

Fakultet tehničkih nauka,

Novi Sad, Trg Dositeja Obradovića 6

Projektni zadatak

Geoprostorna analiza stanovništva opština u Srbiji

Mentori:

Aleksandra Radulović
Dubravka Sladić
Vladimir Pajić

Studenti:

Filip Goldberger RA182/2022
Minja Drakul RA58/2022
Srđan Slokar RA149/2022

Novi Sad, Oktobar 2025.

Sadržaj

➤ Uvod.....	3
Zahtevi projekta.....	3
Pokretanje projekta.....	3
➤ Podaci.....	4
➤ Struktura koda.....	4
Korak 1: Učitavanje podataka.....	4
Korak 2: Kreiranje demografskih podataka.....	4
Korak 3: Spajanje podataka.....	6
Korak 4: Proračun površine i gustine.....	6
Korak 5: Ažuriranje podataka.....	7
Korak 6: Vizualizacija.....	7
➤ Očekivani izlaz.....	8
➤ Zaključak.....	9

Uvod

Ovaj projekat demonstrira osnovne tehnike geoprostorne analize podataka korišćenjem Python biblioteka geopandas, pandas, matplotlib. Cilj projekta je da se spoje geografski podaci o granicama opština u Srbiji sa demografskim podacima (broj stanovnika), izračuna gustina naseljenosti i na kraju vizualizuju rezultati kreiranjem tematskih mapa.

Koraci realizacije projekta:

- Učitavanje geoprostornih podataka iz shapefile formata
- Kreiranje tabelarnih podataka o stanovništvu
- Spajanje geografskih i tabelarnih podataka
- Izračunavanje novih atributa (površina, gustina, naseljenost)
- Ažuriranje podataka
- Kreiranje i prikazivanje mapa

Zahtevi projekta

Za izvršavanje python skripte potrebno je imati instalirane sledeće pakete:

- geopandas: rad sa geoprostornim podacima
- pandas: rad sa tabelarnim podacima (DataFrame)
- matplotlib: za kreiranje i prikazivanje grafika i mapa

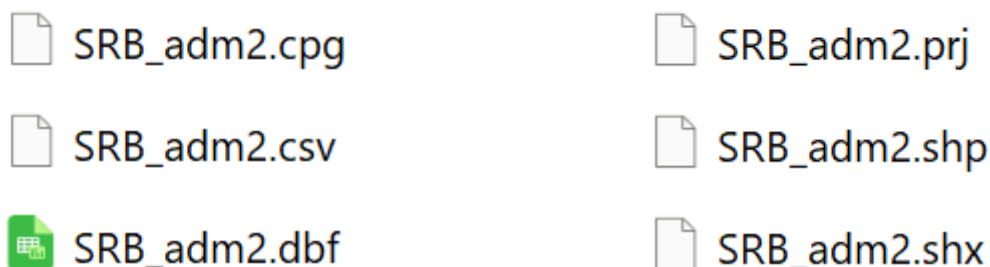
Pokretanje projekta

Ceo projekat se pokrece make komandom u terminalu koja će kreirati .venv direktorijum, instalirati neophodne pakete i pokrenuti skriptu. Neophodni programi:

- python: pokretanje python skripte
- pip: instalacija neophodnih paketa
- make: za automatizaciju instaliranja paketa i pokretanje programa

Podaci

Projekat zahteva shapefile sa administrativnim granicama opština u Srbiji. U skripti se očekuje da se ovaj fajl nalazi na putanji: data/SRB_adm2.shp. Potrebno je osigurati da se svi prateći fajlovi (.shx, .dbf, .prj, itd) nalaze u istom direktorijumu.



Slika 1. Potrebni fajlovi

Struktura koda

Skripta je podeljena u logičke celine koje odgovaraju koracima u analizi podataka.

Korak 1: Učitavanje podataka

Skripta prvo uvozi sve potrebne biblioteke, a zatim se koristi `geopandas.read_file()` za učitavanje shapefile-a. Rezultat je `GeoDataFrame` objekat `opstine_gdf` koji sadrži geometrijske podatke (poligone opština) i prateće atribute iz `.dbf` fajla.

```
7 # Loading data from .shp file (adm1 - opstine Srbije)
8 try:
9     opstine_gdf = gpd.read_file("data/SRB_adm2.shp")
10    print("Shapefile uspešno učitano.")
11 except:
12    raise Exception("Cannot load data")
13
```

Slika 2. Učitavanje fajlova

Korak 2: Kreiranje demografskih podataka

Kreiranje je Python rečnik (dictionary) koji sadrži imena nekoliko opština i njihov (primer) broj stanovnika. Od rečnika se zatim kreira pandas DataFrame stanovništvo_df. Ovo simulira čets slučaj gde se demografski podaci dobijaju iz odvojenog izvora.

```
14 # Kreiranje rečnika sa opstinama i brojem stanovnika
15 podaci_stanovnistvo = {
16     'opstina': ['Čačak', 'Valjevo', 'Novi Sad', 'Subotica', 'Niš', 'Kragujevac'],
17     'stanovnistvo': [105612, 82169, 368967, 127756, 249816, 171186]
18 }
19 stanovnistvo_df = pd.DataFrame(podaci_stanovnistvo)
```

Slika 3. Kreiranje demografskih podataka

Korak 3: Spajanje podataka

Ključni korak gde se dva skupa podataka spajaju u jedan. Metoda .merge() spaja opstine_gdf i stanovnistvo_df.

