

Sveučilište u Rijeci, Odjel za Informatiku

Drugi projektni zadatak – analiza mreže

Seminarski rad

Studentica: Nola Čumlievski

Kolegij: Upravljanje znanjem

Rijeka, 2021.

Sadržaj

1. Opis skupa podataka.....	1
2. Učitavanje podataka u Python i izrada mreže.....	2
3. Globalne mjere.....	5
4. Lokalne mjere	9
5. Dodatne mjere centralnosti i Spearman korelacija	18
6. Zaključak.....	25

1. Opis skupa podataka

Kao skup podataka za ovaj projekt odabran je „fb-pages-tvshow“ unutar kategorije „Društvene mreže“. Skup podataka predstavlja prikupljenje podatke sa društvene mreže Facebook (iz 2017.godine). Čvorovi predstavljaju stranice tv serija, a veze (rubovi) predstavljaju zajedničke lajkove među navedenim serijama. Mreža je **neusmjerena**, s obzirom da jedna veza između dvije serije znači da je korisnik lajkao obe stranice te ne postoji usmjerenje između čvorova. Također, veze nemaju pridružene težine.

Zip arhiva skupa podataka se sastoji od dvije datoteke ekstenzije „nodes“ i „edges“. Elementi datoteka su odvojeni zarezom, te sam radi pogodnosti rada s Python bibliotekom „Networkx“ konvertala datoteke u datoteke ekstenzije „csv“ s obzirom da su elementi unutar originalnih datoteka također odvojeni zarezom.

	A	B	C	D	E
1	id,name,new_id				
2	183215455178728,Orange is the New Black,1852				
3	282808121926963,ç:®åŸæœ%œç! Hello Singapore,592				
4	179653538737148,DC Cupcakes,1370				
5	162816367111296,Gladiatorerna,1263				
6	134115896703387,Globo Rural,3042				
7	154734594579016,Million Dollar Listing,3131				
8	54035698285,Taking the Stage,2144				
9	229468073750527,WDRB in the Morning,3350				
10	174310142817,Australian Story,2280				
11	640672685943394,Medical Center,1190				
12	114060455288958,L'Amour est aveugle,49				

Slika 1. Datoteka čvorova

Datoteka s čvorovima sadrži 3 stupca. Prvi je id serije, drugi predstavlja ime čvora (naziv serije) te treći sadrži novi id koji predstavlja id čvora.

	A	B
1	Source,Target	
2	0,1838	
3	0,1744	
4	0,14	
5	0,2543	
6	1,1009	
7	1,1171	
8	1,1465	
9	1,2069	
10	1,208	
11	1,1856	
12	1,3799	

Slika 2. Datoteka rubova

Datoteka s rubovima sastoji se od dva stupca, od kojih prvi predstavlja prvi čvor, a drugi predstavlja drugi čvor. Čvorovi unutar jednog reda smatraju se povezanim, tj. korisnici su lajkali obe stranice tv serije.

2. Učitavanje podataka u Python i izrada mreže

```
import networkx as nx
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from networkx.algorithms import community
import matplotlib.colors as mcolors
```

Za izradu ovog projekta, instalirane su i učitane slijedeće biblioteke: *networkx*, *pandas*, *numpy*, *matplotlib.pyplot*, *seaborn*, modul *community* koji predstavlja dio biblioteke *networkx.algorithms* te *matplotlib.colors*.

Za početak, izrađena je funkcija pomoću koje ćemo naslove i podatke navedenih dviju datoteka spremiti odvojeno u varijable u obliku liste stringova.

```
def get_data(file):
    #upotreba context managera za učitavanje podataka"
    with open(file, "r", encoding = "utf8") as line:
        lines = line.read().split("\n") #transformacija file-a u string
        data = [line.split(",") for line in lines] #separacija
        header = data[0] #header je prvi red
        data = data[1:] #podaci su svi ostali redovi
    return header, data
```

Parametar file uzima path do datoteke koju želimo učitati. Datoteka se otvara pomoću funkcije *open*, čita se red po red te se podaci dijele ovisno o položaju zareza. Naslov označava prvi red datoteke, dok su podaci svi redovi s početkom od reda s indeksom jedan pa na dalje. Funkcija vraća naslov i podatke.

```
node_header, node_data = get_data("C:/Users/38591/Desktop/Faks/UZ/nodes.csv")
edge_header, edge_data = get_data("C:/Users/38591/Desktop/Faks/UZ/edges.csv")
```

Zatim, učitavamo naslove i datoteke čvorova i rubova te u funkciju prenosimo file path gdje se datoteka nalazi u sustavu.

```
edge_header = [y.strip(' ') for y in edge_header]
node_header = [y.strip(' ') for y in node_header]
for i in range(len(node_data)):
    node_data[i] = [y.strip(' ') for y in node_data[i]]

for i in range(len(edge_data)):
    edge_data[i] = [y.strip(' ') for y in edge_data[i]]
```

Zatim, pregledavajući učitane podatke, primjećeno je da neki podaci, specifično id brojevi, imaju dvostruke navodnike viška unutar stringa što je onemogućavalo učitavanje i procesiranje istih pomoću *networkx* biblioteke. Iz naslova je moguće jednostavno uklanjanje dovstrukih navodnika

