

# Problem izbornih jedinica

---

**Prološćić, Doris**

**Undergraduate thesis / Završni rad**

**2024**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Josip Juraj Strossmayer University of Osijek, School of Applied Mathematics and Informatics / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet primijenjene matematike i informatike**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:126:895626>

*Rights / Prava:* [In copyright](#) / [Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2024-10-06**



*Repository / Repozitorij:*

[Repository of School of Applied Mathematics and Computer Science](#)





SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

ODJEL ZA MATEMATIKU

Sveučilišni prijediplomski studij Matematika i računarstvo

# Problem izbornih jedinica

ZAVRŠNI RAD

Mentor:

**prof. dr. sc. Kristian Sabo**

Kandidat:

**Doris Prološćić**

Osijek, 2024.

# Sadržaj

<b>Sažetak</b>	<b>3</b>
<b>Summary</b>	<b>4</b>
<b>1 Uvod</b>	<b>1</b>
<b>2 Izbori u Republici Hrvatskoj</b>	<b>2</b>
<b>3 Projektna web aplikacija</b>	<b>4</b>
3.1 Django web framework . . . . .	4
3.2 Korisničko sučelje web aplikacije . . . . .	7
3.3 Struktura web aplikacije . . . . .	11
3.3.1 <i>Model-i</i> . . . . .	11
3.3.2 <i>Obrasci</i> . . . . .	13
3.3.3 <i>View-ovi</i> . . . . .	14
3.3.4 <i>Template-i</i> . . . . .	20
<b>4 Mjere ujednačenosti težina biračkih glasova</b>	<b>27</b>
<b>5 Zaključak</b>	<b>30</b>
<b>Literatura</b>	<b>31</b>

# Sažetak

U radu je objašnjen problem nejednake težine biračkog glasa u različitim izbornim jedinicama Republike Hrvatske. Opisano je projektno rješenje koje pomaže u stvaranju raspodjele izbornih jedinica tako da se postigne zakonski zadovoljavajuća ujednačenost težine biračkog glasa. Detaljno je objašnjen način rada i način korištenja stvorene web aplikacije koja omogućava korisnicima da prema osobnom kriteriju stvore raspodjelu izbornih jedinica i dobiju objektivan uvid u kvalitetu raspodjele. Definirani su indeksi koji na različite načine mogu mjeriti ujednačenost težine biračkog glasa.

## Ključne riječi

izborne jedinice, težina biračkog glasa, raspodjela izbornih jedinica, web aplikacija, Python



# The problem of determining constituencies

## Summary

The paper explains the problem of unequal voting weight across different electoral districts in the Republic of Croatia. It describes a web application that assists in creating a distribution of constituencies that achieves voting equality. The paper provides a detailed explanation of how the created web application works and how it can be used, allowing users to create a distribution of constituencies according to their personal criteria and obtain an objective assessment of the distribution's quality. The paper provides an overlook of indices used for measuring the uniformity of voting weight.

## Keywords

constituencies, weight of votes, distribution of constituencies, web application, Python

# 1 | Uvod

Ustavni sud Republike Hrvatske donio je 7. veljače 2023. godine odluku kojom se ukida Zakon o izbornim jedinicama za izbor zastupnika u Zastupnički dom Hrvatskoga državnog sabora zbog ustanovljene nejednakosti u težini biračkog glasa po izbornim jedinicama. Do navedene odluke teritorij Republike Hrvatske bio je podijeljen na 10 izbornih jedinica od kojih se u svakoj biralo 14 zastupnika, no zbog granica izbornih jedinica nije bila postignuta jednakost biračkog prava[1]. Sukladno odluci, Hrvatski sabor morao je pronaći način da osigura jednakost biračkog prava do 1. listopada 2023. godine.

Kako bi se stvorio prijedlog za novu raspodjelu izbornih jedinica koji je u skladu sa Ustavom, osmišljen je projekt koji obuhvaća konstrukciju optimalne raspodjele izbornih jedinica uz pomoć spektralnog klasteriranja i izradu grafičkog korisničkog sučelja koje korisniku omogućava konstrukciju proizvoljne raspodjele i daje povratnu informaciju o kvaliteti konstruiranog modela. Projekt je ostvaren izradom programa u programskom jeziku Python. Rad se temelji na dijelu projekta koji se bavi izradom grafičkog korisničkog sučelja za stvaranje proizvoljne raspodjele.

U glavnom dijelu rada definirani su temeljni politički pojmovi vezani uz izbore i problem raspodjele izbornih jedinica. Opisano je stvoreno grafičko korisničko sučelje za konstrukciju rješenja problema izbornih jedinica. Grafičko korisničko sučelje izrađeno je kao web aplikacija kojoj se pristupa u web pregledniku. Stoga opis obuhvaća osnovne pojmove iz okruženja u kojem je stvorena web aplikacija, strukturu kreirane web aplikacije, objašnjenje pozadinskih procesa web aplikacije, rezultat raspodjele napravljene u web aplikaciji te izgled i upotrebu korisničkog sučelja web aplikacije. Kako bi se razumjeli rezultati koji su dobiveni za konstruiranu raspodjelu izbornih jedinica, definirani su indeksi koje se preporuča koristiti za mjerenje kvalitete raspodjele izbornih jedinica.

## 2 | Izbori u Republici Hrvatskoj

Izbori su proces tvorbe političke volje koji se temelji na mogućnosti birača da se odluče za jednu od više ponuđenih političkih opcija i njihovih nositelja, te na slobodi izbora među tim opcijama. Pravnu osnovicu izbora čini biračko pravo[7].

Biračko pravo temeljno je političko i ustavno pravo državljana da biraju svoje predstavnike i da budu birani u predstavnička tijela i na druge javne položaje[7]. Kako je Republika Hrvatska demokratska država, građanima je zajamčeno opće, jednako, izravno i tajno biračko pravo. Izbornim sustavom pokušavaju se osigurati navedena svojstva biračkog prava, omogućava se biračima da izražavaju svoje političke preferencije u obliku glasova i njime se određuje kako se birački glasovi pretvaraju u mandate.

Pravila koja su uspostavljena izbornim sustavom primjenjuju se unutar izbornog okruga, teritorijalne jedinice u kojoj birači glasuju za kandidate ili liste i u kojoj se ti glasovi pretvaraju u mandate. Podjela države na izborne okruge uvelike utječe na rezultate izbora. U Republici Hrvatskoj u trenutku ukidanja Zakona o izbornim jedinicama zbog dotada definirane podjele na izborne okruge, tj. izborne jedinice, nije bila osigurana jednakost biračkog prava[1, 10]. Jednako biračko pravo podrazumijeva da svi građani imaju jednako pravo glasa (jedan čovjek, jedan glas) i da svaki glas ima jednaku vrijednost (jedan glas, jedna vrijednost)[7].

Da bi se mogla uspoređivati vrijednost biračkog glasa, politolozi su definirali različite indekse koji mjere ujednačenost težine biračkog glasa i oni se mogu opisati matematičkom notacijom. Indeksi koji su korišteni u sklopu projekta definirani su u poglavlju Rezultati.



Slika 2.1: Stara raspodjela izbornih jedinica  
[8]



## 3 | Projektna web aplikacija

Interaktivni program za slaganje izbornih jedinica koji za rezultat daje prikaz kvalitete napravljene raspodjele izrađen je kao web aplikacija u programskom jeziku Python. Web aplikacija ima veliku prednost u odnosu na alternativna korisnička sučelja zbog toga što joj je jednostavno pristupiti. Pristup web aplikaciji omogućen je putem web preglednika i nema potrebe da se preuzima program ili programski paketi. Grafički je prikaz uredan, a korisnik na intuitivan način lako ostvaruje interakciju s programom bez ikakvog predznanja o kodiranju ili samom programu. Na taj način spojeno je projektno rješenje izrađeno u Jupyter Notebook-u s pristupačnijim korisničkim sučeljem za neprogramere te je dodana funkcionalnost samostalnog kreiranja raspodjele izbornih jedinica. Projektno rješenje izrađeno u Jupyter Notebook-u odnosi se na implementaciju algoritma spektralnog klasteriranja te primjenu algoritma u svrhu izrade rješenja problema izbornih jedinica. Rješenje je moguće pogledati u web aplikaciji. Posebno, u navedenom Jupyter okruženju implementirane su funkcije za računanje mjera kvalitete raspodjele izbornih jedinica koje se koriste u sklopu web aplikacije.

