Autor: María José
Gómez Silva

Machine Learning con Python. Semana 1.

CAPÍTULO 1. INTRODUCCIÓN AL MACHINE LEARNING.

1.8 Programación de un Proyecto de Machine Learning con Python

PROGRAMACIÓN DE UN PROYECTO DE MACHINE LEARNING.

Para implementar un proyecto de Machine Learning, será necesario el desarrollo de software (código) que ejecute cada una de sus etapas y módulos. Para desarrollar dicho software, o lo que es lo mismo para programar el código del proyecto, es necesario emplear un *entorno de desarrollo* en el que escribir nuestro código. Además, dicho código será escrito en un determinado *lenguaje de programación*, en el que ya existirán algunas tareas programadas y almacenadas en *librerías*. A continuación, se definen los conceptos enfatizados.

Lenguaje de programación.

Un lenguaje de programación es un lenguaje artificial (con reglas bien definidas) que le proporciona a una persona, en este caso el programador, la capacidad de escribir (o programar) una serie de instrucciones o secuencias de órdenes, para la realización de una determinada tarea. A todo este conjunto de órdenes escritas en un determinado lenguaje de programación se le denomina *programa* o *código*. El lenguaje de programación que se empleará en este módulo es *Python*, del que se darán más detalles en la siguiente sección.

Librerías.

En informática, una librería o biblioteca es un conjunto de archivos con el código necesario para implementar determinadas funcionalidades. Su objetivo es facilitar la programación, al proporcionar funcionalidades comunes, que ya han sido resueltas previamente por otros programadores. En este módulo, emplearemos librerías de Python, como *Matplotlib, Numpy, Pandas*, o *Scikit-learn*, que serán descritas con más detalles, más adelante.

Entorno de desarrollo.

Un entorno de desarrollo de software es una herramienta utilizada para escribir, editar, generar, probar y depurar el código de un programa. También proporcionan a los desarrolladores una interfaz de usuario para desarrollar y depurar el programa. La plataforma que emplearemos en este módulo para desarrollar nuestro código será *Jupyter Notebook*.

PYTHON. LENGUAJE DE PROGRAMACIÓN PARA MACHINE LEARNING.

Python es un lenguaje de programación cuyo uso se ha extendido ampliamente en campos como el de desarrollo de software y aplicaciones web, y recientemente en el área de la Inteligencia Artificial, para implementar programas de Data Science y Machine Learning (ML). Su extenso uso en el contexto científico es debido a que dispone de un gran número de librerías que incluyen herramientas para la preparación de los datos (carga, manipulación, limpieza y procesamiento), su representación gráfica y su posterior análisis, ya sea descriptivo o predictivo.

Python es un lenguaje accesible para un amplio público, no necesariamente con un perfil estrictamente tecnológico. Python es eficiente y fácil de aprender, y se puede ejecutar en muchas plataformas diferentes. La documentación de Python está disponible en la siguiente dirección: https://docs.python.org/3/tutorial/index.html.

Python es un lenguaje interpretado. Esto significa que el código fuente se ejecuta directamente, gracias a un programa llamado interprete que lee cada instrucción en tiempo real y la ejecuta.

Para poder implementar programas con Python, es necesario tener accesible un intérprete de Python y las librerías necesarias. Algunas de las librerías más empleadas en el desarrollo de proyectos de Machine Learning son Numpy, Pandas, Matplotlib y Scikit-learn, que se describen en la siguiente sección.

Anaconda proporciona una distribución libre (gratuita) de todos estos paquetes (interprete, librerías, editor de código) y puede descargarse de su sitio web https://www.anaconda.com/products/individual. Para instalarla, sólo hay que descargar desde su página web la versión de Anaconda adecuada para el sistema operativo que se tenga y seguir los pasos indicados. Una vez completada la instalación, verifíquela abriendo Anaconda Navigator, un programa que se incluye con Anaconda y que deberá aparecer entre las aplicaciones instaladas en su PC.

