Introducción al ciclo de vida de los datos

PID_00265705

Laia Subirats Maté
Diego Oswaldo Pérez Trenard
Mireia Calvo González

Tiempo mínimo de dedicación recomendado: 3 horas

Universitat Oberta de Catalunya

Laia Subirats Maté

Ingeniera de Telecomunicaciones por la Universidad Pompeu Fabra (2008), Máster en Telemática por la Universidad Politécnica de Cataluña (2009) y doctora en Informática por la Universidad Autónoma de Barcelona (2015). Desde 2009, trabaja como investigadora en Eurecat (Centro Tecnológico de Cataluña) aplicando la ciencia de datos a diferentes campos como la salud, el medio ambiente o la educación. Desde 2016, colabora con la UOC como docente en el Máster de Data Science y en el grado de Informática. Es especialista en inteligencia artificial, ciencia de datos, salud digital y representación del conocimiento.

Diego Oswaldo Pérez Trenard

Ingeniero electrónico por la Universidad Simón Bolívar (2015), especialización en High Tech Imaging (HTI) por la Universidad Télécom SudParis (2014) y doctor en Señales, Imágenes y Visión por la Universidad de Rennes 1 (2018). Desde 2014, ha trabajado como ingeniero de investigación y desarrollo en el Instituto Nacional de Salud e Investigación Médica (INSERM) y en el Laboratorio de Procesamiento de Señales e Imágenes (LTSI), aplicando conocimientos en electrónica y en procesamiento de datos al estudio de diferentes enfermedades neurológicas, cardíacas y respiratorias. Desde 2018, colabora como docente en el máster de Data Science de la UOC.

Mireia Calvo González

Ingeniera de telecomunicaciones por la Universidad Politécnica de Cataluña (2011), Máster en Ingeniería Biomédica por la Universidad de Barcelona y la Universidad Politécnica de Cataluña (2014) y Doctora en Procesamiento de señales y telecomunicaciones por la Universidad de Rennes 1 y en Ingeniería Biomédica por la Universidad Politécnica de Cataluña (2017). Desde 2012 ha trabajado como investigadora en diferentes entornos académicos, clínicos e industriales, aplicando el procesamiento de datos al estudio de diferentes enfermedades cardíacas y respiratorias. Desde 2017 colabora con la UOC como docente en el Máster de Data Science.

El encargo y la creación de este recurso de aprendizaje UOC han sido coordinados por la profesora: Isabel Guitart Hormigo (2019)

Primera edición: septiembre 2019

Av. Tibidabo, 39-43, 08035 Barcelona
Realización editorial: FUOC

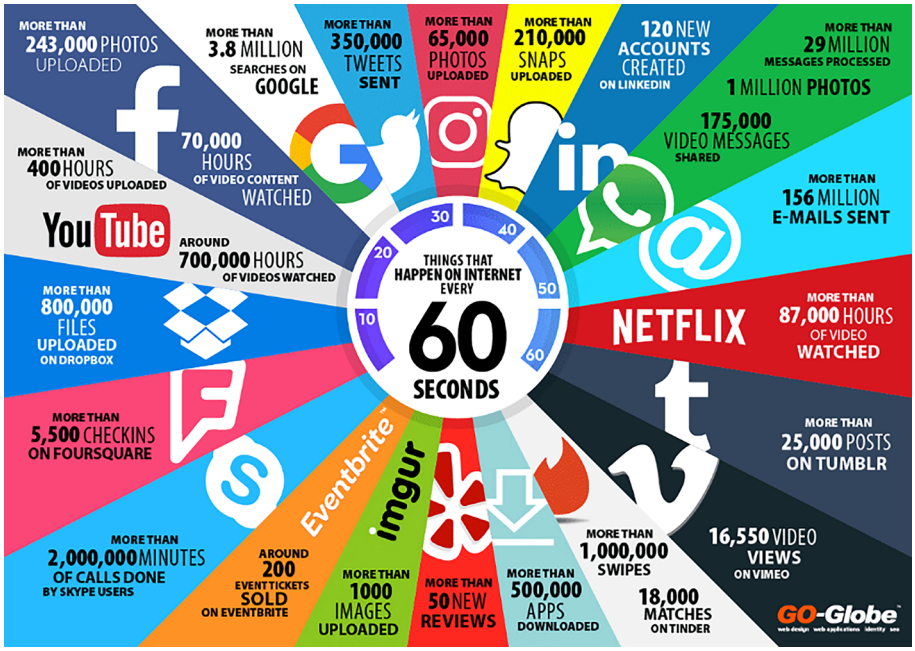
Índice

Introducción

Durante la historia ha habido diferentes revoluciones industriales hasta llegar a la sociedad de la información y el conocimiento. En concreto, la primera revolución industrial surgió en el siglo XVIII cuando apareció la máquina de vapor. La segunda revolución industrial emergió entre los siglos XIX y XX con la producción masiva de electricidad. La tercera revolución industrial, o primera revolución de la información, apareció a finales del siglo XX con la irrupción de las tecnologías de la información y comunicación (TIC). Finalmente, a principios del siglo XXI emergió la cuarta revolución industrial, o segunda revolución de la información, basada en la inteligencia artificial.

Una de las características de la sociedad de la información y el conocimiento es el número creciente de datos generados tanto por individuos como empresas, también conocido como *datificación*. Por ejemplo, Computer Sciences Corporation estima que en el año 2020 habrá 44 veces más datos de los que había en el 2009. Por otro lado, según go-globe.com, cada 60 segundos se genera la cantidad de información mostrada en la figura 1.

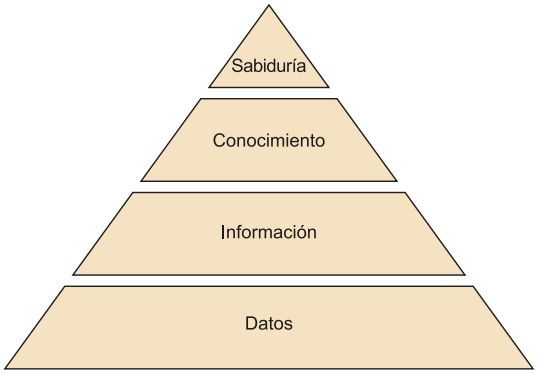
Figura 1. Información generada en Internet cada 60 segundos



Fuente: https://www.go-globe.com

Para acceder a la sabiduría desde los datos definimos la pirámide "Datos, Información, Conocimiento, Sabiduría (DICS)", como se muestra en la figura 2. En esta pirámide partimos de los datos; posteriormente, según el contexto y la conexión de partes, llegamos a la información. El siguiente paso es el conocimiento y finalmente, el último paso, es la sabiduría.

Figura 2. Pirámide DICS



Dentro del ámbito de la ciencia de los datos, diferentes perfiles profesionales hacen posible el tránsito reflejado en la pirámide. A continuación indicamos algunos ejemplos:

Más información en:

La infografía "The data science industry. Who does what".

- **Científico de datos** (*data scientist*). Limpia y organiza los datos. Es un analista de datos curioso.

- **Analista de datos** (*data analyst*). Recoge, procesa y realiza análisis estadísticos de los datos. Intuitivo con capacidad para descifrar problemas y datos.

- **Arquitecto de datos** (*data architect*). Gestiona los sistemas para integrar, centralizar, proteger y mantener fuentes de datos. Crea planos para la gestión de datos para integrar, centralizar, proteger y mantener las fuentes de datos.

- **Ingeniero de datos** (*data engineer*). Desarrolla, construye, testea y mantiene arquitecturas. Desarrolla, construye, prueba y mantiene arquitecturas (como bases de datos y sistemas de procesamiento a gran escala).

- **Estadístico** (*statistician*). Recoge, analiza e interpreta datos cualitativos y cuantitativos con teorías y métodos estadísticos. Es un profesional de la lógica y domina la estadística.

- **Administrador de base de datos** (*database administrator*). Asegura que la base de datos esté disponible para todos los usuarios relevantes, que funcione correctamente y se mantenga segura.

- **Analista de negocio** (*bussiness analyst*). Mejora los procesos de negocio como intermediario entre negocio y tecnologías de la información.

- **Líder de ciencia de datos** (*data science leader*). Gestiona un equipo de analistas y científicos de datos.

En la tabla 1 se muestran los diferentes lenguajes que dominan los diferentes perfiles:

Tabla 1. Lenguajes de programación de los diferentes perfiles de la ciencia de datos.

