



Università Politecnica delle Marche

Facoltà di Ingegneria

Corso di Laurea Magistrale in Ingegneria Informatica e dell'Automazione

Corso di “Data Science” tenuto dal Prof. Domenico Ursino

Analisi di una rete di pagine Facebook verificate attraverso la libreria NetworkX

A large white "facebook" logo centered on a solid blue background.

Realizzato da:

Occhionero Giorgia
Sciarretta Luigi
Sebastianelli Alessandro

Sommario

1.	Introduzione	3
1.1	Social Network Analysis.....	3
2.	Dataset.....	4
3.	Implementazione.....	6
3.1	Statistiche elementari – rete di partenza	9
3.2	Statistiche elementari – rete ridotta	12
3.3	Centralità	13
3.3.1	Degree Centrality.....	14
3.3.2	Betweenness Centrality.....	19
3.3.3	Closeness Centrality	22
3.3.4	Eigenvector Centrality	25
3.4	Community Detection and Network Visualization	27
3.4.1	Network Visualization.....	29
3.4.2	Community Network Visualization.....	30
3.5	Communities report	32
3.5.1	Comunità 1	32
3.5.2	Comunità 2	32
3.5.3	Comunità 3	33
3.5.4	Comunità 12	33
3.6	Cliques detection.....	34
3.7	Ego Network	37
3.7.1	Ego Network Visualization.....	38
3.8	Group Centrality	39
3.8.1	Primo gruppo: Categoria TV Show	40
3.8.2	Secondo gruppo: Categoria Company	40
3.8.3	Terzo gruppo: Categoria Government.....	40
3.8.4	Quarto gruppo: Categoria Politician.....	41
3.8.5	Conclusioni Group Centrality.....	41
3.9	Classificazione multinodo	42
4.	Conclusioni	45
5.	Riferimenti	46

1. Introduzione

NetworkX è un pacchetto Python per la creazione, la manipolazione e lo studio della struttura, della dinamica e delle funzioni di reti complesse. In particolare, risulta adatta ad operare su grandi grafi del mondo reale: ad esempio, grafi con 10 milioni di nodi e 100 milioni di archi. Grazie alla sua dipendenza da una struttura dati "dictionary of dictionary" in Python, NetworkX è un framework ragionevolmente efficiente, molto scalabile e altamente portabile per l'analisi di reti e social network.

La libreria NetworkX viene utilizzata non solo da chi studia le social network, ma da chi studia qualunque tipo di rete. Il potenziale pubblico include quindi matematici, fisici, biologi, informatici e scienziati sociali. Chiaramente trattandosi di una libreria scritta in uno specifico linguaggio di programmazione, non si tratta di un tool user-friendly e di facile utilizzo per i più. Nonostante questo, si parla comunque di un potente strumento che riesce molto bene a svolgere le analisi per cui è stato progettato.

Per installare NetworkX è necessario eseguire il seguente comando:

~ pip install networkX

Mentre per aggiornarlo è possibile eseguire il seguente comando:

~ pip install --upgrade networkX oppure *~ pip install -U networkX*

1.1 Social Network Analysis

L'analisi delle reti sociali è il processo di indagine delle strutture sociali attraverso l'uso delle reti e della teoria dei grafi. Questa branca di studio è piuttosto antica e le sue origini risalgono a prima della nascita di internet e le cui applicazioni riguardo una molteplicità di scienze come ad esempio l'antropologia, la sociologia, l'economia, la psicologia e tante altre. La teoria delle reti sociali vuole che la società sia vista come una rete di relazioni, che possono essere più o meno estese e strutturate.

L'assioma principale è che ogni individuo si relaziona con gli altri e le possibili interazioni plasmino e modifichino il comportamento di entrambi. Lo scopo principale è dunque quello di individuare e analizzare tali legami tra gli individui. Tali legami in un contesto interpersonale possono essere amicizia, fiducia, importanza, affetto, o allo stesso tempo anche conflitto ed ostilità.

2. Dataset

Il dataset preso in analisi è stato scaricato dal seguente link:

<https://snap.stanford.edu/data/facebook-large-page-page-network.html> e rappresenta una rete.

La rete scelta è una rete page-to-page di siti Facebook verificati. I nodi rappresentano le pagine ufficiali di Facebook mentre i link sono i reciproci like tra i siti. Le caratteristiche dei nodi sono estratte dalle descrizioni del sito che i proprietari delle pagine hanno creato. La rete è stata raccolta attraverso la Facebook Graph API nel novembre 2017 e limitato alle pagine di 4 categorie che sono definite da Facebook. Queste categorie sono: politici, organizzazioni governative, spettacoli televisivi e aziende.

I dati si presentano sotto forma di due file principali, un file denominato “musae_facebook_target” ed uno denominato “musae_facebook_edges”.

All'interno di musae_facebook_edges ci sono esclusivamente le informazioni relative ai collegamenti tra i nodi ed è quindi la lista dei nodi della rete tra loro collegati. Viceversa, musae_facebook_target presenta la lista dei nodi della rete ed una serie di attributi descrittivi, quali **id**, **facebook_id**, **page_name** e **page_type**.

Una porzione dei due dataset è mostrata nelle figure seguenti:

id_1	id_2
0	18427
1	21708
1	22208
1	22171
1	6829
1	16590
1	20135
1	8894
1	15785
1	10281
1	22265
1	7136
1	22405
1	10379

Figura 2.1: Dataset “musae_facebook_edges”

Dunque, i campi **id_1** ed **id_2** identificano i nodi e permettono di rappresentare il collegamento esistente tra i nodi della rete.

Mentre una porzione del dataset “musae_facebook_target” è mostrata nella figura seguente:

id	facebook_id	page_name	page_type
0	1.45647E+14	The Voice of China	tvshow
1	1.91483E+11	U.S. Consulate General Mumbai	government
2	1.44761E+14	ESET	company
3	5.687E+14	Consulate General of Switzerland in Montreal	government
4	1.40894E+15	Mark Bailey MP - Labor for Miller	politician
5	1.34465E+14	Victor Dominello MP	politician
6	2.82657E+14	Jean-Claude Poissant	politician
7	2.39338E+14	Deputado Ademir Camilo	politician
8	5.44818E+14	T.C. Mezar-Äz̄erif BaÄýkonsolosluÄýu	government
9	2.85156E+11	Army ROTC Fighting Saints Battalion	government
10	2.95295E+14	NASA Student Launch	government
11	8.37707E+14	Eliziano Gama	politician
12	1.89778E+11	Socialstyrelsen	government
13	1.53345E+14	Brisbane Water LAC - NSW Police Force	government
14	3.74623E+11	NASA's Marshall Space Flight Center	government
15	1.35948E+14	Municipio de Lomas de Zamora	government
16	2.18961E+14	Die Techniker (TK)	company
17	6.03089E+14	Digvijaya Singh	politician
18	1.07219E+11	1st Armored Division Sustainment Brigade	government
19	2.88891E+11	Shapeways	company
20	2.0075E+14	FranÄşoise GuÃ©got	politician
21	3.12379E+14	Hydro Coco	company
22	1.97258E+14	Embassy of the Netherlands in Uganda	government
23	1.21926E+14	Ford Danmark	company

Figura 2.2: Dataset “musae_facebook_target”

Dove:

- **Id:** rappresenta la lista dei nodi, ogni singolo valore è dunque l’identificativo del nodo;
- **Facebook_id:** ogni nodo, oltre al proprio id con cui viene identificato, presenta un corrispondente id facebook;
- **Page_name:** descrive il nome della pagina facebook verificata relativa ad un particolare nodo;
- **Page_type:** rappresenta il tipo di pagina che contraddistingue un certo nodo. Esistono quattro tipi di pagine all’interno di tale dataset e corrispondono ai politici, organizzazioni governative, spettacoli televisivi e aziende.

3. Implementazione

Dal punto di vista del codice, per prima cosa vengono importate le librerie necessarie:

```
import networkx as nx
import matplotlib.pyplot as plt
import matplotlib.colors as mcolors
import random
import seaborn as sns
import os
import pandas as pd
import numpy as np
```

Figura 3.1: Importazione librerie necessarie

Di seguito andremo ad importare il dataset di interesse. Il primo file importato è un file di estensione .csv denominato musae_facebook_target:

```
df4 = pd.read_csv('C:/Users/gigio/Desktop/facebook_large/musae_facebook_target.csv')
```

Figura 3.2: Importazione dataset “musae_facebook_target”

Subito dopo aver importato il dataset separiamo ed inseriamo ogni attributo all’interno di liste, per aggiungere poi tali attributi ai nodi della rete.

Viene creata una lista per gli id e quindi la lista dei nodi della rete:

```
ID=[ ]
ID = df4['id'].values.tolist()
```

Figura 3.3: Creazione lista ID

Viene creata una lista per il campo facebook_id e quindi la lista dei facebook_id dei nodi:

```
facebook_id=[]
facebook_id = df4['facebook_id'].values.tolist()
```

Figura 3.4: Creazione lista facebook_id

Viene creata una lista per il campo page_name e quindi la lista delle pagine:

```
pagename = []
pagename= df4['page_name'].values.tolist()
```

Figura 3.5: Creazione lista pagename

Viene creata una lista per il campo page_type e quindi la lista delle categorie delle pagine:

```
pagetype = []
pagetype = df4['page_type'].values.tolist()
```

Figura 3.6: Creazione lista pagetype

Dopo queste operazioni andremo a creare la rete, a partire quindi dalla lista dei nodi.

Viene istanziato l'oggetto G attraverso la libreria networkx e tramite il metodo add_nodes_from vengono aggiunti i nodi della rete:

```
G=nx.Graph()  
G.add_nodes_from(df4['id'])
```

Figura 3.7: Aggiunta nodi

Inoltre, si andranno ad aggiungere gli altri attributi che sono stati precedentemente inseriti nelle liste. Quindi per ogni nodo avremo il relativo facebook id, il relativo nome della pagina e la categoria ad esso associata.

```
for i in range(len(G.nodes)):  
    G.add_nodes_from([i], ID=ID[i], facebookid=facebook_id[i], pagename = pagename[i], pagetype = pagetype[i])
```

Figura 3.8: Aggiunta attributi

Dopo aver creato la rete a partire dai nodi, bisogna stabilire quali sono i collegamenti tra questi. Per fare ciò importiamo il dataset musae_facebook_edges:

```
df_archi = pd.read_csv('C:/Users/gigio/Desktop/musae_facebook_edges.csv')
```

Figura 3.9: Importazione dataset musae_facebook_edges

Come si nota dalla figura seguente, il dataframe necessita di alcune piccole operazioni di ETL prima di poter essere utilizzato.

	id_1;id_2
0	0;18427
1	1;21708
2	1;22208
3	1;22171
4	1;6829

Figura 3.10: Dataset musae_facebook_edges pre ETL

Dunque, andremo a effettuare le seguenti operazioni al fine di ottenere due campi separati per id_1 e id_2:

```
df_archi[['id1','id2']] = df_archi["id_1;id_2"].str.split(";", expand=True)  
df_archi= df_archi.drop(['id_1;id_2'], axis = 1)
```

Figura 3.11: ETL su musae_facebook_edges

Ottenendo il dataframe nella forma desiderata:

	id1	id2
0	0	18427
1	1	21708
2	1	22208
3	1	22171
4	1	6829

Figura 3.12: Dataset musae_facebook_edges post ETL

A questo punto vengono definiti i nodi di partenza e di arrivo, il cui collegamento è di fatto l'arco che collega i nodi. Quindi definiremo due variabili *nodo_from* e *nodo_to* rispettivamente per la lista dei nodi di partenza e la lista dei nodi di arrivo:

```
nodo_from = df_archi['id1'].values.tolist()
nodo_to = df_archi['id2'].values.tolist()
```

Figura 3.13: Creazione nodo_from e nodo_to

A questo punto, creeremo un'unica lista *edges* all'interno della quale sono inseriti i collegamenti tra i nodi, dunque gli archi:

```
edges = []
for i in range(G.nodes):
    edges.append((int(nodo_from[i]), int(nodo_to[i])))
```

Figura 3.14: Creazione lista edges

Volendo rappresentare una porzione di quello che è la lista *edges* appena creata, si ha il seguente output:

```
[(0, 18427), (1, 21708), (1, 22208), (1, 22171), (1, 6829), (1, 16590), (1, 20135), (1, 8894), (1, 15785), (1, 10281), (1, 2265), (1, 7136), (1, 22405), (1, 10379), (1, 13737), (1, 8533), (1, 14344), (1, 2812), (1, 5755), (1, 16260), (1, 15026), (1, 17370), (1, 17460), (1, 8049), (1, 5307), (1, 4987), (1, 18304), (1, 12305), (1, 19743), (1, 20024), (1, 21729), (1, 10554), (1, 11557), (1, 5228), (1, 9934), (2, 9048), (2, 6353), (2, 2629), (2, 11537), (2, 13205), (2, 22304), (2, 17728), (2, 19337), (2, 126), (2, 17554), (2, 8495), (2, 5857), (3, 16742), (3, 293), (3, 5826), (3, 3479), (3, 19753), (3, 17346), (3, 10945), (3, 22338), (3, 11319), (3, 9654), (4, 13645), (4, 20876), (4, 11446), (4, 16203), (4, 2830), (4, 2004), (4, 20624), (4, 21280), (4, 1182), (4, 21538), (4, 1443), (4, 11423), (4, 187), (4, 5738), (4, 2983), (4, 1489), (4, 6823), (4, 17695), (4, 1102), (4, 6390), (4, 17242), (4, 18018), (4, 5147), (4, 6427), (4, 14628), (4, 1882), (4, 22481), (4, 16128), (4, 12872), (4, 9263), (4, 14155), (4, 21631), (4, 6329), (4, 17507), (4, 2282), (4, 9706), (4, 4738), (4, 3676), (4, 16972), (4, 5356), (4, 8514), (4, 14332), (4, 7212), (4, 8843), (4, 1879), (4, 1377), (4, 1997), (4, 7813), (4, 3891), (4, 2732), (4, 4189), (5, 8288), (5, 9206), (5, 1840), (5, 17845), (5, 17411), (5, 21768), (5, 15735), (5, 18468), (5, 21755), (5, 16406), (5, 14111), (5, 20510), (5, 945), (5, 20271), (5, 14862), (5, 3726), (5, 6946), (5, 12902), (5, 4808), (5, 14241), (5, 7106), (5, 18497), (6, 18893), (6, 1193), (6, 4000), (6, 12625), (6, 290), (6, 22261), (6, 13966), (6, 3300), (6, 22403), (6, 18782), (6, 5066), (6, 17038), (6, 3816), (6, 16052), (6, 12645), (6, 15644), (7, 3305), (7, 12361), (7, 18601), (8, 13872), (8, 14205), (8, 3975), (9, 2773), (9, 14497), (10, 14), (10, 11332), (10, 13511), (11, 20092), (11, 8106), (11, 12130), (11, 19489), (12, 4683), (12, 21430), (12, 18391), (12, 18059), (12, 6008), (12, 19356), (13, 22435), (13, 1744), (13, 395), (13, 16399), (13, 9001), (13, 2280), (13, 6113), (13, 7235), (13, 213), (13, 7431), (13, 501), (13, 11971), (13, 16282), (13, 11209), (13, 1494), (13,
```

Figura 3.15: Lista di coppie nodo_from e nodo_to

Di fatto, abbiamo un'unica lista costituita da un numero di coppie pari al numero degli archi ed ogni coppia è costituita dal nodo di partenza e dal nodo di arrivo.

