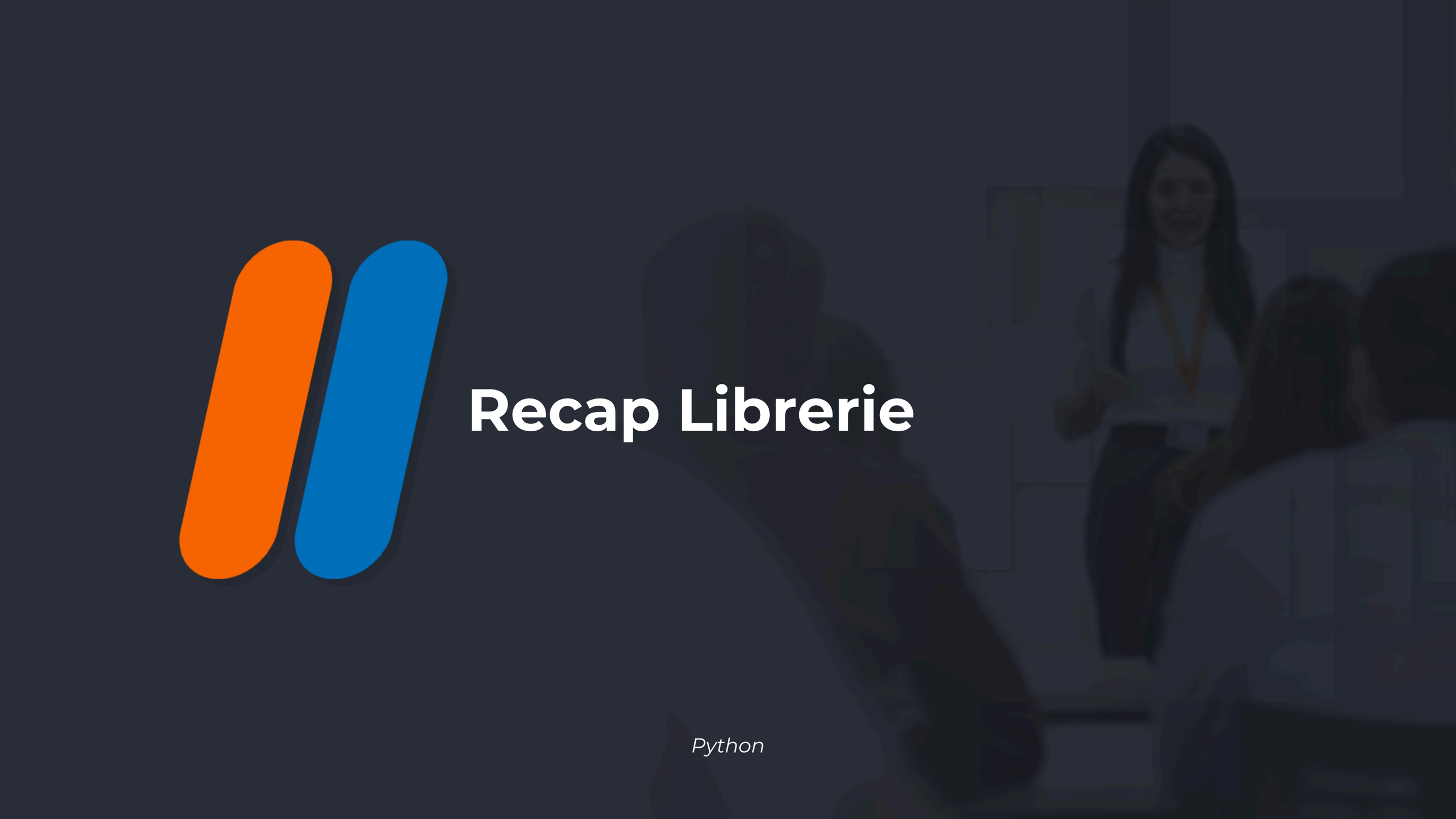


Recap Librerie

Python

Nel panorama della programmazione Python, esistono alcune librerie fondamentali che costituiscono la base per l'analisi dei dati, la manipolazione di dataset complessi e la visualizzazione di informazioni.

Queste librerie sono progettate per semplificare operazioni matematiche, statistiche e grafiche, offrendo strumenti ad alte prestazioni che rendono Python uno degli ambienti preferiti per il Data Science e il Machine Learning.

Tra le più utilizzate troviamo NumPy, Pandas, Matplotlib e Seaborn, ciascuna con un ruolo ben definito ma fortemente complementare alle altre.

