

Introduzione all'analisi in Python

Python è uno dei linguaggi di programmazione più utilizzati per l'analisi dei dati grazie alla sua semplicità e alla vasta gamma di librerie disponibili.

In questa introduzione, esploreremo alcuni degli strumenti principali e le basi delle keyword più comuni utilizzate nell'analisi dei dati con Python.



Strumenti per l'Analisi in Python

Jupyter Notebook:

- Ambiente interattivo ideale per l'analisi dei dati.
- Permette di scrivere codice Python, visualizzare grafici e documentare il processo di analisi in un unico luogo.



Librerie Principali:

- NumPy: Fornisce supporto per array multidimensionali e matrici, insieme a una vasta libreria di funzioni matematiche di alto livello.
- Pandas: Essenziale per la manipolazione e l'analisi dei dati, offre strutture dati come DataFrame per gestire e analizzare grandi insiemi di dati.
- Matplotlib: Utilizzata per la visualizzazione dei dati, permette di creare grafici 2D.
- Seaborn: Basata su Matplotlib, offre un'interfaccia di alto livello per la creazione di grafici statistici.

 Scikit-learn: Libreria per il machine learning che include una serie di strumenti per la modellazione e l'analisi dei dati.

Manipolazione dei Dati con Pandas

Pandas è una libreria open-source di Python che fornisce strutture dati flessibili e facili da usare, come Series e DataFrame, per la manipolazione e l'analisi dei dati. La sua potenza risiede nella capacità di gestire dati eterogenei in modo efficiente e performante.

Grazie alle sue funzioni di indicizzazione avanzate, Pandas permette di selezionare, filtrare, raggruppare e trasformare i dati in maniera intuitiva.



Le operazioni di lettura e scrittura da diverse fonti di dati, come CSV, Excel e database SQL, sono integrate nella libreria, rendendo Pandas uno strumento essenziale per i data scientist e gli analisti.

Manipolazione dei Dati con Pandas

```
1. import pandas as pd
3.df = pd.read_csv('file.csv')
1.print(df.head())
2.print(df.info())
3. print(df.describe())
                              # Selezione di una colonna
1.df['colonna']
2.df[['colonna1', 'colonna2']] # Selezione di più colonne
3.df[df['colonna'] > valore] # Filtraggio delle righe
```



1.df['nuova_colonna'] = df['colonna1'] + df['colonna2'] # Creazione di una nuova colonna 2.df.drop(['colonna'], axis=1, inplace=True) # Rimozione di una colonna 3.df.fillna(valore, inplace=True) # Sostituzione dei valori NaN Visualizzazione dei Dati con Matplotlib e Seaborn

Matplotlib è la libreria di riferimento per la visualizzazione dei dati in Python, nota per la sua capacità di creare grafici statici, animati e interattivi.

Seaborn, costruita su Matplotlib, fornisce un'interfaccia di alto livello per la creazione di grafici statistici più complessi e accattivanti.

La visualizzazione dei dati è cruciale per comprendere le relazioni e i pattern nascosti nei dataset, facilitando l'interpretazione dei risultati dell'analisi.



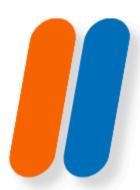
Matplotlib offre flessibilità e controllo dettagliato sui grafici, mentre Seaborn semplifica la creazione di visualizzazioni statisticamente significative con pochi comandi.

Visualizzazione dei Dati con Matplotlib e Seaborn

Matplotlib e Seaborn sono librerie potenti per la visualizzazione dei dati:

Grafici di Base con Matplotlib:

- 1. import matplotlib.pyplot as plt
- 2.
- 3. plt.figure(figsize=(10, 6))
- 4. plt.hist(df['colonna'], bins=30, alpha=0.7, label='Colonna')
- 5. plt.xlabel('Valori')
- 6. plt.ylabel('Frequenza')
- 7.plt.title('Istogramma')
- 8. plt.legend()
- 9. plt.show()



Visualizzazione dei Dati con Matplotlib e Seaborn

Matplotlib e Seaborn sono librerie potenti per la visualizzazione dei dati:

Visualizzazioni Avanzate con Seaborn:

```
1.import seaborn as sns
2.
3.sns.pairplot(df)
4.plt.show()
5.
6.sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
7.plt.show()
8.
```



Analisi Statistica con SciPy e StatsModels

SciPy è una libreria Python che fornisce funzioni e algoritmi per l'analisi matematica e scientifica avanzata, mentre StatsModels offre strumenti per la stima di modelli statistici e la realizzazione di test statistici.

Queste librerie permettono di effettuare analisi statistiche sofisticate, come test di ipotesi, analisi di regressione, e modellazione econometrica.

SciPy è noto per la sua capacità di gestire operazioni matematiche complesse, come l'ottimizzazione e l'algebra lineare, mentre StatsModels è apprezzato per la sua robustezza nella stima di modelli statistici e nella produzione di output statistici dettagliati e interpretativi.



Analisi Statistica con SciPy e StatsModels

SciPy e StatsModels sono librerie utilizzate per eseguire analisi

Statistica Descrittiva:

- 1. from scipy import stats
- 2.
- 3.media = np.mean(df['colonna'])
- 4. mediana = np.median(df['colonna'])
- 5. moda = stats.mode(df['colonna'])



Analisi Statistica con SciPy e StatsModels

SciPy e StatsModels sono librerie utilizzate per eseguire analisi

Test Statistici:

1.t_stat, p_val = stats.ttest_ind(df['colonna1'], df['colonna2'])

Regressione Lineare:

- 1. import statsmodels.api as sm
- 2.
- 3.X = df[['colonna1', 'colonna2']]
- 4.y = df['target']
- 5.X = sm.add_constant(X) # Aggiunta dell'intercetta
- 6. modello = sm.OLS(y, X).fit()
- 7.print(modello.summary())

