Curriculă Pandas

Capitolul 1 – Pandas Series

- Crearea unei serii
  - Dintr-o listă
  - Dintr-un dicționar
  - Dintr-un fișier .csv cu `read_csv`
- Atribute și metode ale obiectului Series
  - Atribute: `index`, `values`, `dtype`, `name`
  - Metode uzuale: `head()`, `tail()`, `describe()`, `value_counts()`, `sort_values()` etc.
- Pandas Series ca argumente în funcții
  - Transmiterea seriilor în funcții Python
  - Vectorizare vs. `apply()`
- Extragerea valorilor seriei
  - După index numeric
  - După etichetă
  - Indexare booleană (ex. `serie[serie > 10]`)

Capitolul 2 – Pandas DataFrame

- Atribute și metode comune cu Series
  - Similarități și diferențe
  - `shape`, `columns`, `info()`, `describe()`, `T`
- Extragerea și adăugarea datelor
  - Pe o coloană: `df['col']`
  - Pe mai multe coloane: listă de coloane
  - Adăugarea unei coloane: `df['noua_coloana'] = ...`
- Filtrarea datelor
  - Condiție unică: `df[df['col'] > 100]`
  - Condiții multiple: `&`, `|`, `~`
- Alte metode utile
  - `sort_values()`, `sort_index()`
  - `unique()`, `nunique()`, `isin()`, `drop_duplicates()`
- Utilizarea `loc` și `iloc`
  - Indexare cu etichete (`loc`) vs. cu poziții (`iloc`)
  - Selectare rânduri și coloane: `df.loc[rows, cols]`
- Ștergerea rândurilor sau coloanelor
  - `drop()` cu `axis`, `inplace`
- Redenumirea coloanelor: `df.rename(columns={...})`
- Crearea unui DataFrame aleatoriu
  - Exemplu cu `numpy.random`

Capitolul 3 – Input/Output de date externe

- Citirea datelor: `pd.read_csv`, `pd.read_excel`, `pd.read_json`
- Salvarea datelor: `df.to_csv`, `df.to_excel`, `df.to_json`
- Conectarea la baze de date SQL: `pd.read_sql`

Capitolul 4 – Agregări, GroupBy și combinare

- GroupBy și agregări
  - `df.groupby('col').agg({'altacol': 'sum'})`
  - Funcții: `sum`, `mean`, `count`, `min`, `max`, `describe`, `apply`
- Pivot Table și Crosstab
  - `pd.pivot_table(...)`
  - `pd.crosstab(df['col1'], df['col2'])`
- Concatenare, Merge și Join
  - `pd.concat([df1, df2], axis=...)`
  - `pd.merge(df1, df2, on='col', how='inner/outer/left/right')`

Capitolul 5 – Curățarea și pregătirea datelor

- Valori lipsă: `isnull()`, `dropna()`, `fillna()`
- Outliers: identificare și tratare
- Transformări de string: `.str.lower()`, `.str.split()`, `.str.replace()`, regex
- Conversii de tip: `astype()` pentru numeric, categoric, datetime

Capitolul 6 – Indexare avansată și MultiIndex

- Crearea MultiIndex: `df.set_index(['col1', 'col2'])`
- Navigare și reindexare: `df.reindex()`, `df.unstack()`
- Sortare ierarhică: `df.sort_index(level=...)`

Capitolul 7 – Vizualizare și date temporale

- Vizualizare
  - Metode Pandas: `df.plot(kind='line'/'bar'/'hist')`
  - Integrare cu Matplotlib/Seaborn
- Date temporale
  - Conversie la `datetime`: `pd.to_datetime()`
  - Extracția componentelor: an, lună, zi, oră
  - `resample()` și agregări pe intervale de timp

Capitolul 8 – Performanță și optimizări

- `apply` vs. funcții vectorizate
- Tipuri de date eficiente: categoric, numeric
- Seturi de date mari: folosirea Dask sau alternative

Capitolul 9 – Mini-proiect aplicativ

- Import date: `read_csv`
- Explorare și curățare: valori lipsă, outliers
- Coloane derivate, filtrare
- GroupBy, Pivot Table, Crosstab
- Vizualizare simplă
- Salvare rezultate: CSV/Excel
- Discuții interpretare și bune practici
- Lucru cu date temporale