pomoću funkcije *strip* no za podatke je trebalo napisati petlju koja pristupa posebno svakom elementu liste (koji je također lista) te izvesti istu operaciju za element unutar svakog indeksa.

```
graf = nx.Graph()
for node in node_data:
    graf.add_node(int(node[2]), name = node[1])

print(graf.nodes.data())

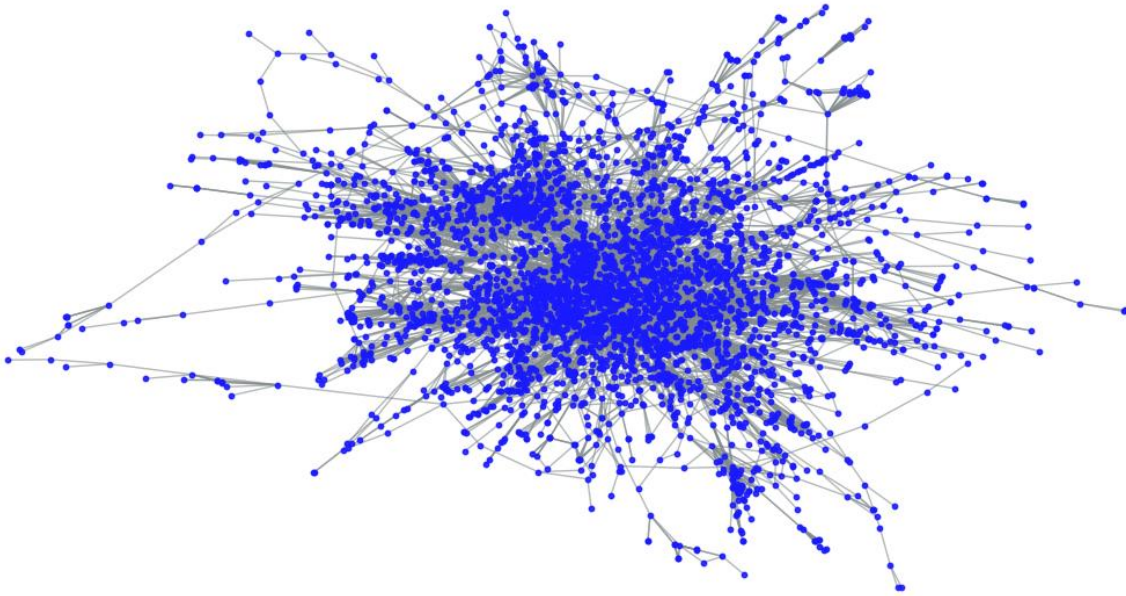
for edge in edge_data:
    graf.add_edge(int(edge[0]), int(edge[1]))

print(graf.edges.data())
```

Zatim, nakon uređivanja podataka, pomoću funkcije *Graph* izrađujemo objekt koji predstavlja mrežu i pomoću funkcija *add_node* i *add_edge* učitavamo zasebno čvorove i rubove mreže. Id čvorova su učitani kao integeri, te je još nadodan atribut (kod učitavanja čvorova) kako bi svaki čvor imao pridruženo odgovarajuće ime. Pokretanjem funkcije *print* provjeravamo da li se čvorovi i rubovi ispravno ispisuju.

```
plt.figure(3,figsize=(200,200))
pos = nx.spring_layout(graf, seed=1)
sc = nx.draw_networkx_nodes(G=graf, pos = pos, nodelist = graf.nodes(), al
pha=0.9, node_size = 10, node_color="#1a1aff")
nx.draw_networkx_edges(G = graf, pos = pos, edge_color='#818a8c', alpha=0.
6, width=1)
plt.show()
```

Za vizualizaciju mreže korištena je biblioteka *matplotlib*. Potrebno je bilo izraditi figuru veće veličine, te prilagoditi poziciju čvorova kao i veličinu zbog same veličine mreže (nezgodna je za vizualizaciju). No, konfiguracijom određenih parametara, kao što je npr. boja čvora i same veze (parametar *alpha* rubova postavljen je na 0.6 da se veze ne ističu previše na vizualizaciji) te i debljina rubova, na vizualizaciji se može vidjeti koliko toliko jasno struktura mreže. Imena čvorova nisu uključena u vizualizaciju također iz estetičkih razloga.



Slika 3. Vizualizacija mreže

Na slici 3. možemo vidjeti samu vizualizaciju mreže. Čvorovi su spojeni te nema odvojenih komponenti što ćemo potvrditi daljnjom analizom.

3. Globalne mjere

U ovom poglavlju opisane su i tablično prikazane globalne mjere koje su korištene za analizu strukture mreže. Prvo slijedi kratak opis svake mjere, zatim su tablično prikazani rezultati koji su dobiveni za prethodno izrađenu mrežu.

Broj čvorova i rubova: broj čvorova i njima pripadajućih veza unutar mreže.

```
graf.number_of_nodes()
```

```
graf.number_of_edges()
```

Prosječni broj veza: prosjek broja veza između čvorova. Izračunala sam prosjek na slijedeći način:

```
susjedi = dict(graf.degree)
prosjek = sum(susjedi.values()) / graf.number_of_edges()
```

Najprije je za svaki pojedini čvor u rječnik spremljen broj čvorova s kojima je pojedini čvor mreže vezan (broj susjeda). Zatim je suma tih susjeda (koje se nalaze pod vrijednostima rječnika) podjeljena sa ukupnim brojem veza kako bi dobili prosječan broj veza.

Broj komponenti: broj povezanih komponenti.

```
nx.number_connected_components(graf)
```

Prosječni najkraći put: predstavlja prosječnu udaljenost čvorova. Računa se kao zbroj svih udaljenosti od čvora do preostalih čvorova u grafu podjeljen sa ukupnim brojem čvorova.

```
nx.average_shortest_path_length(graf)
```

Dijametar: najveća udaljenost među čvorovima mreže, tj. najveća udaljenost između svih najkraćih puteva u mreži.

```
nx.diameter(graf)
```

Ekscentričnost: najveća udaljenost čvora od drugih čvorova u mreži.

```
eccentric = nx.eccentricity(graf)
polja = np.fromiter(eccentric.values(), dtype=float)
avg_e = np.mean(polja)
```

Globalna učinkovitost: mjera koliko učinkovito mreža prenosi informacije. Globalna učinkovitost predstavlja kvantifikaciju prijenosa informacija kroz cijelu mrežu gdje se informacije trenutno prenose.

```
nx.global_efficiency(graf)
```