### 3.1 Django web framework

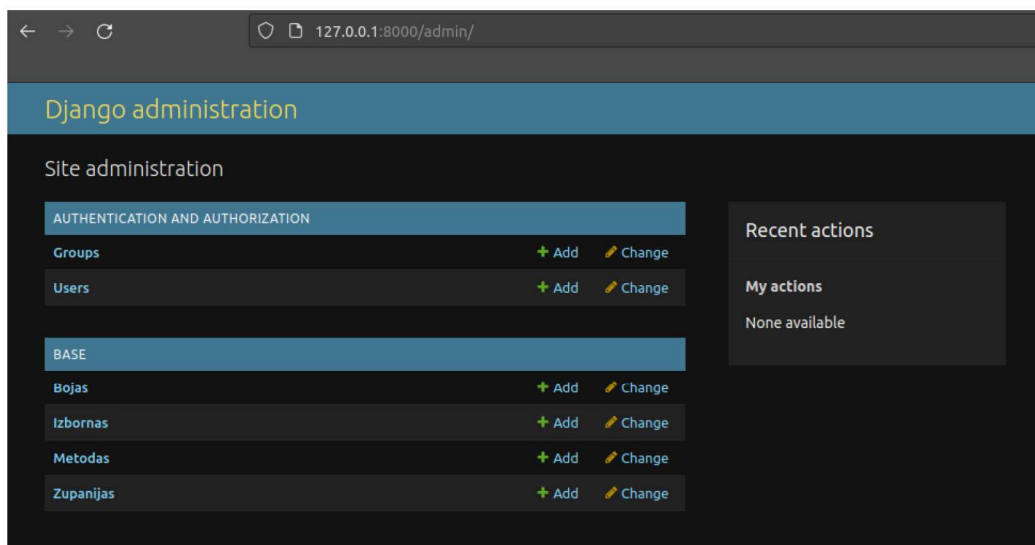
Za izradu web aplikacije korišten je Django web framework za izradu web aplikacija. Web framework je softver s alatima, ugrađenim knjižnicama (eng. *built-in libraries*) i gotovim komponentama za razvoj web aplikacija koje programer nadograđuje funkcionalnostima specifičnima za aplikaciju koju želi izgraditi. Kod se piše programskim jezikom Python. Django koristi *Model-View-Template* (MVT) arhitekturu za web aplikacije. *Model* je podatkovna struktura u Django, *view* logika koja se koristi za prezentaciju, a *template* je sloj prezentacije koji renderira podatke.

Što se tiče *model*-a, Django ima ugrađeni ORM (*Object-Relational Mapping*) sustav koji omogućuje rad s bazama podataka na objektno-orijentiran način[6]. Tako se *model*-i mogu definirati u Pythonu, a Django pozadinski generira odgovarajuće SQL upite nad bazom podataka. Također, aplikacija dolazi sa automatski generiranim administratorskim sučeljem pomoću kojeg programer može upravljati definiranim *model*-ima.

Django *view* naziv je za komponentu koja upravlja logikom za obradu zahtjeva i vraćanje odgovora klijentu[6]. Može biti u obliku funkcije ili klase. Svi *view*-ovi u projektnoj web aplikaciji definirani su kao funkcije. U tom slučaju *view* prima *HttpRequest* objekt (pri čemu *request* znači zahtjev) kao argument i vraća *HttpResponse*

```
class Zupanija(models.Model):
    broj_zupanije = models.IntegerField(null=True)
    naziv = models.CharField(max_length=50, default="županija", null=True)
    slika = models.ImageField(upload_to="", null=True)
    izborna = models.ForeignKey(Izborna, null=True, on_delete=models.SET_NULL, blank=True)
    broj_biraca = models.IntegerField(null=True, blank=True)
    boja = models.CharField(max_length=7, blank=True, null=True)

    def __str__(self):
        return self.naziv
```

Slika 3.1: Primjer *model*-a u Django

Slika 3.2: Django administratorsko sučelje

objekt (*response* u prijevodu znači odgovor). *View* komponenta može komunicirati s *model* komponentama za dohvaćanje podataka iz baze i, osim toga, podatke iz baze može proslijediti *template*-u. *View* je potrebno ukomponirati u aplikaciju na način da ga se navede u URL konfiguraciji. URL konfiguracija definira se unutar Python datoteke u Django projektu, a uspostavlja pravilo za mapiranje URL uzoraka i *view* funkcija. Time se omogućuje usmjeravanje HTTP zahtjeva na obradu odgovarajućem *view*-u. Dakle, kod pristupa određenom URL uzorku u web pregledniku pokreće se funkcija pripojenog *view*-a.

```
147 def najboljaraspodjela(request):
148     indeksi, podpre = f.najboljaraspodjelarender()
149     context = {'nazivi_indeksi': indeksi.keys, 'indeksi': indeksi.values, 'range': range(1,7), 'podpre': podpre, }
150     return render(request, 'base/informativna.html', context)
151
```

