

III. Python for Data Science

Introduction to Data Science

May 18, 2020



Overview I

1. Conceptos Básicos
2. Librerías para Análisis de Datos
3. Clouds
4. Visualización de Datos



1. Conceptos básicos



1. Conceptos básicos

1.1 ¿Qué es Python?



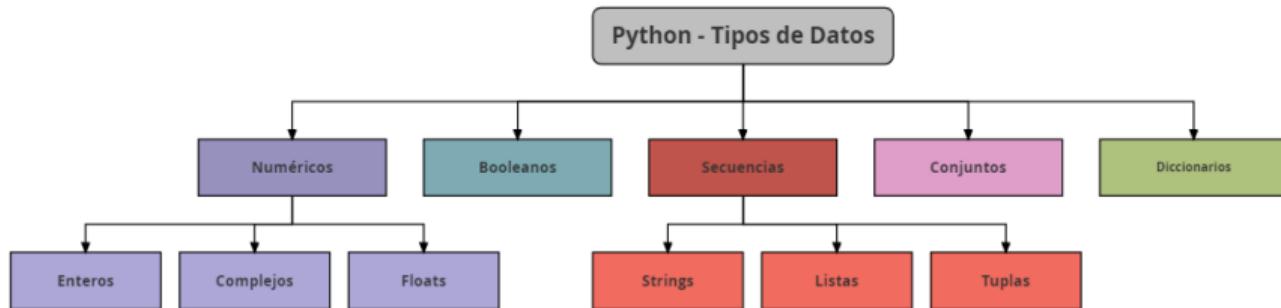
Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. Se trata de un lenguaje de programación multiparadigma, es decir permite varios estilos: programación orientada a objetos, programación imperativa y programación funcional.



1. Conceptos básicos

1.2 Tipos de Datos

Cuando se crea una *variable* en Python, se le debe asignar un tipo de dato. Los tipos de datos existentes en Python quedan resumidos en el siguiente diagrama:

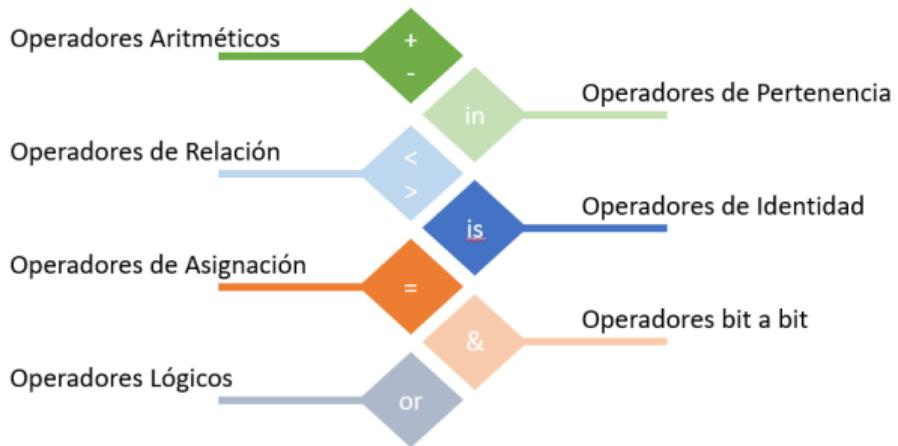




1. Conceptos básicos

1.3 Tipos de Operadores

Un operador es básicamente un simbolo que permite realizar una operación a 1 o más operandos. Existen 7 tipos de operadores:





1. Conceptos básicos

1.4 Sentencias Condicionales

Para escribir programas útiles, casi siempre necesitamos la capacidad de comprobar ciertas condiciones y cambiar el comportamiento del programa como corresponda. Las sentencias condicionales nos dan esta capacidad:

```
x = 7
if x < 10:
    print("El número es menor que 10")
elif x > 10:
    print("El número es mayor que 10")
else:
    print("El número es 10")
```

Si ejecutaramos el código de la izquierda, el resultado sería que en pantalla nos arrojaría "El número es menor que 10".