	Científico de datos	Analista de datos	Arquitecto de datos	Ingeniero de datos	Estadístico	Administrador de base de datos	Analista de negocio	Líder de ciencia de datos
R	X	X		X	X			X
SAS	X			X	X			X
Pytdon	X	X		X	X	X		X
Matlab	X			X	X			X
Hive / Piq	X		X	X	X			
Spark	X		X		X			
HTML / Javascript		X						
C / C++		X		X				
SPSS				X	X			
Java				X		X		X
Rubby				X		X		
Perl				X	X			
XML			X					
C#						X		
Stata					X			

Objetivos

En este material didáctico se proporcionan las herramientas fundamentales que permitirán asimilar los siguientes objetivos:

1. Entender qué es la generación de datos y el concepto *sociedad de la información*.

2. Ser capaz de identificar los diferentes perfiles que intervienen en la ciencia de datos.

3. Conocer el significado y la clasificación de los datos según su estructura, nivel de acceso y tipo de información.

4. Ser capaz de evaluar la calidad de los datos.

5. Conocer las diferentes etapas del ciclo de vida de los datos: captura, almacenamiento, preprocesado, análisis, representación y publicación.

1. ¿Qué son los datos?

Un dato es, en principio, una cantidad o cualidad que describe un atributo de una entidad, dentro de un rango de valores posibles. Es un valor «dado» al respecto de algo observado, de acuerdo con la raíz latina que da origen al término (*datum*).

Bibliografía recomendada

Minguillón, Julià (2016). «Fundamentos de Data Science». Editorial UOC.

Los datos pueden clasificarse de diferentes maneras. En los siguientes subapartados explicaremos los diferentes tipos de datos y los aspectos que determinan su calidad de vida.

1.1. Clasificación de los datos

Los datos se pueden clasificar según su estructura, nivel de acceso y el tipo de información que contienen.

1.1.1. Estructura de los datos

Según su estructura podemos clasificar los datos de la siguiente manera:

- **Datos Simples**. Datos atómicos indivisibles, con un significado propio, de acuerdo con la definición (un valor de un atributo). Se trata de datos reales o cadenas, por ejemplo, el peso corporal.

- **Datos Compuestos o estructurados**. Datos que son una combinación de otros datos simples y/o compuestos, de acuerdo con una estructura fija y conocida *a priori*. Un ejemplo de datos compuestos sería una radiografía torácica.

- **Datos Semiestructurados**. Datos estructurados que siguen una estructura parcial o que puede cambiar según el contexto, o que no siguen ninguna estructura. Un ejemplo sería una página HTML.

1.1.2. Nivel de acceso

En lo referente a la seguridad, según la Universidad Carnegie Mellon los datos pueden clasificarse en tres niveles:

- **Datos restringidos**: los datos deben clasificarse como restringidos cuando la divulgación no autorizada, la alteración o la destrucción de esos datos podría causar un nivel significativo de riesgo para la universidad o sus afi-

liados. Los ejemplos de datos restringidos incluyen datos protegidos por regulaciones de privacidad estatales o federales y datos protegidos por acuerdos de confidencialidad. El nivel más alto de controles de seguridad se debe aplicar a los datos restringidos.

- **Datos privados**: los datos deben clasificarse como privados cuando la divulgación no autorizada, la alteración o la destrucción de esos datos podría resultar en un nivel moderado de riesgo para la institución. De forma predeterminada, todos los datos institucionales que no se clasifican explícitamente como restringidos o datos públicos deben tratarse como datos privados. Se debe aplicar un nivel razonable de controles de seguridad a los datos privados.

- **Datos públicos**: los datos deben clasificarse como públicos cuando la divulgación no autorizada, la alteración o la destrucción de esos datos conlleve un riesgo pequeño o nulo para la institución. Como ejemplos de datos públicos se incluyen comunicados de prensa y publicaciones. No obstante, hay que tener en cuenta que se requiere cierto nivel de control para evitar la modificación o destrucción no autorizada de los datos públicos.

A veces también se clasifican los datos según el posible impacto de seguridad que tengan en la organización. En la tabla 2 se muestran tres objetivos de seguridad (confidencialidad, integridad y disponibilidad) clasificados según el impacto de seguridad de los datos (bajo, moderado o alto).

Tabla 2. Clasificación según el impacto de seguridad de los datos

Objetivo de seguridad	Impacto bajo	Impacto moderado	Impacto alto
Confidencialidad	La divulgación no autorizada de información puede tener un efecto adverso **limitado** en la organización.	La divulgación no autorizada de información puede tener un efecto adverso **grave** en la organización.	La divulgación no autorizada de información puede tener un efecto adverso grave o **catastrófico** en la organización.
Integridad	La modificación o destrucción no autorizada de la información puede tener un efecto adverso **limitado** en la organización.	La modificación o destrucción no autorizada de la información puede tener un efecto adverso **grave** en la organización.	La modificación o destrucción no autorizada de la información puede tener un efecto adverso grave o **catastrófico** en la organización.
Disponibilidad	La interrupción del acceso o uso de la información o un sistema de información puede tener un efecto adverso **limitado** en la organización.	La interrupción del acceso o uso de la información o un sistema de información puede tener un efecto adverso **grave** en la organización.	La interrupción del acceso o uso de la información o un sistema de información puede tener un efecto adverso grave o **catastrófico** en la organización.

Los beneficios de estas clasificaciones son los siguientes:

- Cumplimiento de los datos y una gestión de riesgos más sencilla. Los datos se ubican donde se espera en el nivel de almacenamiento predefinido y «punto en el tiempo».

- Simplificación del cifrado de datos porque no es necesario cifrar todos los datos. Esto ahorra valiosos ciclos de procesador y toda la consecutividad relacionada.

- Indización de datos para mejorar los tiempos de acceso de los usuarios.

- La protección de datos se redefine cuando se mejora el RTO (objetivo de tiempo de recuperación o en inglés *recovery time objective*).

1.1.3. Tipo de información

Desde un punto de vista estadístico, los datos pueden clasificarse en cuantitativos (o numéricos) y cualitativos. Los datos cualitativos pueden clasificarse en ordinales (que se pueden ordenar) o nominales (no hay un orden). Los datos cualitativos también pueden clasificarse en binarios (dicotómicos) o categóricos (multicotómicos).

Ejemplo

- Dato cualitativo nominal: las lenguas que hablas (multicotómico) o si eres mayor de edad (dicotómico).
- Dato cualitativo ordinal: el grado de formación (elemental, formación profesional, graduado, máster, doctor, etc.).
- Datos cuantitativos: la temperatura.

Referència bibliogràfica

Kristin H. Jarman (2013). *The art of data analysis. How to answer almost any question using basic statistics*. Hoboken, NJ: Wiley.

Para tratar y resumir datos cualitativos normalmente se utiliza lo que se llama *análisis de frecuencia*. Esta es simplemente contar cuántos datos hay en cada categoría. También es útil mostrar los datos en frecuencia relativa (porcentaje). Hay diferentes maneras de visualizar datos cualitativos como gráficos circulares o gráficos de barras. En los valores ordinales también se pueden usar percentiles, mediana, moda y el rango intercuartil para resumir sus datos. Para datos cuantitativos se pueden usar histogramas o gráfico de cajas y bigotes (en inglés box plot).

1.2. Calidad de los datos

Hay diferentes factores que influyen en la calidad de los datos: exactitud, completitud, consistencia, atemporalidad, unicidad y validez. A continuación, para cada factor se explica la definición, referencia, medida, el ámbito, la unidad de medida y las dimensiones relacionadas.

Más información en:

Ver el artículo "The Six primary dimensions for data quality assessment. Defining Data Quality Dimensions".

1.2.1. Exactitud

Se define como el grado en que los datos describen correctamente el objeto o evento del «mundo real». Idealmente, la verdad del «mundo real» se establece a través de la investigación primaria. Sin embargo, como a menudo esto no es práctico, es común utilizar datos de referencia de terceros, de fuentes que

se consideran confiables y de la misma cronología, la medida es el grado de representación del mundo real y el ámbito es la base de datos, la unidad de medida es el porcentaje de datos que pasan las reglas de exactitud. Las dimensiones relacionadas son la validez, la unicidad y la consistencia.

Ejemplo

Una enfermera dado que había llegado recientemente de Europa y aún no estaba familiarizada con el formato de fechado estadounidense, entró todas las fechas de nacimiento con el formato europeo (DD/ MM/YYYY), en lugar del estadounidense (MM/DD/YYYY). Los datos pasaron la fase de validación del sistema ya que los valores se mantenían dentro del rango, pero no eran exactos: el sistema entendió que todos los niños cuyos datos se entraron ese día habían nacido el 5 de enero de 2014.

1.2.2. Completitud

Se define como la proporción de datos almacenados frente al potencial «100 % completo». La referencia son las reglas de negocio que definen lo que representa el «100 % completo». La medida es la ausencia de valores en blanco (nulo) o la no presencia de valores en blanco. El ámbito es la base de datos y la unidad de medida es un porcentaje. Las dimensiones relacionadas son la validez y la exactitud.

Ejemplo

Porcentaje de pacientes que tienen todos los elementos de datos mínimos y básicos, según lo definido por el estándar, sin valores en blanco.

