ANALIZA VELIKIH PODATAKA

školska 2024/2025 godina

Vežba 5: Vizuelizacija podataka pomoću Matplotlib i Seaborn

Vizuelizacija podataka je proces predstavljanja informacija u obliku grafova, grafikona i dijagrama. Ona predstavlja most između sirovih podataka i korisnog znanja koje može da se primeni u praksi. Cilj vizuelizacije je da kompleksne skupove podataka učini lako razumljivim kroz grafičke prikaze.

Kada podatke prikažemo vizuelno, mozak ih mnogo brže i efikasnije obradi nego kada ih gledamo u tabelama ili sirovim brojevima.

Ključni razlozi zašto koristimo vizuelizaciju:

- Brže uočavanje trendova i obrazaca.
- Lako poređenje između različitih grupa ili kategorija.
- Otkrivanje anomalija ili neočekivanih pojava u podacima.
- Donošenje boljih i informisanijih odluka.

U Python programiranju, najčešće koristimo dve biblioteke za pravljenje grafika:

- **Matplotlib** — osnovna biblioteka za kreiranje svih vrsta grafova.
- **Seaborn** — proširena biblioteka zasnovana na Matplotlib-u koja omogućava lepši dizajn i jednostavnije kreiranje kompleksnih grafika.

Matplotlib nam daje potpunu kontrolu nad svim aspektima grafikona (boje, veličine, oznake, mreže itd.).

Seaborn pojednostavljuje pravljenje grafika koji automatski izgledaju profesionalno.

Zahvaljujući ovim alatima, analiza i interpretacija podataka postaje brža, intuitivnija i dostupna širem krugu korisnika.

📈 Linijski dijagram (Line Plot)

Linijski dijagram prikazuje **vrednosti u odnosu na vremensku osu** (ili drugu kontinuiranu promenljivu). Tačke koje predstavljaju pojedinačne podatke su povezane linijama.

Kada ga koristiti:

- Praćenje promena kroz vreme (npr. prodaja po mesecima).
- Analiza trenda rasta ili opadanja neke veličine.
- Vizuelizacija sezonskih obrazaca ili ciklusa.

Primer:

```
#Prvo uključimo neophodnu biblioteku
import matplotlib.pyplot as plt

meseci = ['Jan', 'Feb', 'Mar', 'Apr', 'Maj', 'Jun']
prodaja = [200, 250, 220, 270, 300, 320]

plt.plot(meseci, prodaja, marker='o')
plt.title('Prodaja po mesecima')
plt.xlabel('Mesec')
plt.ylabel('Broj prodatih proizvoda')
plt.grid(True)
plt.show()
```

📊 Stubičasti grafikon (Bar Chart)

Stubičasti grafikon koristi **vertikalne ili horizontalne stubove** za prikaz vrednosti različitih kategorija.

Kada ga koristiti:

- Upoređivanje kategorija (npr. broj učenika po školama).
- Prikazivanje razlika između grupa.
- Ilustrovanje relativne veličine između entiteta.

Primer:

```
odeljenja = ['I-1', 'I-2', 'II-1', 'II-2']
broj_ucenika = [25, 28, 30, 27]

plt.bar(odeljenja, broj_ucenika, color='lightblue')
plt.title('Broj učenika po odeljenju')
plt.xlabel('Odeljenje')
plt.ylabel('Broj učenika')
plt.show()
```

📊 Histogram

Histogram je vrsta grafikona koji prikazuje raspodelu numeričkih podataka. Podaci se grupišu u određene intervale (tzv. "binove"), a za svaki interval se crta stub čija visina predstavlja broj vrednosti koje upadaju u taj opseg.

Kada ga koristiti:

- Analiza frekvencije pojava.
- Razumevanje raspodele vrednosti (npr. raspodela ocena, godina života).
- Procena normalnosti (da li podaci prate normalnu distribuciju).

Primer:

```
# Prvo definišemo ocene od 5 do 10
ocene = [6, 7, 8, 8, 9, 10, 7, 6, 9, 10, 8, 7, 6, 8, 9, 7]

plt.hist(ocene, bins=5, color='green', edgecolor='black')
plt.title('Distribucija ocena')
plt.xlabel('Ocena')
plt.ylabel('Broj učenika')
plt.show()
```

⚡ Scatter Plot (Raspršeni dijagram)

Scatter plot (raspršeni dijagram) prikazuje pojedinačne tačke koje predstavljaju parove vrednosti za dve promenljive. Svaka tačka na grafiku odgovara jednom paru podataka.

Kada ga koristiti:

- Otkrivanje veza između dve promenljive (korelacija).
- Vizuelizacija odnosa između faktora (npr. visina i težina).
- Prepoznavanje grupisanja ili klastera podataka.

Primer:

```
visina = [160, 165, 170, 175, 180, 185]
tezina = [55, 60, 65, 70, 80, 85]

plt.scatter(visina, tezina, color='red')
plt.title('Visina vs Težina')
plt.xlabel('Visina (cm)')
plt.ylabel('Težina (kg)')
plt.grid(True)
plt.show()
```

📦 Boxplot (Dijagram kutije)

Boxplot je grafički prikaz koji sažima raspodelu podataka kroz pet ključnih vrednosti: minimum, prvi kvartil (Q1), medijanu (Q2), treći kvartil (Q3) i maksimum. Takođe pomaže u prepoznavanju potencijalnih odstupanja (outliera).

