

Herramientas y ecosistema de ciencia de datos

En este cuaderno se resumen la ciencia de datos y el ecosistema.

Algunos de los lenguajes populares que utilizan los científicos de datos son:

1. **Python:** Python es el lenguaje más popular utilizado por los científicos de datos. Tiene una amplia gama de bibliotecas y herramientas que facilitan el trabajo con datos, incluidas NumPy, pandas y scikit-learn.
2. **R:** R es un lenguaje popular para la informática estadística y es ampliamente utilizado por los científicos de datos. Tiene una amplia gama de bibliotecas y herramientas para visualización de datos, aprendizaje automático y modelado estadístico.
3. **SQL:** SQL (lenguaje de consulta estructurado) es un lenguaje utilizado para administrar y analizar bases de datos relacionales. Los científicos de datos utilizan SQL para extraer y manipular datos de bases de datos.
4. **Julia:** Julia es un nuevo lenguaje que está ganando popularidad entre los científicos de datos. Es conocido por su alto rendimiento y se utiliza para aprendizaje automático, análisis de datos y visualización.
5. **MATLAB:** MATLAB es un lenguaje utilizado para la computación numérica y es ampliamente utilizado por los científicos de datos. Tiene una amplia gama de bibliotecas y herramientas para análisis de datos, aprendizaje automático y visualización.
6. **Scala:** Scala es un lenguaje utilizado para construir sistemas escalables y distribuidos. Es ampliamente utilizado por los científicos de datos para crear aplicaciones de big data utilizando Apache Spark.
7. **Java:** Java es un lenguaje popular utilizado por los científicos de datos para crear aplicaciones de datos a gran escala. Se utiliza ampliamente para crear canalizaciones de datos, modelos de aprendizaje automático y herramientas de visualización de datos.
8. **C++:** C++ es un lenguaje utilizado para crear aplicaciones de alto rendimiento. Los científicos de datos lo utilizan ampliamente para crear modelos de aprendizaje automático, canalizaciones de datos y herramientas de visualización de datos.

Algunas de las bibliotecas comúnmente utilizadas por Data Scientist incluyen:

Proporciona una interfaz de alto nivel para dibujar gráficos estadísticos atractivos e informativos.

1. **NumPy** : la biblioteca NumPy se utiliza para cálculos numéricos. Proporciona soporte para matrices y arreglos multidimensionales grandes, junto con una gran colección de funciones matemáticas de alto nivel para operar en estos arreglos.
2. **Pandas** : la biblioteca Pandas se utiliza para la manipulación y análisis de datos. Proporciona estructuras de datos como Series (matriz etiquetada unidimensional) y

DataFrame (estructura de datos etiquetada bidimensional con columnas de tipos potencialmente diferentes).

3. **Matplotlib** : la biblioteca Matplotlib se utiliza para crear visualizaciones estáticas, animadas e interactivas en Python. Proporciona un conjunto completo de herramientas para crear gráficos 2D y 3D de alta calidad.
4. **Scikit-learn** : la biblioteca Scikit-learn se utiliza para tareas de aprendizaje automático como clasificación, regresión, agrupación y selección de modelos. Proporciona una amplia gama de algoritmos para estas tareas, junto con herramientas para la evaluación y selección de modelos.
5. **TensorFlow** : la biblioteca TensorFlow se utiliza para construir y entrenar redes neuronales artificiales, particularmente redes neuronales profundas. Proporciona un marco flexible para definir, ejecutar y ajustar modelos de aprendizaje automático.
6. **Keras** : la biblioteca Keras se utiliza para crear y entrenar modelos de aprendizaje profundo. Proporciona una interfaz de alto nivel para construir redes neuronales y puede ejecutarse sobre TensorFlow, CNTK o Theano.
7. **OpenCV** : la biblioteca OpenCV se utiliza para tareas de visión por computadora, como procesamiento de imágenes y videos, detección de características, detección de objetos y seguimiento.
8. **Seaborn** : la biblioteca Seaborn se utiliza para la visualización de datos estadísticos. El
9. **Statsmodels** : la biblioteca Statsmodels se utiliza para modelado y análisis estadístico. Proporciona una amplia gama de técnicas estadísticas, incluidas pruebas de hipótesis estadísticas, intervalos de confianza y análisis de regresión.
10. **Scipy** : la biblioteca Scipy se utiliza para aplicaciones científicas y de ingeniería. Proporciona funciones para aplicaciones científicas y de ingeniería, incluida la optimización, el procesamiento de señales y las estadísticas.

Herramientas de ciencia de datos Las tres filas siguientes de la tabla deben indicar tres herramientas de código abierto del entorno de desarrollo utilizadas en la ciencia de datos.

Herramienta Descripción

| Cuaderno Jupyter | Jupyter Notebook es un entorno informático interactivo basado en web que le permite crear y compartir documentos que contienen código en vivo, ecuaciones, visualizaciones y texto narrativo. Los usos incluyen limpieza y transformación de datos, simulación numérica, modelado estadístico, visualización de datos, aprendizaje automático y más. | | RStudio | RStudio es un popular entorno de desarrollo integrado (IDE) para R. Proporciona un conjunto completo de herramientas para análisis, visualización y modelado de datos. RStudio incluye funciones como resaltado de sintaxis, finalización de código y gestión de proyectos. También admite control de versiones, depuración y visualización. | | PyCharm | PyCharm es un IDE

popular para Python. Proporciona un conjunto completo de herramientas para el análisis, visualización y modelado de datos. PyCharm incluye funciones como resaltado de sintaxis, finalización de código y gestión de proyectos. También admite control de versiones, depuración y visualización. PyCharm es particularmente útil para tareas de ciencia de datos que involucran aprendizaje automático, aprendizaje profundo y visualización de datos. |

A continuación se muestran algunos ejemplos de evaluación de expresiones aritméticas en Python.

Ejemplos aritméticos

```
a = 5 b = 2
+a 5 -b -2 a + b 7 a - b 3 a * b 10 a / b 2.5 a %
b 1 a // b 2 a**b 25
```

```
En [ ]: ( 3 * 4 ) + 5
```

```
"Esta es una expresión aritmética simple"
```

Afuera[... 17

```
En [ ]:
```

```
Celda En[4], línea 2
```

```
minutos =
```

```
^
```

```
Error de sintaxis : sintaxis invalida
```

```
En [ ]: def minutos_a_horas(minutos):
```

```
    horas = minutos / 60
```

```
    return horas
```

```
# Convertir 200 minutos a horas
```

```
minutos = 200
```

```
horas_convertidas = minutos_a_horas(minutos)
```

```
print(f"{minutos} minutos son equivalentes a {horas_convertidas} horas.")
```

200 minutos son equivalentes a 3.3333333333333335 horas.

Objectives:

- List popular languages for Data Science
- Discuss the importance of Data Science
- Understand basic Python syntax and functions
- Explore data manipulation with pandas
- Create visualizations with matplotlib and seaborn

Autor

Carlos Palacios Betancourt