LIBRERÍAS DE PYTHON PARA MACHINE LEARNING

Existen numerosas librerías de Python que son ampliamente empleadas para desarrollar proyectos de Machine Learning, y que cubren distintas funcionalidades. Hay librerías para *cálculo numérico y análisis de datos* (Numpy, SciPy, Pandas, Numba), librerías de *visualización* (Matplotlib, Seaborn, Boheh), y librerías para entrenar y ejecutar algoritmos de *Machine Learning* (scikit-learn) y de *Deep Learning* (TensorFlow, Keras, PyTorch), entre otras. A continuación, se describen algunas de ellas.

Numpy

NumPy es una librería de cálculo numérico, que proporciona una estructura de datos universal que posibilita el análisis y el intercambio de datos entre distintos algoritmos. Las estructuras de datos que implementa, llamadas *arrays*, son vectores multidimensionales y matrices con gran capacidad (véase la Figura 1). Además, esta librería permite realizar operaciones matemáticas de alto nivel, sobre las estructuras de datos mencionadas. La documentación de Numpy está disponible en la siguiente dirección: https://numpy.org/doc/stable/.

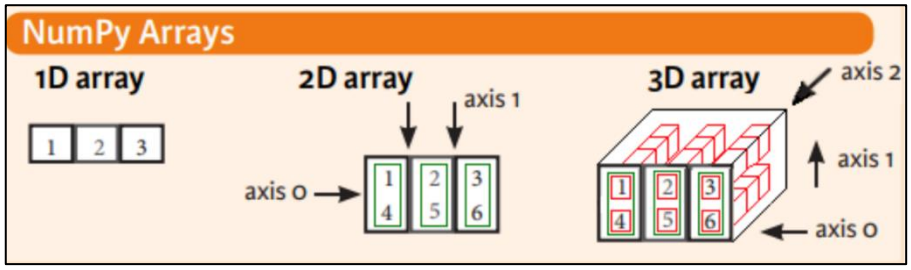

Figura 1. Estruturas de arrays de Numpy.

Pandas

Pandas es una de las librerías de python más útiles en Data Science, por el tratamiento rápido, sencillo y flexible que hace los datos para su análisis. Pandas maneja dos estructuras de datos principales, las *Series* para datos en una dimensión y los *DataFrames* para datos en dos dimensiones. Estas estructuras son muy usadas en multitud de áreas, tales como estadística, finanzas, ciencias sociales y muchas ramas de la ingeniería. Pandas está construido sobre los fundamentos de Numpy, adaptando las operaciones sobre arrays a las nuevas estructuras. Resulta especialmente útil para trabajar con datos heterogéneos representados de forma tabular, como los de la Figura 2. La documentación de Pandas está disponible en la siguiente dirección: https://pandas.pydata.org/docs/.

	name	city	phone-number	date
0	James Bass	Fife Lake	340-848-7354	2010-12-11
1	Cody Werner	Topanga	326-520-2048	1996-12-22
2	Joshua West	Richwoods	265-159-8349	2012-03-12
3	Kenneth Hanson	Northfield Woods	184-496-6411	1979-07-12
4	Michelle Brown	Lake Beulah	554-703-6417	1986-06-04

Figura 2. Ejemplo de DataFrame de Pandas

Matplotlib

Matplotlib es una librería gráfica de Python, que permite generar gráficos de una gran variedad de tipos: series temporales, histogramas, espectros de potencia, diagramas de barras, diagramas de errores, etc. La Figura 3 muestra algunos ejemplos. Los gráficos generados por esta librería ofrecen la calidad necesaria para publicarlos tanto en papel como digitalmente. La documentación de Matplotlib está disponible en la siguiente dirección: https://matplotlib.org/stable/index.html.