Spiegazione delle librerie

- **NumPy: è la libreria di riferimento per il calcolo numerico. Fornisce strutture dati avanzate come gli array multidimensionali e funzioni ottimizzate per operazioni matematiche complesse, consentendo di lavorare in maniera molto più efficiente rispetto alle liste native di Python.**

Spiegazione delle librerie

- **NumPy:**

```
1. import numpy as np
2.
3. # Creazione di un array NumPy e calcolo di media e deviazione standard
4. dati = np.array([10, 20, 30, 40, 50])
5. media = np.mean(dati)
6. deviazione = np.std(dati)
7.
8. print("Media:", media)
9. print("Deviazione standard:", deviazione)
```

Spiegazione:
In questo esempio NumPy gestisce array numerici in modo efficiente. Calcoliamo direttamente la media e la deviazione standard, operazioni che con liste Python standard richiederebbero cicli più lunghi.

Spiegazione delle librerie

- **Pandas: si concentra sulla manipolazione e analisi di dataset. Introduce strutture come Series e DataFrame, che permettono di gestire tabelle di dati in modo flessibile, con strumenti integrati per filtri, aggregazioni e trasformazioni.**

Spiegazione delle librerie

- **Pandas:**

```
import pandas as pd

# Creazione di un DataFrame e filtro dei dati
dati = {
    "Nome": ["Anna", "Luca", "Marco", "Sara"],
    "Età": [23, 30, 18, 27],
    "Città": ["Roma", "Milano", "Roma", "Torino"]
}
df = pd.DataFrame(dati)

# Filtriamo solo chi vive a Roma
roma = df[df["Città"] == "Roma"]
print(roma)
```

Spiegazione:
Pandas consente di rappresentare i dati in forma tabellare con DataFrame. Qui creiamo una tabella con nomi, età e città e poi filtriamo solo i residenti a Roma in modo molto semplice.

Spiegazione delle librerie

- **Matplotlib: è la libreria più diffusa per la creazione di grafici statici. Consente di rappresentare i dati in diverse forme (linee, istogrammi, scatter plot, ecc.), personalizzando ogni dettaglio visivo del grafico.**

Spiegazione delle librerie

- **Matplotlib:**

```
import matplotlib.pyplot as plt

# Creazione di un grafico a linee
anni = [2018, 2019, 2020, 2021, 2022]
vendite = [100, 120, 90, 150, 180]

plt.plot(anni, vendite, marker="o", color="blue")
plt.title("Andamento vendite")
plt.xlabel("Anno")
plt.ylabel("Vendite")
plt.show()
```

Spiegazione:
Matplotlib permette di creare grafici personalizzabili.
Qui rappresentiamo l'andamento delle vendite in cinque anni con una linea, aggiungendo titolo ed etichette agli assi.

Spiegazione delle librerie

- **Seaborn: costruito sopra Matplotlib, è pensato per semplificare la creazione di grafici statisticamente significativi con un design elegante e leggibile. Fornisce funzioni pronte all’uso per analizzare distribuzioni, relazioni e pattern nei dati, riducendo la complessità del codice necessario.**

Spiegazione delle librerie

- **Seaborn:**

```
import seaborn as sns
import matplotlib.pyplot as plt

# Dataset di esempio integrato in Seaborn
tips = sns.load_dataset("tips")

# Grafico che mostra la distribuzione dei totali spesi
sns.histplot(tips["total_bill"], bins=20, kde=True, color="green")
plt.title("Distribuzione dei conti totali")
plt.show()

```

Spiegazione:
Seaborn semplifica le visualizzazioni statistiche.
In questo caso usiamo il dataset integrato tips e rappresentiamo la distribuzione dei conti totali (total_bill) con un istogramma e una curva di densità.