Aggiungiamo ora gli archi alla rete creata, attraverso il metodo *add_edges_from*:

```
G.add_edges_from(edges)
```

Figura 3.16: Aggiunta dei collegamenti

A questo punto G rappresenta la rete completa costituita dai nodi e dai relativi archi.

3.1 Statistiche elementari – rete di partenza

È possibile calcolare alcune informazioni e statistiche elementari che permettono di descrivere la rete creata. In prima battuta, verifichiamo il numero di nodi e di archi che la compongono.

```
print("LUNGHEZZA DEI NODI E' " , len(G.nodes))
print("LUNGHEZZA DEGLI ARCHI E' " , len(G.edges))

LUNGHEZZA DEI NODI E'  22470
LUNGHEZZA DEGLI ARCHI E'  171002
```

Figura 3.17: Dimensioni rete

Possiamo affermare che la rete è costituita da 22.470 nodi e da un numero di archi pari a 171.002.

È possibile verificare se il grafo che identifica la rete è di tipo diretto o indiretto, attraverso la funzione `is_directed()` di networkx:

```
G.is_directed()

False
```

Figura 3.18: Verifica tipologia grafo

Applicando la funzione alla nostra rete, l'output è “False”, questo significa che il nostro grafo è indiretto. Risultato giustificabile dal fatto che la rete rappresenta un collegamento tra pagine facebook in cui i collegamenti sono i mi piace reciproci; quindi essendo collegamenti che costituiscono una relazione simmetrica si ipotizzava il grafo fosse indiretto.

È possibile calcolare anche il valore della densità del grafo, il quale è un numero compreso tra 0 ed 1. La densità è 0 per un grafo senza archi e 1 per un grafo completo.

```
density= nx.density(G)

print(density)

0.000677398715568023
```

Figura 3.19: Calcolo densità

Nel nostro caso la densità del grafo è pari ad un valore 0.00067, quindi presenta una densità piuttosto bassa.

A partire da queste statistiche elementari quindi si è visto che la rete considerata è costituita da 22.470 nodi e 171.002 archi, quindi una rete abbastanza complessa. Risulta difficile visualizzare a colpo d'occhio una rete di queste dimensioni e anche calcolare alcuni parametri diventa oneroso in termini di tempo di calcolo, data la potenza di calcolo disponibile. Quindi sono state effettuate una serie di operazioni volte a ridurre le dimensioni della rete.

Ovviamente è stato seguito un criterio di riduzione, che consiste nel filtrare i nodi in base alla centralità. Nel dettaglio, si sono considerati solo i nodi con un valore di centralità maggiore di 20.

Dal punto di vista implementativo:

```
print("Graph is connected: ")
print(nx.is_connected(G))
print(" ")
print("Number of connected components: ")
print(nx.number_connected_components(G))

components = nx.connected_components

x = nx.node_connected_component(G, 0)
y = max(nx.connected_components(G), key=len)

G_1 = nx.Graph()
G_1 = G.subgraph(nodes=y)

Graph is connected:
True

Number of connected components:
1
```

Figura 3.20: Verifica connessioni

Il codice precedente permette di valutare se la rete presa in esame è connessa ed individuare eventualmente il numero di componenti connesse. G_1 sarà la rete con la componente connessa più grande. Nel nostro caso si ha un'unica componente connessa che coincide con la rete G stessa.

Di fatto la lunghezza dei nodi e degli archi è pari a:

```
print("NODES: ", len(G_1.nodes))
print("EDGES: ", len(G_1.edges))

NODES: 22470
EDGES: 171002
```

Figura 3.21: Dimensioni rete G_1

Dunque, a questo punto inizia la vera e propria riduzione. Come già discusso, andremo ad eliminare tutti quei nodi il cui valore di centralità risulta essere minore di 20. Ricordiamo che il grado di un nodo è il numero complessivo dei legami che esso possiede.

Dal punto di vista del codice si ha:

```
G_reduce = nx.Graph(G_1)
e = list(G_1.nodes)

for cont in e:
    if (G_reduce.degree[cont] < 20):
        G_reduce.remove_node(cont)
```

Figura 3.22: Riduzione e creazione rete G_reduce

A questo punto verifichiamo la nuova dimensione della rete andando a calcolare la lunghezza degli archi e dei nodi:

```
print("NODES: ", len(G_reduce.nodes))
print("EDGES: ", len(G_reduce.edges))
G_reduce.degree
```

```
NODES: 4145  
EDGES: 81246
```

Figura 3.23: Verifica dimensioni rete G_reduce

La rete ridotta presenta un numero di nodi pari a 4.145 ed un numero di archi pari a 81.246. A questo punto andremo a considerare la più grande componente连通的:

```
y_1 = max(nx.connected_components(G_reduce), key=len)  
G_reduce_1 = G_reduce.subgraph(nodes=y_1)
```

Figura 3.24: Scelta della componente连通的 più grande

```
print("NODES: ", len(G_reduce_1.nodes))  
print("EDGES: ", len(G_reduce_1.edges))  
  
NODES: 4138  
EDGES: 81245
```

Figura 3.25: Dimensioni G_reduce_1

In definitiva la rete da considerare nelle successive analisi è G_reduce_1, che presenta 4.138 nodi e 81.245 archi.

3.2 Statistiche elementari – rete ridotta

In analogia alle prime analisi effettuate sulla rete iniziale più complessa, si procede al calcolo di alcune delle statistiche elementari per poter descrivere la nuova rete ridotta.

In primis, viene calcolato il valore della densità della rete, ossia il rapporto tra il numero di collegamenti presenti nella rete e il numero massimo di collegamenti possibili

```
density = nx.density(G_reduce_1)
print(density)
0.00949184486438561
```

Figura 3.26: Calcolo densità G_reduce_1

Notiamo che in questo caso la densità del grafo è pari a 0.0094. È un valore maggiore rispetto al valore della rete non ridotta e questo è giustificabile dall'operazione di riduzione che è stata effettuata. Rimuovendo alcuni nodi e mantenendo solo quelli più centrali il valore di densità è aumentato ma comunque di molto poco.

Vengono anche calcolate anche le misure generali di connessione come la transitività, una misura del grado in cui i nodi di un grafo tendono ad essere connessi fra loro:

```
transitivity= nx.transitivity(G_reduce_1)
print(transitivity)
0.30906317976584097
```

Figura 3.27: Calcolo transitività G_reduce_1

L'indice varia da 0 a 1, dove 1 sta ad indicare che il grafo è completamente transitivo. Nelle reti sociali solitamente l'indice di transitività varia tra 0.3 e 0.6. Nel nostro caso particolare, notiamo che il valore di transitività della rete è 0.30.

3.3 Centralità

È possibile ora entrare nel vivo del calcolo di alcuni parametri interessanti relativi alla centralità.

Il grado, come misura di connessione: maggiore è il grado, maggiore è probabilmente il potere dell'attore, in quanto dispone di maggior libertà nella scelta d'uso dei propri legami o se si preferisce, è meno dipendente dagli altri.

La prossimità, come misura di distanza (vicinanza) dagli altri attori: minore è la distanza (espressa in termini di lunghezza dei percorsi geodeticci), maggiore può essere il potere derivante dall'essere un "punto di riferimento" per gli altri attori, poterli raggiungere facilmente, etc.

La betweenness, come misura del ruolo di connettore di altri attori, assumendo una funzione di broker: maggiore la betweenness, maggiore è probabilmente il potere posseduto.

Le misure di centralità rispondono alla domanda: "Chi è la persona più importante o centrale in questa rete?" In realtà però ci sono molte risposte a questa domanda, a seconda di cosa intendiamo per importanza.

3.3.1 Degree Centrality

Essa è calcolata andando a contare il numero di legami che un nodo ha con gli altri nodi della rete. Quindi tale metrica esprime, nel nostro specifico caso, il grado di connettività di una certa pagina con le altre pagine, permettendo di capire se il nodo considerato ha una posizione strutturale rilevante nella rete di cui fa parte. NetworkX calcola tale metrica normalizzata per il massimo grado possibile di ogni nodo, con il vantaggio di ottenere un valore percentuale con cui è possibile effettuare confronti anche tra nodi appartenenti a sottoreti distinte.

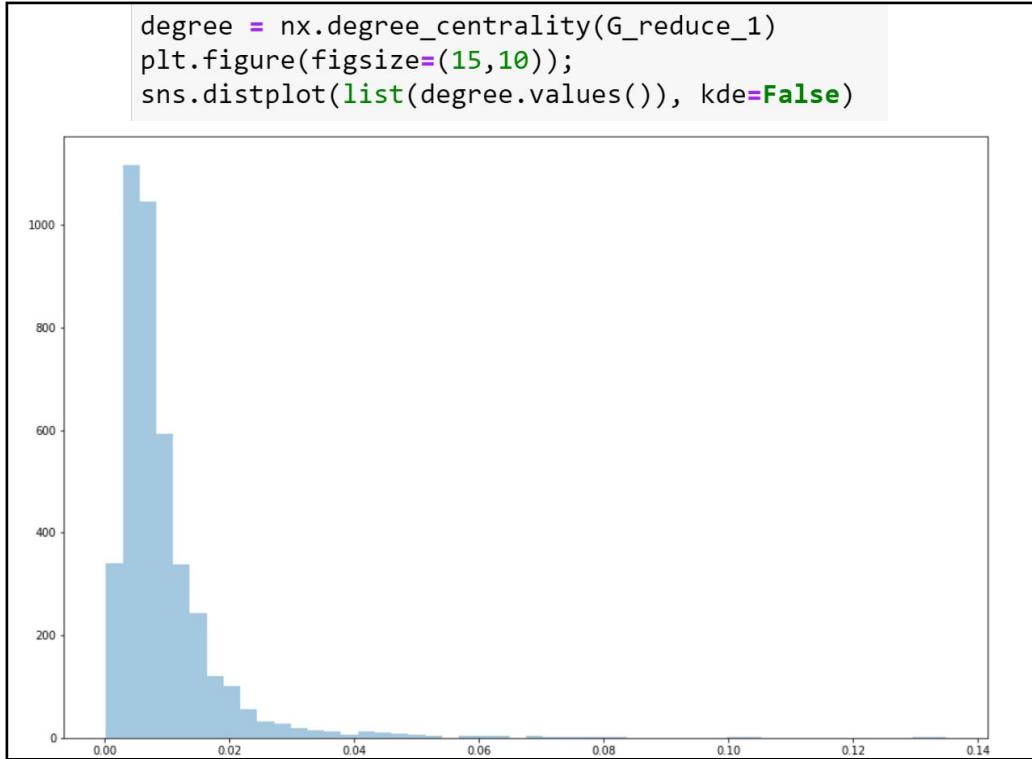


Figura 3.28: Codice e grafico della degree centrality

Da grafico della degree centrality appena calcolata si comprende come alcuni e pochi nodi hanno un valore altissimo di degree centrality e molti invece ne presentano un valore molto basso, tendente allo zero. Questo è un andamento tipico all'interno delle reti sociali ed in gergo, questo andamento esponenziale è noto come power-low. La conseguenza di questa cosa, infatti, è che se si prendono pochissime persone o pagine in una rete (quelle con più follower di tutti) riesce a raggiungere la maggior parte delle persone o delle pagine di quella rete.

Definendo una funzione di questo tipo:

```
def draw(G, pos, measures, measure_name):
    nodes = nx.draw_networkx_nodes(G, pos, node_size=250, cmap=plt.cm.plasma,
                                    node_color=list(measures.values()),
                                    nodelist=measures.keys())
    nodes.set_norm(mcolors.SymLogNorm(linthresh=0.01, linscale=1))

    # labels = nx.draw_networkx_labels(G, pos)
    edges = nx.draw_networkx_edges(G, pos)

    plt.title(measure_name)
    plt.colorbar(nodes)
    plt.axis('off')
    plt.show()
```

Figura 3.29: Definizione funzione draw

Possiamo “disegnare” e rappresentare in maniera differente la rete e le varie centralità. In particolare, possiamo esprimere la Degree Centrality nelle seguenti maniere:

Utilizzando il metodo `spring_layout` possiamo graficare la nostra rete come nella figura seguente. Posiziona i nodi utilizzando l’algoritmo di Fruchterman-Reingold diretto dalla forza. L’algoritmo simula una rappresentazione a forza della rete trattando gli archi come molle che tengono i nodi vicini, e tratta i nodi come oggetti che si respingono. La simulazione continua fino a quando le posizioni sono vicine ad un equilibrio.

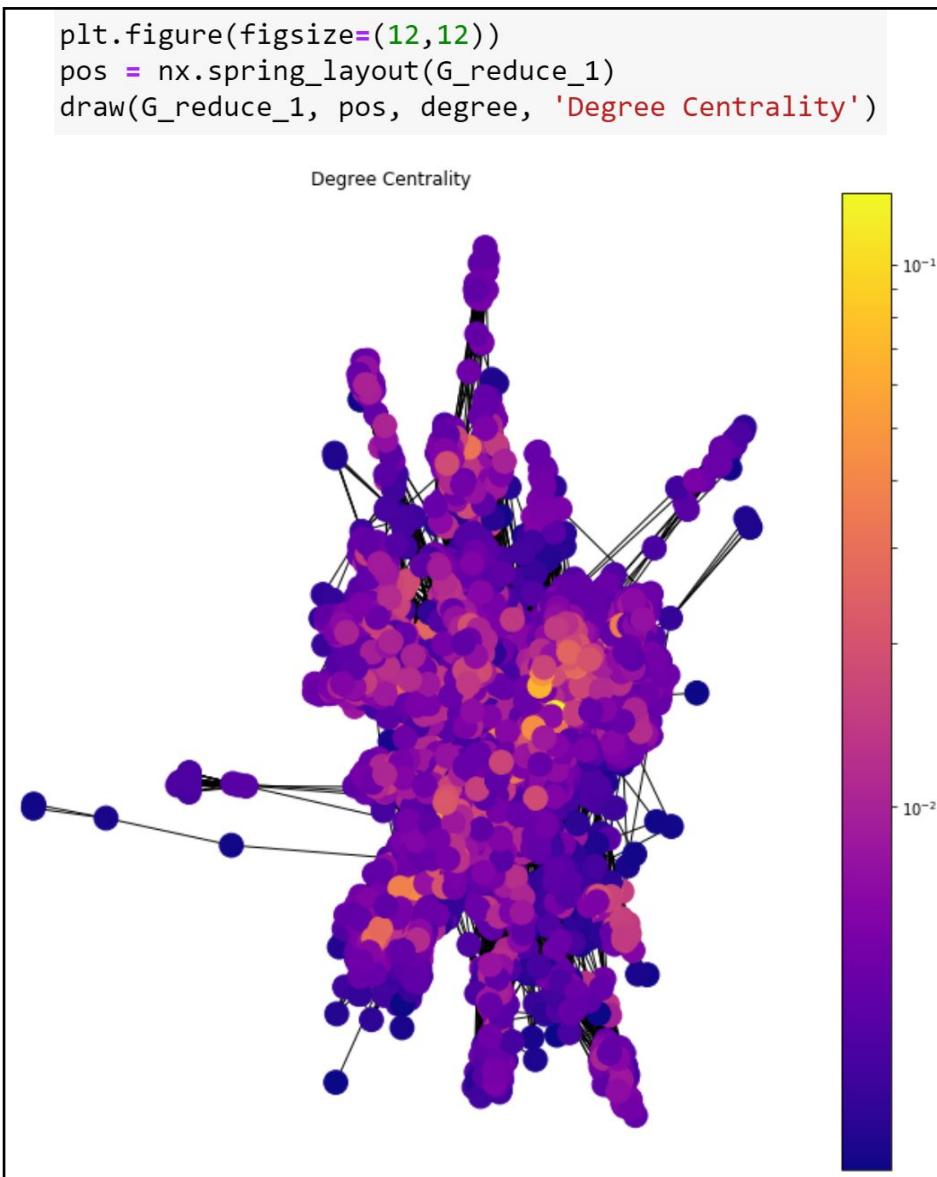


Figura 3.30: Codice e grafico degree centrality attraverso `spring_layout`

Nel nostro caso notiamo che, ogni nodo della rete è associato ad un colore rappresentato dalla scala in figura. Nonostante i punti siano molto addensati tra di loro vediamo come alcuni nodi centrali hanno un colore tendente all’arancione e quindi a media centralità, pochissimi nodi hanno un colore tendente al giallo e quindi con elevata degree centrality. Quasi tutti i nodi periferici tendono ad un colore scuro ad indicare un basso valore di degree centrality. Questa situazione rispecchia a tutti gli effetti il grafico nelle due figure precedenti. Di fatto è un’altra rappresentazione della stessa entità.

