

UNIVERSITÀ POLITECNICA DELLE MARCHE

FACOLTÀ DI INGEGNERIA



*Corso di Laurea Magistrale in
Ingegneria Informatica e dell'Automazione*

Corso di Data Science

Relazione per il progetto di Social Network Analysis

Studenti:

Angelone Mattia Domenico
Romanelli Marco
Giannelli Edoardo
Ali Waqar Badar

Anno Accademico 2022-2023

Social Network Analysis

Indice

| | |
|--|----|
| 1. Introduzione | 3 |
| 2. Strumenti Utilizzati..... | 4 |
| 2.1 Python | 4 |
| 2.2 NetworkX | 4 |
| 2.3 Google Colab..... | 4 |
| 2.4 Github | 5 |
| 3. Analisi Descrittiva..... | 6 |
| 3.1 Inizializzazione | 6 |
| 3.2 Analisi del grado | 7 |
| 4. Analisi delle Centralità..... | 10 |
| 4.1 Degree Centrality | 10 |
| 4.2 Closeness Centrality | 11 |
| 4.3 Betweenness Centrality | 12 |
| 4.4 Eigenvector Centrality | 13 |
| 4.5 Considerazioni sull'Analisi delle Centralità | 15 |
| 5. Analisi delle Strutture..... | 16 |
| 5.1 Clique | 16 |
| 5.2 K-core..... | 18 |
| 5.3 Ego Network | 18 |
| 5.3.1 Musicista '67' | 19 |
| 5.3.2 Musicista '7' | 19 |
| 5.3.3 Musicista '20' | 20 |
| 5.3.4 Musicista '23' | 21 |
| 5.3.5 Considerazioni sulle Ego Network | 21 |
| 5.4 Ponti | 22 |

1. Introduzione

La Social Network Analysis (SNA) è un campo di studio interdisciplinare che analizza le relazioni sociali tra individui, gruppi o organizzazioni. Utilizza concetti e metodi provenienti dalla sociologia, psicologia sociale, matematica, statistica e informatica per comprendere la struttura sociale e il modo in cui le persone interagiscono tra loro.

La SNA si concentra sull'analisi dei legami sociali, come ad esempio le amicizie, le interazioni professionali, le collaborazioni scientifiche o le interazioni online, e li rappresenta tramite grafi o reti. In una rete sociale, gli individui sono rappresentati come "nodi" e le relazioni tra di loro come "archi" o "spigoli". Questi grafi possono essere visualizzati e analizzati per identificare modelli, ruoli chiave, sottogruppi o comunità all'interno di una rete.

Attraverso la Social Network Analysis è possibile misurare e valutare diverse proprietà delle reti sociali, come la centralità di un individuo (quanto sia importante o influente nella rete), la densità della rete (quanto sono connesse le persone tra loro), la coesione di un gruppo (quanto gli individui del gruppo siano fortemente legati tra loro), l'omofilia (la tendenza delle persone a connettersi con individui simili a loro) e la diffusione delle informazioni o delle opinioni attraverso la rete.

La SNA ha molte applicazioni pratiche, ad esempio nell'analisi delle reti sociali online, nella gestione delle organizzazioni, nello sviluppo delle politiche pubbliche, nello studio delle dinamiche di diffusione delle malattie o delle innovazioni, nell'analisi dei mercati finanziari e in molti altri campi in cui le relazioni sociali sono rilevanti.

2. Strumenti Utilizzati

2.1 Python

Per svolgere la nostra analisi abbiamo adottato il linguaggio Python. Python è un linguaggio interpretato molto utilizzato nella data science per via del suo ricco ecosistema di librerie di machine learning e della sua semplicità. In particolare, in questo progetto sono state utilizzate le seguenti librerie: Pandas, Scikit-learn, Matplotlib, Seaborn, e principalmente NetworkX.

2.2 NetworkX

NetworkX è una libreria open-source scritta in Python che fornisce strumenti per la creazione, manipolazione e analisi di strutture di reti complesse. È ampiamente utilizzata per l'analisi delle reti sociali, delle reti di comunicazione, delle reti biologiche e di altri tipi di reti complesse.

NetworkX offre una vasta gamma di funzionalità per lavorare con reti, tra cui:

- 1- Creazione di reti: NetworkX supporta la creazione di diversi tipi di reti, come reti non dirette (grafo) e reti dirette (grafo diretto). Puoi creare una rete vuota o importare dati da file o altre fonti.
- 2- Gestione dei nodi e degli archi: Puoi aggiungere, rimuovere e modificare nodi e archi all'interno di una rete. NetworkX fornisce metodi per accedere alle proprietà dei nodi e degli archi e per eseguire operazioni di ricerca e filtraggio.
- 3- Analisi della rete: La libreria offre una vasta gamma di algoritmi per analizzare reti complesse. Puoi calcolare la centralità dei nodi, identificare i cammini più brevi tra i nodi, individuare comunità o cluster, calcolare misure di similarità tra reti e molto altro ancora.
- 4- Visualizzazione delle reti: NetworkX fornisce funzionalità per la visualizzazione grafica delle reti. Puoi utilizzare librerie come Matplotlib o Plotly per generare grafici e grafici delle reti in modo interattivo.
- 5- Importazione ed esportazione dei dati: NetworkX supporta l'importazione e l'esportazione di dati da e verso diversi formati, come file CSV, GML, GraphML, Pajek e molti altri.

NetworkX è una libreria flessibile e altamente personalizzabile, che consente agli utenti di adattare le funzionalità alle loro specifiche esigenze di analisi delle reti. È ampiamente utilizzata sia in ambito accademico che nell'industria per studiare e comprendere le dinamiche delle reti complesse.

2.3 Google Colab

Google Colab, abbreviazione di Google Colaboratory, è un servizio gratuito offerto da Google che consente di eseguire e scrivere codice Python all'interno di un ambiente di sviluppo basato sul cloud. È un ambiente di notebook interattivo che consente di creare e condividere documenti contenenti codice eseguibile, testo formattato, immagini e altro ancora.

Ci sono diverse caratteristiche che rendono tale strumento utile ai fini del nostro progetto, queste sono: l'esecuzione interattiva del codice, Accesso ai dati e alle librerie, Collaborazione in tempo reale, Integrazione con diversi servizi Google. Inoltre, è particolarmente utile per le persone che desiderano eseguire calcoli intensivi o lavorare con librerie di machine learning senza dover configurare un ambiente di sviluppo locale. Inoltre, essendo basato sul cloud, i notebook Colab possono essere eseguiti su qualsiasi dispositivo con accesso a Internet, rendendo il lavoro più flessibile e accessibile.

