



UNIVERZITET U NOVOM SADU  
PRIRODNO-MATEMATIČKI  
FAKULTET  
DEPARTMAN ZA MATEMATIKU  
I INFORMATIKU



Stefan Živkov, 133/18

Praktični projekat iz predmeta Socijalne mreže

- Izveštaj -

Novi Sad, 2022.

## Sadržaj

Sadržaj.....	2
1. Uvod.....	3
2. Struktura implementacije projekta.....	3
3. Analiza ručno kreirane klasterabilne mreže.....	6
4. Analiza random kreirane neklasterabilne mreže .....	7
5. Analiza wiki mreže.....	10

## 1. Uvod

Zadatak praktičnog projekta jeste implementacija algoritma za proveru klasterabilnosti neusmerenih označenih mreža. Dodatne funkcionalnosti koje su trebale da budu implementirane jesu pronalaženje linkova koji kvare klasterabilnost, kreiranje mreže klastera, pronalaženje koalicija i antikoalicija, isporučivanje klastera kao podmreže i analiza mreža.

Projekat je realizovan u programskom jeziku Python, i korišćene u sledeće biblioteke – networkx, random i itertools.

## 2. Struktura implementacije projekta

Na samom početku, nakon pokretanja programa, korisniku će biti ponuđene različite opcije sa svim mrežama koji mogu da se učitaju i nad kojima može da se izvrši analiza. Nakon što odabere jednu od opcija biće učitana mreža i započeta analiza.

Prvo što se ispisuje jeste koliko učitana mreža ima negativnih, pozitivnih grana, broj čvorova i grana, broj povezanih komponenti u mreži, ispisuje se koliko procenata mreže čini gigantska komponenta povezanosti i indeks asortativnosti mreže. Nakon toga se identifikuju komponente, gde se ignorišu negativne grane i pronalaze se samo one koje su povezane pozitivnim granama. To je realizovano bfs algoritmom, kao što je prikazano na slici 1.

```
visited = []
components = []
def identifyComponents(graph):
    print('-> Identifikovanje klastera u mrezi')
    component = 0
    for node in graph.nodes:
        if node not in visited:
            components.append(bfs(node, graph))
            component += 1
    print('-> Broj klastera u mrezi: ', component)
    return components

def bfs(node, graph):
    comp = []
    queue = []
    comp.append(node)
    visited.append(node)
    queue.append(node)
    while queue:
        current = queue.pop(0)
        neighbors = nx.neighbors(graph, current)
        for n in neighbors:
            sign = graph.get_edge_data(current, n)
            if sign['sign'] == '-':
                continue
            if n not in visited:
                comp.append(n)
                visited.append(n)
                queue.append(n)
    return comp
```

Slika 1 – identifikacija komponenti

Nakon što su identifikovane komponente, na osnovu njih se kreira za svaku komponentu posebno podmreža na osnovu početne mreže. Metoda kojom je to realizovano je prikazana na slici 2, a ideja metode je da se za svaku komponentu kreira novi graf, dodaju čvorovi iz liste

koja predstavlja tu komponentu i nakon toga se kreiraju sve moguće kombinacije grana od čvorova iz liste. Zatim se prolazi kroz te kombinacije i proverava se da li grana postoji u originalnom grafu, ako postoji dodamo je u novi graf i dodelimo joj oznaku.

```
def klasteri_podmreze(clusters, graph):
    print('-> Kreiranje podgrafa za svaki klaster')
    podmreza = []
    for cluster in clusters:
        g = nx.Graph()
        for node in cluster:
            g.add_node(node)
        comb = list(combinations(cluster, 2))
        for c in comb:
            if graph.has_edge(c[0], c[1]):
                edge = graph.get_edge_data(c[0], c[1])['sign']
                g.add_edge(c[0], c[1], sign=edge)
            else:
                continue
        podmreza.append(g)
    return podmreza
```

Slika 2 – kreiranje podmreža za klasterne

Nakon što je kreirana i lista podmreža koja predstavlja klasterne proverava se i klasterabilnost grafova. Metoda za proveru, koja je prikazana na slici 3, je realizovana tako da se prolazi kroz svaki graf i gleda se da li ta podmreža ima negativne grane, ukoliko ima ispisuje se da je mreža neklasterabilna, a ukoliko nijedna podmreža nema negativnu granu da je klasterabilna.

```
def proveraklasterabilnosti2(components, graph):
    print('-> Provera klasterabilnosti mreze')
    for c in components:
        neg_edges = [(u, v) for (u, v, d) in c.edges(data=True) if d['sign'] == "-"]
        if len(neg_edges) > 0:
            print('Mreža nije klasterabilna, postoje negativne grane/grana u nekom od klastera.')
            return False
    print('Mreža jeste klasterabilna, ne postoje negativne grane unutar klastera.')
    return True
```