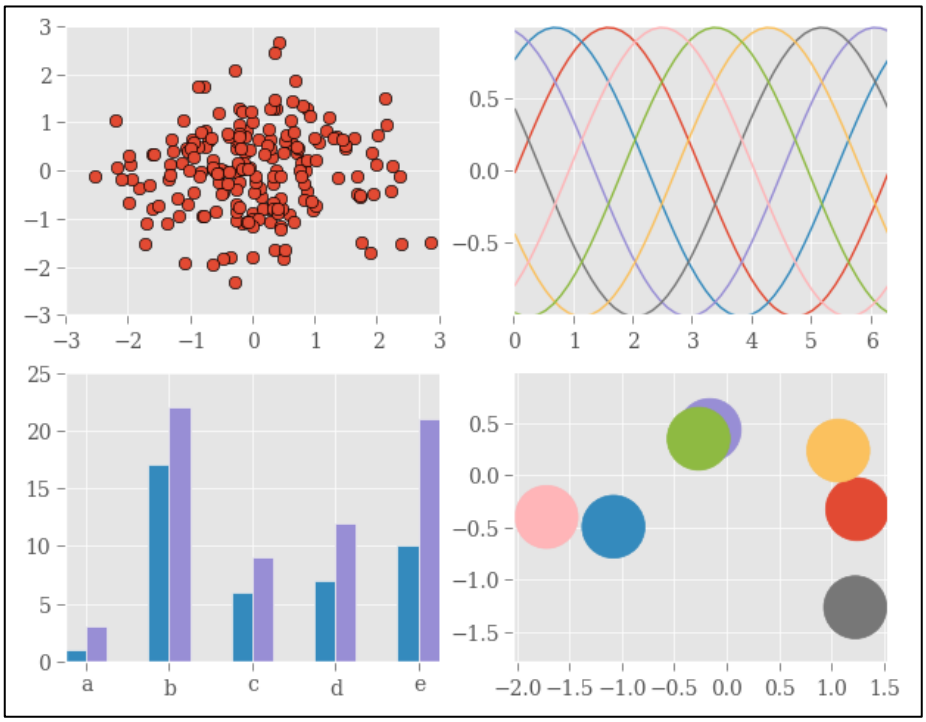

Figura 3. Ejemplos de gráficos obtenidos con Matplotlib

Scikit-learn

Scikit-learn es una librería para desarrollar métodos de Machine Learning y Análisis de Datos. Está basada en otras librerías como NumPy, SciPy y Matplotlib. Una de las mayores cualidades de scikit-learn es su facilidad de uso, gracias a una interfaz simple y muy consistente. Además, permite implementar multitud de técnicas de Machine Learning, tanto con aprendizaje supervisado, como no supervisado. Algunas de las tareas que permite implementar son: regresión (lineal, polinómica y logística), clasificación (máquinas de vectores de soporte, árboles de decisión, bosques aleatorios, clasificadores bayesianos), agrupamiento (clustering), reducción de dimensionalidad, y detección de anomalías, como muestra la Figura 4. La documentación de scikit-learn está disponible en la siguiente dirección: https://scikit-learn.org/stable/.