2.4 Github

GitHub è una piattaforma web-based che consente agli sviluppatori di software di collaborare, condividere e gestire il codice sorgente dei loro progetti. È ampiamente utilizzato per la gestione del controllo di versione del codice sorgente utilizzando Git, un sistema di controllo di versione distribuito. GitHub è diventato un'importante piattaforma per lo sviluppo collaborativo di software open source e viene utilizzato da milioni di sviluppatori in tutto il mondo. Questa piattaforma offre molte funzionalità, per facilitare la collaborazione tra sviluppatori, tra cui le Repository in cui è possibile ospitare ed organizzare il codice sorgente del nostro progetto nel quale c'è il collegamento a Google Colab. Tutti i riferimenti al codice e ai risultati ottenuti sono presenti nel seguente link:

https://github.com/Badar97/Progetto_DataScience/tree/main/3-NetworkX

3. Analisi Descrittiva

3.1 Inizializzazione

Il dataset scelto è stato preso dal seguente link: <https://networkrepository.com/arenas-jazz.php> questo rappresenta una Collaboration Network tra i musicisti Jazz. Ogni nodo è un musicista jazz e un arco tra 2 nodi denota che due musicisti hanno suonato insieme in una band. I dati sono stati raccolti nel 2003. Il dataset è rappresentato dal file "arenas-jazz.edges", in cui sono presenti tutti gli archi che indicano le collaborazioni tra i vari musicisti jazz; i nodi, invece, hanno valori numerici.

Facendo un'analisi preliminare notiamo i seguenti parametri, i quali ci danno una descrizione iniziale della rete che stiamo analizzando:

| | |
|--------------|---------------------|
| NODI | 198 |
| ARCHI | 2742 |
| RAGGIO | 4 |
| DIAMETRO | 6 |
| DENSITÀ | 0.14059375480695277 |
| IS_CONNECTED | True |
| CLUSTERING | 0.6174507021536301 |
| N° DI CICLI | 2545 |

Il grafo nella sua interezza è rappresentato nella figura sottostante attraverso lo *Spring Layout* di NetworkX, ovviamente il grafo può essere rappresentato anche in altri modi come, per esempio, attraverso il *Circular Layout* o *Kamada Kawai Layout*:

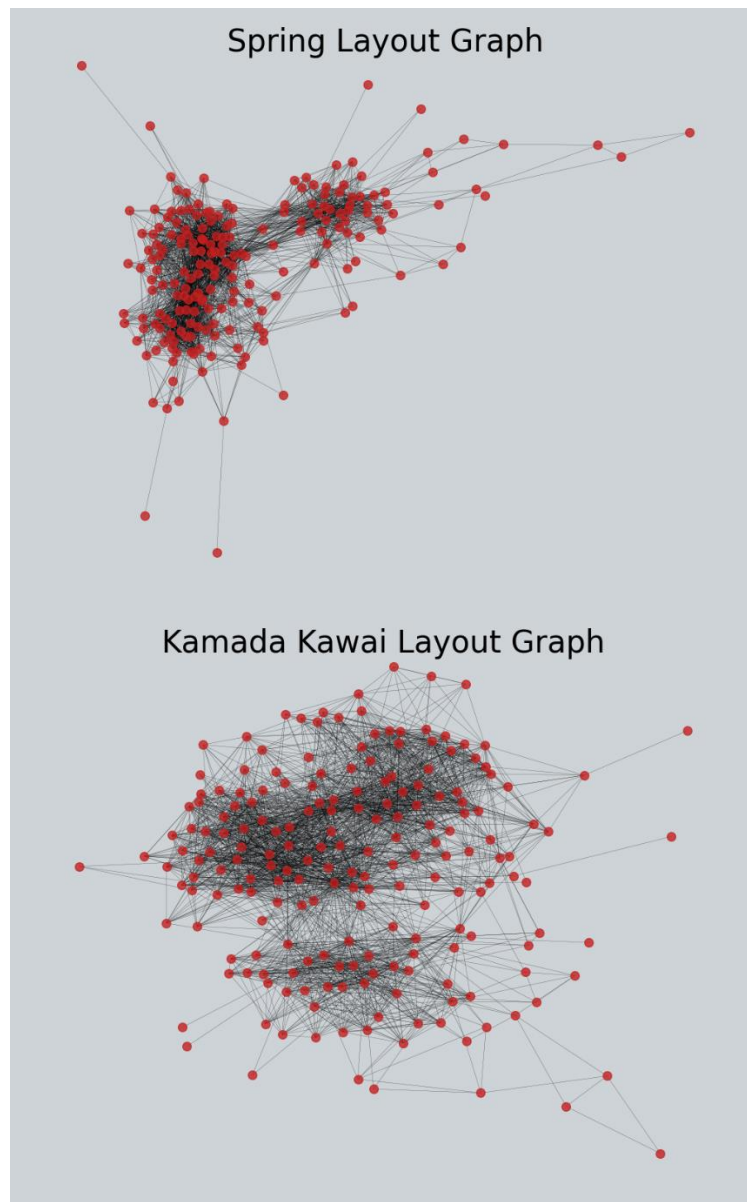


Figura 1: Visualizzazione dei nodi del grafo attraverso Spring Layout e Kamada Kawai Layout

Come si può evincere dai parametri preliminari estratti sopra si ha un grafo connesso, con 198 nodi e 2742 archi, inoltre ha il raggio della parte connessa pari a 4 e il diametro pari a 6. Nella [Figura 1](#) (Spring Layout) si può vedere che c'è una forte concentrazione di nodi nella parte centrale che rende indistinguibile la maggior parte dei nodi.

3.2 Analisi del grado

L'analisi descrittiva di un grafo in NetworkX si concentra sulla descrizione quantitativa del grafo, ad esempio dimensioni, forma, connessione e centralità dei nodi. Il nostro grafo ha le seguenti caratteristiche:

| | |
|---------------------|-------------------|
| Deviazione Standard | 17.41035019313416 |
|---------------------|-------------------|

| | |
|---------|--------------------|
| Media | 27.696969696969695 |
| Mediana | 25.0 |
| Min | 1 |
| Max | 100 |

Questi ci dicono come è in generale il grado del nostro grafo; infatti, notiamo una deviazione standard di circa 17, con il grado minimo 1 e grado massimo 100. Possiamo notare anche che la media dei gradi è circa 27 da questo possiamo dedurre che anche se abbiamo grado massimo 100 la maggior parte dei nodi hanno il grado che si avvicina al 27.

