**Korisničko uputstvo**

Zadatak ovdje je bio predstaviti namjenu zemljišta primjenom Corine Land Cover klasa u opštini Ćićevac putem python skripte.

1. **Učitavanje biblioteka**

import geopandas as gpd

import matplotlib.pyplot as plt

Pomoću komande import uvozimo Python biblioteke neophodne za rad skripti.

U ovom slučaju uvozimo geopandas, biblioteku za rad sa geoprostornim podacima,

matplotlib.pyplot - biblioteka crtanja za kreiranje vizualizacija u Pythonu.

1. **Definisanje lokacija potrebnih shapefajlova**

CLC = gpd.read\_file(r'C:\Users\perko\OneDrive\Desktop\Programiranje-Ispit\CLC18 SRB UTM.shp')

Srb = gpd.read\_file(r'C:\Users\perko\OneDrive\Desktop\Programiranje-Ispit\OpstineSrb-UTM.shp')

CLC se odnosi na Corine Land Cover tj. na klase zemljištnih pokrivača

Srb predstavlja granice države Srbije.

1. **Provjera koordinatnog sistema podataka i izdvajanje opštine interesovanje**

print("Current Projection:", Srb.crs)

print("Current Projection:", CLC.crs)

Sledeći korak je bio provjera koordinatnog sistema podataka. Na osnovu povratne informacije naši podaci su u WGS (World Geodetic System) 84 / UTM zone 34N, ili EPSG:32634 koordinatnom sistemu. Ova projekcija odgovara našim podacima i nema potrebe za daljom transformacijom.

cicevac\_opstina = Srb[Srb['Opstina'] == 'Cicevac'] - ovim kodom smo izdvojili opstinu Ćićevac, sa kojom ćemo i raditi u ovom radu, iz shapefajla Srbije sa svim opštinama.

1. **Spajanje opštine Ćićevac sa Korine klasama - CLC lejerom**

intersection = gpd.overlay(CLC, cicevac\_opstina, how='intersection') – izveli smo operaciju ukrštanja između dva skupa geoprostornih podataka, CLC i cicevac\_opstina, koristeći funkciju gpd.overlay() iz geopandas biblioteke; how='intersection' specifikuje da se u rezultujućem skupu podataka trebaju zadržati samo geometrije koje se sijeku.

dissolved1 = intersection.dissolve(by='code\_18').reset\_index() – ovdje smo izveli operaciju rastvaranja na osnovu vrijednosti u koloni 'code\_18'. To znači da će sve geometrije sa istom vrijednošću iz 'code\_18' biti kombinovane u jednu geometriju. Reset\_index() funkcija se koristi za resetovanje indeksa tj. da nam kolonu 'code\_18' koji smo prethodno iskoristili za dissolve, vrati u tabelu. Jer na osnovu vrijednosti u koloni ' code\_18' ćemo dalje dodijeliti boje.

1. **Dodjela odgovarajućih boja mapi i prikaz rezultata**

code\_18\_values = ['112', '131', '211', '321', '231', '242', '243', '311', '321','324','411', '511'] – napravili smo listu koja sadrži sve vrijednosti iz 'code\_18'

color\_palette=['#ff0000','#a600cc','#ffffa8','#e68000','#e6e64d','#ffe64d','#e6cc4d',"#80ff00",'#ccf24d','#a6f200','#a6a6ff','#00ccf2'] – heksagonalna kolor paleta

value\_color\_map = {value: color\_palette[i] for i, value in enumerate(code\_18\_values)} – ovim kodom smo svakoj vrijednosti iz code\_18\_values dodijeli odgovarajuću boju iz color\_palette

dissolved1['color'] = dissolved1['code\_18'].map(value\_color\_map).fillna('gray') – dodajemo novu kolonu pod nazivom 'color' u dissolved1 varijablu. Vrijednosti iz kolone 'code\_18' će biti prikazane u prethodno zadatoj boji koju smo definisali varijablom value\_color\_map. Ako vrijednosti iz 'code\_18' nemaju odgovarajuću boju pri mapiranju, ona će se popuniti bojom 'siva' ('gray' u kodu iznad)

dissolved1.plot(color=dissolved1['color']) – kreiramo dijagram koji prikazuje prethodno pripojene vrijednosti iz 'code\_18' sa odgovarajućom bojom

plt.show() – prikaz krajnjih rezultata tj. prethodno generisanog dijagrama

output\_path = r'C:\Users\perko\OneDrive\Desktop\Programiranje-Ispit\dissolved1.shp' – definisemo lokaciju na kojoj cemo sacuvati shapefile koji smo kreirali

dissolved1.to\_file(output\_path)