Globalni koeficijent grupiranja: izračunava se prebrojavanjem povezanih trojki unutar mreže te se sam koeficijent računa kao omjer broja zatvorenih trojki grupacije i ukupnog broja zatvorenih trojki.

```
g_k_g = nx.clustering(graf)
a_g_k_g = np.mean(np.fromiter(g_k_g.values(), dtype=float))
```


Asortativnost: svojstvo mreže koje ukazuje na to koliko se čvorovi povezuju sa sebi sličnim čvorovima u odnosu na različita svojstva. Izračunava se pomoću Pearsonove korelacije, tako da se zapravo gleda korelacija u određenom svojstvu čvorova koji se spajaju.

```
nx.degree_assortativity_coefficient(graf)
```

Napomena: Izračunom globalnog koeficijenta grupiranja, dobije se riječnik veličine jednako broju čvorova mreže te je svakom čvoru pridružena određena vrijednost. U tablici sam napisala samo prosječni globalni koeficijent grupiranja, kao globalnu mjeru no u skripti ima i riječnik svih globalnih koeficijenata grupiranja, za svaki čvor.

Tablica 1. Globalne mjere mreže

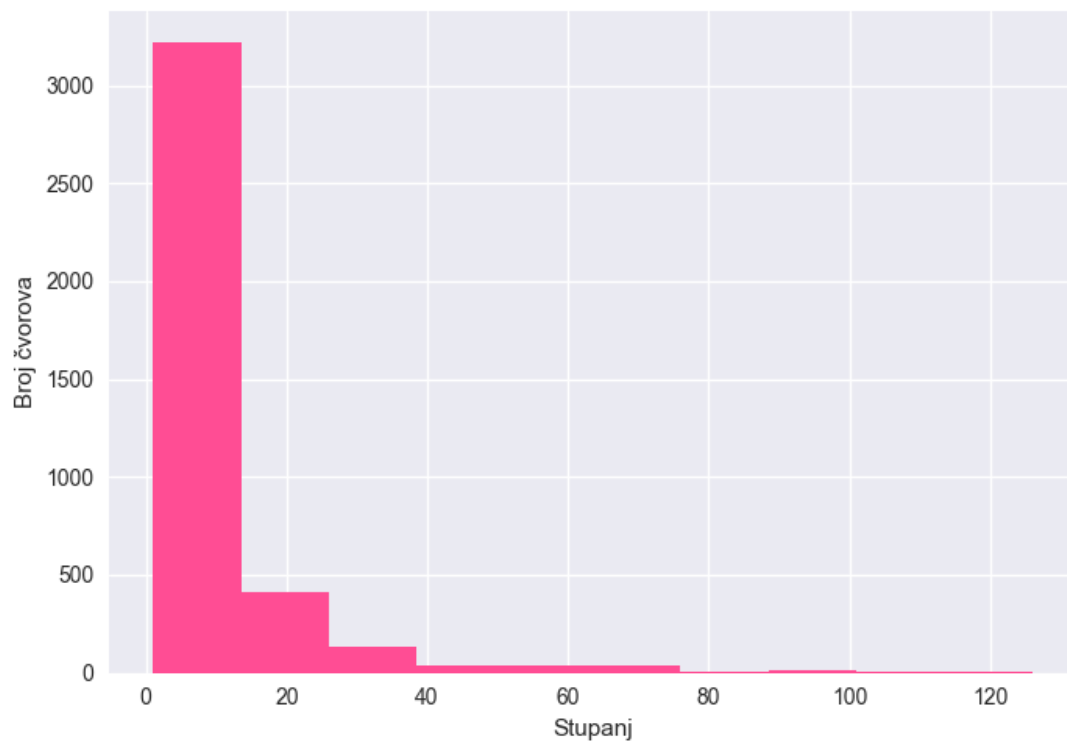
Globalna mjera	Rezultat
Broj čvorova	3892
Broj rubova (veza)	17262
Prosječan broj veza	2
Broj komponenti	1
Prosječni najkraći put	6.28
Dijametar	20
Prosječna ekscentričnost	13.32
Globalna učinkovitost	0.18
Prosječni globalni koeficijent grupiranja	0.37
Asortativnost	0.56
Prosječna centralnost blizine	0.17
Prosječna međupoloženost	0.001

Napomena: s obzirom da je broj povezanih komponenti 1, broj čvorova i veza jednak je broju čvorova i veza iste mreže. Zatim, izrađen je dijagram distribucije stupnjeva koji prikazuje distribuciju čvorova prema stupnju centralnosti, odnosno prema broju povezanih čvorova..

```
def graf_distribucije_stupnjeva(graf):
    stupnjevi = [graf.degree(n) for n in graf.nodes()]
    plt.style.use("seaborn")
    plt.hist(stupnjevi, color = "#ff4d94")
    plt.xlabel("Stupanj")
    plt.ylabel("Broj čvorova")
    plt.show()

graf_distribucije_stupnjeva(graf)
```

Kroz navedenu funkciju, izračunava se zasebno stupanj svakog čvora unutar liste čvorova mreže te se izrađuje histogram.



Slika 4 Distribucija stupnjeva mreže.

Iz navedene vizualizacije možemo primjetiti da većina čvorova sadrži stupanj veličine između 0 i 15, dok su čvorovi s većim vrijednostima izrazito rijetki.

4. Lokalne mjere

U ovom poglavlju opisane su i prikazane lokalne mjere koje su se koristile u analizi prethodno izrađene mreže. Kao i u prethodnom poglavlju, mjere su prvo opisane te je tablično prikazano, za pojedinu mjeru, 10 čvorova s najvećim vrijednostima pojedine mjere.