Facendo una classifica dei migliori 10 nodi con grado maggiore otteniamo i seguenti risultati:

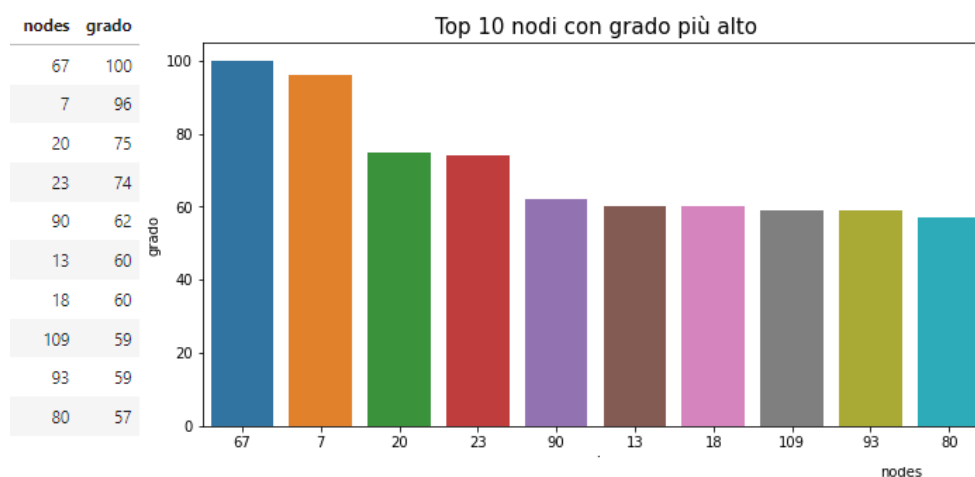


Figura 2: Top 10 nodi con grado più alto

Si vede subito, in [Figura 2](#), che il nodo con grado maggiore, cioè 100, è il nodo '67' si può intuire facilmente che è il musicista che ha avuto più collaborazioni di tutti, ad esso succedono i nodi "7" con grado 96, "20" con grado 75, "23" con grado 74 e "90" con grado 62. Il restante dei nodi ha grado inferiore a 60. Ciò si può ben notare dal seguente istogramma:

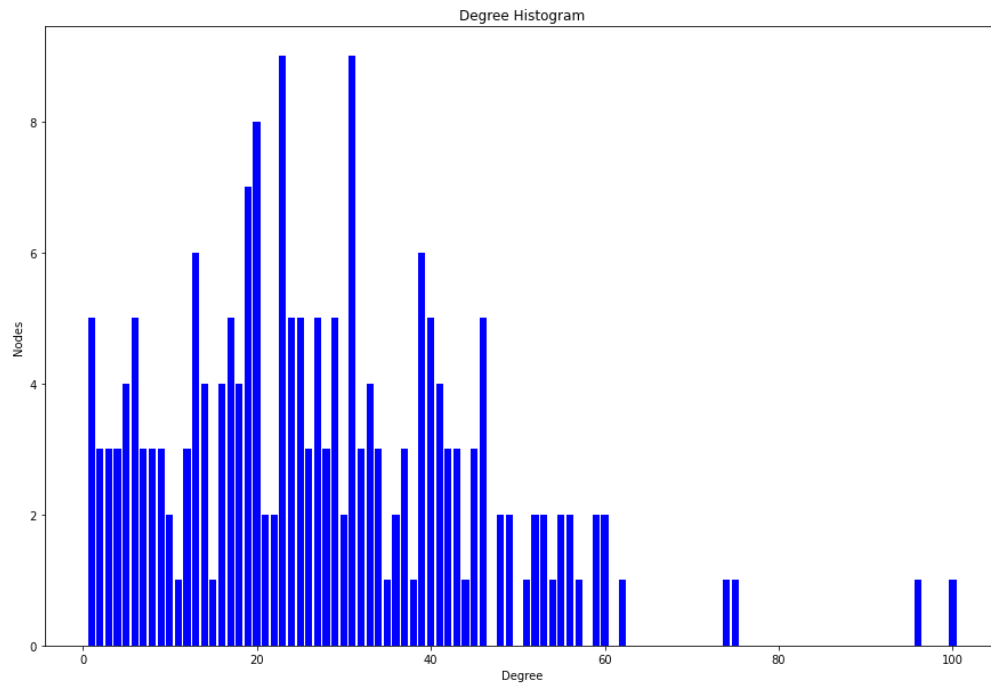


Figura 3: Istogramma grado

Da questo istogramma ([Figura 3](#)) si può che la maggior parte dei nodi hanno grado inferiore a 40, in fatti, c'è una concentrazione di nodi parecchio elevata nel caso di Degree minore di 40.

4. Analisi delle Centralità

Dopo le prime considerazioni sul dataset e sul grafo stesso si è effettuata un'analisi delle centralità associate ai vari nodi, al fine di ricavare quelli che sono i nodi più importanti per la rete stessa. Di seguito verranno analizzate le seguenti metriche di centralità:

- Degree Centrality
- Closeness Centrality
- Betweenness Centrality
- Eigenvector Centrality

4.1 Degree Centrality

La Degree Centrality è la più semplice forma di centralità e misura la popolarità o l'influenza di un nodo contando il numero di connessioni dirette che ha con gli altri nodi della rete. Un nodo con un alto valore di Degree Centrality ha molti collegamenti e può essere considerato un nodo centrale o influente nella rete, poiché è più propenso a essere coinvolto nelle informazioni o nei flussi di comunicazione. Questa centralità può essere calcolata in 2 modi, non normalizzata e normalizzata, nel nostro caso usiamo quella normalizzata, in cui il grado di un nodo viene diviso per il massimo grado possibile in una rete. Questo restituisce un valore compreso tra 0 e 1, che rappresenta la percentuale di connessioni che il nodo ha rispetto al massimo possibile.

La libreria "NetworkX" fornisce un metodo per calcolare la degree centrality e restituirla sotto forma di dizionario: `degree = nx.degree_centrality(G)`, quello che si ottiene è il seguente risultato:

| Posizione | Nodo | Degree Centrality |
|-----------|-------|---------------------|
| 1 | '67' | 0.5076142131979695 |
| 2 | '7' | 0.4873096446700507 |
| 3 | '20' | 0.38071065989847713 |
| 4 | '23' | 0.3756345177664974 |
| 5 | '90' | 0.3147208121827411 |
| 6 | '13' | 0.3045685279187817 |
| 7 | '18' | 0.3045685279187817 |
| 8 | '109' | 0.299492385786802 |
| 9 | '93' | 0.299492385786802 |
| 10 | '80' | 0.2893401015228426 |

In più si può vedere anche la sua rappresentazione grafica ([Figura 4](#)), in cui i nodi ad avere una colorazione tendente al blu sono molto pochi e risultano essere maggiormente concentrati all'esterno poiché rappresentano nodi che hanno poche collaborazioni, mentre la maggior parte sono di colore giallo e più grandi di dimensioni poiché hanno grado più alto