Volendo un'altra rappresentazione della Degree Centrality, attraverso il metodo “spiral_layout” è possibile distribuire i nodi della rete secondo un layout a spirale, come in figura:

```
plt.figure(figsize=(12,12))
pos = nx.spiral_layout(G_reduce_1)
draw(G_reduce_1, pos, degree, 'Degree Centrality')
```

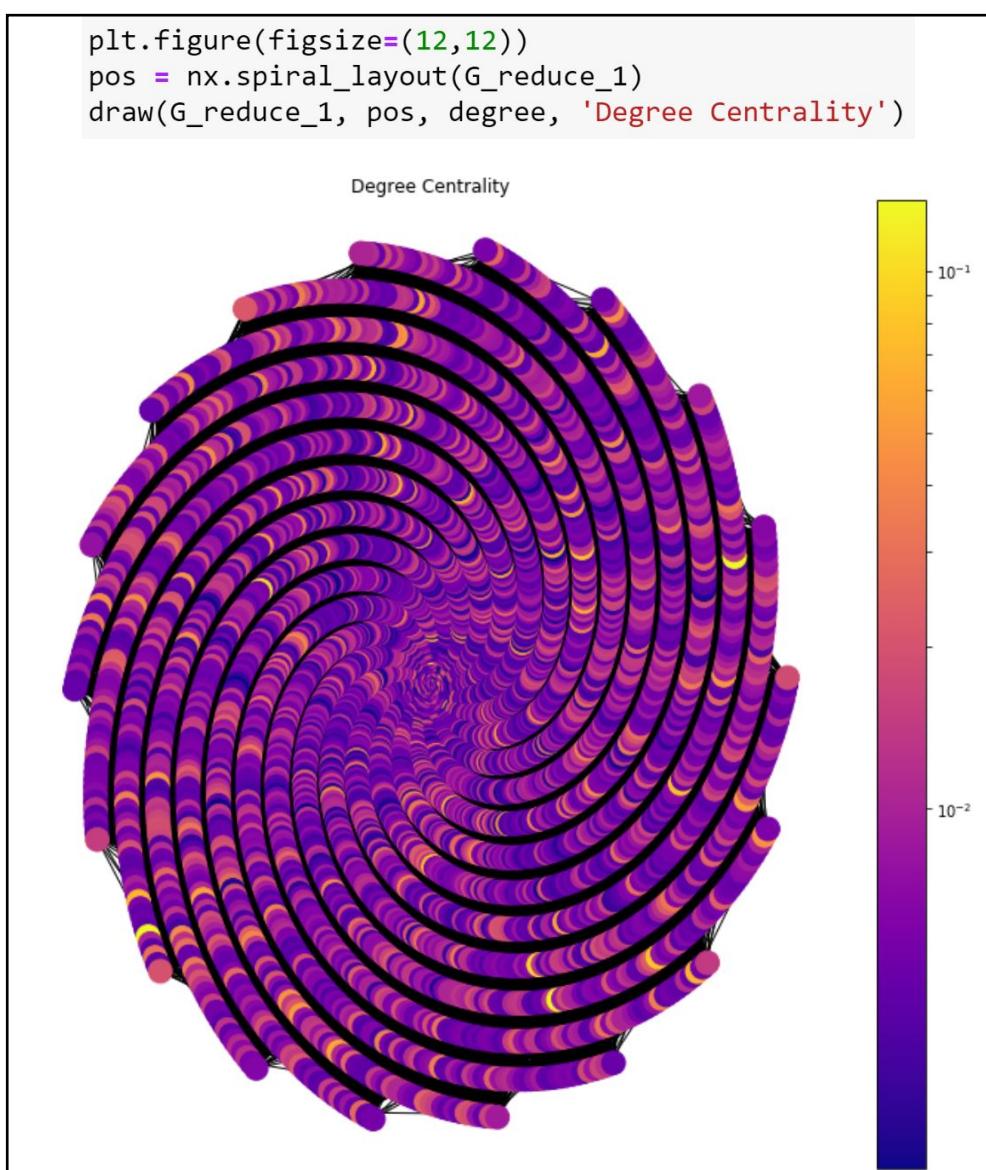


Figura 3.31: Codice e grafico degree centrality attraverso spiral_layout

Successivamente dopo aver calcolato le centralità, quindi, sono stati ordinati i vari nodi prendendo in considerazione i 10 nodi e quindi le 10 pagine Facebook con valore di Degree Centrality più alto:

```
sorted_dc = sorted(degree.items(), key= lambda item: item[1], reverse= True)

print('Best 10 Degree Centrality:')
for i in range(10):
    print(str(i+ 1)+ ' ' + str(sorted_dc[i][0])+ ':' + str(sorted_dc[i][1]))
```

Best 10 Degree Centrality:

1. 19743: 0.1348803480783176
2. 21729: 0.13246313753927969
3. 16895: 0.1310128112158569
4. 1387: 0.10394005317863186
5. 14497: 0.10248972685520909
6. 10379: 0.10055595842397873
7. 19347: 0.08170171621948272
8. 8139: 0.07855934251873338
9. 10426: 0.07855934251873338
10. 15236: 0.077834179357022

Figura 3.32: Ordinamento e stampa dei nodi più centrali

L'output sopra mostra quindi i primi 10 nodi con valore di Degree Centrality più elevato. Abbiamo una lista di 10 elementi in cui per ogni id del nodo si ha il corrispondente valore di centralità. Solo con queste informazioni non riusciamo a comprendere quali siano, di fatto, le pagine con tali valori di centralità. Si ottengono le pagine seguenti con i relativi attributi:

```
1. {'facebookid': 145647315578475, 'pagename': 'The Voice of China 中国好声音', 'pagetype': 'tvshow'}
2. {'facebookid': 191483281412, 'pagename': 'U.S. Consulate General Mumbai', 'pagetype': 'government'}
3. {'facebookid': 144761358898518, 'pagename': 'ESET', 'pagetype': 'company'}
4. {'facebookid': 568700043198473, 'pagename': 'Consulate General of Switzerland in Montreal', 'pagetype': 'government'}
5. {'facebookid': 1408935539376139, 'pagename': 'Mark Bailey MP - Labor for Miller', 'pagetype': 'politician'}
6. {'facebookid': 134464673284112, 'pagename': 'Victor Dominello MP', 'pagetype': 'politician'}
7. {'facebookid': 282657255260177, 'pagename': 'Jean-Claude Poissant', 'pagetype': 'politician'}
8. {'facebookid': 239338246176789, 'pagename': 'Deputado Ademir Camilo', 'pagetype': 'politician'}
9. {'facebookid': 544818128942324, 'pagename': 'T.C. Mezarı Şerif Başkonsolosluğu', 'pagetype': 'government'}
10. {'facebookid': 285155655705, 'pagename': 'Army ROTC Fighting Saints Battalion', 'pagetype': 'government'}
```

Figura 3.33: Stampa dei 10 nodi più centrali con relativi attributi

Dalla lista delle 10 pagine con centralità più alta, possiamo notare come le categorie delle pagine siano molto eterogenee. Troviamo tutte e quattro le categorie presenti e quindi pagine relative sia a politici, a organizzazioni governative, a spettacoli televisivi e ad aziende. Però tra la molteplicità di queste categorie vediamo come quella con centralità più alta è relativa alle pagine televisive, e corrisponde alla pagina "The Voice of Cina". Nel resto di questa top 10, il resto delle pagine è costituita sostanzialmente da pagine di politici e pagine di organizzazioni governative, una sola pagina è relativa alle aziende. Potremmo dedurre che, per quanto riguarda le pagine televisive, di base non hanno un elevata Degree Centrality, ma quelle poche che ne presentano un valore elevato tendono ad avere molto potere e molta visibilità e risultano soprattutto pagine influenti. Mentre, considerando sempre la lista delle top 10 Degree Centrality, abbiamo che i politici e le organizzazioni

governative ricoprono dei ruoli chiave all'interno della rete. In particolare, politici come “Mark Bailey”, “Victor Dominello”, “Jean-Claude Poissant” e “Ademir Camilo” sono politici particolarmente influenti, sono quelli che si trovano vicini all’azione e sono quelli che godono della maggiore visibilità, nonché accesso al maggior quantitativo di risorse, perché il numero di contatti a cui possono attingere è più alto.

The Voice of Cina



The Voice of China è un concorso di canto televisivo cinese trasmesso su Zhejiang Television. Basato sull'originale The Voice of Holland, il concetto della serie è quello di trovare nuovi talenti canori (solisti o duetti) contestati da aspiranti cantanti tratti da audizioni pubbliche. Il vincitore è determinato dai voti espressi da una giuria dei media e dal pubblico dal vivo. Inoltre, tali vincitori ricevono un contratto discografico con varie etichette. Fino ad ora sono state mandate in onda quattro stagioni dal 2012 al 2015 ed i vincitori sono stati: Bruce Liang, Li Qi, Diamond Zhang e Zhang Lei.

3.3.2 Betweenness Centrality

La seconda metrica relativa alle misure di centralità calcolate risulta essere la Betweenness Centrality. Essa è legata al numero di volte in cui un nodo si ritrova lungo il percorso più breve tra le altre coppie di nodi della rete, descrivendo così la capacità di diffusione delle informazioni nella rete attraverso quel nodo. Si può affermare, in altre parole, che misura la strategicità di un nodo nella rete tra due aree importanti della stessa. All'interno delle Social Network, un nodo (un individuo o una pagina) con un'elevata Betweenness Centrality ha una grande influenza nel flusso di informazioni. Inoltre, ha anche un altro ruolo: essa è in grado di identificare gli individui o le pagine che agiscono da bridge, quindi da ponte tra due o più comunità che viceversa non riuscirebbero a comunicare l'una con l'altra.

Ogni nodo del grafo assume dei valori di Betweenness Centrality compresi da 0 a 1: Maggiore è il valore della centralità e quindi tendente ad uno e maggiore sarà la posizione di potere assunta dal nodo all'interno della rete. Significa quindi che la maggior parte delle comunicazioni che avvengono tra elementi delle diverse sottoreti, dovranno passare proprio per il nodo in questione. Inoltre, sono tutti quei nodi la cui rimozione dalla rete interromperà maggiormente le comunicazioni tra gli altri vertici, perché si trovano sul maggior numero di percorsi presi dai messaggi.

Dal punto di vista implementativo possiamo calcolare la Betweenness Centrality e poter visualizzare i suoi valori graficamente:

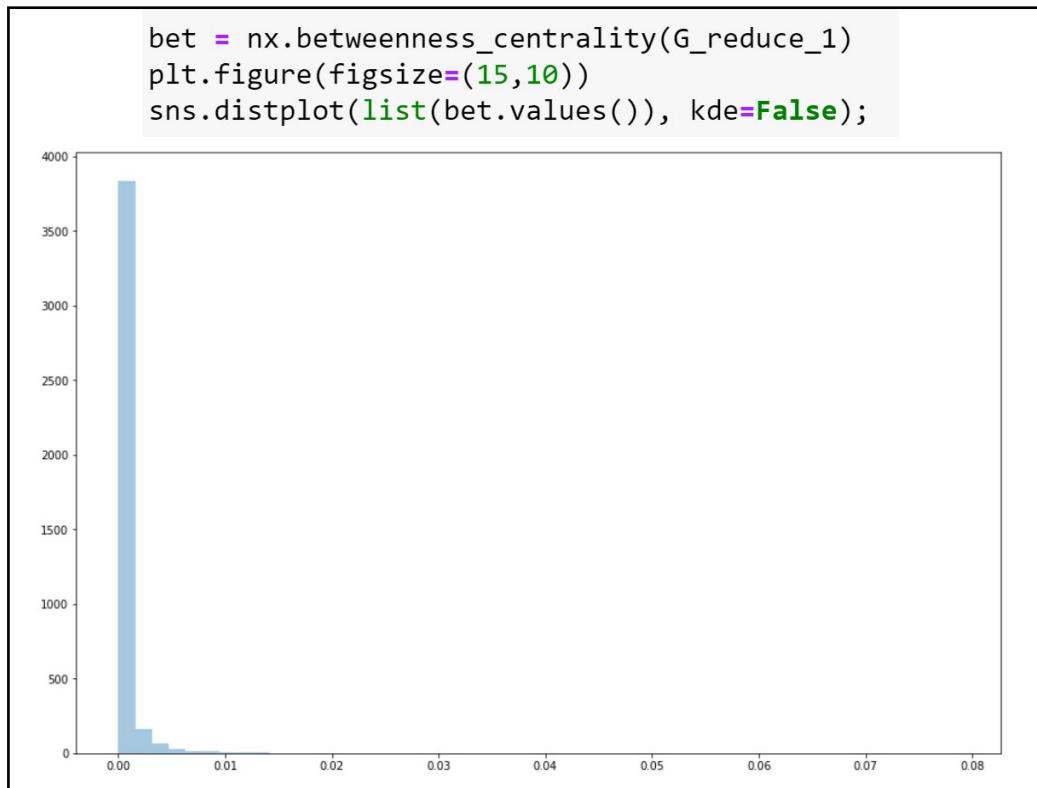


Figura 3.34: Codice e grafico della betweenness centrality

Vediamo dal grafico precedente, che in maniera simile ma non uguale alla Degree Centrality ci sono pochissimi nodi con elevata Betweenness, e quindi pochissimi con un ruolo strategico tipico di questa centralità. La maggior parte degli altri nodi hanno dei valori bassissimi di Betweenness, e quindi ci sono tanti nodi che non hanno una grande influenza nel flusso di informazioni. Anche in questo caso troviamo l'andamento tipo delle reti sociali, noto appunto come power-low. Questa distribuzione in particolare è addirittura molto esponenziale.