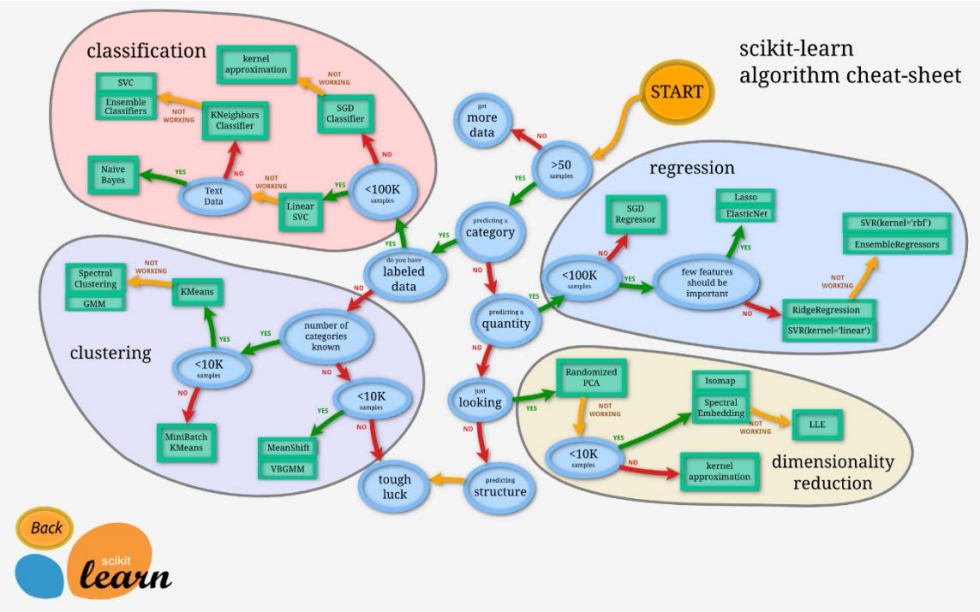

Figura 4. Esquema de ruta de los algoritmos disponibles en scikit-learn

Todas las librerías mencionadas se emplearán en las actividades de este módulo.

JUPYTER NOTEBOOK

Jupyter Notebook es una aplicación web que permite crear documentos (llamados notebooks de Jupyter) que ofrecen una herramienta de ejecución interactiva. Los notebooks contienen código vivo, texto, fórmulas, figuras y medios audiovisuales. Estos documentos se visualizan con un navegador (Explorer, Firefox, Chrome, …) y permiten la ejecución de código escrito en el lenguaje de programación Python, ya que permiten enviar órdenes directamente al intérprete de Python y obtener una respuesta inmediata de cada una de ellas. La documentación de Jupyter está disponible en la siguiente dirección: https://docs.jupyter.org/en/latest/.

Puesta en marcha de Jupyter Notebook.

Una vez realizada la instalación de Anaconda, la ejecución de Jupyter Notebook se puede realizar a través de la aplicación Anaconda Navigator, o directamente, buscando Jupyter Notebook entre las aplicaciones instaladas en nuestro sistema operativo.

La ejecución de Jupiter Notebook abrirá una ventana nueva en el navegador de internet, como se observa en la Figura 5. Esta interfaz web actúa como un explorador de archivos, que inicialmente muestra el contenido de la carpeta en que se abre Jupyter. Jupyter se abre en la carpeta *home* del usuario del equipo. Es recomendable crear una carpeta nueva donde guardar las actividades desarrolladas en este módulo.

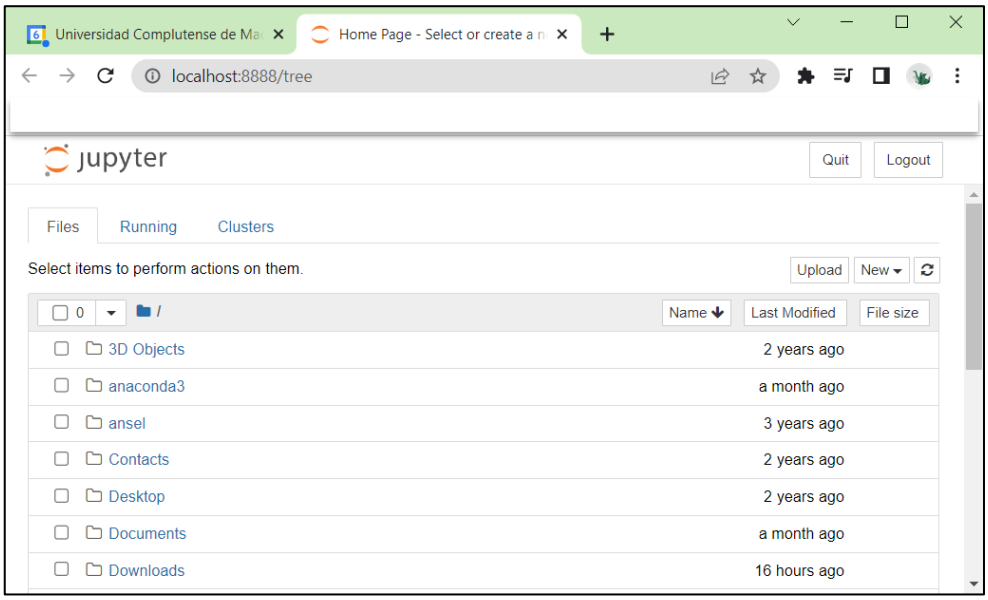

Figura 5. Manejo de los archivos locales a través del entorno de Jupyter