula como la relación entre el número total de verdaderos positivos y verdaderos negativos respecto al número total de pares de instancias de D. Formalmente, se define como:

$$RAND = \frac{TP + TN}{TP + TN + FP + FN} = \frac{TP + TN}{\binom{n}{2}} \tag{20}$$

El índice Rand tiene un valor en el rango $[0, 1]$, donde 0 indica que los dos datos predichos no coinciden en ningún par de puntos con los datos externos, y 1 indica que los datos predichos corresponden exactamente con los datos externos.

6. Apéndice A: Notación

Generalmente, en los métodos supervisados partiremos de un conjunto de datos correctamente etiquetados (D) en el que distinguimos la siguiente estructura:

$$D_{n,m} = \begin{pmatrix} d_1 = & a_{1,1} & a_{1,2} & \cdots & a_{1,m} & c_1 \\ d_2 = & a_{2,1} & a_{2,2} & \cdots & a_{2,m} & c_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ d_n = & a_{n,1} & a_{n,2} & \cdots & a_{n,m} & c_n \end{pmatrix}$$

donde:

- n es número de elementos o instancias,
- m el número de atributos del conjunto de datos o también su dimensionalidad,
- d_i indica cada una de las instancias del juego de datos,
- $a_{i,j}$ indica cada uno de los atributos descriptivos, y
- c_i indica el atributo objetivo o clase a predecir para cada elemento.

En el caso de los métodos no supervisados, el conjunto de datos sigue el mismo patrón y notación, excepto para el atributo de clase (c_i) que no existe (o no se emplea) en el caso de los conjuntos no etiquetados.

La Tabla 2 muestra un resumen de la notación empleada en los distintos módulos didácticos.

Tabla 2. Resumen de la notación básica

Símbolo	Descripción
$D_{n,m}$	Conjunto de datos
n	Número de instancias
m	Número de atributos
k	Número de clases
d_i	Instancia i del conjunto de datos D
a_j	Conjunto del atributo j en D
$a_{i,j}$	Valor del atributo j en la instancia i del conjunto de datos D
c_i	Clase de la instancia d_i

Resumen

En este texto hemos presentado una introducción al aprendizaje automático (*machine learning*, en inglés) o minería de datos (*data mining*). Después de una breve introducción y contextualización de la temática, se han presentado las etapas de un proceso de minería de datos, según la metodología CRISP-DM.

Seguidamente, se han repasado algunos conceptos básicos sobre aprendizaje automático que son necesarios para comprender y profundizar en los modelos de aprendizaje automático. En concreto, hemos revisado los conceptos relacionados con la tipología de métodos y tareas en el aprendizaje automático, así como la generación de los conjuntos de entrenamiento y test.

Hemos finalizado este texto con una revisión de las principales métricas empleadas en la evaluación de los resultados de un modelo de aprendizaje automático.

Glosario

Aprendizaje automático El aprendizaje automático (más conocido por su denominación en inglés, *machine learning*) es el conjunto de técnicas, métodos y algoritmos que permiten a una máquina aprender de manera automática en base a experiencias pasadas.

Aprendizaje no supervisado El aprendizaje no supervisado se basa en el descubrimiento de patrones, características y correlaciones en los datos de entrada, sin intervención externa.

Aprendizaje supervisado El aprendizaje supervisado se basa en el uso de un componente externo, llamado supervisor, que compara los datos obtenidos por el modelo con los datos esperados por éste, y proporciona retroalimentación al modelo para que vaya ajustándose y mejorando las predicciones.

Bibliografía

Géron, Aurélien (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Editio*n. O'Reilly Media, Inc.

de Jonge, Edwin; van der Loo, Mark (2013). *An introduction to data cleaning with R.* The Hague [u. a.]: Statistics Netherlands.

Cichosz, Pawel (2015). *Data mining algorithms. Explained using R.* Wiley.

J. Gironés Roig, J. Casas Roma, J. Minguillón Alfonso, R. Caihuelas Quiles (2017). *Minería de datos: Modelos y algoritmos.* Barcelona: Editorial UOC.