Slika 3 – provera klasterabilnosti

Kada je dobijena informacija da li je mreža klasterabilna ili nije, identifikuju se grane koje kvare klasterabilnost. One će naravno biti identifikovane i prikazane samo ukoliko je mreža neklasterabilna, a u suprotnom biće ispisana adekvatna poruka. Na slici 4 je prikazana metoda koja prolazi kroz podmreže i sve negativne grane u klasterima dodaje u listu.

```
def graneKojekvareKlasterabilnost1(components, graph):
    print('-> Identifikacija grana koje kvare klasterabilnost.')
    wrong_edges = []
    for c in components:
        neg_edges = [(u, v) for (u, v, d) in c.edges(data=True) if d['sign'] == "-"]
        if len(neg_edges) > 0:
            wrong_edges.extend(neg_edges)
    print('Broj grana koje kvare klasterabilnost: ', len(wrong_edges))
    return wrong_edges
```

Slika 4 – pronalaženje grana koje kvare klasterabilnost

Nakon toga se identifikuju koalicije i antikoalicije, koje se ispisuju na ekran. Metoda kojom je to realizovano prikazana je na slici 5.

```
def koalicije_antikoalicije2(components, graph):
    print('-> Pretraga koalicija i nekoalicija')
    coalitions = []
    anti_coalitions = []
    res = []
    for c in components:
        neg_edges = [(u, v) for (u, v, d) in c.edges(data=True) if d['sign'] == "-"]
        if len(neg_edges) > 0:
            anti_coalitions.append(c)
        else:
            coalitions.append(c)
    print('Broj koalicija: ', len(coalitions))
    print('Broj antikoalicija: ', len(anti_coalitions))
    res.append(coalitions)
    res.append(anti_coalitions)
    return res
```

Slika 5 – detektovanje koalicija i antikoalicija

Nakon identifikovanja i ispisivanja koalicija i antikoalicija prelazi se na njihovu analizu. Za njih se samo proverava prosečan stepen čvora i gustina. Realizovano je tako za svaku koaliciju i antikoaliciju dobijemo neki rezultat metrike i na kraju te rezultate saberemo i podelimo sa brojem koalicija/antikoalicija i poredimo ko ima bolju vrednost.

Analizu je potrebno uraditi i za mrežu klastera, koja se kreira odmah nakon završetka analize koalicija i antikoalicija. Kreiranje je realizovano tako što se prolazi kroz sve klastere koji su predstavljeni podmrežama i proverava se da li je bilo koji čvor iz jednog klasera povezan sa bar jednim čvorom iz drugog klastera, ukoliko jeste povezuju se negativnom granom. Implementacija je prikazana na slici 6.

```
def mreza_klastera(graph, clusters):
    print('-> Konstruisanje mreze klastera')
    novi_graf = nx.Graph()
    for k in clusters:
        k.graph['nodes'] = str(k.nodes)
        novi_graf.add_node(k.graph['nodes'])

    for k1 in clusters:
        for k2 in clusters:
            if k1 == k2:
                continue
            povezani = False
            br = 0
            nodes_k1 = list(k1.nodes)
            while br < len(k1.nodes):
                n1 = nodes_k1[br]
                br += 1
                for n2 in k2:
                    if n1 in nx.neighbors(graph, n2):
                        povezani = True
                        br = len(k1.nodes)+1
            if povezani:
                novi_graf.add_edge(k1.graph['nodes'], k2.graph['nodes'], sign='-')
    print('Broj cvorova u mrezi klastera: ', len(novi_graf.nodes()))
    print('Broj grana u mrezi klastera: ', len(novi_graf.edges()))
    return novi_graf
```

Slika 6 – kreiranje mreže klastera

Nakon kreiranja sledi analiza mreže klastera. Metrike koje se primenjuju jestu gustina, prosečan stepen čvora, dijametar i indeks asortativnosti mreže. Pored toga ispisuje se broj čvorova i grana.