Centralnost stupnja: broj čvorova s kojim je određen čvor povezan. Ova mjera, za razliku od mjere stupnja, sadrži normaliziranu vrijednost umjesto samog broja povezanih čvorova.

```
centralnost_stupnjeva = nx.degree_centrality(graf)
centralnost_top10 = sorted(centralnost_stupnjeva, key=centralnost_stupnjeva.get, reverse=True)[:10]
```

Za pristupanje imenima čvorova, može se koristiti slijedeća sintaksa:

```
graf.nodes[1840]['name']
```

Na tablici 2. vidimo prikaz 10 čvorova s najvećim stupnjem centralnosti (najvećim brojem povezanih čvorova).

Tablica 2. Čvorovi s najvećom mjerom centralnosti stupnja

Rank	Čvor	Ime čvora
1.	3254	Queen of the South
2.	2008	Home & Family
3.	3525	So You Think You Can Dance
4.	1177	Glee
5.	1673	New Girl
6.	3156	Family Guy
7.	1595	The Simpsons
8.	3122	Dancing with the Stars
9.	2659	MasterChef
10.	1840	Bones

Možemo primjetiti da su čvorovi s najvećom centralnosti zapravo jako popularne tv serije i reality emisije, kao što su: Bones, The Simpsons, Masterchef, Family Guy, Dancing with the stars i slične.

Zatim slijedi vizualizacija mreže prema navedenoj mjeri. Najprije je napisana funkcija za vizualizaciju iste, kako bi se uzele u obzir veličina i boja čvora prema navedenoj vrijednosti.

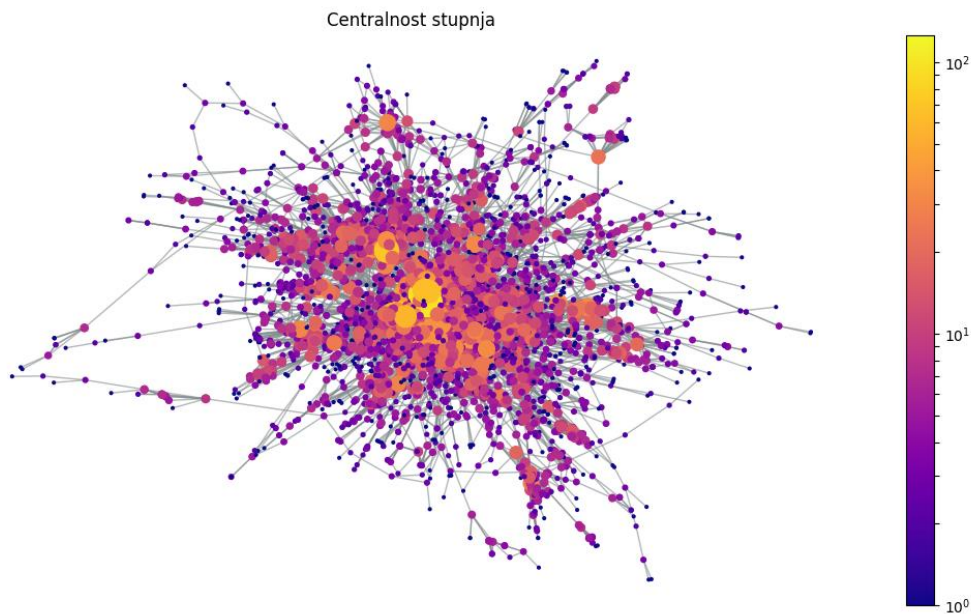
```
def draw(G, pos, node_size, measures, measure_name):
    # grafovi prema centralnosti
    nodes = nx.draw_networkx_nodes(G, pos, node_size=node_size, cmap=plt.cm.plasma,
    node_color=list(measures.values()), nodelist=measures.keys())
    nodes.set_norm(mcolors.SymLogNorm(linthresh=0.01, linscale=1, base=10))
    edges = nx.draw_networkx_edges(G, pos, edge_color='#818a8c', alpha=0.6, width=1)
    plt.title(measure_name)
    plt.colorbar(nodes)
    plt.axis('off')
    plt.show()
```

Funkcija za parametre uzima mrežu, izračunatu poziciju *pos*, veličinu čvorova, mjeru i naziv mjere. Najprije se crtaju čvorovi, gdje se veličina računa u idućem korak uzimajući vrijednosti iz napravljenog riječnika *centralnost_stupnjeva* gdje se nalaze vrijednosti mjere za svaki čvor. Boju i listu čvorova preuzimamo iz rječnika dobivenog primjenom mjere „*measures*“. Zatim se crtaju veze, gdje je navedena boja i debljina veza kako ne bi bila preuočljiva na vizualizaciji. Zatim dodajemo naslov grafikona, boje i prikazujemo vizualizaciju.

```
plt.figure(3, figsize=(200, 200))
node_sizes = []
for n in centralnost_stupnjeva.values():
    node_sizes.append(15000 * n)
draw(graf, pos, node_sizes, dict(graf.degree), 'Centralnost stupnja')
```

Prvo izračunamo veličinu čvorova, gdje se vrijednosti unutar navedenog riječnika množe s određenim brojem jer su vrijednosti normalizirane (jako sitni brojevi). Zatim prenosimo sve

parametre u prethodno napravljenu funkciju, gdje navodimo mrežu, prethodno definiranu poziciju čvorova (prilikom prve vizualizacije mreže), listu s veličinama čvorova, mjeru pomoću koje želimo vizualizirati mrežu te naziv iste.



Slika 5. Vizualizacija centralnosti stupnja.

Na slici 5. vidimo vizualizaciju mreže prema centralnosti stupnja čvorova. Vidimo da čvorovi u sredini mreže imaju najveću centralnost stupnja (u sredini se i nalazi većina čvorova, što objašnjava ovaj rezultat), dok se vrijednost iste smanjuje kako idemo prema krajevima mreže.

Centralnost međupoloženosti: čvorovi posrednici. Određena je kao omjer broja puteva koji prolaze kroz određeni čvor x i ukupnog broja najkraćih puteva. Označava koliko je čvor ključan u povezivanju različitih dijelova mreže.

```
between = nx.betweenness centrality(graf)
between_top10 = sorted(between, key=between.get, reverse=True)[:10]
```

Na tablici 3. vidimo 10 čvorova s najvećom centralnosti međupoloženosti tj. čvorove koji su posrednici najvećeg broja najkraćih puteva.