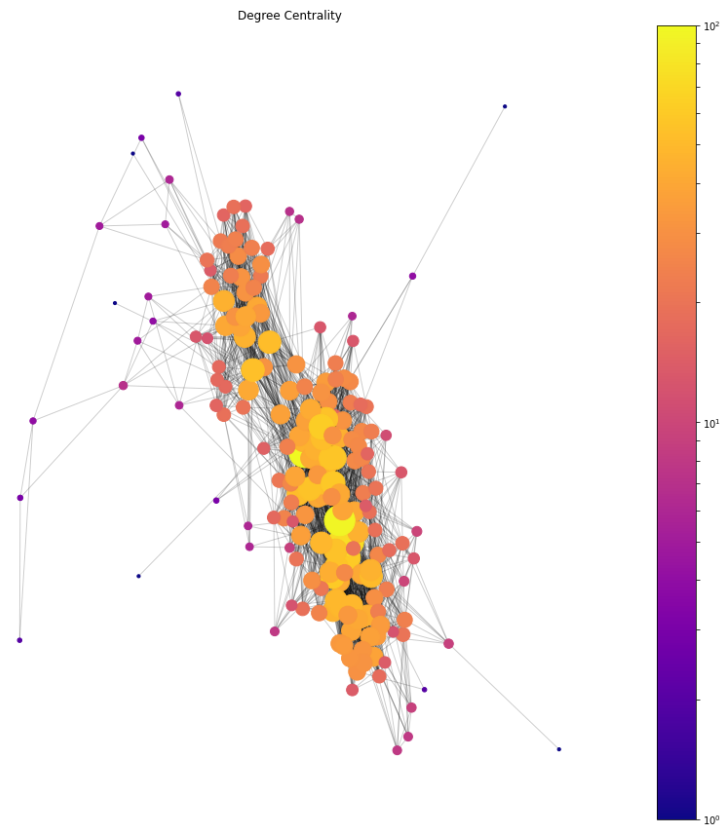


Figura 4: Degree Centrality graph

4.2 Closeness Centrality

La Closeness Centrality è una misura che valuta l'importanza di un nodo sulla base della sua vicinanza agli altri nodi nella rete. Nodi con una Closeness Centrality più alta possono raggiungere più rapidamente altri nodi nella rete, e sono considerati centrali o influenti nella diffusione delle informazioni o delle influenze nella rete.

La libreria "NetworkX" fornisce un metodo per calcolare la Closeness centrality e restituirla sotto forma di dizionario: `closeness = nx.closeness centrality(G)`, quello che si ottiene è il seguente risultato:

| Posizione | Nodo | Closeness Centrality |
|-----------|-------|----------------------|
| 1 | '67' | 0.6480263157894737 |
| 2 | '7' | 0.5898203592814372 |
| 3 | '23' | 0.5811209439528023 |
| 4 | '90' | 0.5596590909090909 |
| 5 | '93' | 0.5580736543909348 |
| 6 | '20' | 0.5549295774647888 |
| 7 | '101' | 0.5487465181058496 |
| 8 | '74' | 0.5487465181058496 |

| Posizione | Nodo | Closeness Centrality |
|-----------|-------|----------------------|
| 9 | '125' | 0.5397260273972603 |
| 10 | '109' | 0.5382513661202186 |

In più si può vedere anche la sua rappresentazione grafica ([Figura 5](#)) tramite la rappresentazione mediante un grafo, in cui si può notare come i primi 4 nodi rimangano invariati rispetto alla degree centrality:

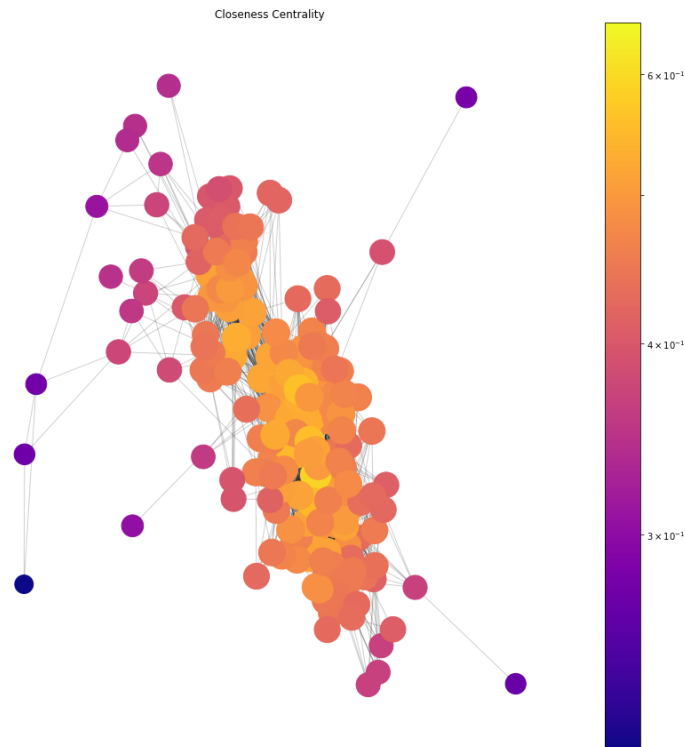


Figura 5: Closeness Centrality graph

4.3 Betweenness Centrality

La Betweenness Centrality è una misura che valuta l'importanza di un nodo come ponte o intermediario nella rete, basandosi sulla frequenza con cui il nodo si trova lungo i cammini più brevi tra tutte le coppie di nodi. Nodi con una Betweenness Centrality elevata sono considerati centrali per il flusso di informazioni o di comunicazione tra altri nodi nella rete.

La libreria "NetworkX" fornisce un metodo per calcolare la Closeness centrality e restituirla sotto forma di dizionario: `betweenness = nx.betweenness centrality(G)`, quello che si ottiene è il seguente risultato:

| Posizione | Nodo | Betweenness Centrality |
|-----------|------|------------------------|
| 1 | '67' | 0.15105615373941586 |
| 2 | '31' | 0.06829206963519824 |
| 3 | '7' | 0.057538917986120146 |

| Posizione | Nodo | Betweenness Centrality |
|-----------|------|------------------------|
| 4 | '70' | 0.04517867657801444 |
| 5 | '23' | 0.03960002124911093 |
| 6 | '32' | 0.039381158328549715 |
| 7 | '47' | 0.02955084339740044 |
| 8 | '30' | 0.028316907207527057 |
| 9 | '62' | 0.02802724501869409 |
| 10 | '93' | 0.02758017663396512 |

Da questi dati si può notare che il nodo con maggiore betweenness è il “67” con un valore pari a circa 0.15.