### 3. Analiza ručno kreirane klasterabilne mreže

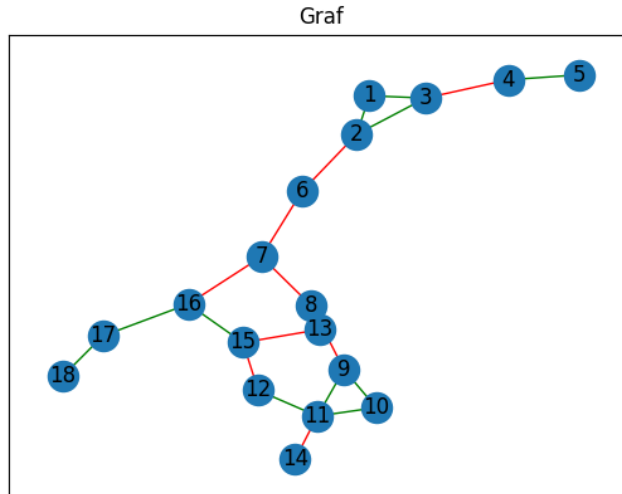
```
Izaberite opciju:
1. Klasterabilan graf
2. Neklasterabilan graf
3. Random graf
4. Wiki graf
5. Epinions graf
6. Slashdot graf
Unesite neku od ponudjenih opcija:1
-> Ucitavanje klasterabilnog grafa
9 negativnih grana
11 pozitivnih grana
Broj cvorova: 18
Broj grana: 20
Broj povezanih komponenti u mrezi: 1
Gigantska komponenta povezanosti cini 100.0 % cvorova grafa.
Indeks asortativnosti pocetnog grafa: -0.3127413127413139
-> Identifikovanje klastera u mrezi
-> Broj klastera u mrezi: 9
-> Kreiranje podgrafa za svaki klaster
-> Provera klasterabilnosti mreze
Mreza jeste klasterabilna, ne postoje negativne grane unutar klastera.
S obzirom da je mreza klasterabilna ne postoje grane koje treba ukloniti.

-> Pretraga koalicija i nekoalicija
Broj koalicija: 9
Broj antikoalicija: 0
-> Koalicije:
[1, 2, 3]
[4, 5]
[6]
[7]
[8]
[9, 10, 11, 12]
[13]
[14]
[15, 16, 17, 18]
-> Antikoalicije:
-> Konstruisanje mreze klastera
Broj cvorova u mrezi klastera: 9
Broj grana u mrezi klastera: 9
-> Analiza mreze klastera
Broj cvorova: 9
Broj grana: 9
Broj komponenti povezanosti: 1
Prosecan stepen cvorova mreze: 2.0
Gustina mreze: 0.25
Indeks asortativnosti mreze: -0.19999999999999962
Dijametar mreze: 6
-> Iscrtavanje grafa.
```

Slika 7 – analiza rucno kreiranog grafa

Proces koji je objašnjen u prethodnom delu je sada i prikazan na slici 7, ali nad ručno kreiranom klasterabilnom grafu.

Ono što možemo da zaključimo na osnovu ovog ispisa jeste da je mreža klasterabilna, što smo i očekivali, da ima 18 čvorova i 20 grana i da ima 9 klastera. S obzirom da je mreža klasterabilna i nemamo antikoalicija, nije bilo moguće ni uraditi poređenje antikoalicija i koalicija. Rezultati koje smo dobili nakon analize mreže klastera jeste da je prosečan stepen čvora 2, da je gustina 0,25, a indeks asortativnosti -0,1999. Grafik koji prikazuje ovu mrežu prikazan je na slici 8.



Slika 8 – prikaz mreže

#### 4. Analiza random kreirane neklasterabilne mreže

Za kreiranje random mreža koriste se dve metode, jedna za kreiranje klasterabilne random mreže, a druga za neklasterabilne i obe su prikazane na slikama 9 i 10. Kreiranje klasterabilne random mreže se vrši po principu da korisnik unese broj čvorova koliko želi da graf ima, a zatim se random određuje broj klastera, kojih može biti između 2 i polovine broja čvorova. Zatim svakom čvoru dodeljujemo random klaster u kojem pripada. Prolazimo kroz sve čvorove i gledamo da li pripadaju istom klasteru. Ako pripadaju gledamo dalje da li je random odabrana vrednost između 0 i 1 manja od  $p=0.5$  i da li je stepen za taj čvor manji od 25. Ako su svi uslovi zadovoljeni spajamo ih pozitivnom granom, a ako ne pripadaju istom klasteru spajamo ih negativnom granom.

```
def randomKlasterabilanGraf():
    numberOfNodes = input('-> Koliko čvorova želite da ima random graf? ')
    num_clusters = randint(2, int(int(numberOfNodes)/2))
    p = 0.5
    graph = nx.Graph()
    for n in range(int(numberOfNodes)):
        cluster_num = randint(0, num_clusters)
        graph.add_node(n, cluster=cluster_num)
    for i in graph.nodes:
        for j in graph.nodes:
            if i == j:
                continue
            if graph.nodes[i]['cluster'] == graph.nodes[j]['cluster']:
                if not graph.has_edge(i, j):
                    if random.uniform(0, 1) < p:
                        if graph.degree(i) < 25:
                            graph.add_edge(i, j, sign='+')
            else:
                if random.uniform(0, 1) < p:
                    if graph.degree(i) < 10:
                        graph.add_edge(i, j, sign='-')
    lista = []
    lista.append(graph)
    lista.append(num_clusters)
    return lista
```