1.2.3. Consistencia

Es la ausencia de diferencia, al comparar dos o más representaciones de una cosa con su definición. La referencia es cada ítem, el ámbito es la base de datos y la unidad de medida es un porcentaje. Las dimensiones relacionadas son la exactitud, la validez y la unicidad.

Ejemplo

Los tres campos que se utilizan para documentar los resultados de una prueba de audición son: «oído izquierdo», «oído derecho» y «general». El campo general está diseñado para adaptarse a casos específicos, cuando parte de la información sobre alguno de los dos oídos no está disponible; este campo debe ser un valor calculado automáticamente basado en resultados de ambos oídos. Por ejemplo, general es OK si y solo si izquierdo y derecho son OK. La evaluación de valores en estos tres campos debe realizarse para asegurar la consistencia.

1.2.4. Puntualidad

Se define la puntualidad (o atemporalidad) como el grado en que los datos representan la realidad desde un punto requerido en el tiempo. La referencia es el tiempo del evento real que ha estado obtenido y la medida es la diferencia en el tiempo. El ámbito es cualquier ítem relacionado con la base de datos y la unidad de medida es el tiempo. Una dimensión relacionada es la exactitud.

Ejemplo

Diferencia horaria entre la finalización de la prueba de audición de un paciente, la visita de diagnóstico y la introducción en el sistema de la información sobre esta visita.

1.2.5. Unicidad

Nos asegura que nada se registra más de una vez. Es el inverso de una evaluación del nivel de duplicación, la referencia es la misma y el ámbito es el conjunto de datos. Se mide como porcentaje y una dimensión relacionada es la consistencia.

Ejemplo

Porcentaje de valores duplicados de un conjunto de datos de audición.

1.2.6. Validez

Se define como el ajuste a la sintaxis predefinida (formato, tipo, rango). La referencia es la base de los datos, metadatos o las reglas de documentación, según los tipos permitidos (cadena, entero, punto flotante, etc.), el formato (longitud, número de dígitos, etc.) y rango (mínimo, máximo o contenido dentro de un conjunto de valores permitidos). La medida es la comparación entre los datos y los metadatos o la documentación. El ámbito es todos los datos y la unidad de medida es el porcentaje de datos válidos o inválidos. Dimensiones relacionadas con la validez son la exactitud, la completitud, la consistencia y la unicidad.

Ejemplo

Un ejemplo de validez a nivel de elemento de datos es el tipo y la severidad de la pérdida auditiva, que deben ser elegidos de una lista dada de valores permitidos.

2. Ciclo de vida de los datos

Con los importantes avances en informática y en tecnologías relacionadas, y su implementación cada vez mayor en todas las áreas del conocimiento, cantidades significativas de datos con diversas características son cada día capturados y almacenados en bases de datos. Por otro lado, el avance del poder de cómputo de los ordenadores no se ha quedado atrás. Esta simbiosis ha permitido que grandes cantidades de información digitalizada sea fácilmente capturada, almacenada y procesada.

Sin embargo, obtener algún tipo de conocimiento a partir de este enorme volumen de datos resulta una tarea ardua y compleja. La razón es que, normalmente, una colección de datos «en bruto» no otorga información relevante. El valor de estos datos se obtiene después de un procesado que permita extraer información útil para facilitar una toma de decisiones o para una correcta comprensión de un fenómeno que gobierne en la fuente de los datos.

Bibliografía recomendada

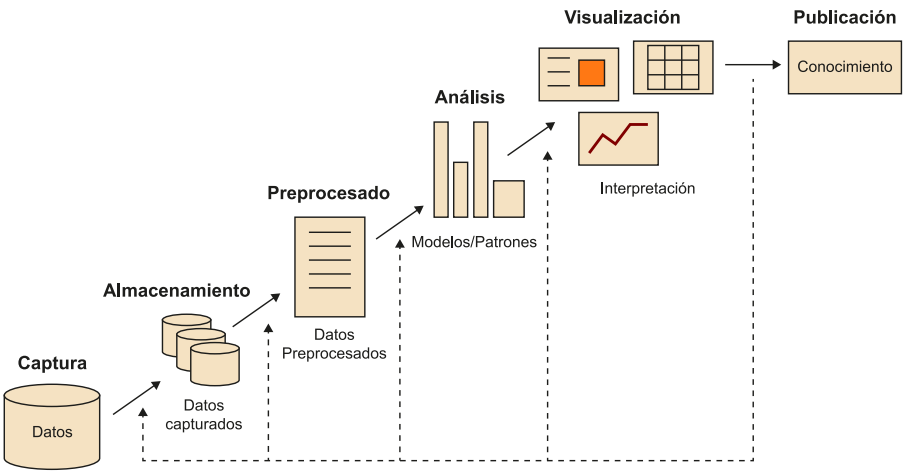
Han, Jiawei; **Kamber, Micheline**; **Pei, Jian** (2012). *Data Mining: Concepts and Techniques*. Waltham: Elsevier.
Riquelme, José Cristóbal; **Ruiz, Roberto**; **Gilbert, Karina** (2006). «Minería de datos: conceptos y tendencias». *Inteligencia artificial: Revista Iberoamericana de Inteligencia Artificial* (vol. 10, n.º 29, págs. 11-18).

La extracción de conocimiento a partir de un grupo dado de datos para la resolución de un cierto problema es un proceso iterativo. En este sentido, se definen seis fases o etapas típicas en el ciclo de vida de los datos: captura, almacenamiento, preprocesado, análisis, visualización y publicación. Cada una de estas etapas tiene un objetivo, generando valor a partir de los datos en cada una de ellas. Aunque, no todas las etapas son estrictamente necesarias. En los siguientes apartados se explicará más en detalle cada una de estas fases (ver figura 6).

Bibliografía recomendada

Fayyad, Usama; **Piatetsky-Shapiro, Gregory**; **Smyth, Padhraic** (1996, marzo). «From data mining to knowledge discovery in databases». *AI Magazine* (vol. 17, n.º 3, pág. 37-54).

Figura 6. Etapas típicas en el ciclo de vida de los datos



2.1. Captura

La primera fase del ciclo de vida de los datos tiene como objetivo la adquisición de todos los datos que se puedan generar durante un proceso dado. Esta fase es conocida como captura y puede subdividirse según el tipo de mecanismo utilizado para dicha adquisición: creación y extracción.

Bibliografía recomendada

Minguillón, Julià (2016). «Fundamentos de Data Science». Editorial UOC.

2.1.1. Creación

Este mecanismo consiste en la implementación de una rutina o algoritmo dentro del proceso que está generando los datos, con el objetivo de almacenar los datos que se consideren relevantes a medida que estos se vayan generando. En otras palabras, los datos son capturados en tiempo real cada vez que se generan.

Este mecanismo puede ser implementado cuando se tiene acceso al proceso o sistema y se puede intervenir en él, ya sea utilizando rutinas preexistentes que nos permitan la obtención de los datos o con la creación de nuevas rutinas que puedan ser añadidas al proceso.

Ejemplo

En una estación meteorológica se desea obtener datos de pluviometría en una zona específica para poder compararlos con otras zonas de interés. Para ello, suponiendo que se tiene acceso al proceso, se decide instalar pluviómetros en la zona a analizar, los cuales envían la información a un servidor que almacena las diferentes medidas que estos aportan durante el día. Como se puede apreciar en este ejemplo, se ha añadido una nueva rutina (implementación de los pluviómetros) a un proceso preexistente (estación meteorológica) al cual se tiene acceso. De este modo, es posible capturar datos completamente nuevos que han sido creados para intentar responder a un problema en específico.

Existen otras técnicas muy utilizadas en la creación de datos como:

a) Creación de formularios: es una técnica que consiste en elaborar una serie de preguntas que van dirigidas a los usuarios con la finalidad de obtener información (datos) sobre un sistema, proceso o servicio. A pesar de ser una técnica relativamente sencilla y eficaz, se considera lenta y costosa, además de presentar limitaciones temporales. Sin embargo, con la aparición de las redes sociales y las diversas herramientas que permiten la creación de formularios en línea, es posible alcanzar a gran cantidad de usuarios aumentando, así, la popularidad de esta técnica.

b) *Crowdsourcing*: se trata de un concepto que fue creado para la resolución de problemas muy complejos que resultan difíciles de resolver por métodos convencionales como la minería de datos o la inteligencia artificial. Se basa en aprovechar el intelecto humano y su capacidad de razonamiento, superior a los métodos actuales, para resolver un problema dado preguntando directamente a los usuarios sobre la posible solución.