Si può fare anche una rappresentazione grafica ([Figura 6](#)), mediante i grafi, come le altre centralità. Dalla figura sottostante si può notare appunto come il nodo giallo che eccede rispetto agli altri sia proprio il nodo “67”, il quale ha betweenness centrality 3 volte rispetto al nodo “31” che è il secondo nodo con betweenness centrality più alta.

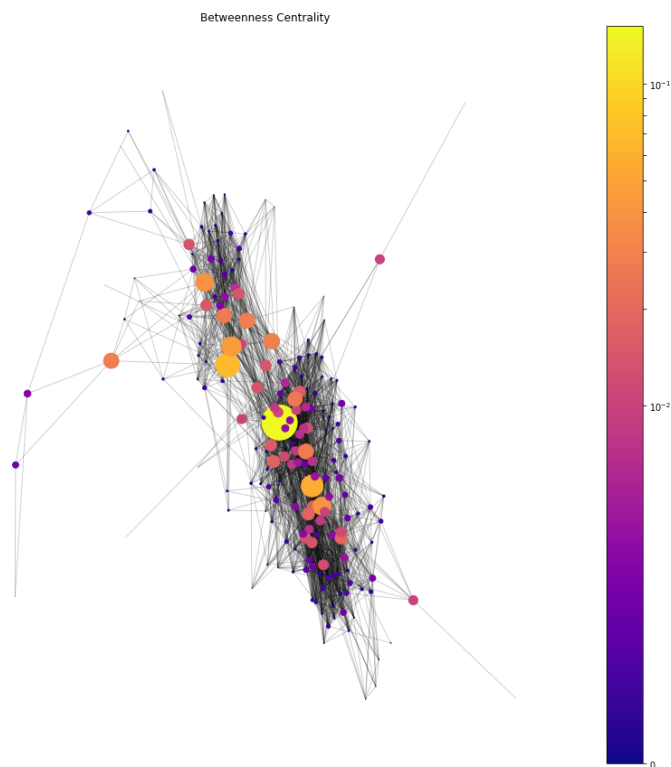


Figura 6: Betweenness Centrality graph

4.4 Eigenvector Centrality

L'Eigenvector Centrality cerca di rappresentare l'importanza e l'influenza di un nodo nella rete, ed invece di contare il numero di connessioni di un nodo effettua una media ponderata, pesando ciascun nodo sulla base del grado dei nodi adiacenti (cioè, i nodi ben connessi a nodi centrali sono pesati di più rispetto a quelli meno connessi ad essi). Attraverso questa metrica, dunque, riusciamo ad individuare le cosiddette "eminenze grigie", cioè i nodi fortemente centrali (prestigiosi, potenti) ma che sono collegati a pochi altri vertici molto connessi.

La libreria "NetworkX" fornisce un metodo per calcolare la Closeness centrality e restituirla sotto forma di dizionario: `eigen = nx.eigenvector_centrality(G)`, quello che si ottiene è il seguente risultato:

| Posizione | Nodo | Eigenvector Centrality |
|-----------|------|------------------------|
| 1 | '7' | 0.210407602557835 |
| 2 | '20' | 0.18434572268355884 |
| 3 | '67' | 0.1777740535663636 |
| 4 | '23' | 0.16687432474667022 |
| 5 | '18' | 0.15531651037408598 |
| 6 | '13' | 0.1481817629488417 |
| 7 | '19' | 0.14656774037270295 |
| 8 | '90' | 0.13523737356536084 |
| 9 | '93' | 0.13457852461030304 |
| 10 | '74' | 0.1337277172726386 |

Da ciò si evince che il nodo con Eigenvector Centrality più alta è il "7", rispetto le altre centralità in cui il nodo con maggiore influenza è il "67".

La rappresentazione grafica di questa centralità è la seguente ([Figura 7](#)):

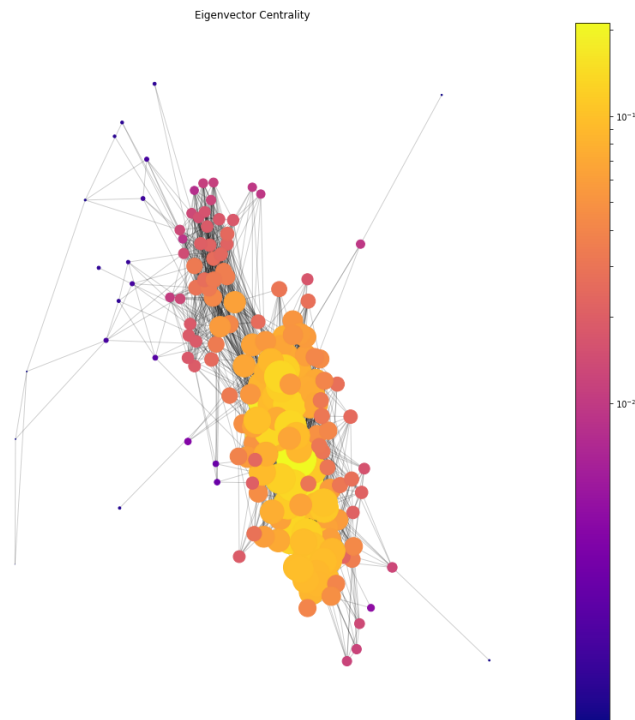


Figura 7: Eigenvector Centrality graph

4.5 Considerazioni sull'Analisi delle Centralità

Osservando le classifiche dei nodi in base alle diverse misure di centralità, possiamo trarre alcune conclusioni:

- Degree Centrality: I nodi con i punteggi più alti di Degree Centrality sono '67' e '7', che indicano che hanno il maggior numero di connessioni nella rete. Questi nodi potrebbero essere considerati nodi centrali o influenti in termini di numero di connessioni.
- Closeness Centrality: I nodi con i punteggi più alti di Closeness Centrality sono ancora '67' e '7', indicando che sono più vicini a tutti gli altri nodi nella rete in termini di distanza geodetica. Questi nodi potrebbero essere considerati importanti per la diffusione delle informazioni o delle influenze nella rete.
- Betweenness Centrality: I nodi con i punteggi più alti di Betweenness Centrality sono '67', '31' e '7', il che significa che questi nodi svolgono un ruolo cruciale nel collegamento di altri nodi nella rete. Possono fungere da ponti o passaggi chiave per la comunicazione tra i nodi.
- Eigenvector Centrality: I nodi con i punteggi più alti di Eigenvector Centrality sono '7', '20' e '67', indicando che sono collegati ad altri nodi importanti o centrali nella rete. Questi nodi potrebbero essere considerati influenti in virtù delle loro connessioni con altri nodi di alto valore di Eigenvector Centrality.