Slika 9 – kreiranje random klasterabilne mreže

Kreiranje neklasteralabilnog random grafa se oslanja na kreiranje klasterabilnog, jer se prvo kreira klasterabilan pa se vrše izmene da bi došli do neklasteralabilnog. Izmene se realizuju tako što se prvo random odaberu klasteri, i zatim se prolazi kroz klastera i proverava se da li imaju više od 3 čvora, ukoliko imaju prolazimo kroz te čvorove i za svaki gledamo da li su prešli granicu od 25 čvorova sa kojima su povezani. Ako nisu, tražimo njegove susede i kreiramo pozitivne grane sa njima da budemo sigurni de grane postoje i između ta dva suseda kreiramo negativnu granu koja će narušiti klasterabilnost.

```
def randomNeklasteralabilanGraf():
    res_list = randomKlasterabilanGraf()
    graph = res_list[0]
    num_clusters = res_list[1]
    randomlist = random.sample(range(0, num_clusters), int(num_clusters/2))
    for c in randomlist:
        cluster_nodes = [u for u in graph.nodes if graph.nodes[u]['cluster'] == c]
        if len(cluster_nodes) > 3:
            for n in cluster_nodes:
                if graph.degree(n) < 25:
                    neigh = [u for u in nx.neighbors(graph, n) if graph.nodes[u]['cluster'] == c]
                    if len(neigh) >= 2:
                        graph.add_edge(n, neigh[0], sign='+')
                        graph.add_edge(n, neigh[1], sign='+')
                        graph.add_edge(neigh[0], neigh[1], sign='-')
    return graph
```

Slika 10 – kreiranje random neklasteralabilne mreže

Na slikama 11, 12, 13 i 14 prikazana je analiza random neklasteralabilne mreže, ali nisu prikazane sve grane koje kvare klasterabilnost, kao i koalicije, jer ih ima previše. Takođe i graf nije grafički prikazan zbog prevelikog broja čvorova.

```
Izaberite opciju:
1. Klasterabilan graf
2. Neklasteralabilan graf
3. Random graf
4. Wiki graf
5. Epinions graf
6. Slashdot graf
Unesite neku od ponudjenih opcija:3
-> Ucitavanje random grafa
-> Da li zelite da random izgenerisan graf bude klasterabilan (y/n)? n
-> Koliko cvorova zelite da ima random graf? 1000
8629 negativnih grana
1296 pozitivnih grana
Broj cvorova: 1000
Broj grana: 9925
Broj povezanih komponenti u mrezi: 1
Gigantska komponenta povezanosti cini 100.0 % cvorova grafa.
Indeks asortativnosti pocetnog grafa: -0.6420559544524338
-> Identifikovanje klastera u mrezi
-> Broj klastera u mrezi: 360
-> Kreiranje podgrafa za svaki klaster
-> Provera klasterabilnosti mreze
Mreza nije klasterabilna, postoje negativne grane/grana u nekom od klastera.
->Identifikacija grana koje kvare klasterabilnost.
Broj grana koje kvare klasterabilnost: 46
-> Grane koje kvare klasterabilnost:
(2, 178)
(3, 32)
(12, 186)
(31, 88)
(36, 251)
(38, 211)
(44, 152)
(68, 312)
(72, 90)
(79, 110)
(82, 261)
(83, 559)
(84, 134)
```

Slika 11 – analiza random neklasteralabilne mreže



```

-> Pretraga koalicija i nekoalicija
Broj koalicija: 314
Broj antikoalicija: 46
-> Koalicije:
[0, 199]
[1, 745, 753]
[4, 321]
[5, 57, 915]
[6, 273, 287]
[7, 148, 179, 992]
[8, 642]
[9]
[10, 105, 119, 182, 416, 797, 970]
[11, 804, 894]
[13, 205]
[14, 184, 259, 532]
[15, 180]
[16]
[17, 402, 596]
[18, 443, 456]
[19]
[20, 124, 433, 686]
[21, 190, 961]
[22, 449]
[23, 318, 561]
[24, 498, 587]
[25, 97, 735]
[26, 314]
[27, 98, 210, 351, 445]
[28]

```

Slika 12 – analiza random neklasterabilne mreže

```

-> Antikoalicije:
[2, 249, 607, 806, 178]
[3, 153, 434, 494, 790, 839, 32]
[12, 492, 708, 942, 186]
[31, 116, 322, 906, 88]
[36, 710, 809, 251]
[38, 823, 944, 211]
[44, 392, 430, 644, 887, 958, 152]
[68, 457, 504, 618, 312]
[72, 348, 666, 90]
[79, 172, 589, 110]
[82, 333, 734, 842, 868, 261]
[83, 567, 579, 559]
[84, 403, 429, 891, 134]
[95, 204, 397, 506, 967, 168]
[100, 208, 223, 326, 799, 159]
[101, 736, 905, 994, 452]
[114, 254, 408, 422, 464, 830, 154]
[122, 247, 485, 129]
[126, 277, 339, 928, 248]
[132, 511, 576, 602, 825, 294]
[145, 516, 622, 252]
[147, 246, 792, 802, 201]
[157, 550, 552, 663, 307]
[169, 731, 772, 871, 949, 627]
[187, 238, 640, 196]
[192, 343, 361, 474, 770, 791, 874, 328]
[198, 308, 586, 234]
[206, 548, 834, 483]
[212, 407, 481, 803, 341]
[221, 393, 760, 789, 974, 296]
[227, 467, 507, 599, 705, 973, 461]
[228, 512, 811, 263]
[236, 505, 893, 465]
[240, 551, 780, 880, 930, 258]
[245, 359, 373, 541, 601, 687, 827, 316]
[264, 629, 969, 377]
[274, 372, 609, 685, 733, 805, 812, 860, 352]
[280, 448, 683, 313]
[285, 729, 912, 918, 691]
[290, 437, 598, 386]
[291, 625, 983, 401]
[356, 771, 878, 741]
[391, 730, 758, 867, 462]
[411, 689, 769, 828, 590]
[497, 775, 820, 937, 572]
[547, 990, 991, 574]

