Autor: María José
Gómez Silva

Machine Learning con Python. Semana 1.

CAPÍTULO 2. INGESTA DE DATOS.

La ingesta de datos (data ingestión, en inglés) es el proceso de recolección de datos de distintas fuentes en un repositorio común de almacenaje. El propósito de este proceso es que los datos necesarios sean fácilmente accesibles para su empleo en el entrenamiento y la evaluación de los algoritmos de Machine Learning.

En este capítulo se estudiarán distintos tipos y posibles fuentes de procedencia de los datos, y como leerlos y almacenarlos en estructuras de tipo *Series* o *DataFrames*, que son estructuras propias de la librería de *Pandas* en Python.

Además, se introducirán algunas de las múltiples funciones y operaciones necesarias para la gestión de los datos, y la combinación de estos cuando proceden de distintas fuentes o archivos.

Por último, y como enlace con el siguiente capítulo de "Preprocesado", se presentarán herramientas para el análisis y visualización de los datos, de modo que faciliten las tareas posteriores de preparación de los datos.

2. 1 Tipos y Fuentes de Datos

Los algoritmos de Machine Learning trabajan con una amplia variedad de tipos de datos procedentes de múltiples fuentes, como bases de datos, vídeos, archivos de audio, etc.

Independientemente de su origen, los datos deben ser almacenados y organizados de forma apropiada para su empleo en los algoritmos de Machine Learning. Para ello, es importante conocer las características de los datos, tales como sus dimensiones, número de variables, si son heterogéneos, si tienen dependencia temporal, o si van acompañados por anotaciones. Además, es necesario tener en cuenta cómo están almacenados o el tipo de archivo en que están contenidos, para establecer el modo adecuado de acceder a ellos. A continuación, se detallan cada una de las consideraciones mencionadas.

Dimensiones de los datos

El número de dimensiones de los datos dependerá de su naturaleza y complejidad. Por ejemplo, una grabación de una señal de audio, como puede ser cualquier canción en formato mp3 o wav, es una señal de una única dimensión, es decir, una simple sucesión de valores. En cambio, en una imagen en escala de grises aparecen datos en 2 dimensiones, ancho por alto, ya que una imagen es una matriz de valores con tantas celdas como píxeles tenga la imagen, como puede observarse en la Figura 1.

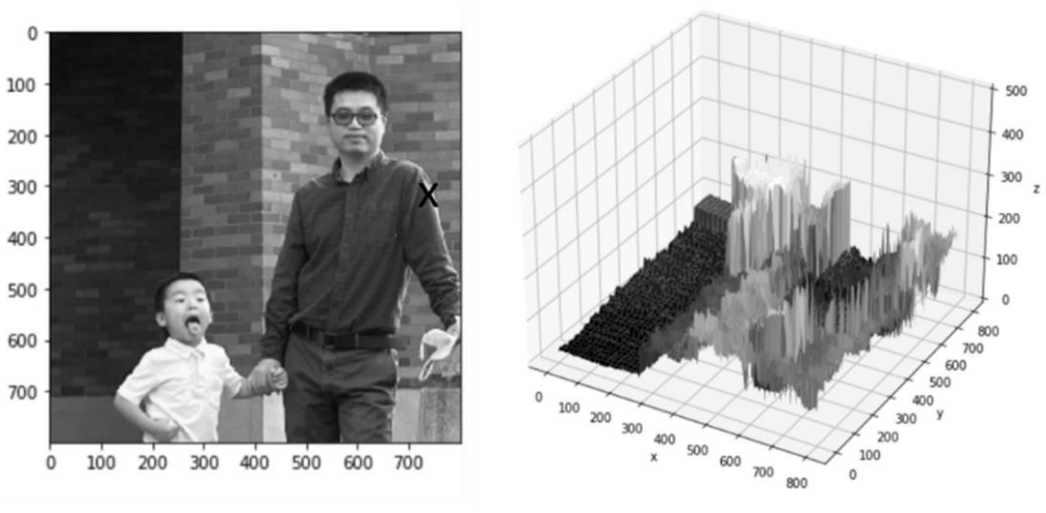

Figura 1. Imagen en escala de grises.

En cambio, si tomamos puntos del entorno con un escáner laser, podemos adquirir una nube de puntos, como la de la Figura 2. Cada punto constituye un dato con 3 dimensiones, sus coordenadas x, y, z.