In generale, possiamo notare che alcuni nodi si ripresentano nelle diverse classifiche, come '67' e '7', suggerendo che questi nodi potrebbero avere un ruolo significativo nella rete in base a diverse misure di centralità.

5. Analisi delle Strutture

Nell'analisi delle reti le strutture rappresentano i modelli e i raggruppamenti di connessioni tra gli attori all'interno di una rete. L'analisi delle strutture mira a identificare e comprendere le configurazioni che emergono dai dati di una rete sociale. Ci sono diverse tipologie di strutture che possono essere analizzate nell'ambito della Social Network Analysis. Per esempio, si può fare analisi dei percorsi o relazione tra i nodi nella nostra rete utilizzando algoritmi di ricerca in modo tale da individuare l'influenza all'interno della nostra rete. Oppure si può fare analisi delle sottoreti con l'obiettivo di trovare specifiche sottoreti della social network. Nello specifico siamo andati ad analizzare le Clique, le Ego Network e i Ponti.

5.1 Clique

Nell' SNA, una Clique è un sottografo completo all'interno di una rete. In altre parole, è un gruppo di nodi in cui ogni coppia di nodi è collegata da un arco. In una clique, tutti i nodi sono fortemente interconnessi tra loro. Le clique sono state utilizzate per identificare gruppi di individui all'interno della nostra rete sociale che hanno interazioni forti e frequenti. Sfruttando la libreria NetworkX siamo riusciti, attraverso l'istruzione `nx.find_cliques(G)`, a capire che la Social Network che stiamo studiando è composta da un totale di 746 Clique, di seguito c'è la [Figura 8](#),

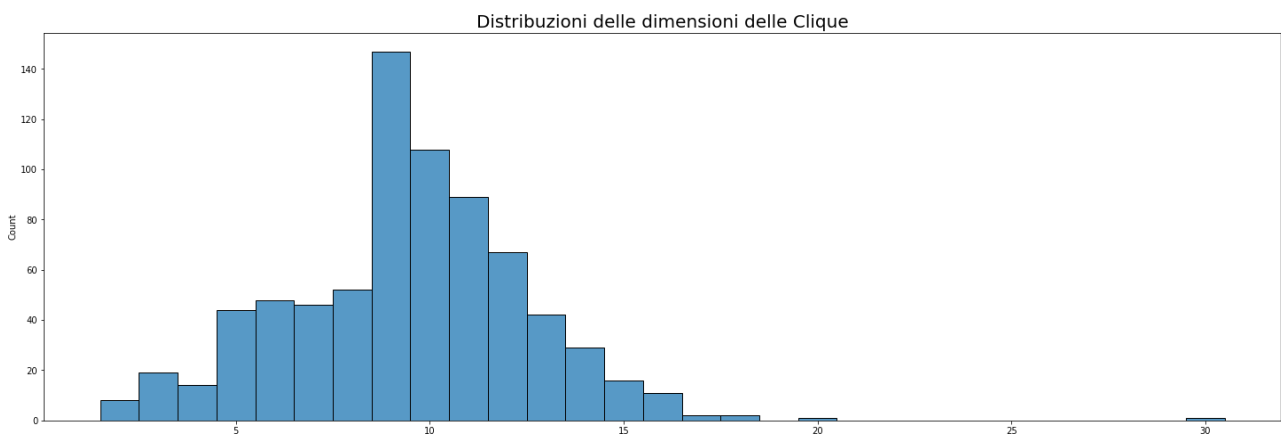


Figura 8: Distribuzione Cliques

dalla quale ad occhio si può notare che la clique massima è composta da 30 nodi, ciò si può facilmente evincere dall'istruzione di NetworkX `nx.graph_clique_number(G)` che conferma che la clique massima è appunto composta da 30 nodi. Inoltre, dall'analisi della distribuzione si può notare che si hanno solo 4 clique che hanno il numero di nodi maggiore o uguale a 18, infatti rielaborando il codice si ottengono i seguenti risultati:

```
Clique with n. nodes >= 18: 4
Graph with 30 nodes and 435 edges
Graph with 20 nodes and 190 edges
Graph with 18 nodes and 153 edges
Graph with 18 nodes and 153 edges
```


Fi

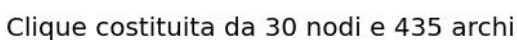


Figura 9: Clique Massima

S

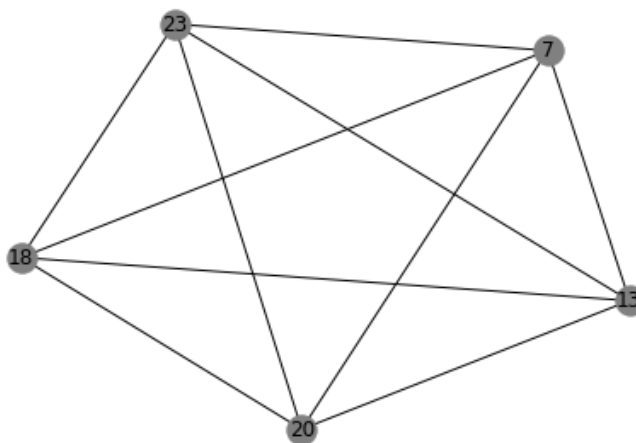


Figura 10: Clique della top 10

Si può concludere dicendo che il nodo '67' anche se è quello con grado superiore agli altri nodi, non è uno tra i nodi più influenti nella clique massima. Mentre i 5 nodi elencati sopra rappresentano un'influenza importante nello studio della clique massima.

5.2 K-core

Un'ulteriore struttura analizzata è stata k-core. K-core costituisce un rilassamento del concetto di clique in quanto calcola il sotto-grafo massimale tale per cui ogni suo nodo ha grado maggiore o uguale a k. Tale sotto-grafo è stato calcolato per mezzo del metodo `k_core()` di NetworkX il quale, non ricevendo l'ordine k all'atto della chiamata, calcola di default il "main core" del grafo di partenza. Il risultato è mostrato in [Figura 11](#). I nodi appartenenti al k-core sono 30; di fatto si tratta proprio della clique massima vista sopra.