```

Slika 13 – analiza random neklasterabilne mreže

```

-> Analiza koalicija i antikoalicija
Prosecan stepen antikoalicija je veci od prosecnog stepena koalicija.
Gustina antikoalicija je veca od gustine koalicija.
-> Konstruisanje mreze klastera
Broj cvorova u mrezi klastera: 360
Broj grana u mrezi klastera: 5323
-> Analiza mreze klastera
Broj cvorova: 360
Broj grana: 5323
Broj komponenti povezanosti: 1
Prosecan stepen cvorova mreze: 29.572222222222223
Gustina mreze: 0.08237387805632931
Indeks asortativnosti mreze: -0.815120975960744
Dijametar mreze: 3
-> Iscrtavanje grafa.

Graf ima vise od 200 cvorova i nije ga moguće graficki prikazati./n

```

Slika 13 – analiza random neklastabilnog grafa

## 5. Analiza wiki mreže

Analiza realne mreže Wiki. Tokom učitavanja fajla povezivanje čvorova je vršeno na osnovu toga da li je source čvor glasao za target čvor. S obzirom da je se za vrednost tog parametra pojavljuje i 0, osim 1 i -1, ona je ignorisana.

```

Izaberite opciju:
1. Klastabilan graf
2. Neklastabilan graf
3. Random graf
4. Wiki graf
5. Epinions graf
6. Slashdot graf
Unesite neku od ponudjenih opcija:4
-> Ucitavanje wiki grafa
39094 negativnih grana
132548 pozitivnih grana
Broj cvorova: 11380
Broj grana: 171642
Broj povezanih komponenti u mrezi: 123
Gigantska komponenta povezanosti cini 98.92794376098418 % cvorova grafa.
Indeks asortativnosti pocetnog grafa: -0.07160730892186953
-> Identifikovanje klastera u mrezi
-> Broj klastera u mrezi: 1397
-> Kreiranje podgrafova za svaki klaster
-> Provera klastabilnosti mreze
Mreza nije klastabilna, postoje negativne grane/grana u nekom od klastera.
-> Identifikacija grana koje kvare klastabilnost.
Broj grana koje kvare klastabilnost: 35514
Bice prikazano samo prvih 100 grana koje treba ukloniti.
('Steel1943\n', 'Jasper Deng\n')
('BDD\n', 'Jusdafax\n')
('BDD\n', 'Kiefer.Wolfowitz\n')
('BDD\n', 'KumiokoCleanStart\n')
('BDD\n', 'DGG\n')
('BDD\n', 'S Marshall\n')
('BDD\n', 'Razionale\n')
('BDD\n', 'Colonel Warden\n')
('BDD\n', 'Hobit\n')
('BDD\n', 'Intothatdarkness\n')
('BDD\n', 'Buggie111\n')
('BDD\n', 'John\n')
('BDD\n', 'Malleus Fatuorum\n')
('BDD\n', 'Vodello\n')
('BDD\n', 'Ihardlythinkso\n')
('BDD\n', 'Kauffner\n')
('Miniapolis\n', 'Piotrus\n')
('Miniapolis\n', 'Go Phightins!\n')
('Miniapolis\n', 'Shadowjams\n')
('Miniapolis\n', 'AutomaticStrikeout\n')
('Miniapolis\n', 'Kierzek\n')
('Miniapolis\n', 'Northamerica1000\n')
('Miniapolis\n', 'MONGO\n')
('Miniapolis\n', 'GiantSnowman\n')
('Miniapolis\n', 'Apteva\n')
('Miniapolis\n', 'Kiefer.Wolfowitz\n')
('Miniapolis\n', 'Spencer\n')
('Miniapolis\n', 'DGG\n')
('Miniapolis\n', 'Salvio giuliano\n')
('Miniapolis\n', 'TParis\n')
('Miniapolis\n', 'Pichpich\n')
('Miniapolis\n', 'Espresso Addict\n')

```

Slika 14 – analiza wiki mreže

```

-> Pretraga koalicija i nekoalicija
Broj koalicija: 1396
Broj antikoalicija: 1
-> Koalicije:
['Liangshan Yi\n']
['Diesel-50\n']
['Andrewman327\n']
['AndyTheGrump\n']
['A. Savin\n']
['Mephistophelian\n']
['JLDW\n']
['George Ponderevo\n']
['Wikipedian Penguin\n']
['Riffic\n']
['Ldvnsx\n']
['SheepNotGoats\n']
['Ypnypn\n']
['Bensdream\n']
['ThePHAdvocate\n', 'angelo1345\n', 'AmericanJohn33\n']
['AppleJack-7\n']
['Thargor Orlando\n']
['Arkon\n']
['Arzel\n']
['GeorgeLouis\n']
['Little green rosetta\n']
['Badmintonhist\n']
['Gryllida\n']
['LittleBenW\n']
['Frietjes\n']
['Dicklyon\n']
['Bidgee\n']
['LightGreenApple\n']
['HaugenErik\n']
['Vanisaac\n']
['Mentoz86\n']
['Mike V\n']
['Ip mask\n']
['Lettik\n']
['Jguy\n']
['Epicgenius\n']
['Estlandia\n']
['MaskedHero\n']
['Curb Chain\n']
['Ansh666\n']
['HerkusMonte\n']
['Deadbeef\n']
['Chris troutman\n']
['Daniel Tomé\n']
['Portillo\n']
['Julia W\n']
['Toddy1\n']
['Sodin\n']
['Cusop Dingle\n']
['Crispmuncher\n']
['Hot Stop\n']