Slika 3.3: Primjer Django *view* funkcije

*Template* u arhitekturi Django aplikacije služi za generiranje HTML koda. Sadrži

```
urls.py 3
base > urls.py > ...
1  from django.urls import path
2  from . import views
3  from django.conf import settings
4  from django.conf.urls.static import static
5
6
7  urlpatterns = [
8      path('', views.start, name="start"),
9      path('metoda/', views.metoda, name="izbor_metode"),
10     path('izbornajedinica/<int:pk>/', views.izbornajedinica, name="izbornajedinica"),
11     path('rezultati/', views.rezultati, name="rezultati"),
12     path('najboljaraspodjela', views.najboljaraspodjela, name="najboljaraspodjela"),
13
14 ] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

Slika 3.4: URL konfiguracija

običan HTML kod za prikaz statičnih dijelova stranice i, u kombinaciji s njime, koristi posebnu sintaksu koja opisuje kako će se u prikaz umetnuti dijelovi stranice koji su promjenjivi[6]. Aplikacija se može služiti s više *template*-ova i pri tome svaki *template* može biti zasebni web prikaz ili ugniježđen u web prikaz nekog drugog *template*-a aplikacije.

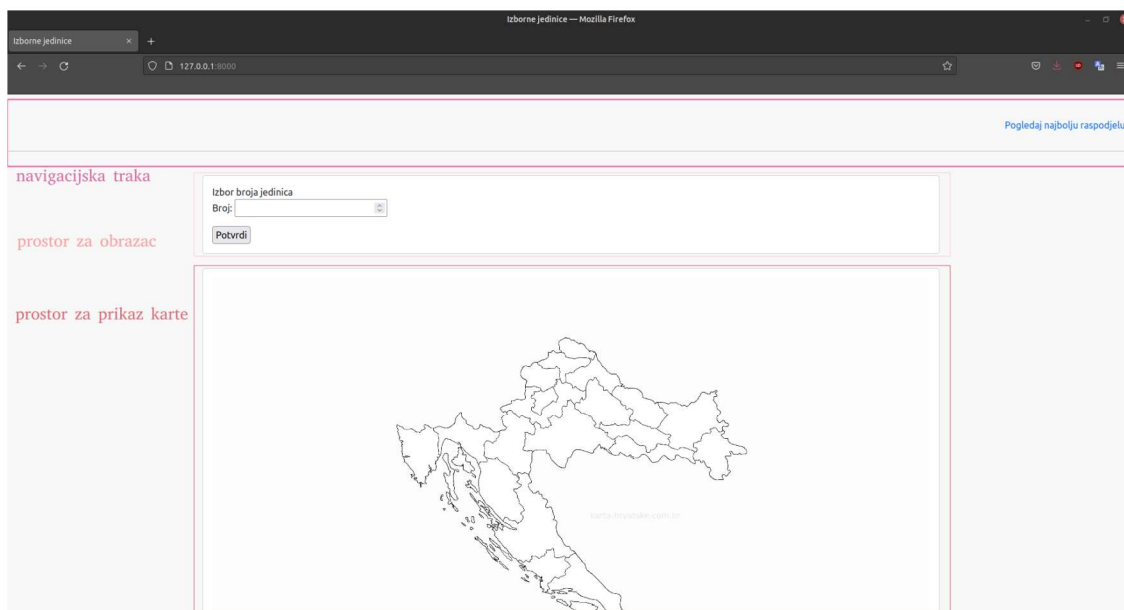
```
1  {% extends 'base/main.html' %}
2  {% load static %}
3
4
5  {% block content %}
6
7
8  <div class="row">
9
10     <div class="col-lg">
11         <div class="row">
12             <div class="col-lg-6">
13
14                 <div class="card card-body">
15                     <canvas id="canvas" width="2000", height="1500"></canvas>
16                 </div>
17             </div>
18             <div class="col-lg-6">
19
20                 <div class="card card-body">
21                     Odaberite županije u {{ pk }}. izbornoj jedinici.
22
23                     <form action="" method="post">
24
25                         {% csrf_token %}
26                         {{ form_zupanije_izborna.as_p }}
27
28                         <input type="submit" name="Submit" value="Potvrđi">
29                         <input type="reset" name="Reset" value="Poništi">
30
31                     </form>
32                 </div>
33             </div>
34         </div>
35     </div>
36 </div>
37
38 </div>
```

Slika 3.5: Primjer Django *template*-a



## 3.2 Korisničko sučelje web aplikacije

Grafičko korisničko sučelje projektne web aplikacije sastoji se od 3 elementa: navigacijske trake koja sadrži poveznicu na prikaz matematički izračunate optimalne raspodjele izbornih jedinica, dijela rezerviranog za prikaz različitih obrazaca ili rezultata konstruirane raspodjele i info-karte Republike Hrvatske s vidljivim granicama županija.

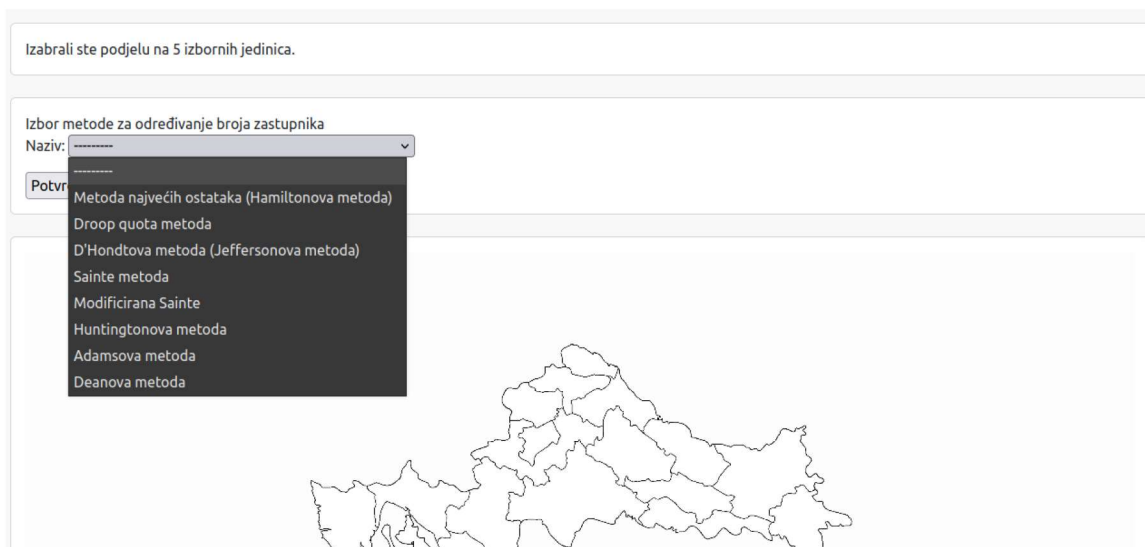


Slika 3.6: Dijelovi u prikazu web stranice

Učitana početna internetska stranica sadrži obrazac za unos željenog ukupnog broja izbornih jedinica. Klikom na gumb za potvrdu broja izbornih jedinica, korisnik je preusmjeren na stranicu s padajućim izbornikom za odabir metode raspodjele broja zastupnika.

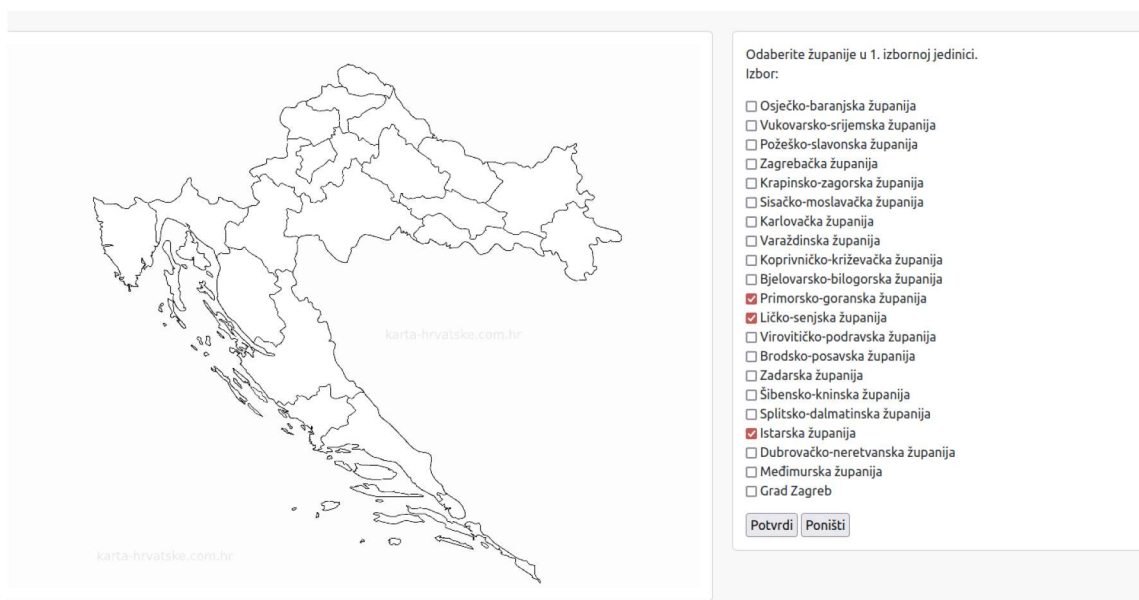
Ponuđeno je osam metoda iz biblioteke *apportionment* koje je moguće koristiti za određivanje broja zastupnika po izornoj jedinici. S obzirom na to da se u hrvatskom izbornom sustavu dosada koristila D'Hondtova metoda, preporučuje se koristiti D'Hondtovu metodu (također se naziva i Jeffersonova metoda)[2]. Ukupni broj zastupnika nije moguće mijenjati od strane korisnika, nego je unutar programa postavljen na 140.





Slika 3.7: Web prikaz pri izboru metode raspodjele zastupnika

Kada korisnik potvrdi izbor metode, preusmjeren je na stranicu koja omogućava izbor županija koje će se naći u I. izornoj jedinici. Na lijevoj polovici zaslona nalazi se info karta s vidljivim granicama županija, a desno je prikazan potvrdni okvir s višestrukim odabirom. Potrebno je odabrati barem jednu županiju za trenutnu izbornu jedinicu da bi se moglo nastaviti klikom na gumb *Potvrdi*. Obrazac se može očistiti klikom na gumb *Poništi*.

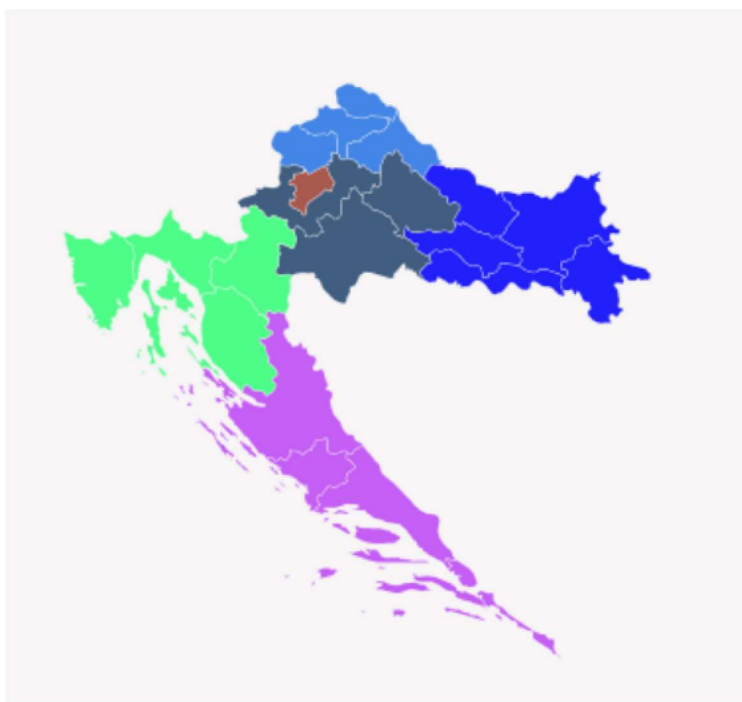


Slika 3.8: Web prikaz izbora županija u izbornu jedinicu

Korisnika se nakon potvrde preusmjeri na izbor županija za iduću izbornu jedinicu i postupak se ponavlja dok se ne popuni i posljednja izborna jedinica, to jest,

postupak se ponavlja onoliko puta koliki je unesen broj izbornih jedinica u prvom koraku. Korisnik može prepravljati izborne jedinice na način da se vrati po jedan korak unazad, odnosno na prethodnu izbornu jedinicu pomoću strelice.

Na ovaj način korisnik samostalno raspoređuje županije po izbornim jedinicama prema proizvoljnom kriteriju bez ikakvih restrikcija. Potvrdom svog izbora na obrascu posljednje izborne jedinice prikazat će se web stranica rezultata. Korisnički stvorena raspodjela izbornih jedinica vidljiva je na karti Republike Hrvatske koja se nalazi na lijevoj polovici stranice. Svaka županija obojena je bojom svoje izborne jedinice.



Slika 3.9: Karta izbornih jedinica u web prikazu rezultata

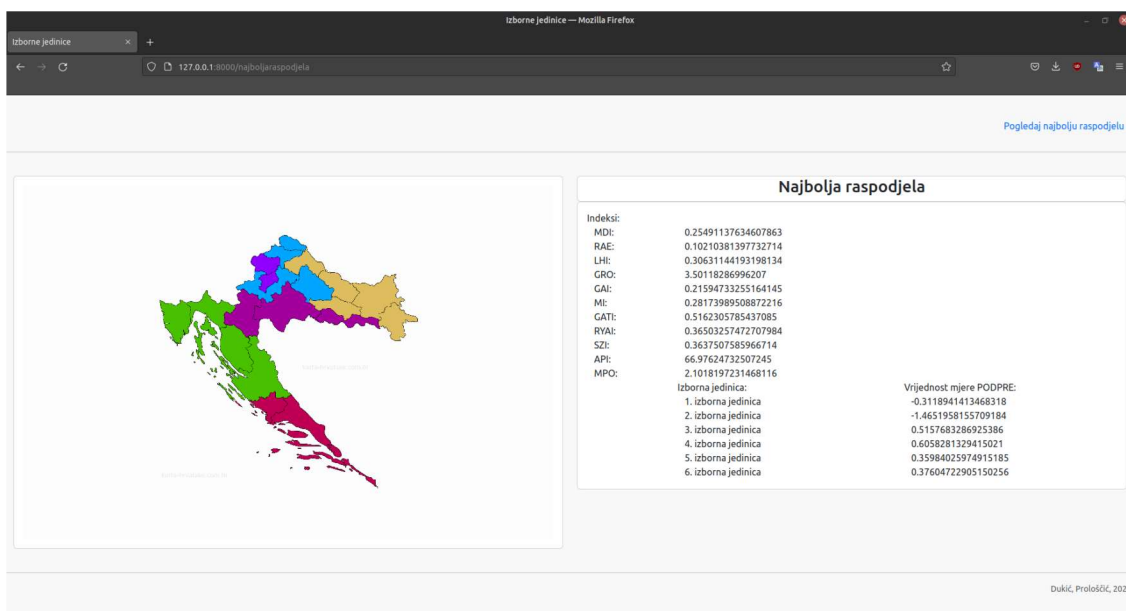
Na drugoj polovici web stranice ispisane su vrijednosti indeksa koji mjere kvalitetu raspodjele te mjera podzastupljenosti/prezastupljenosti za svaku izbornu jedinicu.

[Pogledaj najbolju raspodjelu](#)

<b>Indeksi:</b>	
MDI:	0.38470250926923577
RAE:	0.2752195011248726
LHI:	0.8256585033746177
GRO:	9.630292506019842
GAI:	0.5185114835787762
MI:	0.6774973245235931
GATI:	1.250387433805136
RYAI:	0.8841919943929707
SZI:	0.9956650989049428
API:	50.72442784312137
MPO:	5.016271276547038
<b>Izborna jedinica:</b>	<b>Vrijednost mjere podzastupljenosti/prezastupljenosti:</b>
1. izborna jedinica	-0.3668187522214512
2. izborna jedinica	-2.7565125824971886
3. izborna jedinica	-3.0177331210143565
4. izborna jedinica	1.141530890189126
5. izborna jedinica	1.3578756927586657
6. izborna jedinica	1.847160475780376

Slika 3.10: Mjere za procjenu kvalitete raspodjele u web prikazu rezultata

Na svakom navedenom web prikazu vidljiva je traka na vrhu stranice koja sadrži prečac na rezultate matematički izračunate optimalne raspodjele izbornih jedinica. Prikaz rezultata isti je kao i za korisnički generiranu raspodjelu.



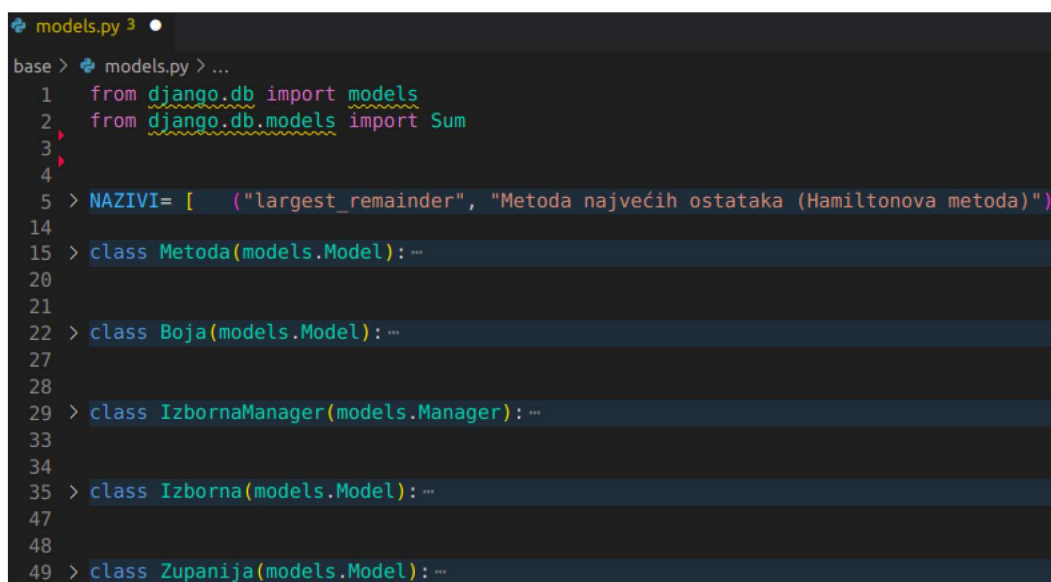
Slika 3.11: Web prikaz rezultata izračunate optimalne raspodjele

## 3.3 Struktura web aplikacije

Web aplikacija za izradu proizvoljne raspodjele izbornih jedinica izrađena je komponiranjem Djangovih *modela*, *view*-ova i *template*-a u funkcionalnu cjelinu. Potrebne komponente definirane su i grupirane u zasebnim datotekama prikladno nazvanima `models.py`, `views.py` te je svaki *template* za renderiranje web stranice zasebna HTML datoteka u mapi `Templates`. Dodatno, uz funkcionalnosti koje proizlaze iz navedenih glavnih komponenti, bilo je potrebno definirati i obrasce (*forms*) koji korisniku služe za unos podataka, to jest, interakciju sa programom.

### 3.3.1 Model-i

Za potrebe aplikacije definirani su modeli *Metoda*, *Boja*, *Izborna* i *Zupanija*.