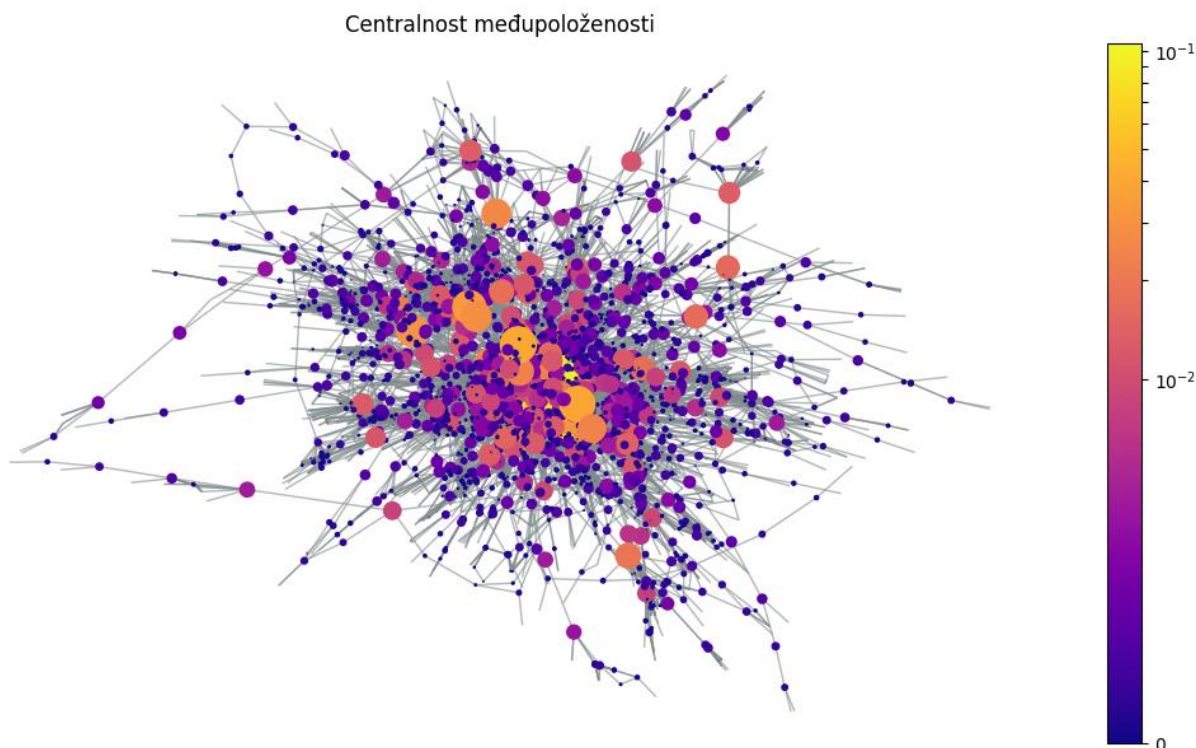
Tablica 3. Čvorovi s najvećom mjerom centralne međupoloženosti

Rank	Čvor	Ime čvora
1.	3254	Queen of the South
2.	2008	Home & Family
3.	819	The Tonight Show Starring Jimmy Fallon
4.	2170	The Voice Global
5.	2751	The Voice
6.	2895	Access
7.	3038	American Idol
8.	2682	Mars
9.	211	The List
10.	2589	Tagesschau

Možemo primjetiti određeni uzorak u čvorovima posrednicima - vidimo da su 4 od 10 serija reality emisije. Ostali čvorovi predstavljaju emisije pod kategorijom vijesti i drame. Nakon definirane tablice, slijedi vizualizacija mreže prema navedenoj mjeri.

```
plt.figure(3,figsize=(200,200))
node_sizes = []
for n in between.values():
    node_sizes.append( 10000 * n )
draw(graf, pos, node_sizes, nx.betweenness centrality(graf), 'Centralnost
međupoloženosti')
```

Vrijednost centralnosti unutar riječnika pomnožena je s određenim brojem kako bi se čvorovi mogli vidjeti na vizualizaciji.



Slika 6. Vizualizacija mreže prema centralnosti međupoloženosti

Na slici 6. prikazana je vizualizacija mreže prema centralnosti međupoloženosti. Vidimo da se najveći broj čvorova posrednika nalazi u sredini mreže, što i ima smisla jer bi većina najkraćih puteva trebalo po određenoj logici prelaziti „sredinu“ mreže.

Centralnost blizine: Određena je kao recipročna vrijednost zbroja svih duljina najkraćih puteva od zadanog čvora do ostalih čvorova. Ako neki čvor ima manje veza sa ostalim čvorovima, a ipak mu je prosječan broj koraka do svih ostalih čvorova malen, taj čvor je blizu svih čvorova, informacija brzo putuje do svih čvorova i ima veliku preglednost mreže.

```
closeness = nx.closeness centrality(graf)
closeness_top10 = sorted(closeness, key=closeness.get, reverse=True)[:10]
```

Na tablici 4. prikazano je prvih 10 rangiranih čvorova prema mjeri centralnosti blizine. tj. 10 čvorova s najvećom preglednosti mreže.

Tablica 4. Tablica čvorova s najvećom mjerom blizine.