```

Slika 15 – analiza wiki mreže

```

-> Antikoalicije:
['Steel1943\n', 'BOD\n', 'Miniapolis\n', 'West.andrew\n', 'Plotrus\n', 'Cuchullain\n', 'TheverCry\n', 'Cncmaster\n', 'Sven Hangard\n', 'Bamakou2000\n', 'RockMagnetist\n', 'Carri\n', 'Someguy1221\n', 'Secret\n', 'Dennis Brown\n', '934\n', 'King of Hearts\n', 'Go HighTits\n', 'Hah21\n', 'Brandely\n', 'Marz\n', 'Pratysha Ghosh\n', 'Film at 11\n', 'Mytend\n', 'Slawing\n', 'Dunkish\n', 'Bluesaberry\n', 'DanielTom\n', 'RightCoeftcross\n', 'Boigi said Zeb deen\n', 'Shadowjams\n', 'Tolly4bolly\n', 'Nintendofan\n', 'Newyorkbrad\n', 'Canyoung54\n', 'MxN\n', 'Wikiwind\n', 'Marcus Quertyus\n', 'Dirtlawyer\n', 'Soap\n', 'MMSolent\n', 'Chris.urs-ols\n', 'Pol430\n', 'Peridon\n', 'Stalwart111\n', 'Mizardman\n', 'Danger\n', 'AutomaticStrikeout\n', 'Rzuu\n', 'B6952497a\n', 'W8scribeln', 'EdJohnston\n', 'Dainomite\n', 'Fucom\n', 'Llamal\n', 'Kurtis\n', 'Inks.K\n', 'Widn\n', 'In ictu oculi\n', 'Thumperward\n', 'Brookiel\n', 'Kizack\n', 'WendyPieChequers\n', 'Het.Prof\n', 'Ultreacrazz\n', 'Grazel\n', 'Stillnotal\n', 'Orlady\n', 'Phillinig\n', 'Romjames\n', 'Ignatiziel\n', 'Melanie\n', 'Rivertorch\n', 'Kaldar\n', 'Scott Martin\n', 'Tharosh of the Mizard\n', 'Dell nkn\n', 'Mr. Stradivarius\n', 'Useight's Public Sock\n', 'CasJern\n', 'RP459\n', 'Favonian\n', 'Monterey Bay\n', 'Minimac\n', 'Angelo1345\n', 'Gilderlen\n', 'Pedro\n', 'Debn\n', 'Technical 13\n', 'Nick\n', 'Amorymeltzer\n', 'Ken ticlen\n', 'TiggerJay\n', 'Doctree\n', 'SarekOfVulcan\n', 'Sparkliam\n', 'Northamerica000\n', 'Hut 8.5\n', 'MOWG\n', 'Adjwilley\n', 'Snow Blizzard\n', 'Mkdw\n', 'Trusilver\n', 'RegentsPark\n', 'Mediran\n', 'Kudung\n', 'The Transhumanis\n', 'Bloom112\n', 'Smekyzoen\n', 'Vintann\n', 'Erik\n', 'Giantsumma\n', 'Reaper Eternal\n', 'Trytyfish\n', 'Salvidrian\n', 'OasiPayanc\n', '78.26\n', 'Dru of Id\n', 'Strike Eagle\n', 'Aptev\n', 'Bilby\n', 'darkkidd\n', 'Jason Quinn\n', 'Legokt\n', 'Lord Roem\n', 'MattyWhitell\n', 'Nthep\n', 'Paulmcdonald\n', 'Shirt58\n', 'Tokyogirl79\n', 'Vablinter\n', 'Banaticus\n', 'Binksternet\n', 'Jasper Deng\n', 'Theopolis\n', 'Basalisk\n', 'Bghitel\n', 'Harria\n', 'KIC\n', 'Monty845\n', 'Sergcross73\n', 'TommyBoy\n', 'Eln', 'Ktrj81\n', 'Luk3\n', 'Jusdafax\n', 'Jebus989\n', 'Anthony Bradbury\n', 'Stfg\n', 'Haukeye\n', 'Gobonob\n', 'Dianna\n', 'Jo Decker\n', 'The Utahrap\n', 'Lectonar\n', 'Stephen\n', '99099\n', 'Tyrois\n', 'Roger\n', 'John F. Leeds\n', 'Torresifcheron\n', 'The ed\n', 'Fillelakeshoen', 'BuickCenturydrive\n', 'Status\n', 'Fred\n', 'Fox\n', '474\n', 'RayAngeln', 'Garamend Lethen\n', 'It Is Me Here\n', 'Swam\n', 'Mlpear\n', 'Inka 888\n', 'Epb123\n', 'K80tm\n', 'Ched\n', 'Mark Arsten\n', 'Chick Bowen\n', 'Kim Dent-Brown\n', 'KillerChihuahua\n', 'Arctic kangaroo\n', 'Chrishonduras\n', 'Elen of the Roads\n', 'Vetrn\n', 'Torchiest\n', 'IComputerSaysNo\n', 