1. Conceptos básicos

1.5 Ciclos

Los ciclos dentro de un programa nos permiten repetir una sentencia siempre y cuando se cumpla una condición, o nos encontremos dentro de algún rango. Existen los ciclos while y for:

```
x = 0
while x < 10:
    print(x)
    x = x + 1
```

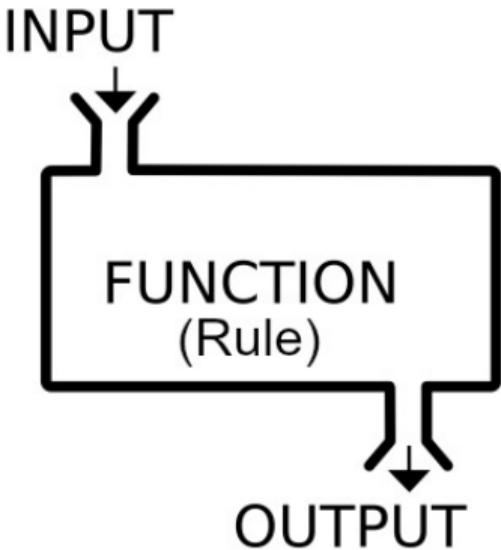
```
L = [0,1,2,3,4,5,6,7,8,9]
for i in L:
    print(i)
```

Si ejecutamos los dos códigos anteriores, ambos imprimirán en pantalla los números desde el 0 al 9.



1. Conceptos básicos

1.6 Funciones



Una función es un bloque de código con un nombre asociado, que recibe cero o más argumentos como entrada: La función sigue una secuencia de sentencias, que ejecutan una operación deseada , y en la mayoría de los casos entregando un valor; además este bloque puede ser llamado cuando se necesite invocando la función.

Python dispone de una serie de funciones integradas al lenguaje, y también permite crear funciones definidas por el usuario para ser usadas en su propios programas.



2. Librerías para Análisis de Datos



2. Librerías para Análisis de Datos



Procesamiento

- NumPy
- Scipy
- Pandas



Modelamiento

- ScikitLearn
- Keras
- Tensorflow



Minería

- Scrapy
- Beautiful Soup
- Selenium

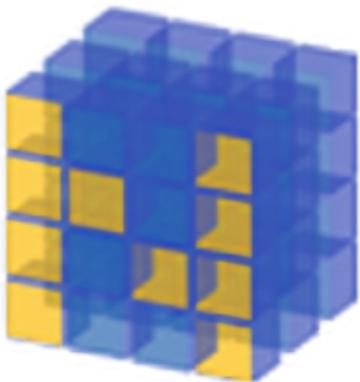


Visualización

- Matplotlib
- Plotly
- Seaborn

2. Librerías para Análisis de Datos

Procesamiento: Numpy



NumPy (Numerical Python) es una herramienta para la computación científica y para realizar operaciones matriciales tanto básicas como avanzadas.

La biblioteca ofrece muchas funciones útiles que realizan operaciones en matrices; ayuda a procesar matrices que almacenan valores del mismo tipo de datos, ademas facilita la realización de operaciones matemáticas en matrices y su vectorización. De hecho, la vectorización de operaciones matemáticas en el tipo de matriz NumPy aumenta el rendimiento y acelera el tiempo de ejecución.



2. Librerías para Análisis de Datos

Procesamiento: SciPy



SciPy (Scientific Python) incluye módulos para álgebra lineal, integración, optimización y estadísticas. Su funcionalidad principal se basó en NumPy, por lo que sus matrices hacen uso de esta biblioteca. SciPy funciona muy bien para todo tipo de proyectos de programación científica (ciencias, matemáticas e ingeniería). Ofrece rutinas numéricas eficientes como la optimización numérica, la integración y otras en submódulos. La extensa documentación hace que trabajar con esta biblioteca sea realmente fácil.