Kada ga koristiti:

- Pregled osnovnih statističkih karakteristika skupa podataka na jednostavan način.
- Poređenje raspodele između više grupa ili kategorija.
- Otkrivanje asimetrije (da li su podaci više koncentrisani prema gornjim ili donjim vrednostima).
- Detektovanje ekstremnih vrednosti (outlieri).

Primer:

```
ocene = [6, 7, 8, 8, 9, 10, 7, 6, 9, 10, 8, 7, 6, 8, 9, 7]

plt.boxplot(ocene, patch_artist=True, boxprops=dict(facecolor='lightblue'))
plt.title('Boxplot ocena')
plt.ylabel('Ocena')
plt.grid(axis='y')
plt.show()
```

🌡️ Heatmap (Toplotna mapa)

Heatmap prikazuje matricu podataka koristeći boje da predstavi intenzitet vrednosti. Tamnije ili svetlije boje ukazuju na više ili niže vrednosti u tabeli.

Kada koristiti:

- Vizuelizacija odnosa između više promenljivih (npr. korelacija između više faktora).
- Brzo otkrivanje obrazaca, anomalija i koncentracija vrednosti.
- Analiza velikih tabela ili matrica podataka (npr. konfuzionih ili korelacionih matrica).

Primer:

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

podaci = np.random.rand(5, 5)

plt.figure(figsize=(8, 6))
sns.heatmap(podaci, annot=True, cmap='coolwarm', linewidths=0.5)
plt.title('Toplotna mapa podataka')
plt.show()
```

Naprednija vizuelizacija pomoću Seaborn-a

Seaborn je napredna biblioteka za vizualizaciju podataka u Pythonu, koja se bazira na Matplotlib-u, ali omogućava:

- automatsko stilizovanje grafova,
- jednostavnije kreiranje složenijih grafika,
- elegantniji i profesionalniji izgled uz minimalno koda.

Jedan od najčešćih primera je kreiranje **scatter plot-a** sa **grupisanjem podataka** po nekoj kategoriji, kao što je **pol** (muško/žensko).

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

# 1. Kreiranje DataFrame-a
data = pd.DataFrame({
    'Visina': [160, 165, 170, 175, 180, 185],
    'Težina': [55, 60, 65, 70, 80, 85],
    'Pol': ['M', 'Z', 'M', 'Z', 'M', 'Z']  # M = muško, Z = žensko
})

# 2. Seaborn scatter plot
sns.scatterplot(x='Visina', y='Težina', hue='Pol', data=data)

# 3. Dodavanje naslova
plt.title('Visina i Težina po polu')

# 4. Prikazivanje grafika
plt.show()
```

Objašnjenje koda:

- sns.scatterplot(...) – pravi **raspršeni dijagram** (scatter plot) gde:
  - **x** osa predstavlja promenljivu **'Visina'**,
  - **y** osa predstavlja promenljivu **'Težina'**,
  - **hue='Pol'** – različitim bojama se automatski označavaju grupe (M i Z).
- data=data – kaže Seabornu da koristi kolone iz DataFrame-a koji smo definisali.
- **Legenda** se automatski generiše i pokazuje koje boje pripadaju kom polu.
- **Prednosti Seaborn-a** u ovom primeru:
  - Grafikon automatski izgleda profesionalno bez potrebe za dodatnim podešavanjem boja, veličina, markera itd.
  - Boje su estetski usklađene i prepoznatljive.
  - Postavljanje "hue" omogućava vrlo lako vizuelno poređenje kategorija.

Seaborn podržava i dodatne parametre, npr.:

- style='whitegrid' za postavljanje svetle mreže iza grafika,
- palette='Set2' za biranje palete boja,
- s=100 za podešavanje veličine tačaka.

📚 Primer (dovršiti samostalno)

Dataset je dostupan ovde: https://www.kaggle.com/datasets/mhdzahier/travel-insurance

🔧 Plan za rad:

1. **Učitavanje podataka sa Kaggle-a:** Prvo ćemo učitati dataset sa Kaggle-a i prikazati osnovne informacije o podacima.
2. **Obrada podataka:**
    - Prepoznavanje i uklanjanje missing vrednosti.
    - Promena tipova podataka ako je potrebno.
    - Priprema za dalje analize (npr. kodiranje kategorijskih varijabli).
3. **Vizualizacija podataka:**
    - Distribucija statusa zahteva.
    - Analiza povezanosti između tipa agencije i broja prodaja.
    - Raspodela starosne dobi i pola osiguravajućih lica.
    - Analiza korelacija između različitih varijabli.

```
# Distribucija statusa zahteva
plt.figure(figsize=(8, 6))
sns.countplot(x='Claim.Status', data=data, palette='Set2')
plt.title('Distribucija statusa zahteva za osiguranje', fontsize=14)
plt.xlabel('Status zahteva', fontsize=12)
plt.ylabel('Broj zahteva', fontsize=12)
plt.show()

# Analiza broja prodaja prema tipu agencije
plt.figure(figsize=(10, 6))
sns.barplot(x='Agency.Type', y='Net.Sales', data=data, palette='Blues')
plt.title('Broj prodaja prema tipu agencije', fontsize=14)
plt.xlabel('Tip agencije', fontsize=12)
plt.ylabel('Ukupna prodaja (Net.Sales)', fontsize=12)
plt.show()

# Raspodela starosne dobi prema polu
plt.figure(figsize=(10, 6))
sns.boxplot(x='Gender', y='Age', data=data, palette='coolwarm')
plt.title('Raspodela starosne dobi prema polu', fontsize=14)
plt.xlabel('Pol', fontsize=12)
plt.ylabel('Starosna dob', fontsize=12)
plt.show()
```