```
models.py 3
base > models.py > ...
1  from django.db import models
2  from django.db.models import Sum
3
4
5  > NAZIVI= [ ("largest_remainder", "Metoda najvećih ostataka (Hamiltonova metoda)"),
14
15 > class Metoda(models.Model): ...
20
21
22 > class Boja(models.Model): ...
27
28
29 > class IzbornaManager(models.Manager): ...
33
34
35 > class Izborna(models.Model): ...
47
48
49 > class Zupanija(models.Model): ...
```

Slika 3.12: Model-i u datoteci `models.py`

Klasa *Metoda* predstavlja *model* s podacima o metodama raspodjele zastupnika po izbornim jedinicama. Ima samo jedan atribut *naziv* koji može biti *string* vrijednost iz liste `NAZIVI` definirane u kodu. Lista sadrži uređene parove kojima je prvi element naziv metode raspodjele zastupnika u Python biblioteci *apportionment*, a drugi je element naziv te metode na hrvatskom jeziku. Biblioteka *apportionment* ima implementirane funkcije različitih metoda za računanje odgovarajućeg broja zastupnika u izbornim jedinicama s obzirom na broj stanovnika po izornoj jedinici.

Klasa *Boja* definira *model* za boje koje reprezentiraju izborne jedinice na informativnoj karti. Sadrži samo atribut *vrijednost* koji pomoću podatka tipa *string* opisuje boju u heksadekadskom formatu za zapis boje u računalu (npr. `#FFFFFF`).



```

5 NAZIVI= [ ("largest_remainder", "Metoda najvećih ostataka (Hamiltonova metoda)" ),
6           ("quota", "Droop quota metoda"),
7           ("dhondt", "D'Hondtova metoda (Jeffersonova metoda)" ),
8           ("saintelague", "Sainte metoda"),
9           ("modified_saintelague", "Modificirana Sainte"),
10          ("huntington", "Huntingtonova metoda"),
11          ("adams", "Adamsova metoda"),
12          ("dean", "Deanova metoda")
13        ]
14
15 class Metoda(models.Model):
16     naziv = models.CharField(max_length=100, choices=NAZIVI)
17
18     def __str__(self):
19         return self.naziv
20
21

```

Slika 3.13: Model Metoda

```

22 class Boja(models.Model):
23     vrijednost = models.CharField(max_length=7, default="#FFFFFF", null=True)
24
25     def __str__(self):
26         return self.vrijednost
27

```

Slika 3.14: Model Boja

Klasa *Izborna* definira *model* za reprezentaciju izbornih jedinica. Ona ima attribute *id*, *naziv* i *boja* te metodu koja računa ukupni broj birača u izornoj jedinici. *Id* izborne jedinice je *integer* vrijednost koja je ujedno *primary key* u *model*-u *Izborna*. Sadrži informaciju o rednom broju izborne jedinice koji se koristi za definiranje *naziva*. U *naziv* se sprema rimski redni broj i pri dohvaćanju izborne jedinice ona se prikazuje kao {*naziv*}. *izborna jedinica* (npr. IV. izborna jedinica). *Model Izborna* povezan je jedan-na-jedan vezom s *model*-om *Boja*. Dakle, svakoj izornoj jedinici pridružena je točno jedna boja i njome će izborna jedinica biti obojena na karti uz rezultate raspodjele. Metoda koja računa ukupni broj birača u izornoj jedinici u kodu je nazvana *f\_broj\_biraca*. Potrebna je za računanje raspodjele broja zastupnika po izbornim jedinicama. Metoda *f\_broj\_biraca* dohvaća županije koje pripadaju određenoj izornoj jedinici i zbraja vrijednosti koje su im spremljene u atribut *broj\_biraca*. Vraća zbroj kao podatak tipa *integer*.