2. Librerías para Análisis de Datos

Procesamiento: Pandas



Pandas es una biblioteca creada para ayudar a los desarrolladores a trabajar con datos etiquetados y relacionales de forma intuitiva. Se basa en dos estructuras de datos principales: "Series" (unidimensional, como una lista de elementos) y "Dataframes" (bidimensional, como una tabla con varias columnas). Pandas permite convertir estructuras de datos en objetos DataFrame, manejar datos faltantes y agregar o eliminar columnas de un DataFrame, ingresar datos faltantes y trazar datos con histograma. Es imprescindible para exploración, manipulación y visualización de datos.



2. Librerías para Análisis de Datos

Modelamiento: ScikitLearn



Scikitlearn es un grupo de paquetes en SciPy Stack que se crearon para funcionalidades específicas, como el procesamiento de imágenes. Scikit-learn utiliza las operaciones matemáticas de SciPy para exponer una interfaz concisa a los algoritmos de aprendizaje automático más comunes.

Se usa además para manejar tareas de aprendizaje automático y minería de datos, como agrupamiento, regresión, selección de modelos, reducción de dimensionalidad y clasificación. documentación de calidad y ofrece un alto rendimiento.



2. Librerías para Análisis de Datos

Modelamiento: TensorFlow

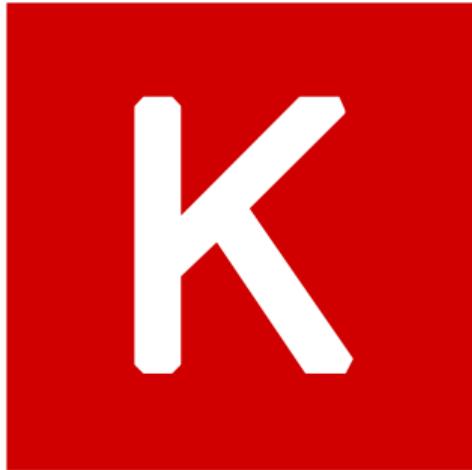


TensorFlow es una librería de Python popular para el aprendizaje automático y el aprendizaje profundo, que se desarrolló en Google Brain. Es la mejor herramienta para tareas como identificación de objetos y reconocimiento de voz entre otras. Ayuda a trabajar con redes neuronales artificiales que necesitan manejar múltiples conjuntos de datos. TensorFlow se expande constantemente con sus nuevas versiones, que incluyen correcciones en posibles vulnerabilidades de seguridad o mejoras en la integración de TensorFlow y GPU.



2. Librerías para Análisis de Datos

Modelamiento: Keras



Keras es una gran biblioteca para construir redes neuronales y modelado. Es muy sencillo de usar y proporciona a los desarrolladores un buen grado de extensibilidad. La biblioteca aprovecha otros paquetes (Theano o TensorFlow) como backends. Además, Microsoft integró CNTK (Microsoft Cognitive Toolkit) para que sirviera como otro backend. Es una gran elección si desea experimentar rápidamente utilizando sistemas compactos, el enfoque minimalista del diseño realmente vale la pena.



2. Librerías para Análisis de Datos

Modelamiento: PyTorch



PyTorch es una librería que es perfecta para realizar tareas de aprendizaje profunda de manera sencilla. La herramienta permite realizar cálculos con tensores utilizando aceleración de GPU; también se usa para otras tareas, por ejemplo, para crear gráficos computacionales dinámicos y calcular gradientes automáticamente. PyTorch se basa en Torch, que es una biblioteca de aprendizaje profundo de código abierto implementada en C, con un contenedor en Lua.



2. Librerías para Análisis de Datos

Visualización: Matplotlib



Es la librería mas popular de visualización de datos en Python; ayuda a generar visualizaciones de datos como diagramas y gráficos bidimensionales (histogramas, diagramas de dispersión, gráficos de coordenadas no cartesianas). Matplotlib es una de esas bibliotecas de visualización que son realmente útiles en proyectos de ciencia de datos: proporciona una API orientada a objetos para incrustar trazados en aplicaciones.