'Cncplayer\n', 'Mooniddengirl\n', 'Cckain14\n', 'PaleAqua\n', 'Mscutbert\n', 'Kiefer.Wolfowitz\n', 'Graham7\n', 'Webclient101\n', 'Spencer\n', 'Ajradatz\n', 'Scotty Wong\n', 'DC12020\n', 'Al amcottwallern', 'Tazardog\n', 'Cauch\n', 'Malcolm\n', 'Godepenn', 'Morning Sunshine\n', 'Brambleberry of Riverclan\n', 'Vacation\n', 'James86\n', 'Quitemusual\n', 'Staberinde\n', 'Philosopher\n', 'Anthonycole\n', 'Kellian\n', 'Mebwail\n', 'Yunchu\n', 'n\n', 'Richie333\n', 'Closedmouth\n', 'Glostedshoen', 'The Devil's Advocate\n', 'The Illusive Man\n', 'Tony\n', 'Fox Wilson\n', 'Wifione\n', 'TCO\n', 'Jackson Peablies\n', 'Zms49\n', 'Hedean\n', 'Lawrence rhoon', 'Crisco 1492\n', 'Kllop\n', 'Pdcock\n', 'Rockfang\n', 'Ottawavever\n', 'Amiruph\n', 'Clarkcj12\n', 'Mllovent\n', 'Malinaccien\n', 'Nativeforeigner\n', 'Johnmiquin', 'Jkmielen', 'Hu12\n', 'Kumkokocleanstart\n', 'Smartsen', 'Malik Shabaz\n', 'Biruitoulin', 'Masterpiece2000\n', 'Fermat\n', '066\n', 'Yannisaroutin', 'DanielFolsom\n', 'Dihydrogen Monoxide\n', 'Elonkai', 'Alex Bakharev\n', 'Kpijs\n', 'Kuklin', 'Kusam\n', 'Retawag1can000\n', 'Mlszai3\n', 'Renata3\n', 'R.O.M. (Ghost In The Machine)\n', 'Goodolopolonius\n', 'TShio12\n', 'Tscout370\n', 'Malibutt\n', 'S Marshall\n', 'Razionale\n', 'I am One of Many\n', 'Cirtin', 'Cool\n', 'Truthkeeper8\n', 'Feydustable\n', 'Salvio giuliano\n', 'Rcapiinter123\n', 'Starblind\n', 'Jayjgn', 'Ragesoss\n', 'The Blade of the Northern Lights\n', 'Stanistan\n', 'Poeticbent\n', 'Vecrumba\n', 'Sandstunk\n', 'Water Squid\n', 'Adoptlesd\n', 'Johnbod\n', 'My very best wishes\n', 'Dr. Biofeld\n', 'Homunculus\n', 'Fabartus\n', 'Agathocleas\n', 'Gwillhickers\n', 'Rocksanddirt\n', 'Everyking\n', 'Antandrus\n', 'Jnln', 'David Gerard\n', 'Neutrality\n', 'Kinturclan', 'Rje\n', 'Raul654\n', 'Bart133\n', 'Ugen4\n', 'Kewington\n', 'Dukn', 'Hwrel\n', 'Tasman\n', 'Danyln', 'Clocking Man\n', 'Michael Snow\n', 'Chris 73\n', 'Alteripse\n', 'JohnLewke\n', 'Horsenwag\n', 'Obliv', 'Amelin', 'Haicr\n', 'Juntung\n', 'Whymless\n', '9218\n', 'Andriol\n', 'Angusciellan\n', 'Rettetast\n', 'Darwineln', 'Siva1979\n', 'IronGargyle\n', 'Sigrandson', 'Majorly\n', 'Cill Wright\n', 'Captain pandan', 'Corps\n', 'Natlin', 'Vegadark\n', 'Alex43223\n', 'Gallim\n', 'Addhoc\n', 'A Train\n', 'Peacevln', 'Mikahid4\n', 'Binguyen', 'dante\n', 'Kathryn Nicholson', 'Coredast\n', 'Somm\n', 'Jersey Devil\n', 'Anas Sallo\n', 'Aeropittico\n', 'Yachielan', 'Jdi145\n', 'Jaradan', 'Naser1451\n', 'Isotop23\n', 'Prot\n', 'Insanephantom\n', 'Terence Ong\n', 'riana dzastan', 'Dionysus\n', 'Garlon96\n', 'A. B.N', 'Sir Nicholas de Mimsy-Porgington', 'The Rambling Man\n', 'The Rambling Man\n', 'Shyam Biharin', 'Trebzor Rountreen', 'Looper5920', 'TheLateDentarthurdent\n', 'Descendalln', 'Tom Morris\n', 'Silkfork\n', 'Mohamed CJ\n', 'Thine Antique Pen\n', 'Cullen320', 'Olinglish\n', 