```

35 class Izborna(models.Model):
36     id = models.IntegerField(primary_key=True)
37     naziv = models.CharField(max_length=3, default="", null=True)
38     boja = models.OneToOneField(Boja, null=True, on_delete=models.SET_NULL)
39
40     def f_broj_biraca(self):
41         return Zupanija.objects.filter(izborna=self).aggregate(Sum('broj_biraca'))['broj_biraca__sum']
42
43     def __str__(self):
44         return self.naziv + " izborna jedinica"
45
46     objects = IzbornaManager()

```

Slika 3.15: Model Izborna

Županije su reprezentirane *model*-om *Zupanija* koji ima sljedeće atribute: *broj\_zupanije*, *naziv*, *slika*, *izborna*, *broj\_biraca*, *boja*. Atributi *broj\_zupanije*, *izborna* i *broj\_biraca* predstavljaju podatke tipa *integer*. Atributi *naziv* i *boja* podaci su tipa *string*. Atribut *slika* posebnog je tipa *ImageField* koji služi za učitavanje slika.

```
class Zupanija(models.Model):
    broj_zupanije = models.IntegerField(null=True)
    naziv = models.CharField(max_length=50, default="županija", null=True)
    slika = models.ImageField(upload_to="", null=True)
    izborna = models.ForeignKey(Izborna, null=True, on_delete=models.SET_NULL, blank=True)
    broj_biraca = models.IntegerField(null=True, blank=True)
    boja = models.CharField(max_length=7, blank=True, null=True)

    def __str__(self):
        return self.naziv
```

Slika 3.16: Model Zupanija

Model *Izborna* povezan je "jedan-na-više" vezom s *model*-om *Zupanija*. Tako atribut *izborna* kod pojedine županije sprema *primary key* izborne jedinice u koju je svrstana. Županija može biti u samo jednoj izbornoj jedinici, dok se u jednoj izbornoj jedinici može naći jedna ili više županija. Također, županija će imati jednaku *boju* kao izborna jedinica kojoj pripada.

### 3.3.2 Obrasci

Dodatna komponenta koja se koristi u radu web aplikacije je Django *form*, tj. obrazac. U sklopu ove web aplikacije definirani su obrasci *BrojIzbornihForm*, *IzborMetode* i *SloziIzbornu*.

```
forms.py 3
base > forms.py > ...
4
5 > class BrojIzbornihForm(forms.Form): ...
7
8 > class IzborMetode(ModelForm): ...
12
13
14 ZUPANIJE = Zupanija.objects.all()
15
16 > class SloziIzbornu(forms.Form): ...
22
```

Slika 3.17: Obrasci u datoteci forms.py

Obrazac *BrojIzbornihForm* postoji kako bi prikupio informaciju o željenom broju izbornih jedinica od korisnika. Ima jedno polje za ispunjavanje u koje se može

upisati cijeli broj iz segmenta [1,21] s obzirom na to da nije moguće imati manje od 1 izborne jedinice i nije moguće imati više izbornih jedinica nego što ima županija.

```
5 class BrojIzbornihForm(forms.Form):
6     broj = forms.IntegerField(max_value=21, min_value=1)
7
```

Slika 3.18: Obrazac za unos broja izbornih jedinica

Obrazac *IzborMetode* prikuplja informaciju o metodi raspodjele zastupnika koju korisnik želi upotrijebiti. Ovaj obrazac ima oblik padajućeg izbornika. Mogući odabiri definirani su unutar modela *Metoda* s kojim je obrazac povezan.

```
8 class IzborMetode(ModelForm):
9     class Meta:
10         model = Metoda
11         fields = ["naziv"]
```

Slika 3.19: Obrazac za izbor metode raspodjele zastupnika

Obrazac *SloziIzbornu* korišten je za spremanje odabira županija za izbornu jedinicu. To je obrazac koji ima oblik potvrdnog okvira s višestrukim izborom. Za označavanje su ponuđene sve županije iz tablice *Zupanija* u bazi podataka. Obrazac zahtijeva odabir barem jedne županije da bi se odgovor spremio.

```
14 ZUPANIJE = Zupanija.objects.all()
15
16 class SloziIzbornu(forms.Form):
17     izbor = forms.ModelMultipleChoiceField(
18         widget=forms.CheckboxSelectMultiple,
19         queryset=ZUPANIJE,
20         error_messages={"required": "Odaberite barem 1 županiju."},
21     )
22
```

Slika 3.20: Obrazac za dodavanje županija u izbornu jedinicu

### 3.3.3 View-ovi

U web aplikaciji za izradu proizvoljne raspodjele izbornih jedinica definirane su *view* funkcije *start()*, *metoda()*, *izbornajedinica()*, *rezultati()* i *najboljaraspodjela()*. Sve primaju argument koji je objekt tipa *HttpRequest* i vraćaju objekt tipa *HttpResponse*. Osim funkcija, u datoteci *views.py* definirana je jedna globalna varijabla *k* koja predstavlja broj izbornih jedinica.



```
views.py 8
base > views.py > ...
1  from django.shortcuts import render
2  from django.http import HttpResponseRedirect
3  from django import forms
4  from .models import *
5  from .forms import *
6  from django.urls import reverse
7
8  from . import funkcionalnosti as f
9  import apportionment.methods as app
10 import numpy as np
11 import pandas as pd
12
13 # globalne
14 global k
15 k = 0 #broj izbornih jedinica
16
17 > def start(request): ...
44
45
46 > def metoda(request): ...
64
65
66 > def izbornajedinica(request, pk): ...
101
102
103 > def rezultati(request): ...
146
147 > def najboljaraspodjela(request): ...
151 |
```

Slika 3.21: View-ovi u datoteci views.py

Funkcija `start()` povezana je s URL uzorkom početne web stranice. Kada se ne događa ništa, obrazac `BrojIzbornihForm` očisti se i funkcija ponovno učitava početnu web stranicu. Kada je obrazac ispunjen i potvrđen, funkcija ga obrađuje prvo provjeravajući je li ispravan. Ako je obrazac ispravan, funkcija sprema obrascem prikupljeni broj izbornih jedinica u globalnu varijablu `k`. Pomoću broja `k` stvara odgovarajući broj boja i izbornih jedinica koje će se dodati u bazu podataka. Boje se generiraju na nasumičan način, a izborne jedinice se definiraju kao I., II., itd. izborna jedinica pri čemu se svakoj izornoj jedinici dodijeli jedinstvena boja. Objekt koji funkcija vraća preusmjerava na web stranicu za izbor metode.



```

25 def start(request):
26
27     if request.method == "POST":
28
29         form_br_jedinica = BrojIzbornihForm(request.POST)
30
31         if form_br_jedinica.is_valid():
32
33             global k
34             k = form_br_jedinica.cleaned_data['broj']
35             Boja.objects.all().delete()
36             generator_boja = ["#" + ''.join([random.choice('0123456789ABCDEF')
37                                     for j in range(6)]) for i in range(k)]
38             Izborna.objects.all().delete()
39
40             for i in range(k):
41
42                 boja = Boja(vrijednost = generator_boja[i])
43                 boja.save()
44                 nova_izborna = Izborna.objects.stvori_izbornu(i+1, str(i+1) + ".", boja)
45                 nova_izborna.save()
46
47             return HttpResponseRedirect(reverse(metoda))
48         else:
49             form_br_jedinica = BrojIzbornihForm()
50
51     context = { 'form_br_jedinica': form_br_jedinica, 'k':k }
52     return render(request, 'base/start.html', context)

```

Slika 3.22: View start()

Funkcija *metoda()* obrađuje zahtjev kada je URL uzorak oblika ".../metoda/". Pri prvom učitavanju, i dok se obrazac za odabir metode raspodjele zastupnika ne ispuni, stranica se osvježava zajedno s obrascem *IzborMetode*. Kada je obrazac potpunjen, brišu se sve instance u tablici *Metoda*. Ako je obrazac ispravan, odgovor se sprema u bazu podataka i funkcija preusmjerava na web stranicu za konstrukciju I. izborne jedinice.