- left_on="NAME_2" specificira da je ključ za spajanje u levom (opstine_gdf) DataFrame-u kolona NAME_2 (koja sadrži imena opština)
- right_on="NAME_2" specificira da je ključ za spajanje u desnom (stanovništvo_df) DataFrame-u kolona opstina.

Rezultat je novi GeoDataFrame merged_gdf koji sada sadrži i geometriju i broj stanovnika za odabrane opštine.

```
23 # Spajanje geografskih podataka sa iz shape file sa demografskim podacima iz DataFrame
24 merged_gdf = opstine_gdf.merge(stanovnistvo_df, left_on='NAME_2', right_on='opstina')
25 print("\nSpojeni podaci (prvih 5 redova):")
26 print(merged_gdf[['NAME_2', 'stanovnistvo', 'geometry']].head())
```

Slika 4. Spajanje podataka

Korak 4: Proračun površine i gustine

Transformacija koordinatnog referentnog sistema (CRS) iz geografskog (stepeni) u projektovani (metri) vrši se pomoću `.to_crs(eps=32634)`. EPSG:32634 je pogodan za Srbiju i omogućava precizno merenje.

Izračunavanje površine svakog poligona u kvadratnim metrima radimo pomoću `.geometry.area`. Vrednost se deli sa 1.000.000 da bi se dobila površina u km². Ovim se kreiraju dve nove kolone: `povrsina_km2` i `gustoca`, gde se gustina računa kao broj stanovnika po km².

```
28 # Transformacija u 2D mapu
29 merged_gdf_proj = merged_gdf.to_crs(eps=32634)
30
31 # Racunanje površine opština
32 povrsina_km2 = merged_gdf_proj.geometry.area / 1000000
33 merged_gdf['povrsina_km2'] = povrsina_km2
34
35 # Racunanje gustoce
36 merged_gdf['gustoca'] = merged_gdf['stanovnistvo'] / merged_gdf['povrsina_km2']
37 print("\nPodaci nakon dodavanja gustine naseljenosti:")
38 print(merged_gdf[['NAME_2', 'stanovnistvo', 'povrsina_km2', 'gustoca']].head())
```

Slika 5. Proračun površine i gustine

Korak 5: Ažuriranje podataka

Ažuriranje podataka u DataFrame-u se vrši pomoću `.loc()` funkcije koja pronalazi red gde je opština „Čačak“ i u tom redu vrednost u koloni stanovništvo se postavlja na 120.000. Nakon ažuriranja broja stanovnika obavezno se ponovo računa i gustina za tu opštinu.

```
40 # Menjanje broja stanovnika za opštinu
41 print("\nAžuriranje broja stanovnika za opštinu Čačak...")
42 merged_gdf.loc[merged_gdf['opstina'] == 'Čačak', 'stanovnistvo'] = 120000
43
44 # Ponovo racunaj gustinu stanovništva
45 merged_gdf.loc[merged_gdf['opstina'] == 'Čačak', 'gustoca'] = \
46     merged_gdf.loc[merged_gdf['opstina'] == 'Čačak', 'stanovnistvo'] / \
47     merged_gdf.loc[merged_gdf['opstina'] == 'Čačak', 'povrsina_km2']
48
49 print("Novi podaci za Čačak:")
50 print(merged_gdf[merged_gdf['opstina'] == 'Čačak'][['opstina', 'stanovnistvo', 'gustoca']])
```

Slika 6. Ažuriranje podataka

Korak 6: Vizualizacija

Kreiramo 2 pod-grafikona (axes) pomoću `plt.subplots(1, 2, ...)` jedan pored drugog. Na oba pod-grafikona se prvo iscrtavaju granice svih opština u Srbiji (`opstine_gdf`) sivom bojom da bi se obezbedio geografski kontekst.

Preko pozadine, na prvom grafiku (`ax1`), crtaju se samo opštine iz `merged_gdf`, obojene prema vrednostima u koloni `stanovnistvo`. Na drugom grafiku (`ax2`), crtaju se opštine, ali obojene prema vrednostima u koloni `gustoca`. `cmap` definiše paletu boja, a `legend` dodaje legendu za lakše tumačenje mape.