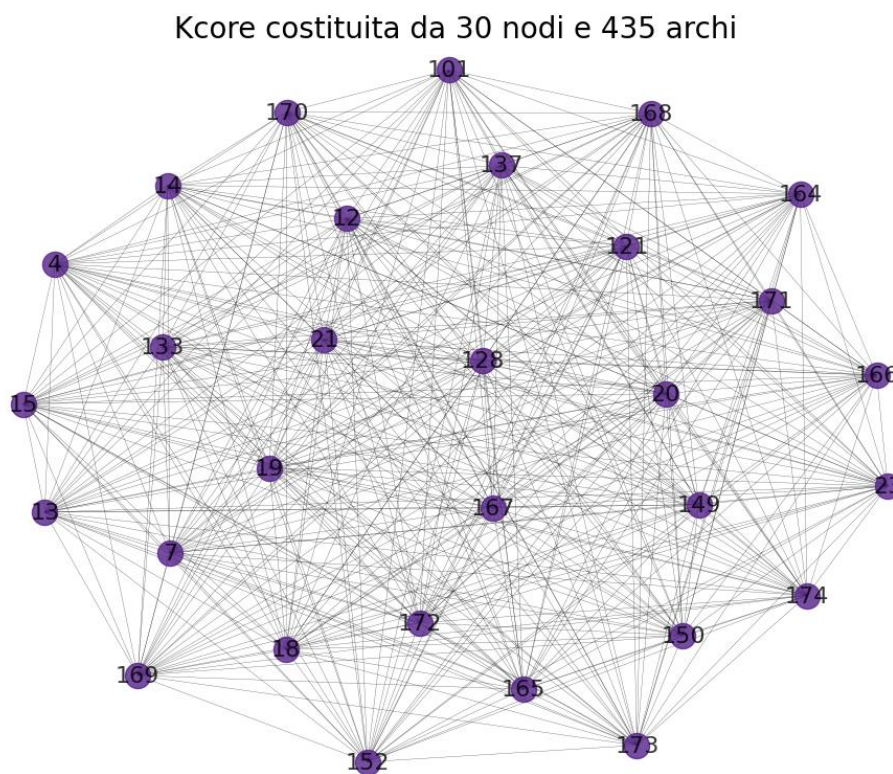


Figura 11: K-core

5.3 Ego Network

Le Ego Networks sono delle sottoreti centrate su uno specifico nodo. Tale rete può essere generata selezionando un nodo e andando a considerare tutte le sue connessioni con altri nodi della rete. Sono state generate 4 Ego Network, centrate sui seguenti nodi o musicisti:

5.3.1 Musicista ‘67’

Ego Network costituita da 101 nodi e 1274 archi

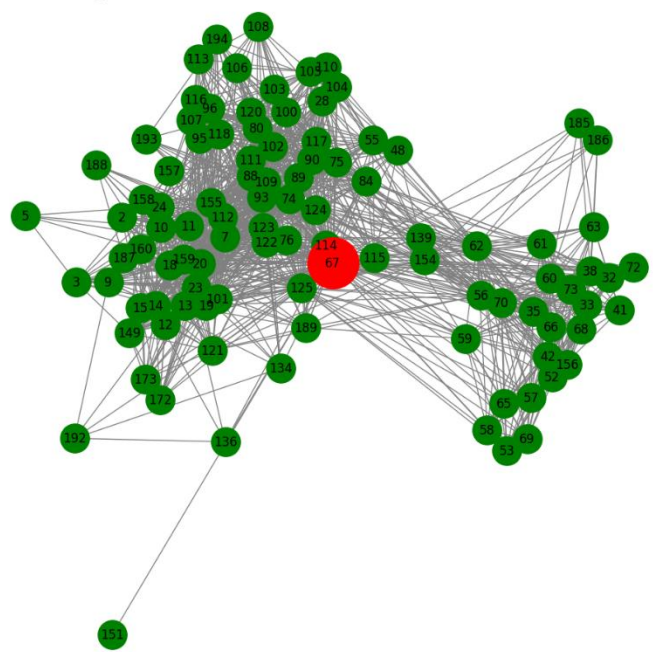


Figura 12: Ego Network nodo 67

| Nodi | Archi | Densità | Clustering |
|------|-------|--------------------|--------------------|
| 101 | 1274 | 0.2522772277227723 | 0.6842198608590029 |

5.3.2 Musicista ‘7’

Ego Network costituita da 97 nodi e 1517 archi

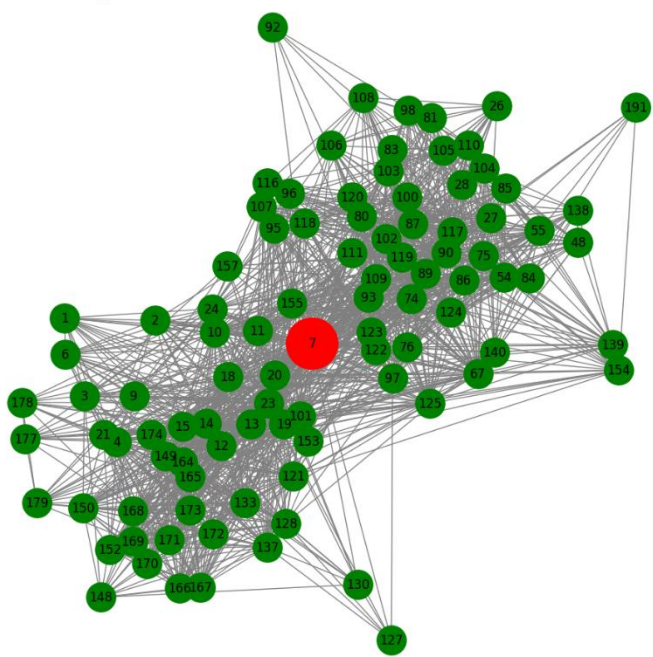


Figura 13: Ego Network nodo 7

| Nodi | Archì | Densità | Clustering |
|------|-------|---------------------|--------------------|
| 97 | 1517 | 0.32581615120274915 | 0.6996607351233486 |

5.3.3 Musicista '20'

Ego Network costituita da 76 nodi e 1196 archi

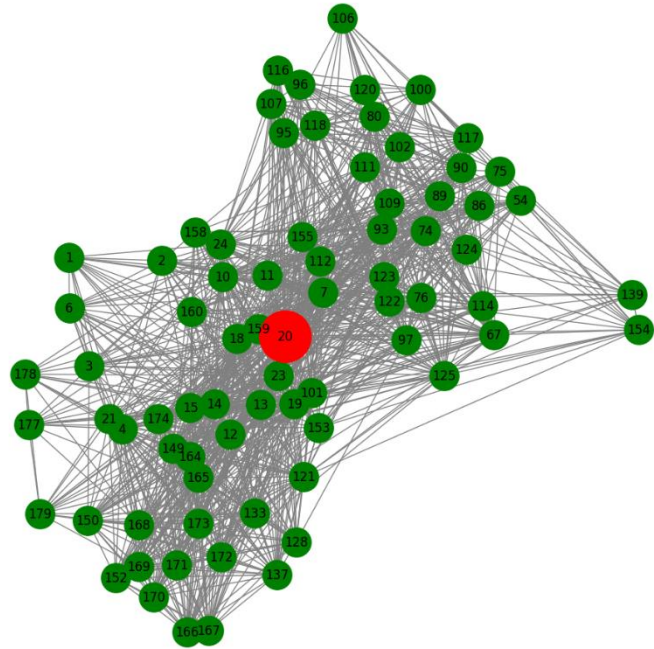


Figura 14: Ego Network nodo 20

| Nodi | Archì | Densità | Clustering |
|------|-------|---------------------|--------------------|
| 76 | 1196 | 0.41964912280701755 | 0.7393252656613751 |

