ANALIZA VELIKIH PODATAKA

školska 2024/2025 godina

Vežba 3: Rad sa tabelarnim podacima pomoću Pandas biblioteke

Pandas je najpopularnija biblioteka u Pythonu za manipulaciju podacima, a najvažniji objekti u Pandas-u su **DataFrame** i **Series**. DataFrame predstavlja tabelarne podatke, dok je Series jednostavan niz podataka, često korišćen za rad sa jednom kolonom u tabeli.

U ovoj vežbi ćemo se fokusirati na rad sa **CSV fajlovima**, učiti kako učitati podatke, prikazivati prve redove, filtrirati podatke, sortirati ih i raditi sa stvarnim datasetovima.

🌟 1. Pandas - Osnovne strukture podataka: Series i DataFrame

🔷 Series

Series je osnovna struktura podataka u Pandas-u koja predstavlja jednostavan niz podataka sa povezanim indeksima. Može da sadrži bilo koju vrstu podataka (npr. int, float, string, itd.) i omogućava lak pristup pojedinačnim vrednostima pomoću indeksa.

- **Pandas Series** je poput Python liste, ali sa dodatnom funkcionalnošću, kao što je mogućnost da svakom elementu dodelimo jedinstveni indeks.
- Može se koristiti za rad sa pojedinačnim kolonama iz DataFrame-a.

Primer koda:

```
import pandas as pd

# Kreiranje Series iz liste
data = [10, 20, 30, 40, 50]
series = pd.Series(data)

print(series)
```

Šta se dešava?

- U ovom primeru, kreiramo Pandas Series iz obične Python liste. Pandas automatski dodeljuje indekse elementima (od 0 do 4 u ovom slučaju).

🔷 DataFrame

DataFrame je najvažnija struktura podataka u Pandas-u i predstavlja tabelu sa redovima i kolonama. Svaka kolona može biti Series, a zajedno čine tabelu podataka.

- DataFrame je dvodimenzionalna tabela (poput baze podataka, Excel tabele ili SQL tabele).
- DataFrame omogućava pristup celokupnim kolonama, redovima, kao i obavljanje kompleksnih operacija na celokupnim datasetovima, uključujući filtriranje, grupisanje, agregaciju i analizu.

Primer koda:

```
# Kreiranje DataFrame-a od liste rečnika
data = {'Age': [25, 30, 35, 40],
        'Sex': ['Male', 'Female', 'Female', 'Male'],
        'BMI': [22.0, 26.5, 30.0, 28.2],
        'Charges': [5000, 6000, 7000, 8000]}

df = pd.DataFrame(data)

print(df)
```

Šta se dešava?

- Kreiramo DataFrame pomoću Python rečnika, gde su ključevi nazivi kolona, a vrednosti su liste koje predstavljaju podatke za svaku kolonu.
- DataFrame sadrži četiri kolone: Age, Sex, BMI, i Charges, sa odgovarajućim vrednostima.

🌟 2. Učitavanje podataka iz CSV fajla

Jedan od najčešćih načina za rad sa stvarnim podacima je uvoz podataka iz CSV fajlova. Pandas funkcija pd.read_csv() omogućava učitavanje podataka iz CSV fajla u DataFrame.

- **CSV (Comma Separated Values)** je format za čuvanje podataka u tekstualnom obliku gde su vrednosti odvojene zarezima.
- Pandas omogućava lako učitavanje, analizu i manipulaciju podacima iz CSV fajlova.

Primer koda:

```
# Učitavanje CSV fajla u DataFrame
url = "https://www.kaggle.com/datasets/willianoliveiragibin/healthcare-insurance"
df = pd.read_csv(url)

# Prikaz prvih 5 redova
print(df.head())
```

Šta se dešava?

- Koristimo funkciju pd.read_csv() da bismo učitali podatke iz CSV fajla sa linka.
- Funkcija head() prikazuje prvih 5 redova DataFrame-a, što nam pomaže da brzo vidimo sadržaj i strukturu podataka.

🌟 3. Prikazivanje prvih nekoliko redova

Kada učitamo dataset, obično želimo da proverimo kako podaci izgledaju i koja je struktura fajla. Za to koristimo funkciju head() koja prikazuje prvih nekoliko redova u DataFrame-u.