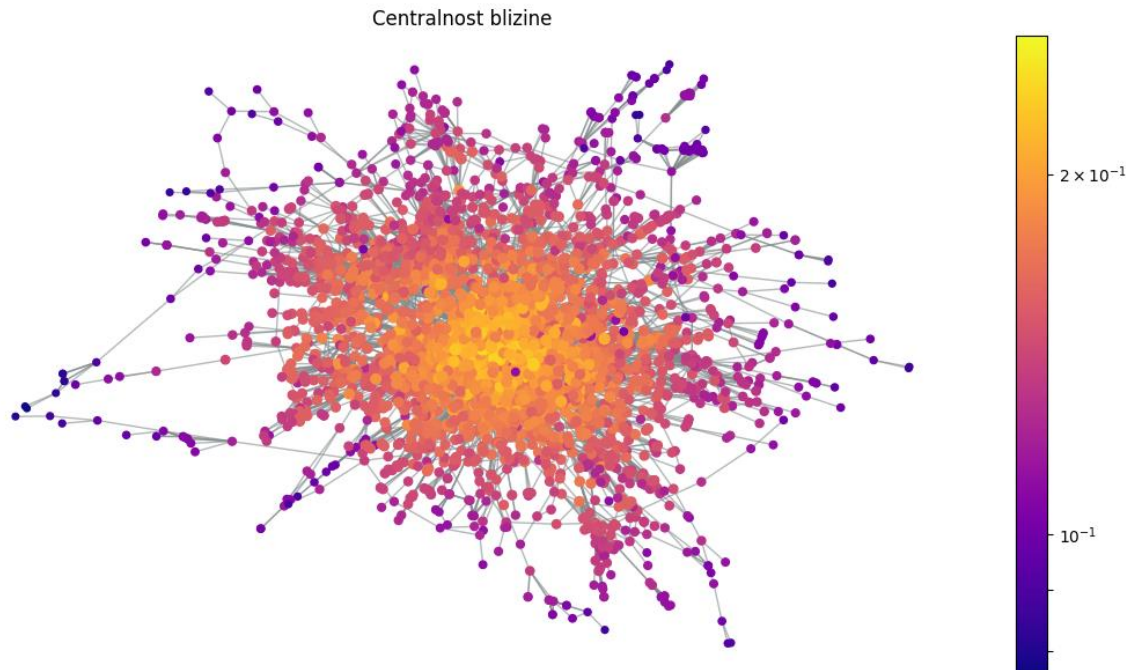
Rank	Čvor	Ime čvora
1.	3254	Queen of the South
2.	2008	Home & Family
3.	2895	Access
4.	819	The Tonight Show Starring Jimmy Fallon
5.	2751	The Voice
6.	211	The List
7.	160	America's Got Talent
8.	3837	Battle Creek on CBS
9.	2885	The Biggest Loser
10.	2035	Parenthood

Kod tablice čvorova s najvećom centralnosti blizine možemo primjetiti već neko pojavljivanje određenih čvorova (Queen of the South, Access, Home & Family, The Voice,...) te sličnost dotične tablice sa tablicom čvorova s najvećom vrijednosti međupoloženosti.

Slijedi vizualizacija mreže prema mjeri centralnosti blizine.

```
plt.figure(3,figsize=(200,200))
node_sizes = []
for n in closeness.values():
    node_sizes.append( 200 * n )
draw(graf, pos, node_sizes, nx.closeness centrality(graf), 'Centralnost bl
izine')
```

Za vizualizaciju ove mreže koristi se ista funkcija i proces vizualizacije. Promjenjena je samo odgovarajuća mjera prema kojoj će se vizualizirati čvorovi.



Slika 7. Vizualizacija mreže prema mjeri centralnosti blizine

Na slici 7. prikazana je vizualizacija mreže prema mjeri centralnosti blizine. Vidimo da su čvorovi s najvećom preglednosti mreže nalaze točno u sredini mreže, dok su čvorovi s manjom preglednosti mreže smješteni većinom na „krajevima“ mreže (što i ima smisla, jer se većina čvorova nalazi u sredini i pri tome su središnji čvorovi bliži većem broju čvorova nego čvorovi na „krajevima“ mreže).

Podjela mreže u zajednice

```
communities = community.asyn_fluidc(graf, 10)
communities = list(communities)
```

Biblioteka *networkx* nudi razne algoritme za izračun zajednica. Prije odabira navedenog algoritma (*Asyn_fluidc*) pokušala sam upotrebiti nekoliko različitih algoritama - *naive_greedy_modularity_communities*, *girvan_newman* te i *lukes_partitioning* algoritam no izračun zajednica je trajalo predugo (kod je pušten da se vrti 30 minuta po algoritmu) te sam odabrala navedeni za izračun istih (uzrok tome mislim da je veličina same mreže no i jačina laptopa na kojem se radio projekt). Parametar k (u ovom slučaju 10) određuje koliko zajednica želimo stvoriti. Kao rezultat dobijemo listu u kojoj se unutar pojedinog indeksa nalaze određene zajednice.

Tablica 5. prikazuje redom zajednice te broj čvorova i veza u pojedinoj zajednici. Brojeve čvorova i veza pojedine zajednice možemo dobiti na slijedeći način:

```
zajednica = graf.subgraph(communities[0])
zajednica.number_of_nodes()
zajednica.number_of_edges()
```

gdje stvaramo „podmrežu“ zajednice naše mreže. Pod indeks liste zajednica navodimo indeks određene zajednice unutar liste.