Come nel caso precedente andiamo ad individuare quali sono i primi 10 nodi con maggiore Betweenness Centrality:

```
sorted_bc = sorted(bet.items(), key= lambda item: item[1], reverse= True)

print('Best 10 Betweennes Centrality:')
for l in range(10):
    print(str(l+ 1)+ ' . ' + str(sorted_bc[l][0])+ ': ' + str(sorted_bc[l][1]))

    Best 10 Betweennes Centrality:
1. 11003: 0.0787759998095439
2. 21729: 0.06949968875077028
3. 19743: 0.06904391335892893
4. 701: 0.03652981596309869
5. 22171: 0.0326337450468189
6. 10379: 0.03194151852481163
7. 21120: 0.03015734211268552
8. 18819: 0.02602851325768117
9. 16895: 0.02512337424423291
10. 11611: 0.024229502777339287
```

Figura 3.35: Ordinamento e stampa dei nodi più centrali

Da questo primo risultato possiamo dedurre che i primi dieci nodi con maggior valore di Betweenness, in realtà hanno comunque dei bassi valori di tale centralità. Il valore per cui una certa pagina Facebook funge da collegamento tra comunità che lavorano su argomenti diversi è sostanzialmente basso.

È possibile individuare quali sono le pagine ed ulteriori informazioni relative a questi nodi con maggiore centralità:

```
{'facebookid': 6815841748, 'pagename': 'Barack Obama', 'pagetype': 'politician'}
{'facebookid': 63811549237, 'pagename': 'The Obama White House', 'pagetype': 'government'}
{'facebookid': 1191441824276882, 'pagename': 'The White House', 'pagetype': 'government'}
{'facebookid': 20531316728, 'pagename': 'Facebook', 'pagetype': 'company'}
{'facebookid': 38802554124, 'pagename': 'U.S. Embassy Ottawa', 'pagetype': 'government'}
{'facebookid': 15877306073, 'pagename': 'U.S. Department of State', 'pagetype': 'government'}
{'facebookid': 178362315106, 'pagename': 'European Parliament', 'pagetype': 'government'}
{'facebookid': 601163706652420, 'pagename': 'Niels Annen', 'pagetype': 'politician'}
{'facebookid': 44053938557, 'pagename': 'U.S. Army', 'pagetype': 'government'}
{'facebookid': 21751825648, 'pagename': 'Justin Trudeau', 'pagetype': 'politician'}
```

Figura 3.36: Stampa dei 10 nodi più centrali con relativi attributi

Notiamo che le pagine con maggiore centralità sono essenzialmente pagine Facebook dei politici o pagine governative. Questo risultato ci fa comprendere come l'entità che funge da ponte tra diverse comunità sono le categorie dei politici o di entità governative. Come nodo particolarmente influente troviamo la pagina relativa all'ex presidente degli Stati Uniti Barack Obama. Se affermiamo che i nodi con valori alti di Betweenness, in un certo senso sono i nodi più efficienti del flusso di comunicazione della rete, possiamo affermare che la pagina ufficiale di Barack Obama, e dunque Barack Obama sia l'individuo più efficiente del flusso di comunicazione di tutta la rete considerata. Sul podio, non a caso, segue la pagina "The Obama White House" e "The White House".

Barack Obama



Barack Hussein Obama nasce in Honolulu il 4 agosto 1961. Dal punto di vista politico, è stato membro del Senato dell'Illinois per tre mandati, dal 1997 al 2004. Dopo essersi candidato senza successo alla Camera dei rappresentanti nel 2000, quattro anni più tardi concorse per il Senato federale, imponendosi a sorpresa nelle primarie del Partito Democratico del marzo 2004 su un folto gruppo di contendenti. La vittoria alle primarie contribuì ad accrescere la sua notorietà; in seguito, il suo discorso introduttivo pronunciato in occasione della convention democratica di luglio lo rese una delle figure più eminenti del suo partito. Obama fu quindi eletto al Senato degli Stati Uniti nel novembre 2004 e prestò servizio come senatore junior dal gennaio 2005 al novembre 2008. Il ruolo di presidente lo ricopre a partire dal gennaio del 2009 e termina dopo due mandati nel gennaio 2017. Barack Obama, da un punto di vista più umano è stato un leader molto amato e simbolo di speranza, e non solo per essere stato il primo uomo di colore a sedere alla Casa Bianca. Della sua lunga presidenza vengono ricordate soprattutto la sua azione per il disarmo nucleare (in particolare in Iran) e la riforma del sistema sanitario, ma anche il suo attivismo in termini di diritti civili, di controllo delle armi e sul tema del cambiamento climatico. Delicata la questione della politica estera in Medio Oriente (in particolare in Iraq e Afghanistan), che gli è valsa più di una critica. Ma risulta innegabile come molte delle sue azioni, soprattutto sul fronte delle relazioni internazionali in ottica di pace, abbiano comunque battuto la strada per il progresso mondiale.

3.3.3 Closeness Centrality

La terza metrica relativa alle “misure di centralità” è quella chiamata Closeness Centrality. Questo tipo di centralità è calcolata come il reciproco della somma delle lunghezze dei percorsi più brevi, tra un nodo e tutti gli altri nodi della rete. Una metrica di questo tipo permette quindi di esprimere il grado di prossimità di un nodo agli altri nodi della rete. In sostanza, più un nodo è centrale secondo questa metrica, più è vicino a tutti gli altri nodi.

Si potrebbe pensare per questa metrica ad una particolare analogia. Alcune persone sono a pochi chilometri da una grande città, altre devono guidare per ore: allo stesso modo, i nodi con bassi valori di Closeness Centrality hanno molti chilometri o meglio connessioni che devono percorrere per raggiungere molti altri nodi nella rete, e vale il viceversa. Comunque, analogia a parte. dal punto di vista del codice, si ha il seguente script che permette di calcolare il valore della centralità discussa e di mostrarne anche un grafico:

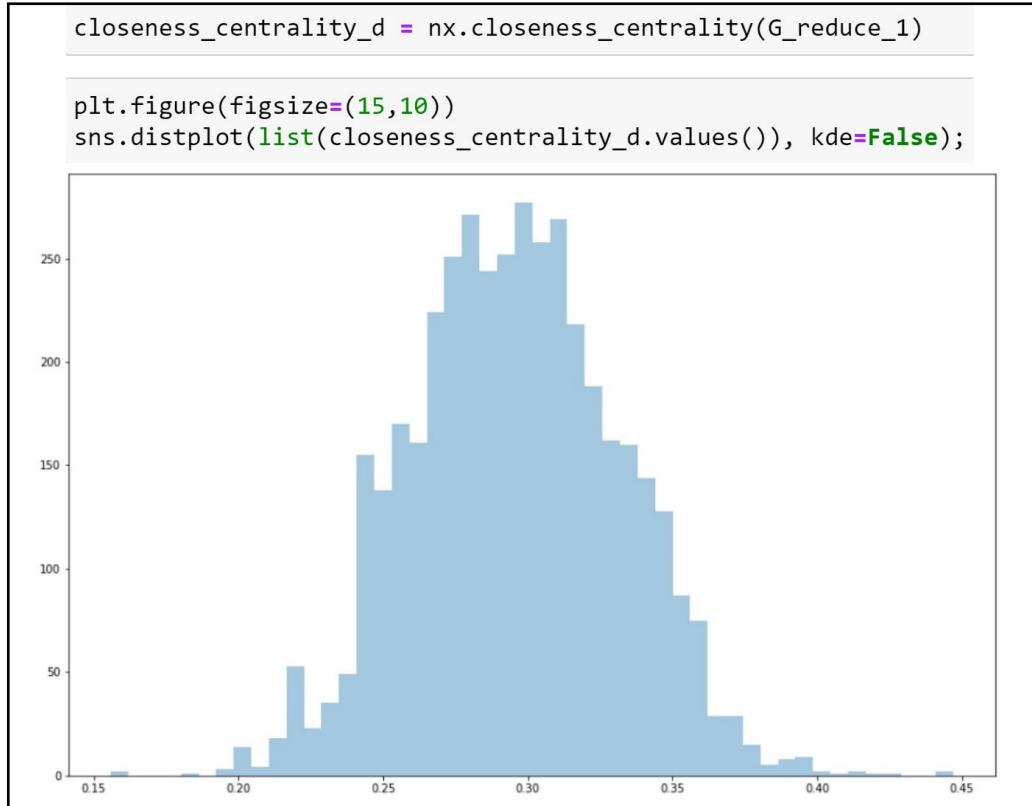


Figura 3.37: Codice e grafico della closeness centrality

L’output relativo è nella figura precedente. Vediamo che in questo caso non si ha più una distribuzione di tipo power-low, ma l’andamento è approssimabile alla forma di una “campana”. La distribuzione, quindi, è complimentamente diversa dalle precedenti.

Come negli altri casi, andremo a individuare quali sono i primi dieci nodi con maggiore Closeness Centrality:

```
sorted_cc = sorted(closeness_centrality_d.items(), key= lambda item: item[1], reverse= True)

print('Best 10 Closeness Centrality:')
for l in range(10):
    print(str(l+ 1)+ '. ' + str(sorted_cc[l][0])+ ':' + str(sorted_cc[l][1]))
```

Best 10 Closeness Centrality:
1. 19743: 0.4470982384091646
2. 21729: 0.4469533275713051
3. 11003: 0.4228764182766023
4. 10379: 0.4219275879653238
5. 22171: 0.4150697301093609
6. 16895: 0.4112734864300626
7. 701: 0.4068246631920543
8. 1387: 0.40455701153921375
9. 8482: 0.4035310183378853
10. 19347: 0.3969868534689569

Figura 3.38: Ordinamento e stampa dei nodi più centrali

Ed otteniamo le seguenti pagine Facebook:

```
{'facebookid': 1191441824276882, 'pagename': 'The White House', 'pagetype': 'government'}
{'facebookid': 63811549237, 'pagename': 'The Obama White House', 'pagetype': 'government'}
{'facebookid': 6815841748, 'pagename': 'Barack Obama', 'pagetype': 'politician'}
{'facebookid': 15877306073, 'pagename': 'U.S. Department of State', 'pagetype': 'government'}
{'facebookid': 38802554124, 'pagename': 'U.S. Embassy Ottawa', 'pagetype': 'government'}
{'facebookid': 44053938557, 'pagename': 'U.S. Army', 'pagetype': 'government'}
{'facebookid': 20531316728, 'pagename': 'Facebook', 'pagetype': 'company'}
{'facebookid': 155837727772692, 'pagename': 'Honolulu District, U.S. Army Corps of Engineers', 'pagetype': 'government'}
{'facebookid': 146599018696771, 'pagename': 'NATO', 'pagetype': 'government'}
{'facebookid': 78922439964, 'pagename': 'FEMA Federal Emergency Management Agency', 'pagetype': 'government'}
```

Figura 3.39: Stampa dei 10 nodi più centrali con relativi attributi

È interessante notare come per quanto riguarda la Closeness, le pagine con maggior centralità risultano essere quelle governative. Si hanno anche due pagine di diversa categoria, una di tipo politico e l'altra relativa alle aziende. Nel dettaglio, la pagina più centrale secondo tale metrica è “The White House”, seguita da “The Obama White House” e dalla pagina ufficiale del politico “Barack Obama”. Questo risultato si incassa con il risultato della metrica relativa alla Betweenness, in cui la pagina Barack Obama era la più centrale. Quindi, un personaggio come Barack Obama è allo stesso tempo sia un personaggio che funge da ponte e che permette quindi di mettere in collegamento diverse comunità e allo stesso tempo ricopre una figura centrale in termini di vicinanza rispetto agli altri nodi. Non a caso, Barack Obama nel periodo temporale di questa analisi, nel 2017, era il presidente degli Stati Uniti d’America, e quindi tali centralità rispecchiano il ruolo ricoperto. Nel dettaglio però, la presidenza di Barack Obama termina nel gennaio del 2017, e il dataset in esame presenta una finestra temporale che termina nel novembre 2017. Questo significa che anche nei mesi in cui Obama non deteneva il titolo di presidente ha continuato a mantenere un ruolo principale e particolarmente centrale all’interno della rete.

The White House



La Casa Bianca (o White House) è la residenza ufficiale del presidente degli Stati Uniti d'America, sede della presidenza stessa. Si trova a Washington, DC ed è stata la residenza di ogni presidente degli Stati Uniti da John Adams, dal 1800 in poi. Il termine "Casa Bianca" è spesso usato per metonimia per riferirsi agli uffici del presidente e dei suoi consiglieri. La residenza è stata progettata dall'architetto irlandese James Hoban in stile neoclassico. Hoban ha modellato l'edificio sulla Leinster House a Dublino, un edificio che oggi ospita l'Oireachtas , la legislatura irlandese. La costruzione avvenne tra il 1792 e il 1800 utilizzando l'arenaria Aquia Creek dipinta di bianco. Dal punto di vista della pagina Facebook, permette sicuramente la diffusione, sull'enorme canale quale è Facebook, di tutte le informazioni pubbliche legate al mondo governativo-amministrativo e risulta dunque il canale attraverso il quale veicolano tutti i provvedimenti, avvisi, indicazioni e informazioni rilasciate dal presidente degli Stati Uniti.

3.3.4 Eigenvector Centrality

La quarta forma di centralità presa in esame si chiama Eigenvector Centrality e dice questo: un nodo è tanto più centrale quanti più sono i suoi collegamenti e quanto più ciascuno dei suoi collegamenti è centrale. In sostanza, un nodo è importante se è collegato ad altri nodi importanti. Quindi capiamo bene che è onerosa da calcolare in quanto per definire la centralità di un nodo devo definire la centralità degli altri nodi. Dato che la nostra rete è stata ridotta risulta meno complesso calcolare quindi il valore di tale centralità.

Dal punto di vista implementativo, si ha il seguente script per calcolare il valore di “Eigenvector Centrality” e successivamente vengono ordinati i valori:

```
eigenvector_centrality = nx.eigenvector_centrality(G_reduce_1)

sorted_ec = sorted(eigenvector_centrality.items(), key= lambda item: item[1], reverse= True)
```

Figura 3.40: Codice dell'eigenvector centrality

Per poter mostrare quali sono i primi dieci nodi con maggior centralità:

```
print('Best 10 Eigenvector Centrality:')
for l in range(10):
    print(str(l+ 1)+ ' . ' + str(sorted_ec[l][0])+ ': ' + str(sorted_ec[l][1]))
```

Best 10 Eigenvector Centrality:
1. 16895: 0.17114325901284388
2. 14497: 0.15123194312910815
3. 1387: 0.13632111675016137
4. 8139: 0.12034415104638818
5. 2442: 0.11756984138162581
6. 19743: 0.11634807401077905
7. 4502: 0.11623413658568531
8. 21729: 0.11540241940884859
9. 15236: 0.10855149631347016
10. 9220: 0.10792544232091943

Figura 3.41: Ordinamento e stampa dei nodi più centrali

Nel dettaglio, le prime dieci pagine Facebook sono le seguenti:

```
{'facebookid': 44053938557, 'pagename': 'U.S. Army', 'pagetype': 'government'}
{'facebookid': 404391086302925, 'pagename': 'U.S. Army Chaplain Corps', 'pagetype': 'government'}
{'facebookid': 155837727772692, 'pagename': 'Honolulu District, U.S. Army Corps of Engineers', 'pagetype': 'government'}
{'facebookid': 136880189673357, 'pagename': 'Defense Commissary Agency', 'pagetype': 'government'}
{'facebookid': 212025308879899, 'pagename': 'Army Training Network (ATN)', 'pagetype': 'government'}
{'facebookid': 1191441824276882, 'pagename': 'The White House', 'pagetype': 'government'}
{'facebookid': 119621891000, 'pagename': 'U.S. Army Materiel Command', 'pagetype': 'government'}
{'facebookid': 63811549237, 'pagename': 'The Obama White House', 'pagetype': 'government'}
{'facebookid': 119105629988, 'pagename': 'United States Air Force', 'pagetype': 'government'}
{'facebookid': 246854871491, 'pagename': 'U.S. Army Garrison Red Cloud', 'pagetype': 'government'}
```

Figura 3.42: Stampa dei 10 nodi più centrali con relativi attributi

Stando alla misura di Eigenvector Centrality notiamo che anche in questo caso i nodi più centrali risultano essere sempre pagine di tipo governativo. In questo caso però il podio è mantenuto da pagine quali “U.S. Army”, “U.S. Army Chaplain Corps” ed “Honolulu District, U.S. Army Corps of Engineers”. Queste sono quindi le pagine che sono più vicine ai nodi centrali. Cioè pagine Facebook riguardanti il settore militare.