5.3.4 Musicista '23'

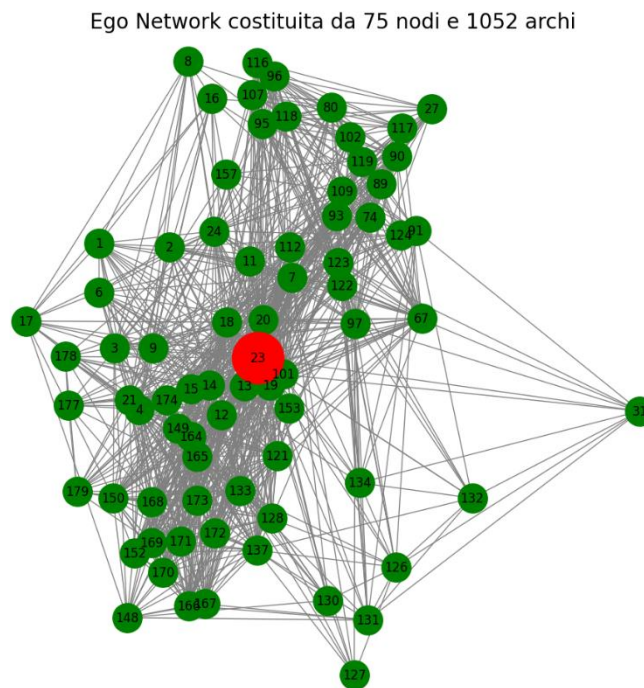


Figura 15: Ego Network nodo 23

| Nodi | Archi | Densità | Clustering |
|------|-------|--------------------|--------------------|
| 75 | 1052 | 0.3790990990990991 | 0.7340157122285705 |

5.3.5 Considerazioni sulle Ego Network

Le Ego Networks Analizzando le Ego Network (EN) dei nodi specificati sopra possiamo fare alcune considerazioni:

EN 67: composta da 101 nodi e 1274 archi, con densità di 0.2522772277227723, il che indica che è relativamente sparsa. Il coefficiente di clustering è 0.6842198608590029, che indica che i nodi tendono a formare cluster, anche se il valore non è molto alto.

EN 7: composta da 97 nodi e 1517 archi. Ha una densità di 0.32581615120274915, che è leggermente superiore rispetto alla ego network del nodo 67. Il coefficiente di clustering è 0.6996607351233486, che indica una tendenza più forte alla formazione di cluster rispetto al nodo 67.

EN 23: composta da 76 nodi e 1196 archi. Ha la densità più elevata tra le quattro ego network analizzate, con un valore di 0.41964912280701755. Questo indica una maggiore densità di connessioni all'interno della rete. Il coefficiente di clustering è 0.7393252656613751, il che indica una forte tendenza alla formazione di cluster all'interno della rete.

EN 23: composta da 75 nodi e 1052 archi. Ha una densità di 0.3790990990990991, che è simile alla ego network del nodo 7. Il coefficiente di clustering è 0.7340157122285705, che indica una forte tendenza alla formazione di cluster, simile alla ego network del nodo 23.

In generale, possiamo notare che tutte e quattro le ego network hanno una certa densità di connessioni, indicando una relativa interconnessione tra i nodi. Inoltre, i coefficienti di clustering indicano che i nodi tendono a formare cluster all'interno delle rispettive Ego Network. La differenza principale tra le ego network risiede nelle dimensioni (numero di nodi e archi) e nella densità delle connessioni.

5.4 Ponti

I Ponti rappresentano i collegamenti tra gruppi o comunità distinti all'interno di una rete. Sono connessioni che collegano nodi appartenenti a diversi sottoinsiemi della rete, consentendo il flusso di informazioni e l'interazione tra di essi. La rimozione o l'indebolimento dei ponti potrebbe influenzare la comunicazione e la diffusione delle informazioni, alterando la struttura e la dinamica della rete stessa. Nel nostro caso attraverso l'uso dell'istruzione di NetworkX `nx.bridges(G)` otteniamo la seguente rappresentazione ([Figura 16](#)):

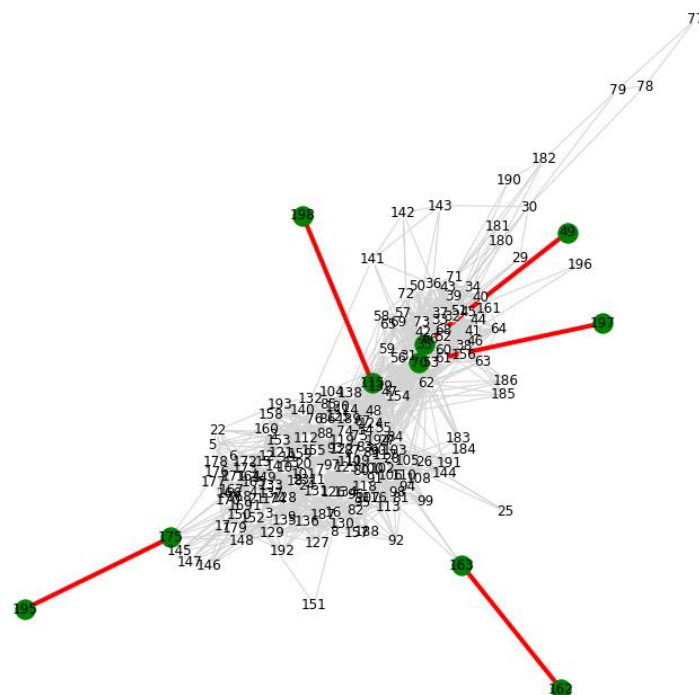


Figura 16: Ponti

In questa rete, i ponti rappresentati in rosso:

```
[('115', '198'), ('35', '49'), ('70', '197'), ('163', '162'), ('175', '195')]
```

non sono di particolare interesse perché coinvolgono solo i nodi più estremi, quindi nodi isolati da quella principale, rappresentanti musicisti coinvolti soltanto una volta in collaborazioni; quindi, la loro rimozione non va a generare componenti connesse dalla fitta rete di nodi centrali.