Tablica 5. Tablica zajednica

Zajednica	Broj čvorova	Broj veza
1.	195	381
2.	463	1020
3.	188	483
4.	528	3002
5.	531	3882
6.	213	488
7.	326	887
8.	475	2078
9.	678	2595
10.	295	1135

Vizualizacija zajednica

```
color_map = []
for node in graf:
    if node in communities[0]:
        color_map.append('#0000ff')
    elif node in communities[1]:
        color_map.append("#00b300")
    elif node in communities[2]:
```

```

        color_map.append("#660066")
    elif node in communities[3]:
        color_map.append("#ff1ac6")
    elif node in communities[4]:
        color_map.append("#e6e600")
    elif node in communities[5]:
        color_map.append("#ff9900")
    elif node in communities[6]:
        color_map.append("#ff0000")
    elif node in communities[7]:
        color_map.append("#664400")
    elif node in communities[8]:
        color_map.append("#5c5c3d")
    else:
        color_map.append('#b3ffb3')

```

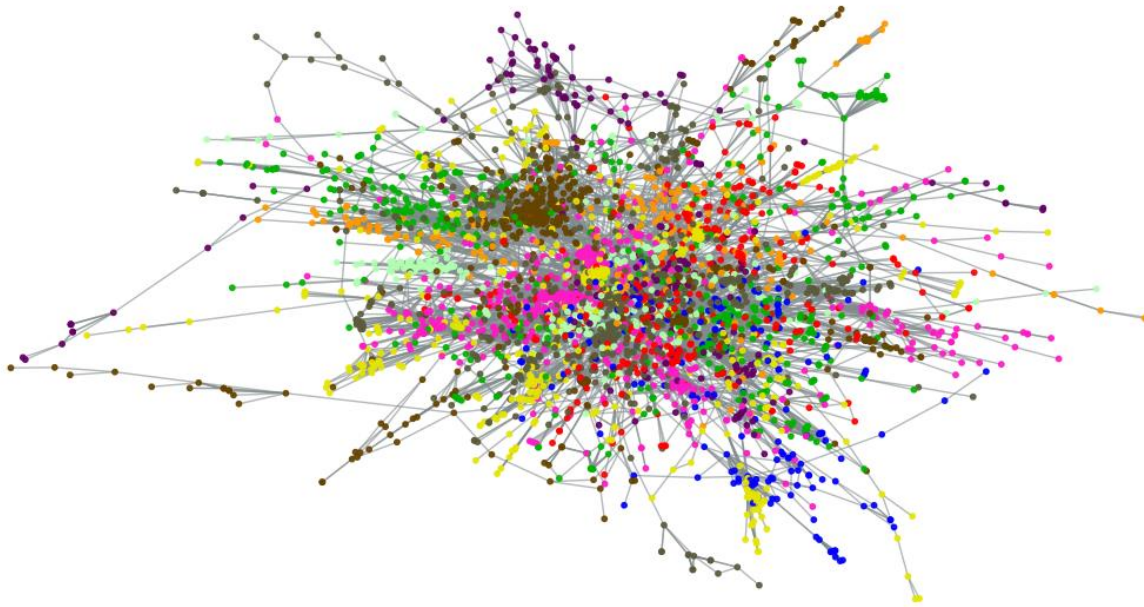
Za početak stvaramo mapu boja gdje stvaramo listu koja odgovara broju čvorova mreže. Pregledava se kojoj zajednici pojedini čvor pripada, te se dodjeljuju određene boje na mjestima indeksa čvorova.

```

plt.figure(3,figsize=(200,200))
sc = nx.draw_networkx_nodes(G=graf, pos = pos, nodelist = graf.nodes(), al
pha=0.9, node_size = 10, node_color=color_map)
nx.draw_networkx_edges(G = graf, pos = pos, edge_color='#818a8c', alpha=0.
6, width=1)
plt.show()

```

Mrežu vizualiziramo na isti način, uz dodatak prethodno izrađene liste boja pod parametar *node_color*.



Slika 8. Vizualizacija zajednica mreže.

Na slici 8. možemo vidjeti vizualizaciju mreže prema pripadnosti određenim zajednicama.

5. Dodatne mjere centralnosti i Spearman korelacija

U ovom poglavlju opisano je 5 dodatnih mjera centralnosti korištenih u projektu i tablično su prikazani 10 čvorova sa najvećom vrijednosti pojedine mjere. Također, prikazan je način izračuna Spearman korelacije između pojedinih rezultata mjera te je prikazana matrica konfuzije.

Katz centralnost: relativni stupanj učinka čvora unutar mreže. Ova mjera uzima u obzir i ukupan broj prolaska između 2 čvora.

```
katz = nx.katz_centrality_numpy(graf)
katz_top10 = sorted(katz, key=katz.get, reverse=True)[:10]
```

Tablica 6. Tablica s 10 čvorova najveće Katz centralnosti

Rank	Čvor	Ime čvora
1.	2797	AL ROJO VIVO
2.	1335	La Querida del Centauro
3.	1847	Que Noche con Angelica y Raul
4.	3362	Un Nuevo Día
5.	3263	Eva La Trailera
6.	279	José José
7.	2263	El Señor de los Cielos
8.	3721	Sangre De Mi Tierra
9.	2452	Bajo el Mismo Cielo
10.	3801	CTVPowerPlay

Informacijska centralnost: harmonični prosjek širine svih puteva koji potječu iz određenog čvora (širina puta je inverzna dužini puta).

```
inf = nx.current_flow_closeness centrality (graf)
inf_top10 = sorted (inf, key=inf.get, reverse=True)[:10]
```

Tablica 7. Tablica s 10 čvorova s najvećom informacijskom centralnošću

Rank	Čvor	Ime čvora
1.	3254	Queen of the South
2.	2008	Home & Family
3.	3525	So You Think You Can Dance
4.	3122	Dancing with the Stars
5.	1177	Glee
6.	1595	The Simpsons
7.	1673	New Girl
8.	3156	Family Guy
9.	2895	Access
10.	2659	MasterChef

Možemo primjetiti sličnost navedene tablice sa tablicaom mjere blizine što ima smisla s obzirom da je navedena mjera nastala kao varijanta mjere blizine.

Centralnost opterećenja: omjer najkraćih puteva koji prolaze kroz određeno čvor i ukupnog broja najkraćih puteva.

```
load = nx.load_centrality(graf)
load_top10 = sorted(load, key=load.get, reverse=True)[:10]
```

Tablica 8. Tablica s 10 čvorova najveće centralnosti opterećenja

Rank	Čvor	Ime čvora
1.	3254	Queen of the South
2.	2008	Home & Family
3.	819	The Tonight Show Starring Jimmy Fallon
4.	2170	The Voice Global
5.	2751	The Voice
6.	2895	Access
7.	3038	American Idol
8.	211	The List
9.	2682	Downton Abbey
10.	2589	Tagesschau

Možemo primjetiti sličnost navedene tablice s tablicom čvorova s najvećom vrijednosti centralnosti međupoloženosti s obzirom da se navedena mjera neznatno razlikuje od mjere centralnosti međupoloženosti.