Abbiamo infatti, tra le pagine trovate sia quelle relative all'esercito, sia all'aeronautica militare, sia ritroviamo la pagina "The White House". Questo perché ovviamente il settore militare è un'organizzazione governativa e quindi avrà dei collegamenti molto stretti con entità che abbiamo visto risultano particolarmente centrali nelle altre misure come appunto "The White House" e "Barack Obama". Nel nostro caso quindi, la pagina "U.S. Army" è quella più vicina e quindi a più stretto contatto con le pagine quali quella del Presidente degli USA e della Casa Bianca.

U.S Army



U.S.Army è la branca terrestre delle forze armate degli Stati Uniti d'America; l'obiettivo principale è quello di fornire le strategie e le capacità necessarie alla sicurezza e alla difesa nazionale degli Stati Uniti d'America e il suo controllo è affidato al Department of the Army, uno dei tre dipartimenti facenti capo al Dipartimento della Difesa. A capo del Dipartimento dell'Esercito è posto un funzionario civile che assume il titolo di Segretario dell'Esercito, equivalente ad un Sottosegretario di Stato alla Difesa italiano con delega all'Esercito, e assume il vertice della gerarchia dell'Esercito. All'apice della gerarchia militare è posto il Capo di Stato Maggiore dell'Esercito, che è il più alto Ufficiale in grado dell'Esercito. La legislazione americana prospetta per tale branca le seguenti funzioni:

- preservare la pace e la sicurezza, assicurare la difesa degli USA e dalle aree da essi occupate;
- sostenere le politiche nazionali;
- realizzare gli obiettivi di politica nazionale;
- combattere ogni nazione che possa nuocere alla pace e alla sicurezza degli Stati Uniti d'America.

Nella pagina Facebook dell'esercito degli Stati Uniti è possibile quindi trovare notizie, video e foto che mostrano le operazioni dei soldati statunitensi in tutto il mondo.

3.4 Community Detection and Network Visualization

Dopo aver calcolato le varie centralità è possibile passare all'individuazione delle comunità presenti nella rete e poter visualizzare la rete stessa. Per quanto riguarda le comunità, possiamo dire che i membri appartenenti ad una community si confrontano e hanno uno scambio di opinioni principalmente attorno ad un settore o, più precisamente, ad un argomento di interesse comune. Le community costituiscono quindi delle vere e proprie comunità virtuali che interagiscono e creano sinergie. In un contesto Facebook, le community costituiscono un prezioso strumento per consentire anche alle aziende di raccontarsi e interagire liberamente con i propri utenti, oltre ad essere molto utili per individuare i target di riferimento.

Dal punto di vista implementativo, si procede quindi ad importare i moduli necessari per il rilevamento delle comunità.

```
from networkx.algorithms import community
```

Figura 3.43: Importazione libreria

```
communities = community.greedy_modularity_communities(G_reduce_1)  
communities
```

Figura 3.44: Calcolo rilevamento comunità

Attraverso il metodo “greedy_modularity_communities” è possibile effettuare il rilevamento di comunità basate sulla modularità. In particolare, la precedente funzione trova comunità in un grafo utilizzando la massimizzazione della modularità golosa di Clauset-Newman-Moore.

L'output non sarà altro che un insieme di frozenset costituiti da tutti i nodi che formano una particolare comunità. Una porzione dei frozenset individuati è riportata nelle figure successive:

```
[frozenset({4097,  
          10246,  
          6151,  
          14,  
          16398,  
          4112,  
          6160,  
          18,  
          6162,  
          2068,  
          10261,  
          14356,  
          18448,  
          8216,  
          16408,  
          26,  
          2075,  
          6171,  
          18454,...},  
         frozenset({1146,  
          2195,  
          2964,  
          6927,  
          8199,  
          10287,  
          10831,  
          12701,  
          13918,  
          14865,  
          15799,  
          16177,  
          17518,  
          17583,  
          18532,  
          20452,  
          20776}),  
         frozenset({72, 1375, 2124, 3829, 3964, 6728, 14205}),  
         frozenset({217, 9000, 10734, 13385})]
```

Figura 3.45: Frozenset ottenuti

Nelle due figure precedenti, l'immagine di sinistra rappresenta il primo frozenset e quindi rappresenta la prima comunità che è costituita da molti altri nodi non entrati in figura. La figura a destra rappresenta invece gli ultimi frozenset individuati, e notiamo come questi siano costituiti da molti meno nodi rispetto il primo frozenset. In particolare, l'ultima comunità individuata risulta costituita da solo quattro nodi.

A questo punto, possiamo ordinare gli elementi presenti in communities e li salviamo all'interno di una variabile denominata “communitiess”.

```
communitiess = sorted(communities, key= len, reverse=True)
```

Figura 3.46: Ordinamento comunità

È possibile individuare quante comunità sono state individuate attraverso il semplice comando:

```
len(communitiess)  
12
```

Figura 3.47: Numero comunità

Dunque, sono state rilevate dodici comunità.

3.4.1 Network Visualization

In ottica “network analysis” è molto interessante poter graficare la rete che si è presi in esame. È possibile fare questo generando il seguente script in cui si definiscono alcuni parametri come, ad esempio, il grafo da rappresentare, la dimensione dei nodi della rete, il colore degli archi e la relativa trasparenza:

```
pos = nx.spring_layout(G_reduce_1, k=0.1)
plt.rcParams.update({'figure.figsize': (15, 10)})
nx.draw_networkx(
    G_reduce_1,
    pos=pos,
    node_size=0,
    edge_color="#444444",
    alpha=0.05,
    with_labels=False)
```

Figura 3.48: Impostazione parametri per visualizzazione rete

Il grafico che ne esce fuori è la rete presa in esame in questo progetto:

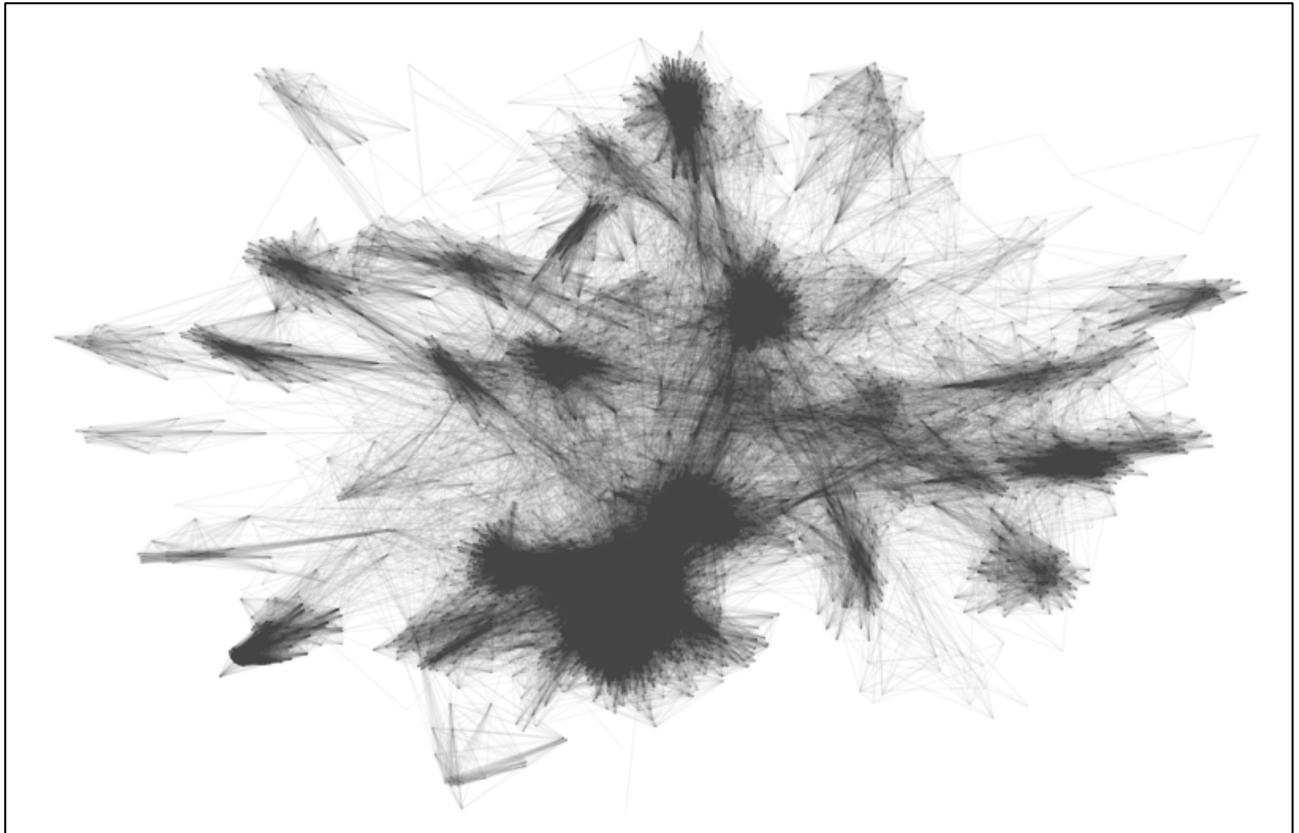


Figura 3.49: Visualizzazione rete

Questa visualizzazione della rete evidenzia i collegamenti esistenti tra i nodi. Essendo una rete abbastanza complessa, infatti, si è pensato di ridurre al minimo la dimensioni dei nodi in modo che a colpo d’occhio si potessero visualizzare in una maniera abbastanza comprensibile, i collegamenti esistenti. Si vede che ci sono zone a maggior densità dove appunto questi collegamenti si addensano, a monito di eventuali comunità.

3.4.2 Community Network Visualization

Essendo riusciti a visualizzare la rete, si è pensati di graficare all'interno della rete stessa, anche le comunità che sono state precedentemente individuate. L'idea è, oltre a visualizzare graficamente le comunità e comprendere quindi come sono distribuite tali comunità nella rete, voler dare anche un tocco artistico alla rete stessa. Per farlo sono stati definiti i seguenti script che permettono sostanzialmente di individuare e visualizzare le diverse comunità con dei colori all'interno del grafo. Nel dettaglio il primo script permette di impostare i nodi e gli archi relativi ad una certa comunità e vengono definiti i colori che verranno assegnati ai vertici del nostro grafo.

```
def set_node_community(G, communities):
    '''Add community to node attributes'''
    for c, v_c in enumerate(communities):
        for v in v_c:
            # Add 1 to save 0 for external edges
            G.nodes[v]['community'] = c + 1

def set_edge_community(G):
    '''Find internal edges and add their community to their attributes'''
    for v, w, in G.edges:
        if G.nodes[v]['community'] == G.nodes[w]['community']:
            # Internal edge, mark with community
            G.edges[v, w]['community'] = G.nodes[v]['community']
        else:
            # External edge, mark as 0
            G.edges[v, w]['community'] = 0

def get_color(i, r_off=1, g_off=1, b_off=1):
    '''Assign a color to a vertex.'''
    r0, g0, b0 = 0, 0, 0
    n = 16
    low, high = 0.1, 0.9
    span = high - low
    r = low + span * (((i + r_off) * 3) % n) / (n - 1)
    g = low + span * (((i + g_off) * 5) % n) / (n - 1)
    b = low + span * (((i + b_off) * 7) % n) / (n - 1)
    return (r, g, b)
```

Figura 3.50: Definizione di funzioni per rappresentazione comunità nella rete

Il secondo script invece, richiama le funzioni che sono state definite nello script precedente e vengono adattate alla nostra rete, considerato appunto il grafo “G_reduce_1” e le comunità che sono salvate all'interno della variabile “communities”. Si impostano quindi le comunità dei nodi e degli archi e si settano i colori per tali comunità.

```

plt.rcParams.update(plt.rcParamsDefault)
plt.rcParams.update({'figure.figsize': (15, 10)})
plt.style.use('dark_background')

# Set node and edge communities
set_node_community(G_reduce_1, communitys)
set_edge_community(G_reduce_1)

# Set community color for internal edges
external = [(v, w) for v, w in G_reduce_1.edges if G_reduce_1.edges[v, w]['community'] == 0]
internal = [(v, w) for v, w in G_reduce_1.edges if G_reduce_1.edges[v, w]['community'] > 0]
internal_color = ["black" for e in internal]
node_color = [get_color(G_reduce_1.nodes[v]['community']) for v in G_reduce_1.nodes]
# external edges
nx.draw_networkx(
    G_reduce_1,
    pos=pos,
    node_size=0,
    edgelist=external,
    edge_color="silver",
    node_color=node_color,
    alpha=0.2,
    with_labels=False)
# internal edges
nx.draw_networkx(
    G_reduce_1,
    pos=pos,
    edgelist=internal,
    edge_color=internal_color,
    node_color=node_color,
    alpha=0.05,
    with_labels=False)

```

Figura 3.51: Codice per visualizzazione delle comunità sulla rete

L’output che vogliamo sarà quindi il seguente, in cui è rappresentata la rete e le relative comunità individuate da un particolare colore:

