Data fundamentals

02/10/2023

Vamos a usar la versión de python 3.7
CRISP-DM
Entender qué es lo que queremos hacer y partir de lo que tenemos.

Siempre hay dos paradigmas a la hora de trabajar
DEV vs PROD → hay una copia a la cual sólo tienen acceso los trabajadores de la empresa, es una copia de la real. Es un entorno controlado para por si la lías, no afecte a todo el mundo.

Mediante variables, guardamos los datos en python. Es como una estantería en la que hay cajas y en cada caja hay una variable. Si vas a buscar una variable que no existe, te lanza un error, entendiendo la estantería como la memoria. Como estamos trabajando en un lenguaje de alto nivel, la etiqueta que le ponemos a la estantería es una palabra.

Tipos de datos en python. Los tipos más sencillos con tipos primitivos y existen 3: los textos, números y los booleanos.

- Los números. Nos centraremos en los enteros y los flotantes. Los números son los números tanto positivos como negativos pero sin decimales. Los flotantes son positivos, negativos pero con decimales. Los decimales en python se ponen con un punto . no con una coma.

- Los textos. Cualquier tecla que tengas en el teclado, no sólo las letras. Cómo se diferencia si escribo *patata*. Para que python sepa que es texto, tiene que ir siempre entre comillas.'Patata', "Patata" comillas triples dobles, a ojos de python es lo mismo. Caracteres especiales también o incluso los números, si pongo un número entre comillas ya no es un número sino un texto.

- Los booleanos. Sólo hay dos y nos ayudan a expresar estados: True y False es sensible a las mayúsculas y minúsculas.

¿Cómo se crea una variable? → nombre_variable = tipo_de_texto
El nombre de la variable no puede comenzar por un número, no puede contener espacios y la tercera norma es que el nombre de la variable no puede ser una palabra reservada, una palabra que ya tiene un uso asignado. Todas las palabras reservadas en python están en inglés.

```
numero = 2
texto = 'Luis'
```

un dataset puede ser una variable, es un tipo de dato más complejo pero la idea es la misma.

Listas
La condición de python para considerar un dato en tipo lista los valores tienen que están separados entre comas y corchetes.

lista = [‘Luis’, ’Sastre’]

Se empieza a contar a partir de 0. El primer elemento está en la posición 0, y el segundo elemento, siempre va a estar el tamaño de la lista 2-1. En el siguiente ejemplo se ve más claro.

lista = [‘Luis’,’Pepe’,’Julian’,’Alex’]

Funciones
Son algo más complejo. Son como las recetas. No podemos comenzar la receta hasta que no hemos comprado todo los ingredientes que necesitamos ya que si nos quedamos cortos, no va a salir y si nos pasamos, saldrá otra cosa. Seguiremos una serie de pasos en el mismo orden y obtendremos un resultado.
Una serie de cálculos que se dan siempre en el mismo orden.
Cuando veamos paréntesis por ahí, eso significa que es una función. Una variable no lleva paréntesis.

suma()

Existen las funciones nativas que son funciones que ya vienen creadas de 0, es decir, alguien las ha creado y las puedes usar. No todas las que necesitaremos están creadas por defecto, por lo que las librerías son un conjunto de utilidades, funciones, que alguien ha creado,la comunidad va haciendo distintas mejoras y tú decides si instalarlas en tu pc. Podemos descargarnos la librería calculadora por ejemplo. Cuando veamos un Pip, nos sirve para instalar una estancia, trabajamos sobre él.

Librerías que son útiles, una de las más importantes es pandas, nos permite trabajar con datasets, es de los más importantes. No viene por defecto, es como el excel de python, nos descargamos la librería y así podemos hacer tratamiento de datos masiva. Sólo depende de la potencia de tu ordenador. Se puede hacer lo que en excel pero más rápido y mejor.
librerías de gráficas → plot

webs que recomiendan
por ejemplo, cuando busquemos algo en Google

how to create a float in python w3schools

05/10/2023

diccionario
Pueden estar indexados no sólo por un número sino por un texto también, esa es la diferencia con las listas.

lista = [‘Hola’,’soy’,’Luis’]
diccionario = {0:’Hola’,1:’soy’,2:’Luis’}

reemplazar valores nulos
np.nan
np.pi
np.

SQL Alchemy
df.to_sql()
read_sql() → le pasamos un string que contenga nuestra query de sql y descarga todos los datos en una sola variable.