```

55 def metoda(request):
56
57     if request.method == "POST":
58
59         Metoda.objects.all().delete()
60         form_metoda = IzborMetode(request.POST)
61
62         if form_metoda.is_valid():
63
64             metoda = form_metoda.save()
65             return HttpResponseRedirect(reverse("izbornajedinica", args=[1]))
66
67         else:
68             form_metoda = IzborMetode()
69
70
71     context = { 'form_metoda': form_metoda, 'k': k}
72     return render(request, 'base/metoda.html', context)

```

Slika 3.23: View metoda()

Funkcija `izbornajedinica()` pripada URL uzorku oblika `".../izbornajedinica/<int:pk>/"`, pri čemu je `<int:pk>` promjenjivi argument tipa `int` (*integer*). Funkcija `izbornajedinica()` jedini je `view` u sklopu aplikacije koji prima dodatni argument tipa `integer` uz `HttpRequest` objekt. Prvo dohvaća iz baze podataka onu izbornu jedinicu čiji je *primary key* jednak argumentu `pk`. Dok obrascu nije predan nikakav odgovor, stranica se osvježava. Kada obrazac dobije odgovor, u slučaju da je valjan, funkcija dohvaća odabrane županije iz odgovora i ažurira njihove podatke u bazi pomoću iterativnog procesa. U procesu se svakoj županiji mijenjaju atributi *boja* i *izborna*. Na poslijetku, funkcija provjerava ima li još izbornih jedinica za popunjavanje. Ako ima, funkcija vraća objekt koji preusmjerava na web stranicu za konstrukciju iduće izborne jedinice, a ako nema, preusmjeravanje vodi na web stranicu rezultata.

```
75 def izbornajedinica(request, pk):
76
77     izborna = Izborna.objects.get(id=pk)
78
79     if request.method == "POST":
80
81         form_zupanije_izborna = SloziIzbornu(request.POST)
82
83         if form_zupanije_izborna.is_valid():
84
85             zupanije = form_zupanije_izborna.cleaned_data['izbor']
86
87             for zupanija in zupanije:
88                 naziv = zupanija.naziv
89                 broj_zupanije = zupanija.broj_zupanije
90                 broj_biraca = zupanija.broj_biraca
91                 boja = izborna.boja.vrijednost
92                 slika = zupanija.slika
93                 id_item=Zupanija.objects.filter(naziv=naziv).values('id')[0]
94                 nova_zupanija = Zupanija(id=id_item['id'], naziv=naziv,
95                                         izborna=izborna, broj_zupanije=broj_zupanije,
96                                         broj_biraca=broj_biraca, boja=boja, slika=slika)
97                 nova_zupanija.save()
98
99             global k
100            if (pk+1) <= k :
101                return HttpResponseRedirect(reverse("izbornajedinica", args=[pk + 1]))
102            else:
103                return HttpResponseRedirect(reverse("rezultati"))
104
105        else:
106
107            form_zupanije_izborna = SloziIzbornu()
108
109        context = { 'form_zupanije_izborna': form_zupanije_izborna, 'pk': pk}
110
111        return render(request, 'base/izbornajedinica.html', context)
```

Slika 3.24: `View izbornajedinica()`

Funkcija *rezultati()* povezana je s URL uzorkom ".../rezultati/". Nakon inicijaliziranja različitih varijabli, *view* kreira listu s brojem birača po izbornoj jedinici. Da bi se kreirala lista s brojem zastupnika po izbornoj jedinici, koristi se funkcija *compute()* iz biblioteke *apportionment*. Ona uzima za argumente: metodu koju je korisnik odabrao, listu s brojem birača po izbornoj jedinici, ukupan broj zastupnika i listu izbornih jedinica. Lista s brojem zastupnika po izbornoj jedinici koju *compute()* vraća i lista s brojem birača po izbornoj jedinici koriste se za računanje indeksa koji mjere kvalitetu raspodjele izbornih jedinica. Funkcije koje se koriste za računanje indeksa implementirane su u zasebnoj Python datoteci, a u *view*-u *rezultati()* samo se pozivaju. Spremaju se vrijednosti indeksa koje te funkcije vraćaju kako bi se mogle renderirati na web stranici. Dodatno se, u svrhu prikaza podataka, dohvaćaju slike i boje svih županija. Objekt koji vraća funkcija *rezultati()* usmjerava na web stranicu s prikazom rezultata korisnički generirane raspodjele izbornih jedinica.



```

114 def rezultati(request):
115     k = Izborna.objects.count()
116     rad2 = pd.read_excel("staticfiles/media/rad2.xlsx")
117     # ukupan broj punoljetnih građana (birača u zemlji)
118     Q = rad2["Punoljetni državljani"][21]
119     # ukupan broj zastupnika
120     S = 140
121
122     # broj birača u i-toj izbornoj jedinici odgovara broju na i-tom indeksu
123     Qi = []
124     # dodajemo podatke u Qi
125     for i in range(k):
126         izborna = Izborna.objects.get(id=(i+1))
127         broj_biraca = izborna.f_broj_biraca()
128         Qi.append(broj_biraca)
129
130     # broj zastupnika u i-toj izbornoj jedinici odgovara broju na i-tom indeksu
131     Si = []
132     # dodajemo podatke u Si
133     metoda = Metoda.objects.first().naziv
134     Si = app.compute(metoda, Qi, S, list(range(1,k)))
135
136     funkcije_indeksi = {'MDI':f.MDI, 'RAE':f.RAE, 'LHI':f.LHI, 'GRO':f.GRO,
137                       'GAI':f.GAI, 'MI':f.MI, 'GATI':f.GATI, 'RYAI':f.RYAI,
138                       'SZI':f.SZI, 'API':f.API, 'MPO':f.MPO}
139     nazivi_indeksi = ['MDI', 'RAE', 'LHI', 'GRO', 'GAI', 'MI', 'GATI',
140                     'RYAI', 'SZI', 'API', 'MPO']
141     indeksi = []
142
143     Qi = np.array(Qi)
144     Si = np.array(Si)
145
146     for i in nazivi_indeksi:
147         indeksi.append(funkcije_indeksi[i](Qi, Si))
148
149     podpre = f.PODPRE(Qi, Si, Q, S)
150
151     boje=[]
152     zupanija_objects = Zupanija.objects.all()
153     image_sources=[]
154     for zupanija in zupanija_objects:
155         image_sources.append(zupanija.slika.url)
156         boje.append(zupanija.boja)
157
158     context = {'Qi': Qi, 'nazivi_indeksi': nazivi_indeksi, 'indeksi':indeksi,
159              'podpre':podpre, 'boje':boje, 'image_sources':image_sources,
160              'range':range(1, 1+k), }
161     return render(request, 'base/info.html', context)

```

Slika 3.25: *View rezultati()*

Funkcija *najboljaraspodjela()* povezana je s URL uzorkom oblika ".../najboljaraspodjela". Ovaj *view* sprema rezultate dobivene metodom spektralnog klasteriranja tako da ih se može upotrijebiti u prikazu web stranice. Funkcija za računanje optimalne raspodjele izbornih jedinica, koja se ovdje poziva, implementirana je u zasebnoj datoteci. Objekt koji vraća funkcija *najboljaraspodjela()* usmjerava na web stranicu s prikazom rezultata matematički izračunate najbolje raspodjele izbornih jedinica.

```
147 def najboljaraspodjela(request):
148     indeksi, podpre = f.najboljaraspodjelarender()
149     context = {'nazivi_indeksi': indeksi.keys, 'indeksi': indeksi.values, 'range': range(1,7), 'podpre': podpre, }
150     return render(request, 'base/informativna.html', context)
151
```

Slika 3.26: *View najboljaraspodjela()*

### 3.3.4 *Template-i*

U aplikaciji je definirano nekoliko *template-a*: main.html, start.html, metoda.html, izbornajedinica.html, info.html i informativna.html.

*Template* main.html služi za stvaranje HTML koda koji određuje općeniti izgled web prikaza. U body dijelu dokumenta definirana je izbornička traka u kojoj se nalazi poveznica na pregled najbolje raspodjele izbornih jedinica, poseban blok *content* za umetanje sadržaja u web prikaz na Django specifičan način i zaglavlje web stranice.

```
<body>
  <br>
  <div class="container-fluid navbar justify-content-end">
    <ul class="nav nav-pills">
      <li class="nav-item">
        <a class="nav-link btn btn-light" href="{% url 'najboljaraspodjela' %}">Pogledaj najbolju raspodjelu</a>
      </li>
    </ul>
  </div>
  <hr>
  <br>
  <div class="container-fluid">
    {% block content %}
    {% endblock %}
  </div>
  <br>
  <hr>
  <div class="card-footer text-muted float-end" style="font-size: 0.875em;">Đukić, Prološćić, 2023.</div>
  <br>
</body>
```

Slika 3.27: *Template* main.html

*Template* start.html nadovezuje se na main.html *template*, odnosno preuzima sve što se nalazi u kodu main.html datoteke i nadodaje sadržaj koji je poseban za start.html web prikaz. U ovoj datoteci definiran je sadržaj bloka *content*: renderirat će se obrazac za određivanje broja izbornih jedinica i informativna karta s prikazom županija u Republici Hrvatskoj.