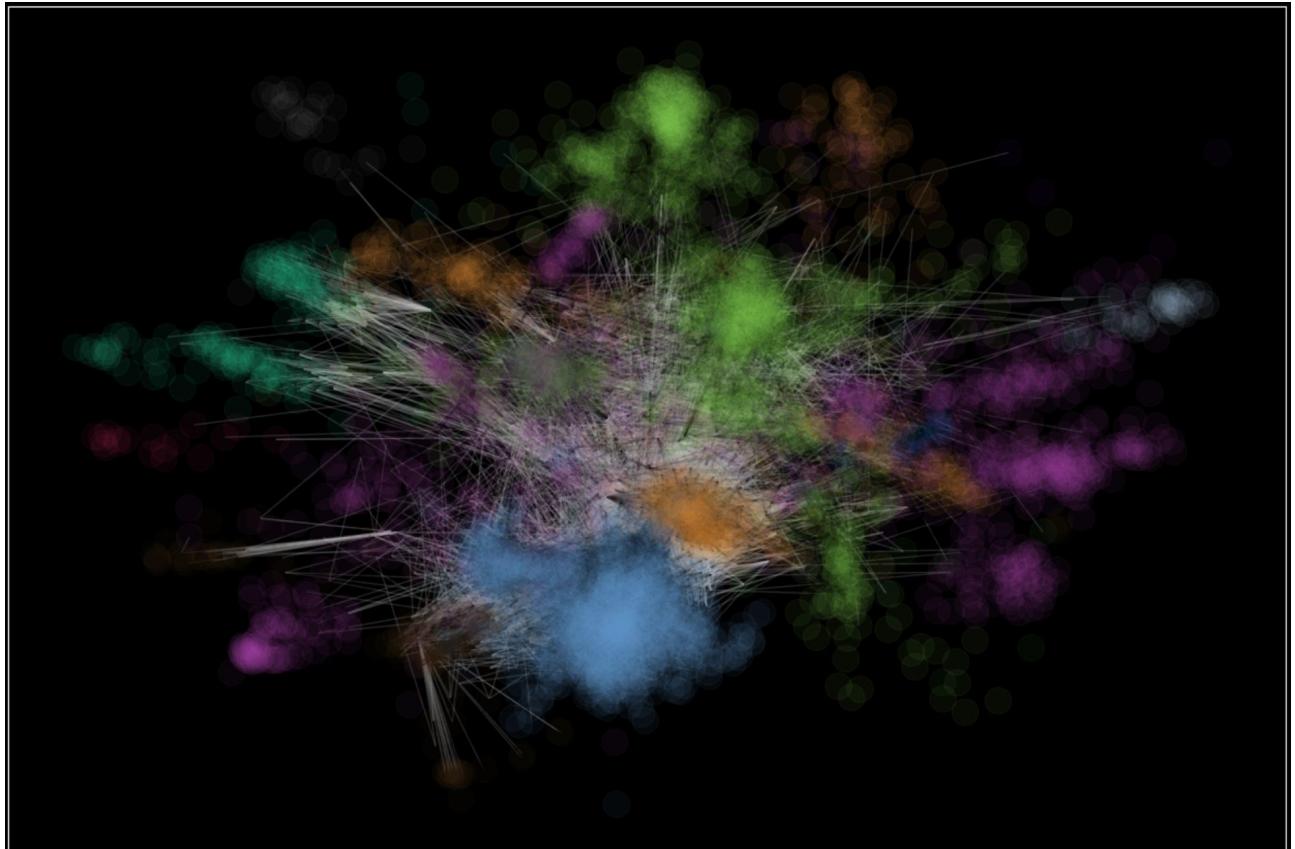


Figura 3.52: Visualizzazione grafica delle comunità

3.5 Communities report

Rappresentate quindi le comunità nella rete, bisogna poter anche individuare quali pagine facciano riferimento alle particolari comunità. Delle dodici comunità che sono state individuate se ne è fatta una selezione e sono state prese in esame le prime tre più grandi, ma anche la comunità più piccola e dunque la meno popolata. Per realizzare questo si è partiti dai frozenset individuati, e sono stati scelti casualmente alcuni dei nodi per ogni frozenset, in modo che siano rappresentativi di quella particolare comunità.

3.5.1 Comunità 1

Per quanto riguarda la comunità maggiormente popolata, si sono scelti casualmente cinque dei nodi della comunità.

```
print(G_reduce_1.nodes[4097])
print(G_reduce_1.nodes[10246])
print(G_reduce_1.nodes[6151])
print(G_reduce_1.nodes[14])
print(G_reduce_1.nodes[16398])
```



```
{'ID': 4097, 'facebookid': 114364722909, 'pagename': 'U.S. Army Space and Missile Defense Command (SMDC)', 'pagetype': 'government', 'community': 1}
{'ID': 10246, 'facebookid': 130918506984758, 'pagename': 'US National Weather Service Columbia South Carolina', 'pagetype': 'government', 'community': 1}
{'ID': 6151, 'facebookid': 19728909942, 'pagename': 'U.S. Army John F. Kennedy Special Warfare Center and School', 'pagetype': 'government', 'community': 1}
{'ID': 14, 'facebookid': 374623305761, 'pagename': "NASA's Marshall Space Flight Center", 'pagetype': 'government', 'community': 1}
{'ID': 16398, 'facebookid': 127036233989057, 'pagename': 'Coaching Into Care', 'pagetype': 'government', 'community': 1}
```

Figura 3.53: Cinque nodi della comunità 1

Notiamo come la comunità più grande è costituita da pagine di tipo governativo e nello specifico troviamo pagine come “U.S. Army Space and Missile Defense Command (SMDC)”, “US National Weather Service Columbia South Carolina”, “U.S. Army John F. Kennedy Special Warfare Center and School”, “NASA’s Marshall Space Flight Center” e “Coaching Into Care”. Se le comunità sono un insieme di pagine o persone che tendono a confrontarsi e a comunicare riguardo ad interessi comuni, i temi centrali saranno sicuramente temi politici, organizzativi e militari.

3.5.2 Comunità 2

```
print(G_reduce_1.nodes[10241])
print(G_reduce_1.nodes[2051])
print(G_reduce_1.nodes[8197])
print(G_reduce_1.nodes[4106])
print(G_reduce_1.nodes[2059])
```



```
{'ID': 10241, 'facebookid': 218434258171162, 'pagename': 'EU Law and Publications', 'pagetype': 'government', 'community': 2}
{'ID': 2051, 'facebookid': 598216760241817, 'pagename': 'Anton Hofreiter', 'pagetype': 'politician', 'community': 2}
{'ID': 8197, 'facebookid': 439052496231941, 'pagename': 'Ministero delle politiche agricole alimentari e forestali', 'pagetype': 'government', 'community': 2}
{'ID': 4106, 'facebookid': 125817437499137, 'pagename': 'Embassy of the Netherlands in Rwanda', 'pagetype': 'government', 'community': 2}
{'ID': 2059, 'facebookid': 408541229160361, 'pagename': 'Campus France', 'pagetype': 'government', 'community': 2}
```

Figura 3.54: Cinque nodi della comunità 2

La seconda comunità per popolosità è quella costituita anche in questo caso da pagine di tipo governativo. Notiamo però come alcune delle pagine siano “EU Law and Publications”, “Anton Hofreiter”, “Ministero delle politiche agricole alimentari e forestali”, “Embassy of the Netherlands in Rwanda”, “Campus France”. Stando

alle pagine individuate, le tematiche su cui si avvita tale comunità potrebbero essere ad esempio, oltre a temi organizzativi e politici, anche tematiche volte all’ambiente, al territorio, ai trasporti o all’edilizia.

3.5.3 Comunità 3

```
print(G_reduce_1.nodes[49])
print(G_reduce_1.nodes[101])
print(G_reduce_1.nodes[106])
print(G_reduce_1.nodes[113])
print(G_reduce_1.nodes[129])

{'ID': 49, 'facebookid': 130096562306, 'pagename': 'Digicel', 'pagetype': 'company', 'community': 3}
{'ID': 101, 'facebookid': 104513569620273, 'pagename': 'The X Factor (USA)', 'pagetype': 'tvshow', 'community': 3}
{'ID': 106, 'facebookid': 145794842231097, 'pagename': 'Chad Griffith - Lake Macquarie', 'pagetype': 'politician', 'community': 3}
{'ID': 113, 'facebookid': 404948826259784, 'pagename': 'Senator James McGrath', 'pagetype': 'politician', 'community': 3}
{'ID': 129, 'facebookid': 173347701125, 'pagename': 'Governor Jan Brewer', 'pagetype': 'politician', 'community': 3}
```

Figura 3.55: Cinque nodi della comunità 3

La terza comunità per popolosità è quella costituita in questo caso da pagine di diverso tipo. Notiamo che le pagine sono essenzialmente di tipo politico ma anche relative agli spettacoli televisivi. Tra queste troviamo “Digicel”, “The X Factor (USA)”, “Chad Griffith - Lake Macquarie”, “Senator James McGrath”, “Governor Jan Brewer”. Tale comunità è composta quindi sia da pagine di politici, senatori e governatori, sia da pagine che riguardano il mondo dello spettacolo. Questo è molto interessante in quanto è monito che il mondo politico comunica ed interagisce con il mondo televisivo e dello spettacolo. E forse questa è una delle potenzialità dei social network, e dunque sia poter dar vita a comunità in cui si discute riguardo a particolari tematiche di interesse, sia poter creare sinergia tra entità e pagine di natura del tutto differente.

3.5.4 Comunità 12

```
print(G_reduce_1.nodes[217])
print(G_reduce_1.nodes[9000])
print(G_reduce_1.nodes[10734])
print(G_reduce_1.nodes[13385])

{'ID': 217, 'facebookid': 330250343871, 'pagename': 'Jeremy Corbyn', 'pagetype': 'politician', 'community': 12}
{'ID': 9000, 'facebookid': 667427310036806, 'pagename': 'Kirsten Oswald - East Ren', 'pagetype': 'politician', 'community': 12}
{'ID': 10734, 'facebookid': 904819546225376, 'pagename': 'Drew Hendry MP', 'pagetype': 'politician', 'community': 12}
{'ID': 13385, 'facebookid': 1580633408837996, 'pagename': 'Neil Gray MP', 'pagetype': 'politician', 'community': 12}
```

Figura 3.56: Comunità 12

L’ultima comunità presa in esame è quella più scarsamente popolata e costituita solo da quattro nodi. Le relative pagine sono tutte di tipo politico e corrispondono a “Jeremy Corbyn”, “Kirsten Oswald - East Ren”, “Drew Hendry MP”, e “Neil Gray MP”. Notiamo come questa comunità sia a tutti gli effetti molto piccola e costituita dalle pagine ufficiali di alcuni politici. Potrebbe rappresentare ad esempio politici appartenenti ad uno stesso partito politico che sono interessati a tematiche comuni.

3.6 Cliques detection

Una “clique”, o in italiano, una cricca, è un sottoinsieme di vertici di un grafo non orientato in modo tale che ogni due vertici distinti siano adiacenti; cioè, il suo sotto grafo indotto è completo. Una cricca è quindi un insieme di nodi totalmente connessi tra di loro e in un certo senso rappresentano comunità strette in cui ogni nodo è connesso tra loro.

Per individuare quindi tali cliques opereremo con il seguente script, che permette anche di individuare la cricca più grande, quindi costituita dal maggior numero di nodi strettamente connessi tra di loro. Lo script permette inoltre anche la visualizzazione della rete, in maniera diversa dalle precedenti.

```
plt.rcParams.update(plt.rcParamsDefault)
plt.rcParams.update({'figure.figsize': (15, 10)})
cliques = list(nx.find_cliques(G_reduce_1))
max_clique = max(cliques, key=len)
node_color = [(0.5, 0.5, 0.5) for v in G_reduce_1.nodes()]
for i, v in enumerate(G_reduce_1.nodes()):
    if v in max_clique:
        node_color[i] = (0.5, 0.5, 0.9)
nx.draw_networkx(G_reduce_1, node_color=node_color)
```

Figura 3.57: Codice per individuazione delle cliques

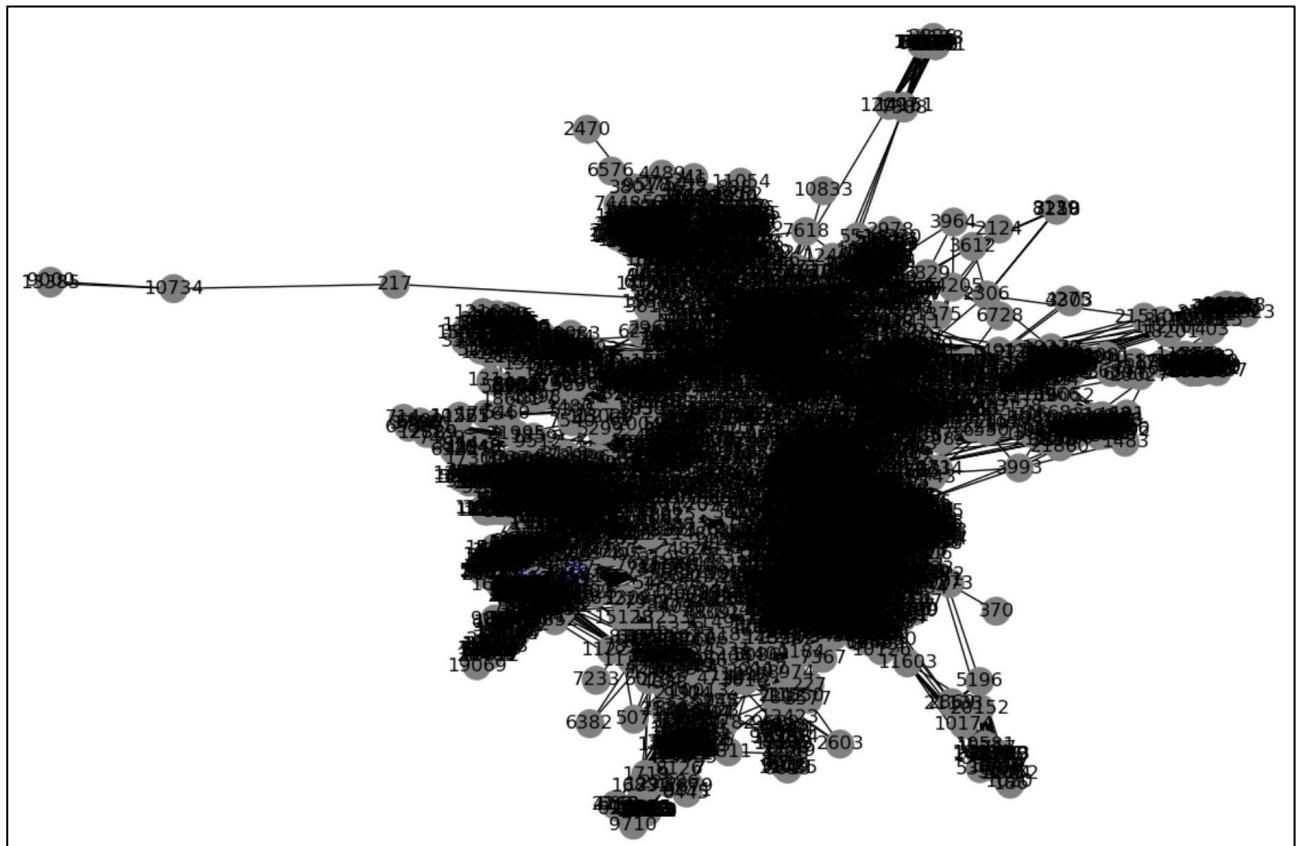


Figura 3.58: Visualizzazione dei Cliques sulla rete

La figura precedente mostra un'altra visualizzazione della rete per mostrare le differenze tra una semplice visualizzazione in questa maniera rispetto alla visualizzazione vista precedentemente. In questo caso, ogni nodo viene etichettato con il relativo id, e quindi le zone più scure nella rete sono zone a maggior densità dei nodi.