Ejemplo

Un ejemplo de esta técnica sería en la clasificación de imágenes que pueden ser posteriormente utilizadas para entrenar un algoritmo de *machine learning*. Esta técnica se observa, típicamente, en los sistemas de control basados en CAPTCHA (que es el acrónimo en inglés de *completely automated public Turing test to tell computers and humans apart*), que evitan que una máquina (*bot*) se registre de manera automática a un servicio. Como se puede ver en la figura 7, el sistema le pregunta al usuario qué imágenes corresponden a señales de tránsito. La imagen se trata de la salida de un sistema de reconocimiento de vídeo en el cual ciertos fragmentos no fueron procesados correctamente, por lo que a medida que los usuarios seleccionan los fragmentos de la imagen donde se encuentran estas señales, el algoritmo aprende y le permite optimizar su rendimiento.

Figura 7. Ejemplo de un CAPTCHA

c) Datos cualitativos: se implementa cuando los datos que se desean obtener son difícilmente cuantificables, como sentimientos, motivaciones o emociones, por lo que se recurre a entrevistas donde, por medio de preguntas, se obtiene la información directamente del usuario. Las entrevistas son normalmente grabadas para después ser transcritas según criterios preestablecidos y poder identificar la información relevante que se busca obtener.

2.1.2. Extracción

Este mecanismo es utilizado cuando no se tiene acceso al sistema o no se puede intervenir en el proceso de generación de los datos. Consiste en capturar los datos que ya fueron generados por un proceso por medio de una búsqueda de los datos de interés. Idealmente, se intenta que estos datos encontrados sean almacenados inmediatamente después de su generación.

Ejemplo

Se puede diseñar un algoritmo que capture los títulos de las diferentes noticias que aparecen en una cierta página web informativa. De este modo, al no tener acceso al proceso de generación de datos, podemos de igual manera extraerlos según su orden de aparición.

En la mayoría de los casos, intervenir en el proceso de generación de los datos no es posible, por lo que las técnicas de extracción de datos son los mecanismos más habituales. Existen diversas técnicas de extracción, siendo las siguientes las más utilizadas.

a) Acceso mediante repositorios: consiste en la adquisición de los datos por medio de una descarga de un repositorio digital publicado en abierto. Numerosas instituciones o páginas web publican sus datos y permiten su acceso a todos los usuarios que deseen utilizarlos. Estos datos vienen normalmente organizados en ficheros según una clasificación preestablecida.

Ejemplo

El World Bank Group es una institución que realiza análisis económicos y estadísticos de los diferentes países. Esta organización permite el acceso y descarga de sus datos en ficheros presentados en diferentes formatos, lo que permite a los usuarios la realización de análisis independientes.

b) Extracción mediante una *application programming interface* (API): en ciertos casos, las instituciones o páginas web ofrecen a los usuarios más que la simple posibilidad de la descarga y acceso de sus datos por medio de un fichero, y permiten la realización de consultas específicas dentro de dichos datos de manera que se obtenga solo un conjunto que cumpla con un cierto parámetro de consulta. Estas interfaces de consulta resultan interesantes para la programación de algoritmos que recopilen datos de forma automatizada.

c) Manipulación de parámetros de búsqueda: se utiliza cuando los sitios web no disponen de una API para realizar consultas específicas, pero se puede acceder a los ficheros de datos mediante su URL. En ciertas ocasiones, los parámetros de la URL indican la arquitectura de la web. Esta característica permite que se puedan diseñar *scripts* que automaticen un proceso de búsqueda, otorgando al usuario la capacidad de seleccionar subconjuntos de datos de interés.

Ejemplo

El Instituto Nacional de Estadística y Censo (INEC) de Panamá permite el acceso a sus datos directamente desde su página web, por lo que se pueden manipular los parámetros del URL para acceder a los datos deseados. La sección de datos se encuentra estructurada de la siguiente forma: `http://www.contraloria.gob.pa/INEC/Avance/Avance.aspx?ID_CATEGORIA=<CAT>&ID_CIFRAS=<CIF>&ID_IDIOMA=<IDI>`. Donde:

- <CAT>
  - Indicadores de coyuntura = 1
  - Sector real = 2
  - Sectores fiscal y financiero = 3
  - Etc.
- <CIF>
  - Índice Mensual de Actividad Económica = 1
  - Indicadores de comercio exterior = 2
  - Etc.
- <IDI>
  - Castellano = 1

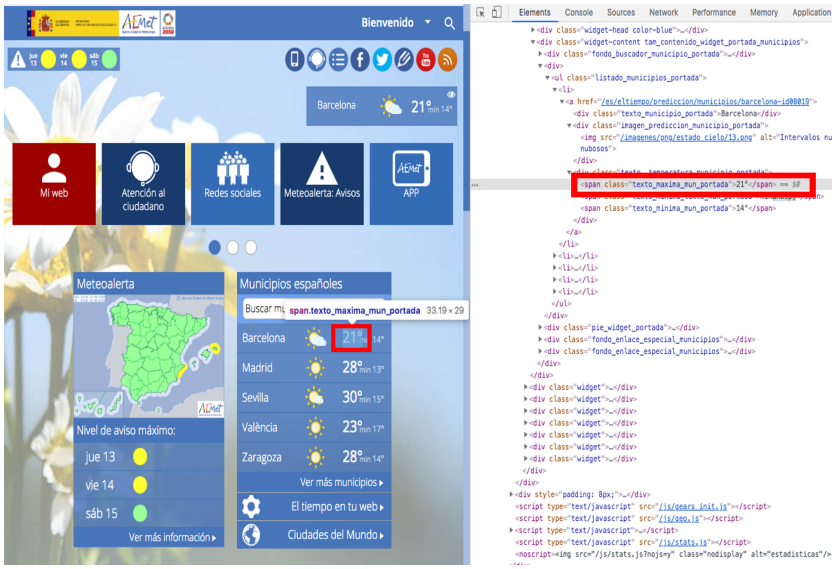
Todos son parámetros que pueden ser manipulados. De esta manera, si se desea acceder a los datos del sector fiscal, se coloca: `http://www.contraloria.gob.pa/INEC/Avance/Avance.aspx?ID_CATEGORIA=3&ID_CIFRAS=16&ID_IDIOMA=1`

d) *Web scraping*: en realidad, los sitios web no disponen de una API para la realización de consultas ni la posibilidad de descarga y acceso a ficheros. Por esta razón, la técnica más utilizada de extracción de datos de forma semiautomática es la conocida como *web scraping*. Esta técnica consiste en la implementación de herramientas conocidas como *bots* que simulan la navegación de un usuario real dentro de una página web. Estas herramientas, además de navegar dentro del sitio, extraen el contenido y la información de este para poder ser utilizada posteriormente. Este mecanismo es posible gracias a la estructura coherente de las páginas web y la posibilidad de su inspección para determinar el formato de dicha estructura interna. De esta forma, al conocer la arquitectura de la página, es posible programar un *bot* capaz de buscar y acceder a datos específicos dentro del contenido.

Ejemplo

En la figura 8 podemos observar cómo la página de la Agencia Estatal de Meteorología de España (AEMET) está estructurada en un formato HTML, en el cual se puede acceder a datos específicos dentro del contenido de la página. En este caso, podemos ver cómo se podría acceder a la temperatura actual en Barcelona moviéndose a través de la estructura del sitio. Un *bot* puede realizar esta tarea de forma automática, con lo que es posible obtener dicho dato cada vez que se requiera.

Figura 8. Página HTML de la Agencia Estatal de Meteorología de España



2.2. Almacenamiento

La siguiente fase del ciclo de vida de los datos, después de la captura, es el almacenado de los datos en un formato o representación adecuada que, según su topología, permita su utilización posterior de la manera más simple posible. A grandes rasgos, existen dos formas típicas de almacenar los datos, en ficheros simples o en bases de datos.

Los ficheros simples son estructuras donde los datos se almacenan según unos criterios preestablecidos. Por ejemplo, un fichero plano que nos dé información de la consola del computador sobre todos los procesos realizados durante un plazo definido.

Por otro lado, las bases de datos consisten en una colección de datos interrelacionados que suelen manipularse mediante un sistema de manipulación de bases de datos, o *database management system* (DBMS), el cual consiste en la colección de datos, y un conjunto de programas de software para gestionar y acceder a dichos datos.

Dentro del concepto de bases de datos, se suele hablar típicamente de dos tipos:

a) Bases de datos relacionales: se trata de una base de datos que consiste en organizar la información mediante una colección de tablas, en la que a cada una se le asigna un nombre único. Cada tabla consiste en un conjunto de atributos (columnas o campos) y normalmente almacena un gran conjunto de tuplas (registros o filas). Cada tupla de una tabla relacional representa un objeto identificado por una clave única y descrito por un conjunto de valores de atributo. En la actualidad, las bases de datos relacionales más utilizadas son MySQL, Oracle, SQL Server y PostgreSQL.

Ejemplo

Supongamos que se tiene una plataforma de cadenas de televisión, en la cual los clientes se suscriben a cada una y pueden elegir su forma de pago. Para ello, empezamos con una primera tabla que contiene solo la información de cada cliente.

Client_ID	Client_FirstName	Client_LastName
1	Juan	González
2	Luis	Pérez
3	José	Hernández

Después, tenemos una tabla con las diferentes cadenas de televisión disponibles y otra con la forma de pago.