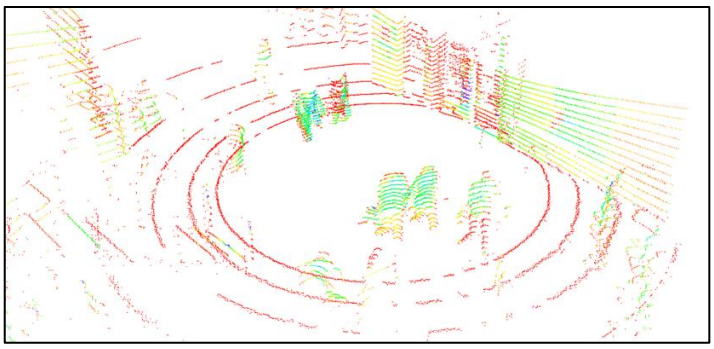

Figura 2. Nube de puntos tridimensionales

Sin embargo, el número de dimensiones de los datos no tiene por qué estar necesariamente ligado al número de dimensiones espaciales que representan. Por ejemplo, si tenemos una imagen en color, en lugar de en escala de grises, la matriz de

píxeles adquiere profundidad. Esto significa que la matriz tiene una dimensión más para poder representar el color como combinación de tres valores (rojo, verde y azul), en lugar de uno sólo (nivel de gris), como se observa en la Figura 3. Además, si los datos son de video, es decir, una sucesión de imágenes en el tiempo, la nueva coordenada de tiempo estaría introduciendo una dimensión más.

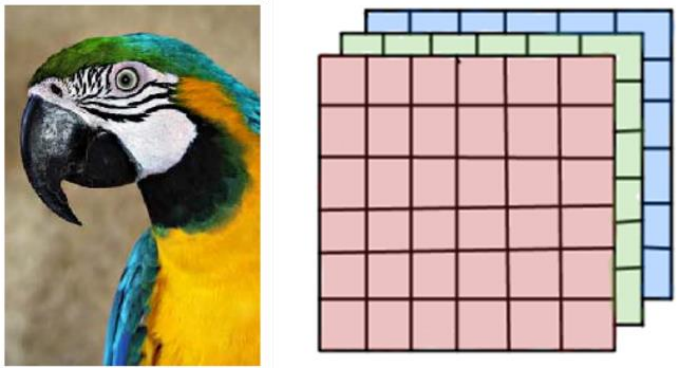

Figura 3. Imagen en color y esquema de la matriz imagen con 3 canales.

Numero de variables

De cada muestra o entrada de nuestra lista de datos, podemos tener almacenada una o más variables. Por ejemplo, si tenemos un listado de viviendas en venta, en el que cada muestra es un inmueble, y lo único que almacenamos de cada inmueble es su precio de venta, en ese caso, tendremos *datos con una única variable*. Los datos con una variable se pueden representar simplemente como una serie de valores. Por el contrario, si aumentamos el número de variables a tener en cuenta (como el número de habitaciones, el número de baños, la superficie total en metros cuadrados, la dirección, los gastos de comunidad, el año de construcción o la fecha en que se puso a la venta cada inmueble) estaremos generando *datos multivariable*. La forma más usual de organizar y almacenar datos multivariable es mediante tablas, en las que cada muestra se almacena en una fila de la tabla, y donde cada columna se corresponde con el valor de una determinada variable.

Heterogeneidad

Cuando tenemos datos con múltiples variables, puede que todas ellas tengan el mismo tipo de dato, por ejemplo, numérico. En ese caso los datos son *homogéneos* en cuanto a formato o tipo de variable. En cambio, es bastante habitual que las bases de datos contengan variables de distintos tipos (numéricos, cadenas de caracteres, binarios, etc.). En ese caso hablamos de datos *heterogéneos*, también llamados a veces, *multimodales*.

Consistencia temporal

Existen algunos tipos de datos, como los de video, que necesariamente deben presentar consistencia temporal, es decir, el orden cronológico de los datos les da sentido. En otros casos, la información de tiempo es opcional, como en el ejemplo anterior del listado de inmuebles, en el que una variable más podría ser la fecha de puesta en venta. Normalmente, las bases de datos, en numerosos y diferentes ámbitos, añaden información temporal, ya que suele ser una variable muy útil para organizar los datos.

Anotaciones

Como ya se explicó en el capítulo anterior, los algoritmos de aprendizaje supervisado requieren de datos etiquetados. Es muy habitual que las anotaciones o etiquetas de los datos, se almacenen como una variable más, en una columna extra de la tabla de datos.