```
1  {% extends 'base/main.html' %}
2  {% load static %}
3
4
5  {% block content %}
6  <div class="row">
7      <div class="col-lg-2"></div>
8      <div class="col-lg-8">
9          <div class="card card-body">
10             Izbor broja jedinica
11             <form action="" method="POST">
12                 {% csrf_token %}
13                 {{ form_br_jedinica.as_p }}
14                 <input type="submit" name="Submit" value="Potvrđi">
15             </form>
16         </div>
17         <br>
18         <div class="card card-body">
19             <canvas id="canvas" width="3000", height="2000">
20             </canvas>
21         </div>
22     </div>
23     <div class="col-lg-2"></div>
24 </div>
25
26 <script>
27     let canvas = document.getElementById("canvas");
28     let ctx = canvas.getContext("2d");
29     let image = new Image();
30     image.src = "{% static 'media/slijepa_karta.png' %}";
31     image.classList.add("img-fluid")
32     image.onload = () => {
33         ctx.drawImage(image, 0, -30, 3000, 2000);
34     }
35 </script>
```

Slika 3.28: *Template* start.html

Datoteka metoda.html također je *template* koji preuzima HTML definiran u *template*-u main.html te definira što će se naći u bloku content. Dakle, web prikaz *template*-a metoda.html sastoji se od izborničke trake s poveznicom, informacije o odabranom broju izbornih jedinica, obrasca za izbor metode raspodjele zastupnika, informativne karte s prikazom županija te od zaglavlja.

```
1  {% extends 'base/main.html' %}
2  {% load static %}
3
4
5  {% block content %}
6
7  <div class="row">
8      <div class="col-lg-2"></div>
9      <div class="col-lg-8">
10         <div class="card card-body">
11             Izabrali ste podjelu na {{ k }} izbornih jedinica.
12         </div>
13         <div><br></div>
14         <div class="card card-body">
15             Izbor metode za određivanje broja zastupnika
16             <form action="" method="post">
17                 {% csrf_token %}
18                 {{ form_metoda.as_p }}
19                 <input type="submit" name="Submit" value="Potvrđi">
20             </form>
21         </div>
22         <div><br></div>
23         <div class="card card-body">
24
25             <canvas id="canvas" width="3000", height="2000">
26             </canvas>
27
28         </div>
29     </div>
30     <div class="col-lg-2"></div>
31 </div>
```

Slika 3.29: *Template metoda.html*

Datoteka *izbornajedinica.html* još je jedan *template* ugniježđen u *main.html*. Sadržaj web prikaza koji je poseban za ovaj *template* definiran je u bloku *content* i renderira informativnu kartu na lijevoj polovici zaslona uz obrazac za odabir županija u izbornoj jedinici na desnoj polovici zaslona.



```
1  {% extends 'base/main.html' %}
2  {% load static %}
3
4
5  {% block content %}
6
7
8  <div class="row">
9
10     <div class="col-lg">
11         <div class="row">
12             <div class="col-lg-6">
13
14                 <div class="card card-body">
15                     <canvas id="canvas" width="2000", height="1500"></canvas>
16                 </div>
17             </div>
18             <div class="col-lg-6">
19
20                 <div class="card card-body">
21                     Odaberite županije u {{ pk }}. izbornoj jedinici.
22
23                     <form action="" method="post">
24
25                         {% csrf_token %}
26                         {{ form_zupanije_izborna.as_p }}
27
28                         <input type="submit" name="Submit" value="Potvrđi">
29                         <input type="reset" name="Reset" value="Poništi">
30
31                     </form>
32                 </div>
33             </div>
34         </div>
35     </div>
36 </div>
37
38 </div>
```

Slika 3.30: *Template* izbornajedinica.html

*Template* info.html služi za prikaz sadržaja kada se blok *content* ugniježđen u *template* main.html treba renderirati kao karta Republike Hrvatske podijeljena na odabrane izborne jedinice uz vrijednosti indeksa koji mjere kvalitetu odabrane raspodjele.



```

1  {% extends 'base/main.html' %}
2  {% load static %}
3
4
5  {% block content %}
6
7
8  <div class="row">
9
10     <div class="col-lg">
11         <div class="row">
12             <div class="col-lg-6">
13
14                 <div class="card card-body">
15                     <canvas id="canvas" width="612", height="408"></canvas>
16                 </div>
17
18             </div>
19             <div class="col-lg-6">
20
21                 <div class="card card-body">
22                     Indeksi:
23                     <div class="row">
24                         <div class="col-2">
25                             {% for i in nazivi_indeksi %}
26                                 <div class="container">{{i}}: </div>
27                             {% endfor %}
28                         </div>
29                         <div class="col">
30                             {% for i in indeksi %}
31                                 <div class="container">{{i}} </div>
32                             {% endfor %}
33                         </div>
34                         <div class="row">
35                             <div class="col">
36                                 Izborna jedinica:
37                                 {% for i in range %}
38                                     <div class="container">{{i}}. izborna jedinica</div>
39                                 {% endfor %}
40                             </div>
41                             <div class="col">
42                                 Vrijednost mjere podzastupljenosti/prezastupljenosti:
43                                 {% for i in podpre %}
44                                     <div class="container">{{i}} </div>
45                                 {% endfor %}

```

Slika 3.31: *Template info.html*

```

46     </div>
47 </div>
48
49 </div>
50
51 </div>
52 </div>
53 </div>
54
55 </div>
56
57 <script src="{% static 'scripts/help_script_boje.js' %}" defer ></script>
58 {{ boje|json_script:"boje_data" }}
59 {{ image_sources|json_script:"image_sources_data" }}
60
61 {% endblock %}

```

Slika 3.32: *Template info.html*

*Template* informativna.html na mjestu bloka *content* u *template*-u main.html renderira kartu s prikazom najbolje raspodjele izbornih jedinica prema rezultatima koje daje metoda spektralnog klasteriranja uz dobivene vrijednosti indeksa kvalitete za tu raspodjelu.

```
1  {% extends 'base/main.html' %}
2  {% load static %}
3
4
5  {% block content %}
6
7
8  <div class="row">
9
10     <div class="col-lg">
11         <div class="row">
12             <div class="col-lg-6">
13
14                 <div class="card card-body">
15                     <canvas id="canvas" width="3000", height="2000"></canvas>
16                 </div>
17             </div>
18             <div class="col-lg-6">
19                 <div class="card">
20                     <h3 class="text-center">Najbolja raspodjela</h3>
21                 </div>
22                 <div class="card card-body">
23                     Indeksi:
24                     <div class="row">
25                         <div class="col-2">
26                             {% for i in nazivi_indeksi %}
27                                 <div class="container">{{i}}: </div>
28                             {% endfor %}
29                         </div>
30                         <div class="col">
31                             {% for i in indeksi %}
32                                 <div class="container">{{i}} </div>
33                             {% endfor %}
34                         <div class="row">
35                             <div class="col">
36                                 Izborna jedinica:
37                                 {% for i in range %}
38                                     <div class="container">{{i}}. izborna jedinica</div>
39                                 {% endfor %}
40                             </div>
41                         </div>
42                     </div>

```

Slika 3.33: *Template* informativna.html