Tornando all’analisi delle cliques, siamo interessati quindi ad individuare quali pagine appartengano alla più grande cricca. Poiché ci siamo salvati all’interno della variabile “max_clique”, la cricca massima, con un semplice “print” possiamo individuare tutti i nodi che ci appartengono.

```
print(max_clique)  
[44, 3074, 9996, 3981, 4751, 5010, 13074, 5525, 20889, 21274, 3103, 20516, 19621, 7978, 4527, 4401, 21681, 6706, 7606, 17591, 5  
68, 18235, 20037, 6398, 5317, 839, 20168, 12745, 21324, 18638, 11473, 1618, 13140, 5589, 8795, 9567, 21087, 2532, 1126, 16105,  
18539, 4077, 18167, 1517, 14703, 7919, 11248, 12025, 19964, 8062, 18624, 11850, 7288, 2160, 19127]
```

Figura 3.59: Nodi della “cricca” più grande

La precedente è la lista dei nodi appartenenti alla cricca più grande; dunque, le informazioni associate a tali nodi sono estratte tramite il seguente codice:

```
nodi = []  
for i in range(len(max_clique)):  
    nodi.append(max_clique[i])  
  
for i in range(len(nodi)):  
    print(G_reduce_1.nodes[nodi[i]])
```

Figura 3.60: Individuazione nodi della cricca massima

E quindi avremo in output una lista delle relative pagine facebook, insieme alle altre informazioni associate al nodo, come il tipo di pagina e la comunità a cui appartiene il nodo:

```
{
  "ID": 44, "facebookid": 1507698529534072, "pagename": "APB FOX", "pagetype": "tvshow", "community": 3}
  {"ID": 3074, "facebookid": 14176232250, "pagename": "24: Legacy", "pagetype": "tvshow", "community": 3}
  {"ID": 9996, "facebookid": 262298737144442, "pagename": "Hotel Hell", "pagetype": "tvshow", "community": 3}
  {"ID": 3981, "facebookid": 212411822583153, "pagename": "Ghosted", "pagetype": "tvshow", "community": 3}
  {"ID": 4751, "facebookid": 126204090726016, "pagename": "MasterChef", "pagetype": "tvshow", "community": 3}
  {"ID": 5010, "facebookid": 1669295546644858, "pagename": "Shots Fired", "pagetype": "tvshow", "community": 3}
  {"ID": 13074, "facebookid": 1132457260127719, "pagename": "Kicking & Screaming", "pagetype": "tvshow", "community": 3}
  {"ID": 5525, "facebookid": 1859450307624772, "pagename": "Showtime At The Apollo", "pagetype": "tvshow", "community": 3}
  {"ID": 20889, "facebookid": 998610450211794, "pagename": "Making History", "pagetype": "tvshow", "community": 3}
  {"ID": 21274, "facebookid": 144734902571821, "pagename": "Party Over Here", "pagetype": "tvshow", "community": 3}
  {"ID": 3103, "facebookid": 658424124220288, "pagename": "Wayward Pines", "pagetype": "tvshow", "community": 3}
  {"ID": 20516, "facebookid": 55482772043, "pagename": "Glee", "pagetype": "tvshow", "community": 3}
  {"ID": 19621, "facebookid": 693029617393698, "pagename": "Gotham", "pagetype": "tvshow", "community": 3}
  {"ID": 7978, "facebookid": 805168456293232, "pagename": "You The Jury", "pagetype": "tvshow", "community": 3}
  {"ID": 4527, "facebookid": 274729849664498, "pagename": "LA to Vegas", "pagetype": "tvshow", "community": 3}
  {"ID": 4401, "facebookid": 913711325328002, "pagename": "Houdini & Doyle", "pagetype": "tvshow", "community": 3}
  {"ID": 21681, "facebookid": 1071460706219923, "pagename": "Son of Zorn", "pagetype": "tvshow", "community": 3}
  {"ID": 6706, "facebookid": 1697323807231225, "pagename": "The Resident", "pagetype": "tvshow", "community": 3}
  {"ID": 7606, "facebookid": 1679646295611249, "pagename": "My Kitchen Rules", "pagetype": "tvshow", "community": 3}
  {"ID": 17591, "facebookid": 13427984738, "pagename": "Hell's Kitchen", "pagetype": "tvshow", "community": 3}
  {"ID": 568, "facebookid": 127428534470290, "pagename": "The Orville", "pagetype": "tvshow", "community": 3}
  {"ID": 18235, "facebookid": 1513309918974102, "pagename": "The Mick", "pagetype": "tvshow", "community": 3}
  {"ID": 20037, "facebookid": 329991720461519, "pagename": "MasterChef Junior", "pagetype": "tvshow", "community": 3}
  {"ID": 6398, "facebookid": 932744616757576, "pagename": "Grandfathered", "pagetype": "tvshow", "community": 3}
  {"ID": 5317, "facebookid": 1486627808230653, "pagename": "The Last Man on Earth", "pagetype": "tvshow", "community": 3}
  {"ID": 839, "facebookid": 787059354700187, "pagename": "Scream Queens", "pagetype": "tvshow", "community": 3}
  {"ID": 20168, "facebookid": 877327628991411, "pagename": "Lucifer", "pagetype": "tvshow", "community": 3}
  {"ID": 12745, "facebookid": 1871217713123778, "pagename": "Love Connection FOX", "pagetype": "tvshow", "community": 3}
  {"ID": 21324, "facebookid": 992028154196876, "pagename": "The Exorcist FOX", "pagetype": "tvshow", "community": 3}
  {"ID": 18638, "facebookid": 387234808124690, "pagename": "Cooper Barrett's Guide to Surviving Life", "pagetype": "tvshow", "community": 3}
  {"ID": 11473, "facebookid": 1311830868853597, "pagename": "Superhuman", "pagetype": "tvshow", "community": 3}
  {"ID": 1618, "facebookid": 29534858696, "pagename": "The Simpsons", "pagetype": "tvshow", "community": 3}
  {"ID": 13140, "facebookid": 222039131156080, "pagename": "New Girl", "pagetype": "tvshow", "community": 3}
  {"ID": 5589, "facebookid": 1071010932932624, "pagename": "American Grit", "pagetype": "tvshow", "community": 3}
  {"ID": 8795, "facebookid": 102666876445950, "pagename": "Bob's Burgers", "pagetype": "tvshow", "community": 3}
  {"ID": 9567, "facebookid": 1432687680122316, "pagename": "The Gifted", "pagetype": "tvshow", "community": 3}
  {"ID": 21087, "facebookid": 1507259829580025, "pagename": "Coupled", "pagetype": "tvshow", "community": 3}
  {"ID": 2532, "facebookid": 822368007822596, "pagename": "The Grinder", "pagetype": "tvshow", "community": 3}
  {"ID": 1126, "facebookid": 109028969735, "pagename": "FOX Teen Choice Awards", "pagetype": "tvshow", "community": 3}
  {"ID": 16105, "facebookid": 611603115587740, "pagename": "Bordertown", "pagetype": "tvshow", "community": 3}
  {"ID": 18539, "facebookid": 122916548170738, "pagename": "The F Word", "pagetype": "tvshow", "community": 3}
  {"ID": 4077, "facebookid": 789058757865133, "pagename": "STAR", "pagetype": "tvshow", "community": 3}
  {"ID": 18167, "facebookid": 19097964496, "pagename": "Bones", "pagetype": "tvshow", "community": 3}
  {"ID": 1517, "facebookid": 310718132392309, "pagename": "Brooklyn Nine-Nine", "pagetype": "tvshow", "community": 3}
  {"ID": 14703, "facebookid": 52268280111, "pagename": "Prison Break", "pagetype": "tvshow", "community": 3}
  {"ID": 7919, "facebookid": 9748634303, "pagename": "So You Think You Can Dance", "pagetype": "tvshow", "community": 3}
  {"ID": 11248, "facebookid": 221127344750774, "pagename": "Empire", "pagetype": "tvshow", "community": 3}
  {"ID": 12025, "facebookid": 1455278684801466, "pagename": "Second Chance", "pagetype": "tvshow", "community": 3}
  {"ID": 19964, "facebookid": 1608458186056725, "pagename": "Rosewood", "pagetype": "tvshow", "community": 3}
  {"ID": 8062, "facebookid": 1131252606983272, "pagename": "Beat Shazam", "pagetype": "tvshow", "community": 3}
  {"ID": 18624, "facebookid": 244266119255283, "pagename": "Famous", "pagetype": "tvshow", "community": 3}
  {"ID": 11850, "facebookid": 467026763375961, "pagename": "Sleepy Hollow", "pagetype": "tvshow", "community": 3}
  {"ID": 7288, "facebookid": 325305484327071, "pagename": "New Year's Eve on FOX", "pagetype": "tvshow", "community": 3}
  {"ID": 2160, "facebookid": 1547869018790094, "pagename": "World's Funniest", "pagetype": "tvshow", "community": 3}
  {"ID": 19127, "facebookid": 24609282673, "pagename": "Family Guy", "pagetype": "tvshow", "community": 3}
}
```

Figura 3.61: Pagine totali della cricca più grande

Notiamo da questa lunga lista di pagine Facebook appartenenti alla cricca più grande come tutte queste pagine appartengano alla stessa comunità, il che risulta logicamente corretto in quanto far parte di una cricca in un certo senso rappresenti l'appartenenza ad una comunità, in cui ogni nodo è connesso con gli altri della comunità. In aggiunta, si osserva come la categoria di pagina dominante sia quella di tipo televisivo. Tra queste pagine Facebook, tra le più popolari troviamo ad esempio "The Simpson", "MasterChef", "MasterChef Junior" e tante altre. Possiamo concludere che quindi, la cricca più grande all'interno della nostra rete è rappresentata da pagine di tipo televisivo e quindi rappresenta un gruppo di pagine che interagiscono intensamente tra loro e condividono quindi gli stessi argomenti e gli stessi interessi, in questo caso relativi al mondo dello spettacolo e della televisione.

3.7 Ego Network

Una Ego Network è una rete incentrata su un nodo. Quindi c'è un nodo centrale chiamato "ego" ed i suoi vicini si chiamano "alters"; la rete sarà quindi costituita dagli archi che vanno dall'ego agli alters e gli archi che collegano gli alters tra di loro. In un contesto Facebook le Ego Network sono molto importanti, soprattutto quando si fa “profiling”. Quindi, ad esempio, se si vuole cercare di creare una rete incentrata su una persona o una pagina, il principio di omofilia è fondamentale se si vuole profilare quella persona. Quindi si avrà il nodo ego che è la persona o la pagina di interesse e tra i relativi alters ci saranno gli amici di quella persona.

Nel nostro caso abbiamo come nodi le pagine Facebook verificate, e centeremo la Ego Network sul nodo a maggior centralità, con massimo valore di “degree”. Per realizzare ciò, si procede in questa maniera:

```
nodes_degrees = G_reduce_1.degree()
top_deg_node = max(nodes_degrees, key= lambda item: item[1])
print( '\033[1m' + 'Page with the gratest number of connections:' + '\033[0m' + str(top_deg_node[0])+
      ' (' + str(top_deg_node[1])+ ')')
Page with the gratest number of connections:19743 (558)
```

Figura 3.62: Nodo con più connessioni

Dunque, il nodo con maggior degree risulta essere il nodo il cui id è 19743. Quindi individuiamo la pagina relativa al nodo attraverso il seguente comando:

```
print(G_reduce_1.nodes[19743])
{'facebookid': 1191441824276882, 'pagename': 'The White House', 'pagetype': 'government', 'modularity': 0}
```

Figura 3.63: Pagina con più connessioni

Dunque, il nodo a maggior centralità risulta essere la pagina “The White House”, ovviamente di tipo governativo. La pagina mostra ben 558 connessioni. Questo risultato è sottolineato dal fatto che, come visto in precedenza, ci sia una importante posizione assunta da tale pagina all'interno della rete presa in esame. Quindi, “The White House” ha contatti con la maggior parte delle altre pagine che appartengono a tale rete.

3.7.1 Ego Network Visualization

Con lo script seguente si andrà a graficare la Ego Network, dove appunto il nodo “Ego” è rappresentato dalla pagina “The White House”.

```
pos = nx.spring_layout(ego_network, k=0.1)
plt.rcParams.update({'figure.figsize': (15, 10)})
nx.draw_networkx(
    ego_network,
    pos=pos,
    node_size=0,
    edge_color="#444444",
    alpha=0.05,
    with_labels=False)
```

Figura 3.64: Codice per visualizzazione Ego network

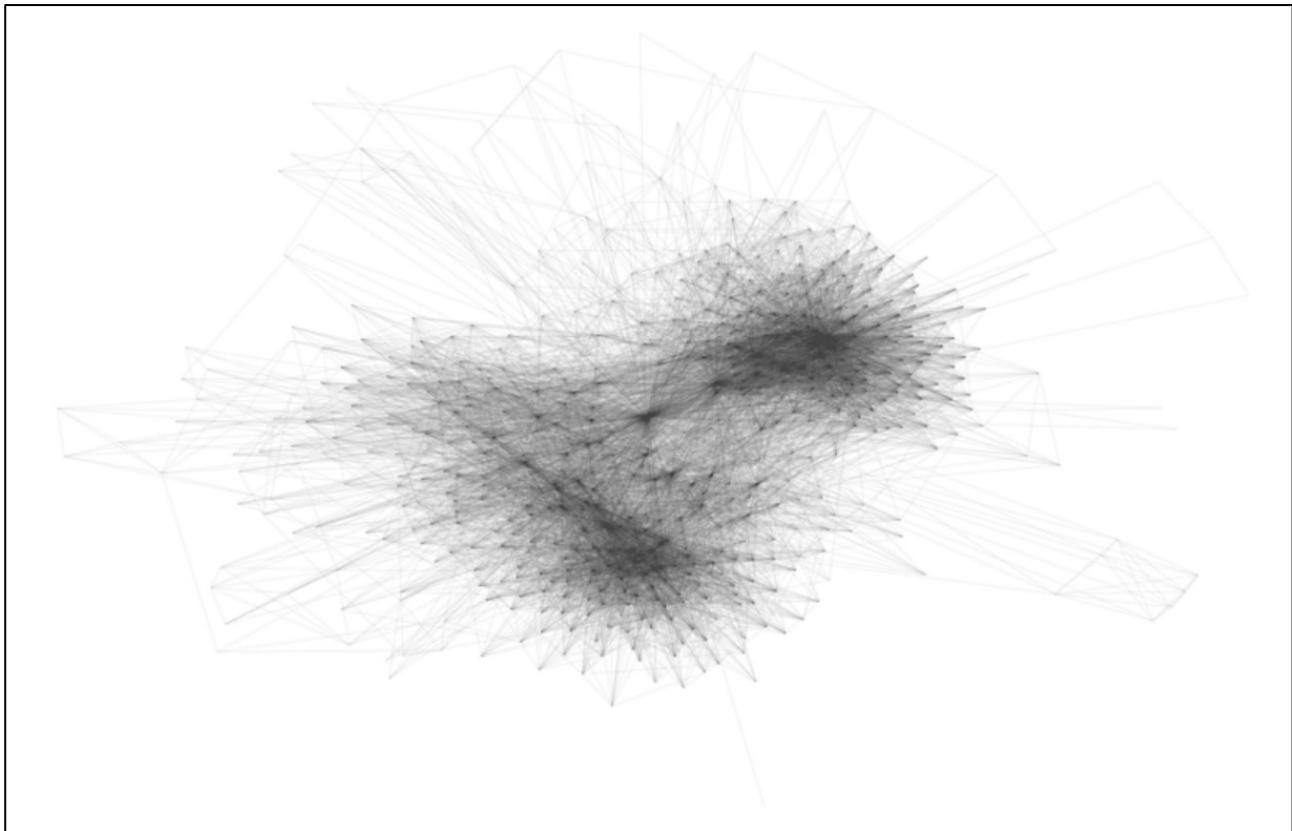


Figura 3.65: Visualizzazione grafo Ego network

Per questa visualizzazione si è scelto di annullare le dimensioni dei nodi per marcare meglio i collegamenti della Ego Network considerata. Vediamo come tutti questi collegamenti sono relativi alla pagina Facebook "The White House".