- **head()** metoda prikazuje prvih nekoliko redova DataFrame-a, što nam pomaže da brzo proverimo podatke bez da bismo morali da gledamo ceo dataset.
- Možemo da promenimo broj redova koji želimo da prikažemo, na primer df.head(10) za prikaz prvih 10 redova.

Primer koda:

```
# Prikazivanje prvih 10 redova
print(df.head(10))
```

Šta se dešava?

- Prikazujemo prvih 10 redova DataFrame-a kako bismo dobili bolji uvid u podatke, posebno kada imamo veliki dataset.

🌟 4. Filtriranje podataka

Filtriranje podataka omogućava da selektujemo samo određene redove iz DataFrame-a koji zadovoljavaju određene uslove. Ovo je ključno za analizu specifičnih podskupova podataka.

- Filtriranje podataka se može vršiti korišćenjem uslova na kolonama.
- Na primer, možemo selektovati sve osobe koje su pušači ili one koji imaju određeni BMI.

Primer koda:

```
# Filtriranje podataka – osobe sa BMI većim od 30
high_bmi = df[df['BMI'] > 30]

print(high_bmi)
```

Šta se dešava?

- Ovaj kod filtrira DataFrame i prikazuje sve redove u kojima je vrednost u koloni BMI veća od 30.
- Pandas omogućava upotrebu složenih uslova za filtriranje podataka, kao što su logičke operacije (AND, OR).

🌟 5. Sortiranje podataka

Sortiranje je važan proces kada želimo da analiziramo podatke u određenom redosledu, na primer, od najnižih do najviših vrednosti, ili obrnuto.

- Pandas omogućava jednostavno sortiranje podataka po vrednostima u jednoj ili više kolona.
- **sort_values()** funkcija omogućava sortiranje podataka u rastućem ili opadajućem redosledu.

Primer koda:

```
# Sortiranje podataka prema koloni 'Charges' u rastućem redosledu
df_sorted = df.sort_values(by='Charges', ascending=True)

print(df_sorted)
```

Šta se dešava?

- Ovaj kod sortira DataFrame po vrednostima u koloni Charges, u rastućem redosledu.
- Ako želimo opadajući redosled, možemo postaviti argument ascending=False.

🌟 6. Rad sa realnim podacima - Healthcare Insurance Dataset

Ovaj dataset sadrži informacije o osiguranju u zdravstvenoj industriji i povezanost između osobina korisnika (kao što su starost, pol, BMI, pušenje, itd.) i troškova zdravstvenog osiguranja. Korišćenjem ovog dataset-a možemo proučiti kako različiti faktori utiču na troškove osiguranja i razviti modele za predviđanje ovih troškova.

🔷 Učitavanje i analiza:

Prvo ćemo učitati ovaj dataset iz Kaggle-a i analizirati osnovne karakteristike podataka, uključujući filtriranje podataka prema određenim uslovima (npr. osobe koje puše ili osobe sa višim BMI-om) i sortiranje prema troškovima osiguranja.

Primer koda:

```
# Učitavanje podataka sa Kaggle-a
df = pd.read_csv('path_to_file.csv')

# Prikazivanje osnovnih informacija o datasetu
print(df.info())

# Prikazivanje statističkog pregleda podataka
print(df.describe())

# Filtriranje podataka: pušači sa BMI manjim od 30
pusaci_bmi30 = df[(df['smoker'] == 'yes') & (df['bmi'] < 30)]
```

Šta se dešava?

- info() metoda prikazuje osnovne informacije o DataFrame-u, kao što su broj redova, broj ne-null vrednosti i tipovi podataka u svakoj koloni.
- describe() metoda daje statistički pregled numeričkih kolona (proseci, standardne devijacije, min, max, kvartili).

📝 **Zadatak za samostalni rad**

1. Učitajte podatke iz CSV fajla sa Kaggle-a.
2. Prikazujte prvih 10 redova podataka.
3. Filtrirajte sve osobe koje imaju BMI veći od 30 i sortirajte ih prema troškovima osiguranja (Charges).
4. Kreirajte vizualizaciju koja prikazuje korelaciju između BMI i Charges.