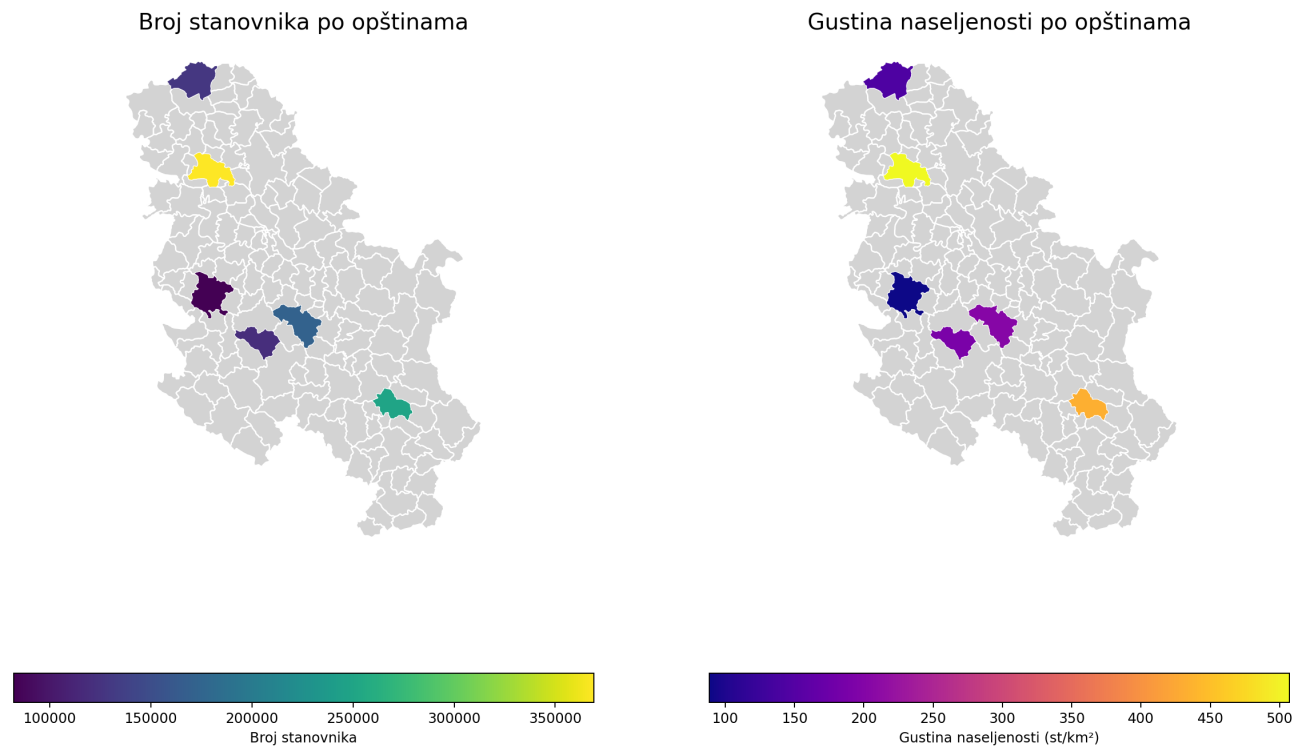
```
52 # Napravi 2 siva subplot-a
53 fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(20, 10))
54 opstine_gdf.plot(ax=ax1, color='lightgray', edgecolor='white')
55 opstine_gdf.plot(ax=ax2, color='lightgray', edgecolor='white')
56
57 # Oboj opstine po broju stanovnika
58 merged_gdf.plot(column='stanovnistvo', ax=ax1, legend=True,
59                 cmap='viridis',
60                 legend_kwds={'label': "Broj stanovnika",
61                             'orientation': "horizontal"})
62 ax1.set_title('Broj stanovnika po opštinama', fontdict={'fontsize': '16', 'fontweight': '3'})
63 ax1.set_axis_off()
64
65 # Oboj opstine po gustini naseljenosti
66 merged_gdf.plot(column='gustoca', ax=ax2, legend=True,
67                 cmap='plasma',
68                 legend_kwds={'label': "Gustina naseljenosti (st/km²)",
69                             'orientation': "horizontal"})
70 ax2.set_title('Gustina naseljenosti po opštinama', fontdict={'fontsize': '16', 'fontweight': '3'})
71 ax2.set_axis_off()
72
73 # Prikazi plotove
74 plt.suptitle('Analiza stanovništva odabranih opština u Srbiji', fontsize=20)
75 plt.show()
```

Slika 7. Vizualizacija

Očekivani izlaz

1. Konzolni ispis: u terminalu će biti ispisani koraci izvršavanja, uključujući kreirani `DataFrame`, prvih nekoliko redova spojenih podataka, kao i ažurirane podatke za opštinu Čačak
2. Grafički prozor: otvoriće se prozor sa dve mape:
 1. Leva mapa: Opštine obojene na osnovu ukupnog broja stanovnika
 2. Desna mapa: Opštine obojene na osnovu gustine naseljenosti

Analiza stanovništva odabranih opština u Srbiji



Slika 8. Izlaz programa

Zaključak

Ovaj projekat uspešno demonstrira upotrebu Python biblioteka geopandas za izvođenje osnovne geoprостorne analize kroz seriju koraka od učitavanja, spajanja i ažuriranje podataka do vizualizacije istih.