Si las anotaciones son complejas, y no se pueden almacenar como una simple variable, se pueden usar archivos independientes para almacenarlas, siempre que se tenga de algún modo la trazabilidad entre los datos y sus anotaciones. Por ejemplo, en visión por computador, donde los datos de entrada son imágenes, es muy habitual, tener archivos conocidos como *archivos de ground truth*, donde se guardan las etiquetas correspondientes a cada imagen, pudiendo ser esas etiquetas, por ejemplo, la localización de cada objeto en la imagen y su clase.

Fuentes y tipos de archivos

El origen o fuente de los datos (por ejemplo, de audio, visuales, inventarios, encuestas, hojas de características, etc.) determina en gran medida la forma de almacenarlos y manejarlos. Sin embargo, para un caso genérico, en el que necesitemos guardar grandes cantidades de datos multivariable, disponemos de multitud de formatos o tipos de archivos. A continuación, se describen algunos de ellos.

En primer lugar, una *base de datos* es un conjunto de datos que pertenecen a un mismo contexto. Las bases de datos se encargan no solo de almacenar datos, sino también de conectarlos entre sí. Se emplean para administrar de forma electrónica grandes cantidades de información. Las bases de datos pueden ser relacionales, como las de tipo *SQL* (Structured Query Language). Esto significa que almacenan datos *estructurados*, en forma de tabla, organizándose en registros y campos. Cada fila es interpretada como un registro y cada columna contiene los campos de cada registro. Sin embargo, también existen bases de datos no relacionales, como las de tipo *JSON* (acrónimo de JavaScript Object Notation), para gestionar datos *no-estructurados* o *semi-estructurados*. El formato JSON permite representar los datos de forma jerárquica (en forma de árbol).

Por otro lado, los lenguajes de marcado, como *XML* (Extensible Markup Language) y *HTML* (HyperText Markup Language), son lenguajes empleados en el desarrollo web, para transporta datos entre aplicaciones y servidores. Normalmente XML se utiliza para almacenar datos estructurados como documentos, facturas, catálogos, libros, etc., y datos en aplicaciones web, como formularios. HTML, además, se ocupa de mostrar datos a los visitantes de los sitios web. Cuando se necesita alojar datos en un sitio web, estos pueden ser integrados en la propia página a través de su código HTML.

Por su parte, el *HDF* (Hierarchical Data Format), es un conjunto de formatos, como el HDF5, diseñados para almacenar y organizar grandes cantidades de datos. Emplea una estructura jerárquica con dos tipos de objetos, bases de datos y grupos. Estos últimos son contenedores de bases de datos u otros grupos.

Además, los datos provenientes de ficheros de texto y hojas de cálculo, con formato tabular, se pueden almacenar en archivos de tipo *CSV* (Comma Separated Values) y *XLSX* (estándar de Excel desde la versión de Microsoft Office 2003). Se considera que un archivo CSV es cualquier archivo de texto en el que los caracteres están separados por comas, formando una tabla con filas y columnas. Los archivos XLSX, además de tablas, tienen soporte para almacenar texto formateado, imágenes y gráficos.

Almacen de los datos

Dependiendo de la cantidad de datos que necesitemos, de las características de los mismos, y de la capacidad de almacenaje de nuestros equipos, habrá opciones de almacenaje que se adapten mejor que otras a nuestro caso. Cuando la cantidad de datos es relativamente pequeña o disponemos de mucho espacio en la memoria de nuestro equipo, podemos emplear un almacenaje *local* de los datos. Si ese no es el caso, podemos optar por un almacenaje en la *nube* (almacenaje en cloud) gracias a plataformas que ofrecen tales servicios (como AWS o AZURE). Una solución que se está extendiendo entre las empresas con departamentos de I+D+i y en los grupos y centros de investigación, es el uso de sus propios clusters. Un *cluster* es un conjunto de equipos (CPUs, GPUs, etc.) que funcionan como un servidor para un cierto número de usuarios.

REFERENCIAS

https://docs.python.org/3/tutorial/index.html

https://www.anaconda.com/products/individual

https://docs.jupyter.org/en/latest/

https://numpy.org/doc/stable/

https://pandas.pydata.org/docs

https://matplotlib.org/stable/index.html

Machine learning con Python y Scikit-learn by Joaquín Amat Rodrigo, available under a Attribution 4.0 International (CC BY 4.0) at https://www.cienciadedatos.net/documentos/py06_machine_learning_python_scikitlearn.html

https://www.drivendata.org/