'Spillbricken', 'Lankivelln', 'Parisln', 'Rschen7754\n', 'Pipchpin', 'Dank\n', 'C678n', 'Espresso Addict\n', 'Titoz itlan', 'Zan missfeldt\n', 'Eric Junior Mouters\n', 'Airplaneman\n', 'Bill William Compton', 'Evermore2\n', 'Andrew Gray\n', 'Henrythyme\n', 'Forgot to put name\n', 'Addshore\n', 'Antiquary\n', 'Retrolord\n', 'S. Albert Square\n', 'Blinghurst\n', 'Glxn', 'SarahStierch\n', 'Mabdu\n', 'DeltaQuad\n', 'ErrantX\n', 'Fain', 'Gfoley\n', 'Gimme danger\n', 'Gonzonoln', 'Guerrillero\n', 'Ironholds\n', 'January\n', 'Qwynxian', 'Slon02\n', 'Ctj83\n', 'Logan\n', 'Steven Zhang\n', 'Grondeman\n', 'Leyo\n', 'Pandy\n', 'Presln', 'Vernomhitneyn', 'Access Denied\n', 'Redthoreau', 'Armbrust\n', 'Acalamarln', 'Catfish Jimn', 'Molelover\n', 'Cyberpower678\n', 'Achowatn', 'Mr Little Irish\n', 'Fastlyn', 'Robofish\n', 'Katarzyna', 'Bulwersateln', 'Mikayachn', 'Colonel Warden', 'Bbz3\n', 'Emblife-v', 'Scrym', 'Mathm Johnson\n', 'Kallimma', 'Greene Bertlett\n', 'Casilber\n', 'Floquenbeam', 'Platonlou27\n', 'B.N', 'Guy Macdon', 'Isellian', 'LadofShalott\n', 'Mrt366\n', 'Whataimdoing\n', 'Beageln', 'Hhhippo', 'Bagumbaln', 'Berean Hunter\n', 'Opaukn', 'Ekabishek\n', 'Fayenatic london', 'Jenks24\n', 'Zagalejo\n', 'Avicennaslin', 'Calabre992\n', 'Figureskatingfan\n', 'Ryan Veseyln', 'SuisterMisterln', 'Anonieln', 'Bahamut0013\n', 'Catfish Jim and the soapdishn', 'Redrose64\n', 'Salvio giuliano\n', 'BusterDn', 'Dayewalkern', 'NS11n', 'Ric Handcavelln', 'Robtln', 'Intotatdarkness\n', 'Thamrospace\n', 'Alcornn', 'Paul Erkin', 'Axln', 'MlyDn', 'Denurge000\n', 'Hexn', 'Shuddeln', 'Ellin', '28bytes\n', 'Guettardn', 'Bayer406\n', 'Buster\n', 'Eldadrfrn', 'Sill en serenon', 'Hicrabyrn', 'DWA11n', 'Omiealn', 'Theoverlorn', 'Alfleylhaas1aj000', 'Spartazn', 'Clabn', 'Sb Johnny', 'Sisals488\n', 'Edison', 'Kevsln', 'Spitln', 'Only in death', 'Volunteer Harekn', 'Santa Claus of the Futurn', 'De728631n', 'Bearialn', 'Xln', 'HSG1n', 'NastCelln', 'Stifeln', 'Msd000Dn', 'Hiberniantearsn', 'Ched Davis\n', 'Ged UK\n', 'Soahyn', 'Bencherlife\n', 'Neutralhome\n', 'Basket of Puppies\n', 'John Nevada

```

Slika 16 – analiza wiki mreže

```
-> Analiza koalicija i antikoalicija
Prosecan stepen antikoalicija je veci od prosecnog stepena koalicija.
Gustina antikoalicija je manja od gustine koalicija.
-> Konstruisanje mreze klastera
Broj cvorova u mrezi klastera: 1397
Broj grana u mrezi klastera: 1286
-> Analiza mreze klastera
Broj cvorova: 1397
Broj grana: 1286
Broj komponenti povezanosti: 123
Prosecan stepen cvorova mreze: 1.8410880458124552
Gustina mreze: 0.0013188309783756843
Indeks asortativnosti mreze: -0.9383835714489013
Nije moguće odrediti dijametar mreze, jer ima više od jedne komponente povezanosti.
-> Iscrtavanje grafa.

Graf ima više od 200 cvorova i nije ga moguće grafički prikazati./n
```

Slika 17 – analiza wiki mreže