3.8 Group Centrality

Come altro tipo di analisi sono stati calcolati alcune delle centralità per i maggiori gruppi di pagine, dove con gruppo si intende l'insieme delle pagine appartenenti ad una stessa categoria. Quindi si hanno fondamentalmente quattro gruppi e relativi al gruppo dei politici, delle organizzazioni governative, degli spettacoli televisivi e delle aziende. Prima di procede all'analisi vera e propria, vengono create 4 liste, una per ogni gruppo ed inoltre questa analisi non è ristretta alla rete ridotta ma verranno calcolate le centralità dei gruppi considerando la rete originaria, più complessa costituita da un maggior numero di nodi e archi e quindi considerando il grafo G:

```
nome_nodo_politician = []
for i in range(len(G.nodes)):
    if (G.nodes[i]["pagetype"] == "politician"):
        nome_nodo_politician.append(i)

nome_nodo_tvshow = []
for i in range(len(G.nodes)):
    if (G.nodes[i]["pagetype"] == "tvshow"):
        nome_nodo_tvshow.append(i)

nome_nodo_government = []
for i in range(len(G.nodes)):
    if (G.nodes[i]["pagetype"] == "government"):
        nome_nodo_government.append(i)

nome_nodo_company = []
for i in range(len(G.nodes)):
    if (G.nodes[i]["pagetype"] == "company"):
        nome_nodo_company.append(i)
```

Figura 3.66: Creazione liste per categorie di pagina

Le metriche di centralità di gruppo assumono lo stesso significato di quelle già viste nelle sezioni precedenti, ma in questo caso gli algoritmi che le calcolano sono adattati da quelli originali per considerare un insieme di nodi anziché un singolo nodo e quindi avremo:

group_degree_centrality: calcolata come il rapporto tra la somma del numero dei nodi non appartenenti al gruppo ma connessi a uno dei membri del gruppo (grado del gruppo), e il numero totale di nodi che non appartengono al gruppo.

group_closeness_centrality: calcolata come il rapporto tra numero di nodi non appartenenti al gruppo e la somma delle distanze tra il gruppo e ogni nodo non appartenente al gruppo (si considera la distanza minima tra quelle misurate dal singolo nodo del gruppo al nodo non appartenente al gruppo).

3.8.1 Primo gruppo: Categoria TV Show

Alla categoria “Tv Show” appartengono tutte quelle pagine Facebook verificate relative al mondo dello spettacolo e televisivo. Esempi ne sono, pagine come “X-Factor”, “The Voice of Cina”, “The Simpson” e altre. Vengono calcolate la Degree Centrality e la Closeness Centrality per questo gruppo ed i relativi valori sono espressi in figura:

```
tvshow_degree = nx.group_degree_centrality(G, nome_nodo_tvshow)
print( '\033[1m' + 'Degree Centrality for TV show: ' + '\033[0m' + str(tvshow_degree))

Degree Centrality for TV show: 0.12720054327952776

tvshow_closeness = nx.group_closeness_centrality(G, nome_nodo_tvshow)
print( '\033[1m' + 'Closeness Centrality for TV show:' + '\033[0m' + str(tvshow_closeness))

Closeness Centrality for TV show: 0.4424081349664895
```

Figura 3.67: Calcolo centralità categoria TV show

3.8.2 Secondo gruppo: Categoria Company

Alla categoria “Company” appartengono tutte le pagine Facebook verificate relative al mondo delle aziende. Esempi, ne sono, pagine come “Facebook”, “Ford Company” e tante altre. Vengono calcolate la Degree Centrality e la Closeness Centrality per questo gruppo ed i relativi valori sono espressi in figura:

```
company_degree = nx.group_degree_centrality(G, nome_nodo_company)
print( '\033[1m' + 'Degree Centrality for company: ' + '\033[0m' + str(company_degree))

Degree Centrality for company: 0.21120500782472612

company_closeness = nx.group_closeness_centrality(G, nome_nodo_company)
print( '\033[1m' + 'Closeness Centrality for company:' + '\033[0m' + str(company_closeness))

Closeness Centrality for company: 0.48756294826796887
```

Figura 3.68: Calcolo centralità categoria TV show

3.8.3 Terzo gruppo: Categoria Government

Alla categoria “Government” appartengono tutte le pagine Facebook di tipo governativo. Tra le principali, come visto, si hanno ad esempio “The White House”, “U.S Army”, “EU Law and Publications” e altre simili. Vengono calcolate la Degree Centrality e la Closeness Centrality per questo gruppo ed i relativi valori sono espressi in figura:

```
government_degree = nx.group_degree_centrality(G, nome_nodo_government)
print( '\033[1m' + 'Degree Centrality for Government: ' + '\033[0m' + str(government_degree))

Degree Centrality for Government: 0.277421423989737

government_closeness = nx.group_closeness_centrality(G, nome_nodo_government)
print( '\033[1m' + 'Closeness Centrality for Government ' + '\033[0m' + str(government_closeness))

Closeness Centrality for Government 0.483560794044665
```

Figura 3.69: Calcolo centralità categoria TV show

3.8.4 Quarto gruppo: Categoria Politician

Alla categoria “Politician” appartengono invece tutte le pagine facebook relative ai politici. Ne sono state viste alcune come ad esempio “Barack Obama”, “Jeremy Corbyn” e altre relative a governatori, senatori e politici. Vengono calcolate la Degree Centrality e la Closeness Centrality per questo gruppo ed i relativi valori sono espressi in figura:

```
politician_degree = nx.group_degree_centrality(G, nome_nodo_politician)
print( '\033[1m' + 'Degree Centrality for Politician: ' + '\033[0m' + str(politician_degree))

Degree Centrality for Politician: 0.2005747814632978

politician_closeness = nx.group_closeness_centrality(G, nome_nodo_politician)
print( '\033[1m' + 'Closeness Centrality for Politician: ' + '\033[0m' + str(politician_closeness))

Closeness Centrality for Politician: 0.4374197941492287
```

Figura 3.70: Calcolo centralità categoria TV show

3.8.5 Conclusioni Group Centrality

Dunque, come si può interpretare a partire dai valori delle centralità individuate, i relativi valori per tutti i gruppi risultano abbastanza simili tra di loro. Comunque, nel dettaglio, tra tutte e quattro le categorie si può affermare che la categoria delle **pagine governative** spicca in termini di Degree Centrality e quindi tale gruppo ricopre all’interno della rete sicuramente una posizione strutturale rilevante. Non a caso, anche in tutte le analisi che sono state effettuate precedentemente, la maggior parte delle centralità erano cappeggiate da pagine la cui tipologia era di tipo governativo. Quindi tale risultato conferma anche ciò che è stato visto fin ora. Inoltre, sempre il gruppo di pagine governative ha il più alto valore di Closeness Centrality, alla pari con il gruppo delle **pagine delle aziende**. Sappiamo che in accordo alla metrica di Closeness Centrality, che più un gruppo è centrale secondo questa metrica, più è vicino a tutti gli altri nodi e dunque le pagine governative e le pagine delle aziende sono quelle più vicine a tutti gli altri nodi. In un certo senso si può affermare che la velocità del flusso di informazioni attraverso questi gruppi rispetto gli altri nodi risultano maggiori. Per quanto riguarda la categoria di **pagine** relativo al **mondo della televisione**, il valore della Degree Centrality è circa la metà rispetto a quello degli altri gruppi, quindi possiamo concludere che le pagine relative al mondo della TV e dello spettacolo, per questa rete, non ricoprono un ruolo particolarmente rilevante. La Closeness Centrality invece, si avvicina molto ai valori degli altri gruppi, superando di pochissimo il valore di Closeness Centrality relativo al **gruppo dei politici**. Quindi possiamo affermare che la velocità del flusso di informazioni per il gruppo degli spettacoli televisivi e per il gruppo dei politici risulta pressoché simile, diffondono quindi informazioni alla stessa velocità.

3.9 Classificazione multinodo

Obiettivo di questa ultima sezione è la classificazione multiclasse del nodo. Nel nostro caso ogni nodo ha diversi attributi, tra cui “facebookid,” il “nome della pagina” e la “categoria della pagina”. Noi siamo interessati alla tipologia di pagina che caratterizza il nodo. Quindi, in particolare si vuole predire, a partire dalle caratteristiche del nodo quale sarà la tipologia di pagina per quel nodo. Per farlo si usa un approccio mulitscala, tramite un algoritmo di rete embedding, o meglio una classe di algoritmi che acquisiscono informazioni su un nodo dalla distribuzione locale sugli attributi del nodo attorno ad esso.

Gli attributi di un nodo e quelli del suo dintorno possono contenere informazioni utili. Tali vicini possono essere considerati a diverse lunghezze di percorso, o scale in modo che, ad esempio, in una rete sociale, i vicini più vicini possono essere gli amici o le pagine con cui si interagisce di più, mentre i nodi separati da scale maggiori possono avere associazioni più deboli di amici. Gli attributi dei vicini su scale diverse possono essere considerati separatamente utilizzando un approccio multi-scala. Mentre, gli attributi dei nodi possono identificare diverse strutture della rete, ad esempio i nodi con attributi simili sono tali per cui hanno più probabilità di essere connessi (noto come principio di omofilia) così che i modelli di attributi simili dei nodi possono identificare una comunità.

Dal punto di vista implementativo, si procede ad importare le librerie necessarie:

```
from scipy import sparse
from sklearn.metrics import f1_score
from sklearn.decomposition import NMF, TruncatedSVD
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression, ElasticNet
```

Figura 3.71: Importazione librerie necessarie

Di seguito viene aggiunto anche il dataset delle features, che contiene due colonne: la prima colonna identifica il nodo, la seconda colonna è un identificatore delle caratteristiche. Essa descrive una matrice sparsa. In aggiunta vengono rinominati i dataset di partenza relativi ai nodi e agli archi.

```
features = pd.read_csv("C:/Users/Desktop/musae_facebook_features.csv")
df4 = target
df_archi = edges
```

Figura 3.72: Importazione dataset “musae_facebook_features” e rinomina dei dataset

La funzione seguente permette di trasformare i features associate ai nodi in forma di matrice sparsa:

```
def transform_features_to_sparse(table):
    table["weight"] = 1
    table = table.values.tolist()
    index_1 = [row[0] for row in table]
    index_2 = [row[1] for row in table]
    values = [row[2] for row in table]
    count_1, count_2 = max(index_1)+1, max(index_2)+1
    sp_m = sparse.csr_matrix(sparse.coo_matrix((values,(index_1,index_2)),shape=(count_1,count_2),dtype=np.float32))
    return sp_m
```

Figura 3.73: Definizione di funzione per matrice sparsa

La funzione seguente normalizza la matrice di adiacenza, che contiene i collegamenti tra i nodi di partenza e i nodi di arrivo:

```
def normalize_adjacency(raw_edges):
    raw_edges_t = pd.DataFrame()
    raw_edges_t["id_1"] = raw_edges["id_2"]
    raw_edges_t["id_2"] = raw_edges["id_1"]
    raw_edges = pd.concat([raw_edges, raw_edges_t])
    edges = raw_edges.values.tolist()
    graph = nx.from_edgelist(edges)
    ind = range(len(graph.nodes()))
    degs = [1.0/graph.degree(node) for node in graph.nodes()]
    A = transform_features_to_sparse(raw_edges)
    degs = sparse.csr_matrix(sparse.coo_matrix((degs, (ind, ind)), shape=A.shape, dtype=np.float32))
    A = A.dot(degs)
    return A
```

Figura 3.74: Definizione di funzione per matrice di adiacenza

Un ulteriore funzione ausiliaria per associare un valore al nodo in base alla categoria di appartenenza:

```
def mapper(x):
    if x == "politician":
        y = 0
    elif x == "company":
        y = 1
    elif x == "government":
        y = 2
    else:
        y = 3
    return y
```

Figura 3.75: Definizione di funzione mapper

Si definiscono quali sono le variabili X su cui si addestra il modello e la variabile Y di interesse da predire, ossia la categoria della pagina e quindi il valore di “pagetype”:

```
target = target["page_type"].values.tolist()
y = np.array([mapper(t) for t in target])
A = normalize_adjacency(edges)
X = transform_features_to_sparse(features)
X_tilde = A.dot(X)
```

Figura 3.76: X e Y del modello

Si definisce una funzione “eval_factorization” che prende in input due variabili (W e y) su cui poi viene addestrato un modello di regressione logistica. Le variabili di input vengono spartite secondo il Train/Test split considerando test_size= 0.3. In output viene restituito il valore dell'accuratezza del modello valutato in termini di F-1 score:

```
def eval_factorization(W,y):
    scores = []
    for i in range(10):
        X_train, X_test, y_train, y_test = train_test_split(W, y, test_size=0.3, random_state = i)
        model = LogisticRegression(C=0.01, solver = "saga",multi_class = "auto")
        model.fit(X_train, y_train)
        y_pred = model.predict(X_test)
        score = f1_score(y_test, y_pred, average = "weighted")
        scores.append(score)
    print(np.mean(scores))

    model = TruncatedSVD(n_components=16, random_state=0)
    W = model.fit_transform(X)
    model = TruncatedSVD(n_components=16, random_state=0)
    W_tilde = model.fit_transform(A)

    eval_factorization(W, y)
    eval_factorization(np.concatenate([W,W_tilde],axis=1), y)
```

0.6606980488328041
0.684474532422351

Figura 3.77: Creazione e valutazione del modello

L'accuratezza del modello realizzato è valutata tramite l'F-1 score ed è pari a 0.66 e 0.68 per i due modelli.

Gli algoritmi di embedding si è visto quindi che prendono come input un grafico di rete e gli attributi dei nodi (di addestramento). La performance di classificazione è valutata addestrando la regressione logistica per prevedere un attributo di test dato un nodo di incorporazione, che nel nostro caso corrisponde al tipo di pagina che caratterizza il nodo.

4. Conclusioni

La libreria NetworkX analizzata presenta quindi grandi potenzialità che permettono di lavorare con grafi eterogenei sia nella loro grandezza, sia nelle loro caratteristiche. Il fatto che risulti scritta in Python è decisamente un punto a favore data la possibilità di integrare la molteplicità dei moduli già pronti all'uso offerti da questo linguaggio in continua evoluzione, un vero e proprio ecosistema. In conclusione, lo strumento studiato offre tutte quelle funzionalità necessarie per affrontare la Network Analysis nel migliore dei modi riuscendo molto bene a svolgere le analisi per cui è stato progettato.

5. Riferimenti

<https://snap.stanford.edu/data/facebook-large-page-page-network.html>

https://en.wikipedia.org/wiki/The_Voice_of_China

https://it.wikipedia.org/wiki/Barack_Obama

<https://www.army.mil/>

https://en.wikipedia.org/wiki/White_House

<https://arxiv.org/abs/1909.13021>