2. Librerías para Análisis de Datos

Visualización: SeaBorn

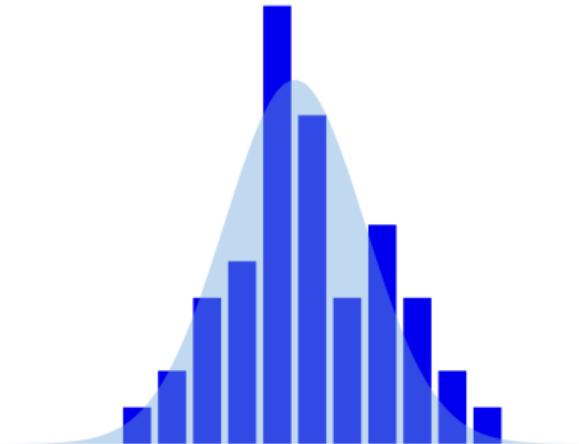


Seaborn se basa en Matplotlib y sirve como una herramienta útil de aprendizaje automático de Python para visualizar modelos estadísticos: mapas de calor y otros tipos de visualizaciones que resumen los datos y representan las distribuciones generales. Al usar esta biblioteca, puede beneficiarse de una amplia galería de visualizaciones (incluidas las complejas, como series de tiempo, diagramas de Venn y diagramas de violín).



2. Librerías para Análisis de Datos

Visualización: Plotly



Esta herramienta basada en desarrollo web para la visualización de datos, ofrece muchos gráficos útiles listos para usar, los cuales pueden ser encontrados en Plot.ly, el sitio web de la librería; esta funciona muy bien en aplicaciones web interactivas. Sus creadores están ocupados expandiendo la biblioteca con nuevos gráficos y características para soportar múltiples vistas vinculadas, animación e integración de diafonía.



3. Clouds



3. Clouds



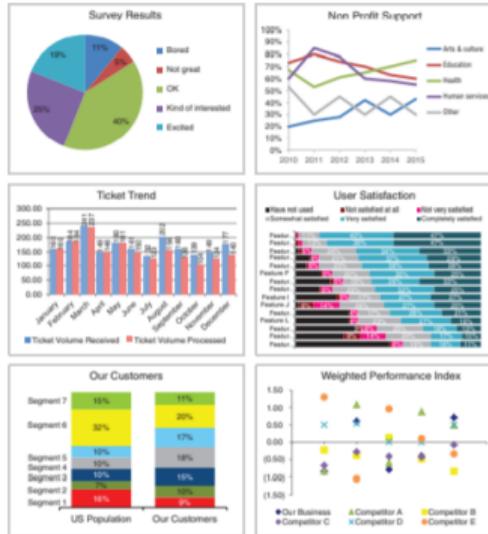


4. Visualización de Datos



4. Visualización de Datos

¿Qué hacer con los malos gráficos?



Uno de los problemas más grandes a la hora de visualizar la información es saber escoger la herramienta correcta para mostrarla; y no solo con respecto al tipo de gráfico, si no que también sus detalles, colores, y muchas otras variables que influyen para una buena exposición de los datos.



4. Visualización de Datos

4.1 La importancia del contexto



Antes de comenzar el camino de la visualización de datos, hay un par de preguntas que uno debe poder responder de manera concisa: ¿Quién es nuestra audiencia? ¿Qué necesitamos que sepan o hagan? Para comenzar una visualización efectiva es de suma importancia comprender el contexto, incluyendo la audiencia, el mecanismo de comunicación y el tono deseado. Si comprendemos de manera sólida del contexto y la situación que rodea a los datos, se reducen las iteraciones en el futuro y tendremos un manejo de lo que estemos mostrando.



4. Visualización de Datos

4.2 Escogiendo una visualización efectiva



¿CUÁL ES LA MEJOR MANERA DE MOSTRAR LOS DATOS QUE DESEA COMUNICAR?

Existen muchísimas visualizaciones distintas para el mismo tipo de dato, es por esto que debemos saber cual de estas será más efectiva a la hora de mostrar nuestra información.