```
35     {% endfor %}
36     <div class="row">
37         <div class="col">
38             Izborna jedinica:
39             {% for i in range %}
40             <div class="container">{{i}}. izborna jedinica</div>
41             {% endfor %}
42         </div>
43         <div class="col">
44             Vrijednost mjere PODPRE:
45             {% for i in podpre %}
46             <div class="container">{{i}} </div>
47             {% endfor %}
48         </div>
49     </div>
50
51 </div>
52 </div>
53 </div>
54 </div>
55 </div>
56 </div>
57
58 </div>
59
60 <script>
61     let canvas = document.getElementById("canvas");
62     let ctx = canvas.getContext("2d");
63     let image = new Image();
64     image.src = "{% static 'media/najbolja_karta.png' %}";
65     image.classList.add("img-fluid")
66     image.onload = () => {
67         ctx.drawImage(image, 0, 0, 3000, 2000);
68     }
69 </script>
70
71 {% endblock %}
```

Slika 3.34: *Template* informativna.html

## 4 | Mjere ujednačenosti težina biračkih glasova

Pretpostavimo da je odabrana podjela Republike Hrvatske na  $k$  izbornih jedinica. Za definiranje indeksa koji mjere ujednačenost težine biračkih glasova potrebno je uvesti oznake:

- $Q$  - ukupni broj birača u državi
- $k$  - broj izbornih jedinica u državi
- $S$  - ukupni broj zastupnika
- $Q_i$  - broj birača u  $i$ -toj izornoj jedinici,  $i = 1, \dots, k$
- $S_i$  - broj zastupnika u  $i$ -toj izornoj jedinici,  $i = 1, \dots, k$ .

Za uvedene oznake vrijede sljedeće jednakosti:

$$\sum_{i=1}^k S_i = S, \sum_{i=1}^k Q_i = Q.$$

U izbornim jedinicama na teritoriju Republike Hrvatske bira se ukupno 140 zastupnika za Hrvatski sabor. Stoga je

$$\sum_{i=1}^k S_i = S = 140.$$

Opća težina biračkog glasa računa se tako da se ukupan broj zastupnika podijeli ukupnim brojem birača:  $\frac{S}{Q}$ , a težina biračkog glasa u  $i$ -toj izornoj jedinici računa se kao  $\frac{S_i}{Q_i}$ ,  $i = 1, \dots, k$ . Kada bi svaki birački glas u državi imao jednaku težinu, vrijedilo bi:

$$\frac{S_i}{Q_i} = \frac{S}{Q}, i = 1, \dots, k.$$

Možemo označiti:

- $w = \frac{S}{Q}$  - opća težina biračkog glasa
- $w_i = \frac{S_i}{Q_i}$  - težina biračkog glasa u  $i$ -toj izornoj jedinici,  $i = 1, \dots, k$

- $q_i = \frac{Q_i}{Q}$  - udio broja birača  $i$ -te izborne jedinice u ukupnom broju birača
- $s_i = \frac{S_i}{S}$  - udio broja zastupnika  $i$ -te izborne jedinice u ukupnom broju zastupnika.

Tada je uvjet opće jednakosti težine biračkog glasa zadovoljen ako je:

$$w_i = w, \forall i = 1, \dots, k.$$

Prema [11], može se uočiti da uvjet vrijedi ako je cijela zemlja jedna izborna jedinica ( $k = 1$ ), dok općenito (za proizvoljni  $k$ ) taj uvjet neće biti ispunjen. Drugim riječima, nemoguće je postići da birači iz različitih izbornih jedinica imaju u potpunosti jednaku težinu biračkog glasa. Stoga se za odabir adekvatne raspodjele izbornih jedinica mogu izračunati različiti indeksi koji na neki način mjere ujednačenost težine biračkog glasa i odabirati ona raspodjela izbornih jedinica koja ima najbolje moguće vrijednosti za sve indekse. Indekse koji su korišteni u okviru projekta moguće je definirati sljedećim matematičkim formulama:

1. Indeks maksimalne devijacije:

$$MDI = 100 \max_{i=1, \dots, k} |s_i - q_i|.$$

2. Raeov indeks:

$$RAE = \frac{100}{k} \sum_{i=1}^k |s_i - q_i|.$$

3. Loosemore-Hanbyjev indeks:

$$LHI = 50 \sum_{i=1}^k |s_i - q_i|.$$

4. Grofmanov indeks:

$$GRO = \frac{100}{\sum_{i=1}^k q_i^2} \sum_{i=1}^k |s_i - q_i|.$$

5. Gallagherov indeks:

$$GAI = 100 \sqrt{\frac{1}{2} \sum_{i=1}^k (s_i - q_i)^2}.$$

6. Monroeov indeks:

$$MI = 100 \sqrt{\frac{\sum_{i=1}^k (s_i - q_i)^2}{1 + \sum_{i=1}^k q_i^2}}.$$



7. Gatev indeks:

$$GATI = 100 \sqrt{\frac{\sum_{i=1}^k (s_i - q_i)^2}{\sum_{i=1}^k (s_i^2 + q_i^2)}}.$$

8. Ryabtsev indeks:

$$RYAI = 100 \sqrt{\frac{\sum_{i=1}^k (s_i - q_i)^2}{\sum_{i=1}^k (s_i + q_i)^2}}.$$

9. Szalaiev indeks:

$$SZI = 100 \sqrt{\frac{\sum_{i=1}^k \left(\frac{s_i - q_i}{s_i + q_i}\right)^2}{k}}.$$

10. Aleskerov-Platonov indeks:

$$API = \frac{100}{k} \sum_{i \in I_0} \frac{s_i}{q_i}, \quad I_0 = \{i : w_i > w\}.$$

Manja vrijednost indeksa znači da su ujednačenije težine biračkih glasova [11]. Navedeni indeksi preuzeti su iz literature (Gallagher, Mitchell [3], Gwizdalla [5], Karpov [9] i Sabo, Scitovski [11]), pri čemu valja napomenuti da su indeksi iz tehničkih razloga pomnoženi sa 100.

Neadekvatna rješenja moguće je eliminirati pomoću indeksa maksimalnog postotnog odstupanja težina biračkih glasova:

$$MPO = 100 \frac{\max_{i=1, \dots, k} w_i - \min_{i=1, \dots, k} w_i}{\min_{i=1, \dots, k} w_i}.$$

*MPO* predstavlja relativnu razliku između najveće i najmanje težine biračkog glasa. Nedovoljno dobre raspodjele izbornih jedinica su one raspodjele za koje je vrijednost indeksa *MPO* veća od zakonski propisanog raspona tolerancije na nejednakost u težini biračkog glasa (izraženog u postocima).

Za kreiranu raspodjelu izbornih jedinica izražene su i vrijednosti mjera podzastupljenosti/prezastupljenosti za svaku izbornu jedinicu:

$$PODPRE_i = 100 \frac{w_i - w}{w}, i = 1, \dots, k.$$

Podzastupljenost izborne jedinice znači da je težina glasa birača iz te izborne jedinice manja od opće težine biračkog glasa. S druge strane, izborna jedinica prezastupljena je kada birač u toj izornoj jedinici ima veću težinu biračkog glasa nego što je opća težina biračkog glasa. Pomoću ovih mjera jasno se vidi koje je izborne jedinice potrebno pokušati prekrojiti kako bi se izjednačila težina biračkog glasa. Idealna vrijednost indeksa *PODPRE* iznosi 0.

## 5 | Zaključak

Završni rad na temu "Problem izbornih jedinica" objašnjava problem raspodjele izbornih jedinica u Republici Hrvatskoj na način koji osigurava jednakost težine biračkog glasa. Temelji se na izrađenom projektu koji obuhvaća matematički model određivanja izbornih jedinica te izradu grafičkog korisničkog sučelja za raspodjelu izbornih jedinica i mjerenje kvalitete modela. Objašnjeni su osnovni politički pojmovi vezani uz izbore i izborni sustav Republike Hrvatske s naglaskom na problem neravnomjerne vrijednosti glasova u različitim izbornim jedinicama. Detaljno je opisana izrada interaktivne web aplikacije za slaganje izbornih jedinica koja kao rezultat prikazuje kvalitetu raspodjele. Aplikacija je izrađena u Pythonu koristeći Django web framework. Opisana je njena struktura koju čine *model*-i, obrasci, *view*-ovi i *template*-i. Objašnjen je način na koji web aplikacija funkcionira, tj. opisan je način na koji kreirano grafičko korisničko sučelje omogućava korisnicima da sami kreiraju raspodjelu izbornih jedinica. Definirani su indeksi koji se koriste za mjerenje ujednačenosti težine biračkih glasova. Ti indeksi pomažu u ocjenjivanju kvalitete različitih raspodjela izbornih jedinica. Koristeći izrađeni program, moguće je pronaći dovoljno dobro ili optimalno rješenje za raspodjelu izbornih jedinica, točnije, raspodjelu koja umanjuje nejednakost u težini biračkih glasova do prihvatljive razine. Rad može dati doprinos u području analize izbornih sustava i pruža alat koji može pomoći u poboljšanju pravednosti izbora u Hrvatskoj.

# Literatura

- [1] Čular, G. (2018). Metodološki izazovi ustavnog sudovanja: učinci podjele na izborne jedinice na rezultate izbora u Hrvatskoj 2000–2016. *Anali Hrvatskog politološkog društva*, 15(1), 7–28, <https://doi.org/10.20901/an.15.01>
- [2] Gallagher, M. (1992). Comparing proportional representation electoral systems: Quotas and thresholds and paradoxes and majorities. *British Journal of Political Science*, 22(4), 469–496, <https://doi.org/10.1017/S0007123400006499>
- [3] Gallagher, M., & Mitchell, R. (2008). *The politics of electoral systems*. Oxford & New York: Oxford University Press.
- [4] Grofman, B., & Hanley, L. (2008). *Redistricting in comparative perspective*. Oxford & New York: Oxford University Press, <https://doi.org/10.1093/oso/9780199227402.001.0001>
- [5] Gwizdalla, T. M. (2008). Gallagher index for sociophysical models. *Physica A: Statistical Mechanics and its Applications*, 387(12), 2937–2951, <https://doi.org/10.1016/j.physa.2008.01.02>
- [6] <https://docs.djangoproject.com/en/5.1/intro/overview/>
- [7] <https://enciklopedija.hr/clanak/izbori>
- [8] <https://www.sabor.hr/hr/zastupnici/izborne-jedinice>
- [9] Karpov, A. (2008). Measurement of disproportionality in proportional representation systems. *Mathematical and Computer Modelling*, 48(9-10), 1421–1438. <https://doi.org/10.1016/j.mcm.2008.05.027>
- [10] Kasapović, M. (2017). Jesu li izborni sustavi sredstva dramatična utjecaja na sudbine zemalja? *Političke analize*, 8(32), 17–21.
- [11] Sabo, K. i Scitovski, R. (2023). Nova metoda za definiranje izbornih jedinica u Hrvatskoj. *Hrvatska i komparativna javna uprava*, 23 (4), 645-672. <https://doi.org/10.31297/hkju.23.4.3>