TV_ID	TV_Name	TV_Price
1	Informative TV	20
2	Cartoon TV	30
3	Music TV	15

Payment_ID	Payment_Type
1	Efectivo
2	Transferencia bancaria

Payment_ID	Payment_Type
3	Tarjeta de crédito

Por último, se tiene una tabla que relaciona todas las tablas anteriores.

ID	Client_ID	TV_ID	Payment_ID
1	1	3	3
2	2	3	2
3	2	3	2
4	3	1	1

Como se puede observar, con la información de la última tabla podemos extraer cualquier dato que se requiera sobre cada cliente. Por ejemplo, para el cliente de nombre Luis Pérez, sabemos que su Client_ID es 2 y podemos ver que aparece en dos filas ya que se ha suscrito a dos cadenas de televisión (ID = 2 e ID = 3) correspondientes a Cartoon TV y Music TV. Además, se observa que ha elegido pagar con transferencia bancaria en ambos casos (Payment_ID = 2). Nótese que no es necesario incluir la tarifa en la tabla final ya que al conocer el TV_ID se puede calcular fácilmente el precio.

B) Bases de datos no relacionales: son un tipo de base de datos que se implementa cuando la información a almacenar es muy compleja para poder ser expresada en una tabla. A diferencia de las bases de datos relacionales, no es necesario conocer *a priori* qué es lo que se desea almacenar, ya que las bases de datos no relacionales son más flexibles y pueden almacenar cualquier tipo de dato sin importar su estructura. En ellas, no se tiene un identificador que sirva para relacionar los conjuntos de datos ni un esquema exacto de lo que se va a almacenar, por lo que se utilizan datos del tipo JSON. Actualmente, las bases de datos no relacionales más utilizadas son MongoDB, Redis y Cassandra.

Ejemplo

Supongamos que se tienen diferentes plataformas que incorporan diferentes sensores y cámaras para dar información sobre el tránsito en una vecindad y qué tipo de vehículo circula. Cada cierto tiempo, una estación recibe un reporte con los resultados de cada plataforma en ficheros de tipo JSON, ya que no se sabe *a priori* qué tipo de sensores tiene cada plataforma ni cómo serán los datos que se recibirán. Los reportes tienen la siguiente forma:

```
{
"Plataforma_ID":0001,
"Ubicacion":"6.054, 987.0, 69.78",
"Vistas":[
"auto",
"bicicleta",
"desconocido",
"auto",
"auto",
],
},
"Sonido":{
"min":15,
"max":44
}
}
```

```
{
"Plataforma_ID":0002,
"Ubicacion":"8.022, 767.0, 29.87",
"Vistas":[
"bicicleta",
"bicicleta",
"desconocido",
"desconocido",
],
},
"Viento":{
"min":120,
"max":298
}
}
```

Como podemos apreciar, estas plataformas ofrecen reportes distintos a la otra y se tiene muy poca o ninguna información sobre sus estructuras, por lo que no es recomendable diseñar una base de datos relacional para almacenar estos datos. En este ejemplo, lo mejor es utilizar una base de datos no relacional que almacene toda la información tal cual se le presente. Es importante destacar que es posible procesar estos datos posteriormente para transformarlos en una base de datos relacional para su futuro análisis, si así se desea. Sin embargo, típicamente, este paso no es necesario.

En algunos casos, cuando se dispone de múltiples bases de datos pertenecientes a una misma organización, se suelen implementar almacenes de datos conocidos como *data warehouses*, que tienen como objetivo optimizar las consultas y generar informes a partir de un resumen de los datos provenientes de diferentes fuentes. En estos almacenes se recopila la información de todas las bases de datos de una forma preestablecida, de manera que facilite sus consultas. Debe asegurarse la disponibilidad, la consistencia y la preservación de los datos a lo largo del tiempo.

En el caso de necesitar un subconjunto de datos provenientes del almacén, con el propósito de brindar soporte a un área específica, se puede implementar un *datamart*. Este tipo de base de datos es simplemente una capa de acceso al almacén, que filtra y selecciona los datos necesarios para el análisis en un área en específico, facilitando así la toma de decisiones.

2.3. Preprocesado

Es la etapa en la que se preparan los datos para su análisis posterior. En ella, se aplica una serie de técnicas de modo que los analistas no deban preocuparse por la calidad ni procedencia de estos datos. A continuación, se presentan las técnicas habitualmente utilizadas en la etapa de preprocesado. Sin embargo, la implementación de las mismas va a depender del tipo de datos a tratar y del análisis que se desee realizar posteriormente.

Lectura recomendada

Han, Jiawai; **Kamber, Micheline**; **Pei, Jian** (2012). *Data Mining: Concepts and Techniques*. Waltham: Elsevier.

2.3.1. Integración

La integración es la combinación de datos provenientes de fuentes diferentes, con el objetivo de agruparlos en una estructura única que facilite sus análisis y permita realizar inferencias más profundas.

Ejemplo

Si se conocen los identificadores dentro de una base de datos «A» de un grupo de usuarios suscritos a un cierto servicio y también se sabe, inequívocamente, que el mismo grupo de usuarios está suscrito a otros servicios descritos en la base de datos «B». Entonces, podemos fusionar esta información en una sola tabla «C» que nos permita hacer análisis más pertinentes sobre dicho grupo.

2.3.2. Selección

Esta técnica se basa en el filtrado de los datos que se encuentran dentro de un fichero o base de datos con el objetivo de seleccionar solo los datos de interés. Para ello, se adoptan criterios de búsqueda que nos permitan discernir, entre el grupo de datos, aquellos que son realmente necesarios para el análisis posterior.

Ejemplo

En una base de datos clínica, se desea analizar a los pacientes con riesgo de infarto. De este modo, se filtran los datos por medio de una simple búsqueda de los pacientes mayores de 30 años, ya que se considera que los pacientes más jóvenes no presentan un riesgo significativo por lo que pueden sesgar los análisis.

2.3.3. Reducción de datos

Esta técnica se utiliza cuando se pretende analizar grandes volúmenes de datos. Consiste en aplicar ciertos métodos sobre un conjunto de datos para obtener un menor volumen o una representación reducida de estos, manteniendo en todo momento la integridad de los datos originales. En otras palabras, los análisis realizados sobre la representación reducida de los datos deben producir los mismos (o muy aproximados) resultados que al aplicarlos sobre el conjunto original de datos. A continuación, se presentan tres métodos típicamente utilizados para reducir la dimensionalidad:

a) Reducción de dimensionalidad: son los métodos que tienen como objetivo reducir el número de variables aleatorias o atributos bajo consideración. Entre los más utilizados están las transformaciones *wavelet* o el análisis de componentes principales (ACP), los cuales transforman o proyectan los datos originales en un espacio más reducido. También es posible utilizar el método de selección de subconjuntos de atributos (*attribute subset selection*), con el que se detectan y eliminan atributos o dimensiones irrelevantes o redundantes. Otros modelos paramétricos se utilizan para estimar los datos, de modo que se almacenan solo sus parámetros. Por ejemplo, modelos de regresión o logarítmicos.

b) Reducción de cantidad: consiste en sustituir el volumen original de datos por una representación alternativa de menor volumen. Entre los modelos más utilizados, tenemos los modelos no paramétricos como, por ejemplo, histogramas, agrupamiento (*clustering*) y muestreo (*sampling*).

c) Compresión de datos: son técnicas que consisten en aplicar transformaciones a un conjunto de datos para obtener una representación comprimida de los datos originales. Estas técnicas pueden dividirse en técnicas con pérdidas o sin pérdidas, dependiendo de si se pueden reconstruir los datos originales a partir de los datos comprimidos manteniendo la integridad de la información. Nótese que los métodos de reducción de dimensionalidad o reducción de cantidad pueden ser considerados como formas de compresión de datos.

2.3.4. Conversión

También conocida como transformación, la conversión de datos implica una serie de técnicas que tienen como objetivo convertir el formato original de los datos a un formato más plano y entendible, capaz de facilitar el análisis posterior o minería de datos. Entre las técnicas más utilizadas están:

a) Suavizado (*smoothing*): tiene como objetivo eliminar el ruido de los datos. Las técnicas incluyen el filtrado, *binning*, regresiones y *clustering*.

b) Construcción de atributos: consiste en agregar nuevos atributos resultantes de operaciones sobre otro conjunto de atributos para facilitar el proceso de minería. En otras palabras, se crean nuevos atributos a partir de cálculos sobre las variables disponibles.

Ejemplo

Al calcular la ratio entre dos variables o realizar conversiones en las unidades.

c) Agregación de datos: se aplican operaciones de síntesis o de adición a un conjunto de datos para permitir múltiples niveles de abstracción en el proceso de minería de datos y resumir el conjunto de atributos en uno solo.