Harmonična centralnost: suma recipročne vrijednosti udaljenosti pojedinog čvora do svakog drugog čvora unutar mreže.

```
harm = nx.harmonic centrality(graf)
harmonic_top10 = sorted(harm, key=harm.get, reverse=True)[:10]
```

Tablica 9. Tablica s 10 čvorova najveće harmonične centralnosti

Rank	Čvor	Ime čvora
1.	3254	Queen of the South
2.	2008	Home & Family
3.	2895	Access
4.	819	The Tonight Show Starring Jimmy Fallon
5.	2751	The Voice
6.	160	America's Got Talent
7.	3837	The Insider
8.	211	The List
9.	2885	The Biggest Loser
10.	3122	Dancing with the Stars

Eigenvector centralnost: mjera utjecaja pojedinog čvora unutar mreže. Pridružuje relativne bodove svakom čvoru unutar mreže koji su bazirani na konceptu da povezanost pojedinog čvora sa čvorom s većim brojem bodova više doprinosi rezultatu pojedinog čvora nego povezanost dotičnog čvora sa čvorovima s niskim brojem bodova.

```
eigen = nx.eigenvector_centrality(graf)
eigen_top10 = sorted(eigen, key=eigen.get, reverse=True)[:10]
```


Tablica 10. Tablica s 10 čvorova najveće Eigenvector centralnosti

Rank	Čvor	Ime čvora
1.	3525	So You Think You Can Dance
2.	1673	New Girl
3.	1840	Bones
4.	2659	MasterChef
5.	3159	In Plain Sight
6.	566	Bob's Burgers
7.	1595	The Simpsons
8.	2036	Brooklyn Nine-Nine
9.	1177	Glee
10.	1073	Hell's Kitchen

Izračun Spearmanovog koeficijenta korelacije

```
central = list(centralnost_stupnjeva.values())
bet = list(between.values())
close = list(closeness.values())
katzc = list(katz.values())
info = list(info.values())
loadc = list(load.values())
harmo = list(preloc.values())
eige = list(eigen.values())
```

Najprije su napravljene liste vrijednosti svih listi navedenih mjera. Ove liste su izrađene kako bi kasnije mogli izraditi pandas DataFrame za olakšan izračun korelacije između stupaca.

```
data = {"degree":central, "betweenness":bet, "closeness":close, "katz": katzc, "information":info, "load": loadc, "harmonic": prelo, "eigenvector": eige}
```

Zatim izrađujemo rječnik sa nazivima i njima pripadajućim listama.

```
df = pd.DataFrame.from_dict(data,orient='index').transpose()
res = df.corr(method = "spearman")
```

Nakon izrađenog rječnika, možemo pomoću pandas funkcije *DataFrame.from_dict* pretvoriti rječnik u DataFrame i zatim izračunati Spearmanov koeficijent korelacije između pojedinih stupaca (od kojih svaki predstavlja listu vrijednosti čvorova dobivenih primjenom algoritma određene mjere centralnosti).

Index	degree	betweenness	closeness	katz	information	load	harmonic	eigenvector
degree	1	0.644514	0.551579	0.0288236	0.867377	0.640031	0.576796	0.51061
betweenness	0.644514	1	0.418352	0.065861	0.520651	0.999479	0.429151	0.349382
closeness	0.551579	0.418352	1	0.125715	0.792823	0.416195	0.998452	0.902329
katz	0.0288236	0.065861	0.125715	1	0.0728445	0.0683853	0.119395	0.0795513
information	0.867377	0.520651	0.792823	0.0728445	1	0.517806	0.812976	0.728335
load	0.640031	0.999479	0.416195	0.0683853	0.517806	1	0.426876	0.34728
harmonic	0.576796	0.429151	0.998452	0.119395	0.812976	0.426876	1	0.904096
eigenvector	0.51061	0.349382	0.902329	0.0795513	0.728335	0.34728	0.904096	1

Slika 9. Matrica konfuzije Spearmanovih koeficijenata

Na slici 9. vidimo matricu konfuzije Spearmanovih koeficijenata između pojedinih mjera. Svaka od pojedinih mjera sadržavala je listu od onoliko elemenata koliko ima čvorova unutar mreže. Vidimo izrazito visoku korelaciju između nekoliko mjera: load centrality – betweenness centrality (0.9994), harmonic – closeness (harmonična centralnost zapravo je i nastala iz centralnosti blizine, 0.998), eigenvector centrality – closeness centrality (0.9), information centrality – degree centrality (0.87), harmonic centrality – information centrality (0.81), harmonic centrality – eigenvector centrality (0.9). Također, možemo primjetiti da ne postoji negativna korelacija između niti jedne od navedenih mjera.

6. Zaključak

Kod mjerenja pojedinih mjera centralnosti može se primjetiti učestala zastupljenost određenih čvorova, kao što su tv serije: Bones, The Simpsons, The List i slične koje imaju velik udio u većini izmjerenih mjera što znači da su korisnici koji lajkaju te stranice najčešće i lajkali druge, što može značiti da su to određene popularnije serije koje općenito i imaju više lajkova na facebook-u. S obzirom na učestalo ponavljanje čvorova unutar različitih tablica te i matricu konfuzije rezultata, možemo primjetiti da postoji određena povezanost između većine navedenih mjera te da isti čvorovi u velikoj mjeri imaju najveći utjecaj na protok informacija unutar navedene mreže (ovaj zaključak temelji se na dobivenih prvih deset čvorova s najvećom vrijednosti pojedine mjere).