Introducere în Pandas (Python)

Ce este Pandas?

Pandas este o bibliotecă Python utilizată pentru lucrul cu seturi de date. Aceasta oferă funcții pentru analizarea, curățarea, explorarea și manipularea datelor.

Numele „Pandas" provine atât din „Panel Data", cât și din „Python Data Analysis". Biblioteca a fost creată de Wes McKinney în 2008.

De ce să folosim Pandas?

Pandas ne permite să analizăm seturi mari de date și să tragem concluzii pe baza teoriilor statistice.
De asemenea, Pandas poate curăța seturi de date dezordonate și le poate transforma în date lizibile și relevante.
Datele relevante sunt foarte importante în știința datelor.

Ce este Data Science?

Data Science este o ramură a informaticii care studiază modul de stocare, utilizare și analiză a datelor pentru a extrage informații utile din acestea.

Ce poate face Pandas?

Pandas oferă răspunsuri despre date, cum ar fi:

- Există o corelație între două sau mai multe coloane?
- Care este valoarea medie?
- Care este valoarea maximă?
- Care este valoarea minimă?

  Pandas poate, de asemenea, să șteargă rândurile care nu sunt relevante sau care conțin valori greșite, cum ar fi valorile goale (NULL). Acest proces se numește curățarea datelor.

Instalarea Pandas

Dacă Python și PIP sunt deja instalate, instalarea Pandas este foarte ușoară.
Folosește următoarea comandă în terminal sau Command Prompt:

pip install pandas

Dacă această comandă nu funcționează, poți folosi o distribuție Python care are deja Pandas instalat, precum Anaconda sau Spyder.

Importarea Pandas

După instalare, Pandas poate fi importat în aplicațiile tale folosind cuvântul cheie import:

```
import pandas as pd
```

Acum Pandas este gata de utilizare.

Exemplu simplu

```
import pandas as pd
mydataset = {
'cars': ["BMW", "Volvo", "Ford"],
'passings': [3, 7, 2]
}
myvar = pd.DataFrame(mydataset)
print(myvar)
```

Ce este o Serie (Series)?

O Serie Pandas este similară cu o coloană dintr-un tabel. Este un array unidimensional care poate conține date de orice tip.

Exemplu:

```
import pandas as pd
a = [1, 7, 2]
myvar = pd.Series(a)
print(myvar)
```

Indici (Labels)

Dacă nu se specifică altceva, valorile sunt etichetate cu numărul indexului lor: prima valoare are index 0, a doua index 1 etc.
Acest index poate fi folosit pentru a accesa o anumită valoare.

Exemplu:

```
print(myvar[0])
```

Noțiuni de bază în Pandas

Obiecte principale în Pandas:

- `Series`: coloană unică de date
- `DataFrame`: tabel bidimensional (similar cu un Excel)

Funcții uzuale:

- `df.head()` → afișează primele rânduri
- `df.info()` → informații generale despre setul de date
- `df.describe()` → statistici descriptive
- `df.dropna()` → șterge valorile lipsă
- `df.fillna()` → completează valorile lipsă
- `df.sort_values()` → sortează datele
- `df.groupby()` → grupează datele după o coloană
- `df.loc[] și df.iloc[]` → selectează date pe baza etichetelor sau a pozițiilor.

Ce este un DataFrame?

Un Pandas DataFrame este o structură de date bidimensională, similară cu un tabel cu rânduri și coloane (ca un tabel Excel sau un array bidimensional).

Exemplu - Crearea unui DataFrame simplu:

```
import pandas as pd
data = {
"calories": [420, 380, 390],
"duration": [50, 40, 45]
}
df = pd.DataFrame(data)
print(df)
```

Vizualizarea datelor

O metodă foarte utilizată pentru a obține rapid o privire de ansamblu asupra unui DataFrame este metoda head().
Metoda head() returnează anteturile și un număr specificat de rânduri, începând de sus.

Exemplu - Afișarea primelor 10 rânduri:

```
df = pd.read_csv('data.csv')
print(df.head(10))
```

Notă: dacă nu se specifică un număr de rânduri, metoda head() returnează implicit primele 5 rânduri.

Exemplu - Afișarea primelor 5 rânduri:

```
df = pd.read_csv('data.csv')
print(df.head())
```

Metoda tail()

Există și metoda tail(), care afișează ultimele rânduri ale DataFrame-ului.
Aceasta returnează anteturile și un număr specificat de rânduri, începând de jos.

Exemplu - Afișarea ultimelor 5 rânduri:

```
print(df.tail())
```

Citirea și scrierea fișierelor CSV

O modalitate simplă de a stoca seturi mari de date este utilizarea fișierelor CSV (Comma Separated Values).
Fișierele CSV conțin text simplu și pot fi citite ușor de oricine, inclusiv Pandas.

Exemplu - Încărcarea unui fișier CSV într-un DataFrame:

```
df = pd.read_csv('data.csv')
print(df.to_string())
```

Sfaturi utile pentru citirea fișierelor CSV cu Pandas:

- `df = pd.read_csv("file.csv", delimiter=";")` → dacă separatorul nu este „,"
- `df = pd.read_csv("file.csv", encoding="utf-8")` → dacă fișierul are caractere speciale`
- `df.to_csv("nume_fisier.csv", index=False)` → pentru a salva un DataFrame în CSV fără coloana index.

Alte funcții utile în Pandas

Funcții utile pentru explorarea datelor:

- `df.shape` → afișează dimensiunea DataFrame-ului (rânduri, coloane)
- `df.columns` → afișează numele coloanelor

- `df.describe()` → oferă statistici descriptive
- `df.info()` → oferă informații despre tipurile de date și valorile lipsă
- `df.value_counts()` → numără valorile unice dintr-o coloană
- `df.isnull().sum()` → verifică câte valori lipsă există pe coloană
- `df.drop(columns=\["col"\])` → șterge o coloană
- `df.sort_values(by="col")` → sortează după o coloană
- `df.rename(columns={"vechi": "nou"})` → redenumește coloane.