4. Visualización de Datos

4.3 Eliminando el desorden

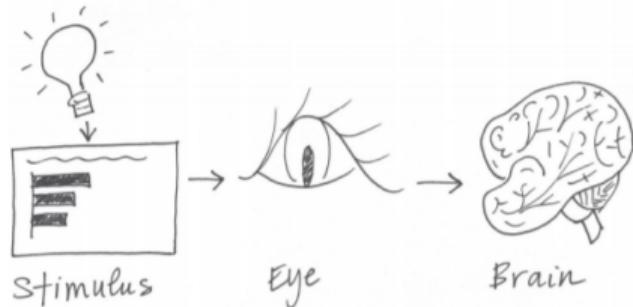
Cada elemento o gráfico que le mostramos al espectador toma una carga cognitiva por parte de este, por ello debemos ser exigentes a la hora de identificar que debemos dejar y que no a la hora de mostrar nuestros datos, y también donde deben ir estos en nuestra presentación. La ubicación de cada imagen o gráfico es importante para la atención del espectador.





4. Visualización de Datos

4.4 Saber enfocar la atención

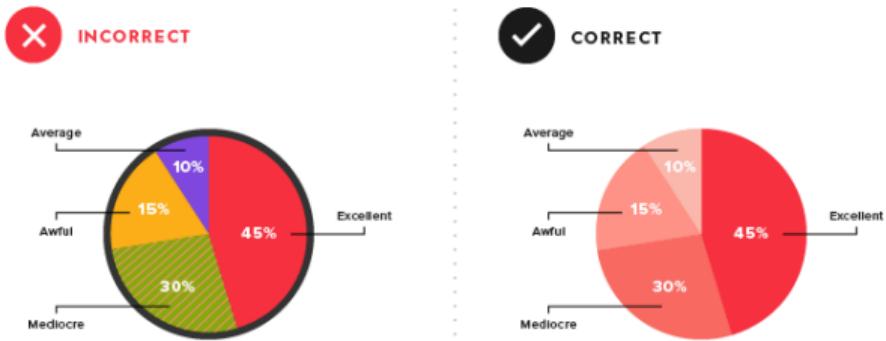


Hay gráficos que deben sobresalir más que otros, ya que debemos hacer que el objetivo general de nuestro proyecto sea el que el espectador entienda por sobre los otros. Entender como la vista y la memoria actuan hace que podamos utilizarlas para enfocar la atención a los detalles en medida que nos pongamos. Es importante además una jerarquía visual de componentes para que sea simple el flujo de la información y de cómo esto se debe procesar.



4. Visualización de Datos

4.5 ¿Cómo piensa un diseñador?



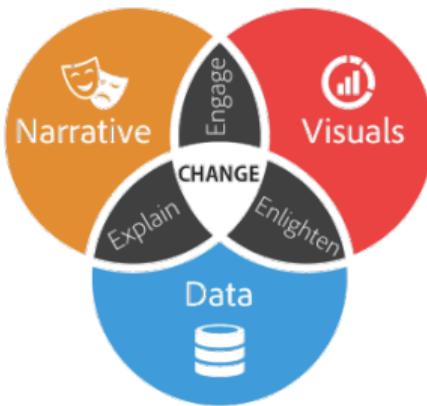
El hecho de tener un gráfico seleccionado y nuestra presentación de información esquematisada y ordenada, no necesariamente hace que la visualización este del todo prolíja. Se deben analizar detalles estéticos dentro de sus presentaciones, si los colores y enfasis en los gráficos son acordes a lo que se quiere mostrar, entre otros detalles.



4. Visualización de Datos

4.6 Contar una historia

Las historias resuenan y se quedan con nosotros de una manera que los datos por sí solos no pueden, por lo que uno debe tener estructurada su historia para mostrar la visualización con un comienzo, un medio y un final claro; esto es válido tanto para exposiciones científicas como para presentaciones de negocios.



III. Python for Data Science

Introduction to Data Science

May 18, 2020



DataOwl