Ejemplo

Sumar el número de suscripciones diarias a un servicio determinado para obtener las suscripciones mensuales.

d) Normalización: consiste en modificar la escala de los atributos de modo que se encuentren dentro de un rango más pequeño, típicamente, entre -1.0 y 1.0 o entre 0.0 y 1.0.

e) Discretización: se basa en reemplazar los valores numéricos de los atributos por etiquetas, lo que resulta en un nivel más alto de abstracción. Estas etiquetas pueden ser en intervalos o conceptuales.

Ejemplo

Al hablar de las edades de un grupo de pacientes, en vez de colocar dentro del atributo el valor numérico de la edad, podemos colocar una etiqueta con intervalos de 0-20, 21-40, 41–60, o colocar etiquetas conceptuales como: infante, niño, adolescente, adulto o anciano. Nótese que estas etiquetas pueden organizarse a su vez de forma recursiva, lo que resulta en la generación de una jerarquía de conceptos.

f) Generación de jerarquía de conceptos para datos nominales: se utiliza, a diferencia del caso anterior, cuando los atributos no poseen valores numéricos sino nominales.

Ejemplo

Una dirección puede reemplazarse por etiquetas conceptuales de un nivel superior, como: ciudad, región o país.

2.3.5. Limpieza

La limpieza de datos o *data cleaning* se considera uno de los pasos más importantes del preprocesado de los datos, ya que la calidad y la veracidad de los resultados van a depender en gran parte del correcto desarrollo de esta fase. La limpieza de datos consiste en eliminar las inconsistencias de estos, ya sea de forma automática a través de una detección, o de forma manual tras una inspección más detallada. Este proceso se realiza ya que, normalmente, el dato en «bruto» viene contaminado con ruido e inconsistencias que no permiten la obtención de conocimiento con la aplicación directa de los métodos de análisis.

Entre las inconsistencias típicamente encontradas en los datos tenemos: la presencia de datos perdidos, datos no definidos (cero, vacío y nulo) o la aparición de valores extremos. Los métodos aplicados para la resolución de estos problemas van a depender del origen y del tipo de datos a tratar, como veremos a continuación.

a) Análisis de datos perdidos: la denominación de datos perdidos o *missing data* se emplea cuando, para una variable u observación, no se tiene ningún dato. Este es uno de los problemas más comunes que se dan en el momento de la verificación de los datos, antes de realizar su limpieza. Pueden surgir por el mal funcionamiento de los dispositivos o procesos de captura, errores (olvidos) humanos o por errores de transmisión de datos. Dependiendo de su naturaleza, existen diferentes soluciones para resolver este inconveniente. Entre ellas tenemos:

- **Ignorar el atributo:** se aplica cuando no se tiene ninguna referencia sobre el atributo ni su procedencia. Es una técnica muy poco efectiva ya que, dependiendo de su importancia en el análisis, puede resultar en la pérdida completa de un individuo o clase, lo que conlleva a la pérdida de otros atributos que sí podrían estar definidos.

- **Completado manual:** es una técnica que consiste en rellenar manualmente el dato faltante por un dato conocido *a priori*. También se considera poco efectiva, ya que requiere mucho tiempo de verificación y resulta inaplicable cuando se trata de grandes volúmenes de datos.

- **Completado con una constante global:** consiste en reemplazar los valores perdidos de un atributo por una misma constante o etiqueta (por ejemplo, «Desconocido» o «∞»). Sin embargo, tampoco se considera una técnica infalible ya que, si no se toman las precauciones necesarias en el programa de análisis, este puede interpretar estos datos como atributos de interés debido a su valor en común.

- **Completado a partir de una medida de tendencia central:** se reemplazan los datos faltantes por un mismo valor que es el resultado de una medida de tendencia central (media o mediana), dependiendo del tipo de distribución de los datos.

- **Completado con el valor más probable:** se trata de la implementación de métodos probabilistas para determinar o predecir el valor que podría tomar dicho atributo dando un conjunto de datos. Entre las herramientas más utilizadas en este caso tenemos: regresiones, inferencias basadas en modelos bayesianos o a partir de árboles de decisión.

b) Análisis de datos no definidos: típicamente, al referirse a datos no definidos, nos referimos a los ceros, vacíos y nulos. Sin embargo, el entendimiento y la definición de estos van a depender del contexto y el origen de los datos.

Para poder entender cuándo se debe utilizar este tipo de datos, es necesario entender las diferencias entre ellos. En primer lugar, los ceros se consideran valores legítimos y válidos para una medida numérica si el cero es parte del dominio (por ejemplo, resultado final en un partido de fútbol: 0 a 1).

Por otro lado, un dato vacío representa la existencia de un atributo del que se desconoce su valor. En otras palabras, este puede ser hallado. Por ejemplo, una fecha que se olvidó colocar en un campo de una tabla puede generar un valor vacío, ya que no se sabe su valor, pero este valor se sabe que existe. Al trabajar con cadenas de caracteres o *strings* suelen utilizarse los datos vacíos como equivalentes a un valor cero, es decir, significa que el valor de ese campo para esa instancia es «nada», lo cual es completamente legítimo.

Por último, los datos nulos representan la no existencia del valor, ni siquiera de la «nada». Por ejemplo, un campo de una tabla indicando si un cliente contrató cierto servicio puede colocarse como nulo si este cliente nunca ha contratado ningún servicio a la compañía (el dato no existe).

Como se puede apreciar, el uso de este tipo de datos va a depender del mensaje o significado que se desee transmitir dentro del conjunto de datos. Nótese que los datos perdidos pueden ser reemplazados por este tipo de valores, si se considera necesario. Además, durante esta etapa de preprocesado, los datos de

Bibliografía recomendada

Squire, Megan (2015). «Data Mining». Birmingham: Packt Publishing.

este tipo pueden reemplazarse entre sí dependiendo del análisis que se realizará posteriormente. Por ejemplo, reemplazar los ceros por datos nulos para evitar que interfieran en ciertas pruebas estadísticas.

c) Análisis de valores extremos: se consideran valores extremos aquellos valores que están muy por encima de la distribución normal para una variable o población. También puede entenderse como observaciones que se desvían tanto del resto que pueden despertar sospechas de si fueron generadas por un mecanismo diferente. Por lo general, se consideran valores extremos los que superan ±3 desviaciones estándar cuando la muestra es suficientemente grande. La aparición de estos valores extremos puede afectar a los resultados de forma adversa de diferentes maneras:

- Incrementando el error en la varianza y reduciendo el poder en test estadísticos, alterando así sus resultados.
- Si estos valores no son generados de forma aleatoria, pueden alterar significativamente las probabilidades de errores de tipo I y tipo II (ver la definición en el apartado del glosario).
- Pueden afectar los resultados sobre correlaciones entre variables o regresiones.
- Sesgan los cálculos y estimaciones sobre un conjunto de datos al no pertenecer a la población de interés.

Estos valores pueden aparecer por diferentes razones y deben tratarse dependiendo de cada caso. Por ejemplo, una posible razón de su aparición es por errores en la captura de los datos, como errores humanos de transcripción (en una encuesta, el entrevistador olvida colocar un punto decimal en el campo, lo que resulta en un atributo con un valor de 100 en vez de 1,00). Estos errores pueden solucionarse fácilmente volviendo a los registros originales o volviendo a realizar el cálculo.

Bibliografía recomendada

Osborne, Jason W. (2010, marzo). «Data cleaning basics: Best practices in dealing with extreme scores». *Newborn and Infant Nursing Reviews* (vol. 10, n.º 1, págs. 37-43).

Otro ejemplo de aparición de este tipo de valores es por errores o sesgo en el muestreo, es decir, cuando los valores son producidos por la captura de datos que no pertenecen a la misma población a analizar. En un análisis de los salarios de los trabajadores de una compañía, observamos que ciertos individuos presentan valores extremos; al verificarlos, comprendemos que corresponden a los salarios de los accionistas (dueños) de la compañía. En este caso, los datos no pueden considerarse como legítimos ya que no pertenecen a la misma población de trabajadores. En estos casos, deben eliminarse los datos que no pertenecen a la población.

Nótese, que no todos los valores extremos son ilegítimos o deben eliminarse, por lo que es importante conocer su origen para poder tomar una decisión sobre qué procedimiento realizar sobre estos.

2.4. Análisis

Esta etapa, también conocida como fase de minería de datos, tiene como objetivo crear diferentes modelos que permitan explicar cómo son los datos y cuáles son sus características principales. De esta manera, a partir de estas respuestas se pretende responder a las preguntas que se plantean en el marco de un proyecto donde dichos datos fueron extraídos.

Dependiendo de la naturaleza de los datos y de los objetivos del proyecto, se pueden aplicar diferentes tipos de análisis. Los más utilizados se presentan a continuación.

Bibliografía recomendada

Minguillón, Julià (2016). «Fundamentos de Data Science». Editorial UOC.