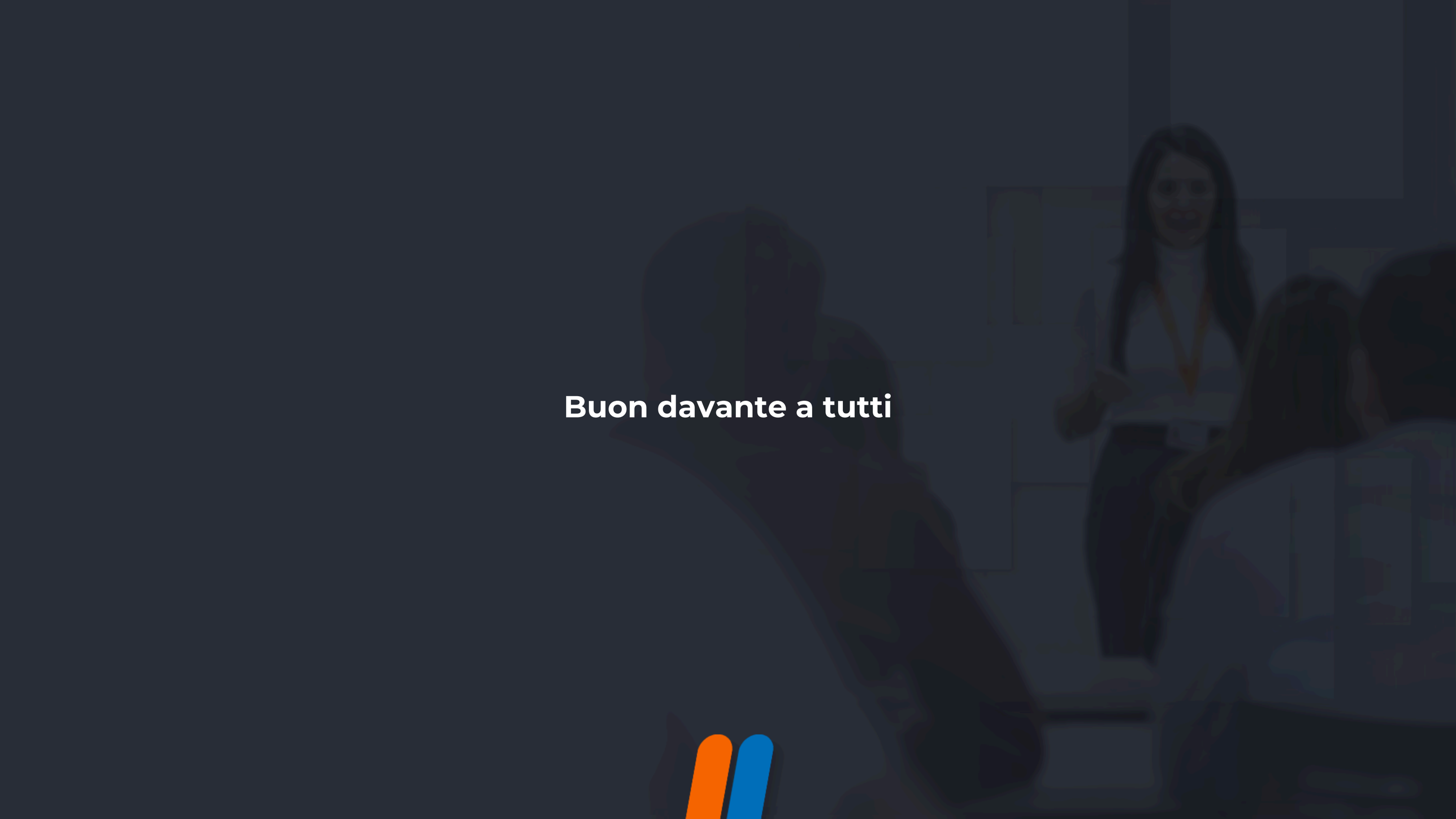
Libreria	Punto di forza principale	Casi d'uso tipici
NumPy	Calcolo numerico veloce con array multidimensionali	Algebra lineare, operazioni su matrici, funzioni matematiche
Pandas	Gestione di dataset tabellari (DataFrame)	Analisi di dati, pulizia dataset, trasformazioni e filtri
Matplotlib	Creazione di grafici personalizzati	Grafici statici (linee, barre, istogrammi, scatter plot)
Seaborn	Visualizzazioni statistiche eleganti e semplici	Heatmap, distribuzioni, correlazioni tra variabili

Dopo aver consolidato l’uso delle librerie basilari, in ambito Data Science e Machine Learning si incontrano strumenti più evoluti che ampliano le possibilità analitiche.

Scikit-learn è una libreria fondamentale per il machine learning classico: offre algoritmi già pronti per classificazione, regressione, clustering e riduzione dimensionale, oltre a strumenti per valutazione e validazione dei modelli.

TensorFlow e PyTorch, invece, rappresentano i pilastri per il Deep Learning, consentendo di costruire e addestrare reti neurali complesse in modo scalabile, anche su GPU.

Per la visualizzazione interattiva troviamo Plotly, che permette grafici dinamici utili in dashboard e applicazioni web, mentre per la gestione dei big data emergono librerie come Dask e PySpark, progettate per distribuire i calcoli su più core o cluster.


Buon davante a tutti