- **Análisis estadístico descriptivo**. Consiste en modelar los datos a partir de un conjunto reducido de valores de acuerdo a alguna distribución conocida. El objetivo es describir adecuadamente las características intrínsecas de dicho conjunto. En este caso, no se tiene en cuenta si el conjunto en tratamiento forma parte o no de un conjunto de datos superior. Por ejemplo, cálculo de medias, medianas, desviación estándar, etc.

- **Análisis estadístico inferencial**. A diferencia del caso anterior, en este tipo de análisis se intenta modelar los datos a través de una distribución desconocida. La razón, es que se asume que el conjunto de datos a analizar representa solo una fracción de la totalidad de una población. Por consiguiente, el objetivo en este caso es inferir cómo es la población. Para ello, se asume un grado de error en las estimaciones debido a que no se dispone de todos los datos. Ejemplos de este tipo de análisis son: contrastes de hipótesis, regresiones, correlaciones, etc.

- **Extracción de características**. Se sirve de todos los atributos disponibles dentro de un conjunto de datos para, a través de diferentes operaciones, obtener atributos nuevos que, manteniendo la integridad de los datos originales, permitan representar de una mejor manera los aspectos que se desean analizar dentro del proyecto. Este nuevo grupo de atributos se conoce también como características, o *features*, en inglés.

Ejemplo

Si se desea determinar el consumo eléctrico de una serie de circuitos electrónicos, pero solo se dispone de datos de las corrientes «I» y de los componentes resistivos «R». Entonces, por medio de la Ley de Ohm, es posible extraer una característica nueva llama potencia eléctrica ($P = I^2 \cdot R$) que se encuentra más ligada al concepto de consumo, de manera que se facilita la interpretación y análisis de estos datos.

Por otro lado, existen también otros métodos que permiten extraer características de un conjunto de datos a partir de análisis específicos (por ejemplo, análisis de la varianza) sobre los datos disponibles. Entre los ejemplos

más comunes de este tipo de análisis tenemos: el análisis de componentes principales (ACP), redes neuronales simples y *deep learning*, el cual, por medio de una combinación de redes neuronales, busca extraer un conocimiento a partir del conjunto dado de datos.

- **Modelos supervisados**. Son métodos que se aplican cuando se dispone de datos en los que uno o varios atributos representan el objetivo del problema que se pretende resolver. De esta manera, se diseñan modelos que buscan predecir los valores de nuevas variables de entrada correspondientes a dichos atributos a partir de las otras variables del conjunto de datos. De forma general, las variables que se busca predecir son conocidas como variables dependientes, mientras que las demás son conocidas como variables independientes. Entre los métodos basados en modelos supervisados más utilizados tenemos: clasificadores, árboles de decisión, redes neuronales y modelos lineales.

- **Modelos no supervisados**. Se utilizan cuando no se dispone de una variable o atributo objetivo. Por esta razón, los modelos intentan comparar los datos entre sí con el objetivo de encontrar diferencias o similitudes que permitan detectar algún tipo de estructura interna que pueda formar agrupaciones en función de un criterio de evaluación. Estos modelos pueden ser utilizados para extraer nuevas características que, a su vez, puedan ser utilizadas en otro tipo de modelos como, por ejemplo, modelos supervisados. Ejemplos de métodos utilizados para el diseño de estos modelos son: algoritmos de *clustering*, redes neuronales y mapas autoorganizativos.

2.5. Visualización

También conocida como fase de representación, la visualización consiste en aprovechar la capacidad del sistema visual humano para la detección de patrones, tendencias y extracción de un conocimiento a partir de gráficos, modelos y cualquier otra herramienta que permita interactuar con los resultados de la fase de análisis.

Para una implementación más eficiente de esta fase, es necesario que las representaciones de los datos posean una interfaz de navegación sobre ellos, permitiendo operaciones como, por ejemplo, selección, comparación, agregación, marcado, etc. De este modo se facilita la extracción de conocimiento a través de la combinación de un conjunto de datos, sin sobrecargar de información al usuario.

Existen siete tareas básicas que permiten un nivel más alto de abstracción para la visualización de datos:

- **Panorama general** (*overview*): permitir una vista general de la colección de datos.
- **Acercamiento** (*zoom*): permitir el acercamiento a un punto de interés.
- **Filtrado** (*filter*): permitir el filtrado de los elementos no interesantes según un cierto criterio.
- **Detalles a petición** (*details on demand*): obtener detalles sobre un elemento o grupo de ser necesario.
- **Relaciones** (*relate*): permitir la visualización de tendencias entre los elementos.
- **Historial** (*history*): mantener un historial de acciones para poder revenir a pasos anteriores o repetir acciones.
- **Extracción** (*extract*): permitir la extracción de un subconjunto y sus detalles correspondientes.

Bibliografía recomendada

Shneidermann, Ben (1996). «The eyes have it: a task by data type taxonomy for information visualization». *Proceedings of the 1996 IEEE Symposium on Visual Languages* (vol. 96, págs. 336-343).

De esta manera se combina la capacidad humana con la potencia de un sistema informático que permita este tipo de visualizaciones sobre un conjunto de datos y sobre los resultados obtenidos después del análisis, para así extraer un verdadero conocimiento del conjunto de datos original y responder a las preguntas del proyecto en cuestión.

2.6. Publicación

Esta fase final del ciclo de vida de los datos tiene como objetivo documentar los resultados obtenidos, de forma que sea posible su reutilización para la resolución de nuevos proyectos.

Para una correcta publicación de estos datos, es necesario utilizar espacios que estén optimizados para ello. Por ejemplo, los repositorios digitales. La razón de esto es que este tipo de espacios garantizan la disponibilidad de dichos datos a lo largo del tiempo y, además, son capaces de adaptar el formato y la estructura interna de los datos para que sea posible su acceso a través de otras herramientas. Esta característica se conoce como preservación y es un aspecto que se busca cumplir a la hora de una publicación de datos.

Otra característica a tener en cuenta para una correcta publicación es la diseminación. Esta tiene como objetivo asegurar la reutilización de los datos facilitando su búsqueda, acceso y entendimiento. Para lograrlo, es importante que se disponga de metadatos adecuados que permitan una descripción correcta de los datos, indicando sus condiciones y limitaciones de uso para, así, permitir a terceros el entendimiento de su origen y naturaleza.

Bibliografía recomendada

Minguillón, Julià (2016). «Fundamentos de Data Science». Editorial UOC.

Como se ha explicado anteriormente, en la fase de captura, idealmente, un repositorio digital debería incorporar un API que facilitase el acceso a los datos. Además, es recomendable que también se permita seleccionar el tipo de formato en el que se descargan los datos, como por ejemplo *comma-separated values* (CSV), *JavaScript Object Notation* (JSON), *Extensible Markup Language* (XML), etc. De este modo se garantiza que los datos y sus resultados puedan ser utilizados en nuevos proyectos, y que permitan extraer un valor añadido de ellos.

Resumen

En este módulo didáctico se han revisado los conceptos básicos de la sociedad de la información. En primer lugar, en la Introducción, se han descrito los perfiles principales de la ciencia de datos (científico de datos, ingeniero de datos, arquitecto de datos, analista de datos, estadístico, administrador de base de datos, analista de negocio y líder de ciencia de datos).

En el apartado "¿Qué son los datos?" se han descrito las diferentes clasificaciones de datos (según estructura, nivel de accesoy tipo de información), y los diferentes parámetros para evaluar la calidad de la información en los datos, mostrando ejemplos reales para entender las diferentes dimensiones (exactitud, completitud, consistencia, atemporalidad, unicidad y validez).

Finalmente, en el apartado "Ciclo de vida de los datos" se han descrito las principales fases del ciclo de vida, que son captura (que incluye creación y extracción), almacenamiento, análisis preprocesado (que incluye integración, selección, reducción de datos, conversión y limpieza), análisis (que puede ser descriptivo, inferencial, de extracción de características, modelos supervisados y no supervisados), representación y publicación.

Ejercicios de autoevaluación

1. ¿Qué diferencias hay entre un *data scientist* y un *data engineer*?

2. Imagínate que creas una empresa. Pon un ejemplo de cada uno de los tres tipos de datos que tu empresa podría gestionar. Coge un ejemplo de los tres que has mencionado anteriormente y explica cada fase del ciclo de vida de los datos.

3. Explica con tus propias palabras cuándo es útil realizar *web scraping*. Imagina que tienes un negocio, explica cuándo podría ser útil aplicar *web scraping*.

Solucionario

1. Las diferencias entre un *data scientist* y un *data engineer* son:

- Un *data scientist* es una persona capaz de plantearse las preguntas adecuadas a partir de un conjunto de datos relativo a un dominio, y establecer qué métodos y técnicas son los más adecuados para extraer el conocimiento necesario para responder a dichas preguntas, para posteriormente realizar dicha tarea. Este perfil está orientado principalmente a resolver el «¿qué?».
- Un *data engineer* es una persona capaz de preparar un conjunto de datos de forma que tenga la estructura e información adecuadas para su posterior análisis, así como de llevar a la práctica una solución o prueba de concepto y convertirla en una implementación que pueda usarse en un entorno productivo real. Este perfil está más orientado a resolver el «¿cómo?».

2. La empresa que se estudia es una empresa que predice posibles complicaciones en las revisiones de salud anuales que se realizan a los empleados de una empresa. Un ejemplo de cada uno de los 3 tipos de datos que esta empresa podría gestionar es:

- Datos simples: el peso corporal de un empleado en kilogramos.
- Datos compuestos o estructurados: una imagen médica como, por ejemplo, una radiografía de fémur.
- Datos semiestructurados o no estructurados: el documento de análisis de la revisión anual periódica.

Las etapas del ciclo de vida de los datos son los siguientes (se coge como ejemplo el tipo de dato simple «peso corporal»):

- **Captura:** una persona se pesa en una báscula e indica su peso.
- **Almacenamiento:** la persona almacena el valor en una aplicación (*app*) que monitorea el seguimiento de su peso.
- **Preprocesado:** fusión (se fusiona con otros datos ya almacenados la aplicación), conversión (se convierte a kilogramos si es necesario), limpieza (se detectan inconsistencias, por ejemplo, pesos que no son razonables, que son *outliers*), agregación (se agregan los diferentes valores del peso para poder conocer la evolución, por ejemplo, se podría hacer una agregación mensual), creación de nuevas variables (si se dispone de la altura se podría calcular, por ejemplo, el índice de masa corporal).
- **Análisis:** descriptivo (describir a la persona que se está pesando), estadístico inferencial (quién está realizando la medida), extracción de características (cálculo del índice de masa corporal), reducción de dimensionalidad (si hay varias medidas se puede realizar un *principal component analysis* para reducir la dimensionalidad), modelos supervisados (se intenta predecir cuál será el peso del próximo mes a partir de datos obtenidos), modelos no supervisados (se realiza *clustering* para agrupar los datos), visualización (visualización de los patrones, irregularidades u *outliers*).
- **Visualización:** la aplicación puede proporcionar una visualización temporal distinta de los resultados y, por qué no, también geoespacial.
- **Publicación:** los datos se pueden publicar para que otras personas los puedan analizar o utilizar.

3. Es útil realizar *web scraping* cuando no disponemos de API para acceder a los datos web. Como ejemplo de un negocio, supongamos que tengo una tienda que vende zapatos y quiero hacer un seguimiento de los precios de mi competencia. Podría ir al sitio web de mi competidor todos los días para comparar el precio de cada zapato con el mío, sin embargo, esto tomaría mucho tiempo y no escalaría si vendiera miles de zapatos o si necesitase controlar los cambios de precios con más frecuencia. O tal vez solo quiero comprar un zapato cuando esté a la venta. Podría volver y revisar el sitio web del zapato cada día hasta que tenga suerte, pero el zapato que quiero podría no estar en oferta durante meses. Estos dos procesos manuales repetitivos podrían reemplazarse con una solución automatizada utilizando las técnicas de *web scraping*.

Glosario

ACP *m* Véase ***principal component analysis***.

API *f* Véase ***application programming interface***.

application programming interface *f* Véase **interfaz de programación de aplicaciones**.

CAPTCHA *m* Véase ***completely automated public turing test to tell computers and humans apart***

comma-separated values *m* Formato de archivo de texto donde se utilizan comas para separar los campos.
sigla **CSV**

completely automated public Turing test to tell computers and humans apart *m* Test de Turing completamente automático y público para diferenciar ordenadores de humanos.
sigla **CAPTCHA**

CSV *m* Véase ***comma-separated values***.

data, information, knowledge and wisdom Véase **datos, información, conocimiento y sabiduría**.

database management system *m* Sistema para crear y gestionar bases de datos. Estos sistemas proporcionan a los usuarios y programadores una forma sistemática de crear, recuperar, actualizar y gestionar datos.
sigla **DBMS**

datos, información, conocimiento y sabiduría Pirámide que permite obtener valor de los datos hasta llegar a la sabiduría.
sigla **DIKW**

DBMS *m* Véase ***database management system***.

Error tipo I *m* El error tipo I, también denominado error de tipo alfa (α) o falso positivo, es el error que se comete cuando el investigador rechaza la hipótesis nula siendo esta verdadera en la población.

Error tipo II *m* El error tipo II, también llamado error de tipo beta (β) (β es la probabilidad de que exista este error) o falso negativo, se comete cuando el investigador no rechaza la hipótesis nula siendo esta falsa en la población.

extensible markup language *m* Véase **lenguaje de marcado extensible**.

HTML *m* Véase ***HyperText Markup Language***.

HyperText Markup Language *m* Véase **lenguaje de marcas de hipertexto**.

interfaz de programación de aplicaciones *f* Conjunto de rutinas que permiten acceder a funciones de un determinado software; en Internet, las API permiten acceder al contenido de un sitio web.
sigla **API**

JavaScript object notation Formato de archivo de estándar abierto que utiliza texto legible por el ser humano para transmitir objetos de datos que consisten en pares de atributos-valor y arreglos de datos. Es un formato de datos muy común utilizado para la comunicación asíncrona navegador-servidor.
sigla **JSON**

JSON Véase ***JavaScript object notation***.

lenguaje de marcado extensible *m* Metalenguaje extensible de etiquetas, desarrollado por el World Wide Web Consortium (W3C) y adaptado del SGML (*Standard Generalized Markup Language*).
sigla **XML**

lenguaje de marcas de hipertexto *m* Lenguaje de marcado utilizado para la elaboración de páginas web.

sigla **HTML**

lenguaje estructurado de consultas *m* Lenguaje de acceso a una base de datos.
sigla **SQL**

minería de datos *f* Proceso de análisis para descubrir patrones en conjuntos de datos. Se aplican métodos de aprendizaje automático, estadísticos, entre otros.

paquete estadístico para las ciencias sociales *m* Paquete estadístico utilizado para analizar datos.
sigla **SPSS**

principal component analysis *m* Procedimiento estadístico que utiliza una transformación ortogonal para convertir un conjunto de observaciones o variables posiblemente correlacionadas en un conjunto de valores de variables linealmente no correlacionadas, llamadas componentes principales.
sigla **ACP**

recovery time objective *m* Es la duración específica del tiempo y un nivel de servicio dentro del cual se debe restaurar un proceso comercial después de un desastre (o interrupción) para evitar consecuencias inaceptables asociadas con una interrupción en la continuidad del negocio.
siglas **RTO**

RTO *m* Véase ***recovery time objective***.

SAS *m* Véase ***statistical analysis system***.

sistema estadístico de análisis *m* Lenguaje de programación utilizado para analizar datos.
sigla **SAS**

SPSS *m* Véase ***statistical package for the social sciences***.

SQL *m* Véase ***structured query language***.

statistical analysis system *m* Véase **sistema estadístico de análisis**.

statistical package for the social sciences *m* Véase **paquete estadístico para las ciencias sociales**.

structured query language *m* Véase **lenguaje estructurado de consultas**.

URL Véase ***uniform resource locator***.

uniform resource locator Referencia a un recurso web que especifica su ubicación en una red informática y un mecanismo para recuperarla.
siglas **URL**

XML *m* Véase ***extensible markup language***.

Bibliografía

Fayyad, Usama; **Piatetsky-Shapiro, Gregory**; **Smyth, Padhraic** (1996, marzo). *«From data mining to knowledge discovery in databases»*. AI Magazine (vol. 17, n.º 3, págs. 37-54).

Han, Jiawai; **Kamber, Micheline**; **Pei, Jian** (2012). *Data Mining: Concepts and Techniques*. Waltham: Elsevier.

Jarman, Kristin H. (2013). *The art of data analysis. How to answer almost any question using basic statistics*. Hoboken, NJ: Wiley.

Minguillón, Julià (2016). *«Fundamentos de Data Science»*. Editorial UOC.

Osborne, Jason W. (2010, marzo). *«Data cleaning basics: Best practices in dealing with extreme scores»*. Newborn and Infant Nursing Reviews (vol. 10, n,º 1, págs. 37-43).

Riquelme, José Cristóbal; **Ruiz, Roberto**; **Gilbert, Karina** (2006). *«Minería de datos: conceptos y tendencias»*. Inteligencia artificial: Revista Iberoamericana de Inteligencia Artificial (vol. 10, n.º 29, págs. 11-18).

Shneidermann, Ben (1996). *«The eyes have it: a task by data type taxonomy for information visualization»*. Proceedings of the 1996 IEEE Symposium on Visual Languages (vol. 96, págs. 336-343).

Squire, Megan (2015). *«Data Mining»*. Birmingham: Packt Publishing.