

UNIVERSITÀ POLITECNICA DELLE MARCHE

FACOLTÀ DI INGEGNERIA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE



**Social Network Analysis applicata ad un servizio di  
bike-sharing nella città di Helsinki mediante l'uso  
della libreria NetworkX**

Studenti:

Bedetta Alessandro  
Ascani Christian

Docenti:

Ursino Domenico  
Bonifazi Gianluca

A.A. 2022-2023

# Sommario

1. Introduzione.....	2
1.1 Social Network Analysis.....	2
1.2 Helsinki City Bikes Dataset .....	2
2. Analisi della rete .....	4
2.1 Pre-processing.....	4
2.2 Descrizione del grafo e metriche .....	4
3. Data visualization.....	6
3.1 Visualizzazione standard .....	6
3.2 Visualizzazione geografica e mappa .....	7
4. Analisi delle centralità.....	9
4.1 Degree Centrality .....	9
4.2 Betweenness Centrality .....	10
4.3 Eigenvector Centrality.....	12
4.4 Closeness Centrality.....	13
5. Analisi delle strutture.....	15
5.1 Community detection .....	15
5.2 Triadi .....	18
6. Conclusioni.....	21

# 1. Introduzione

Lo scopo di questa relazione è quello di illustrare i risultati da noi ottenuti nell'ambito del progetto riguardante la social network analysis. A tal fine, come ambienti di sviluppo, sono stati utilizzati Google Colab e Visual Studio Code; l'intero codice è stato sviluppato secondo il linguaggio Python e di fondamentale importanza è stata la libreria NetworkX, la quale ci ha permesso di creare, manipolare ed analizzare reti complesse tramite l'uso di metodi predefiniti.

## 1.1 Social Network Analysis

L'analisi delle reti sociali è il processo di indagine delle strutture sociali tramite l'uso della teoria dei grafi e rappresenta una delle branche più importanti della Data Science. Secondo la teoria delle reti sociali, la società è vista come una rete di relazioni e la Social Network Analysis studia sia gli individui sia la struttura delle relazioni tra questi. La SNA affonda le sue radici diversi anni prima dell'avvento dei social network (come Facebook o Twitter) ed in particolare nel campo della sociologia, con l'intento di studiare i rapporti sociali che si creano tra le persone. Poiché si basa sulla teoria dei grafi, la SNA deve considerare alcuni elementi: i nodi (persone, devices, oggetti) e gli archi (le relazioni tra i nodi, come amicizia, odio). Mappando una rete su un grafo, si possono analizzare importanti proprietà, come l'influenza delle componenti, la coesione tra i nodi e la presenza di pattern interessanti. Si utilizza la Social Network Analysis anche nell'ambito dei trasporti, in quanto l'astrazione permette di porre l'attenzione sulla struttura del sistema stesso. Perciò, i singoli punti geografici vengono identificati come nodi, mentre i percorsi come archi. Solitamente, questo tipo di reti presenta archi fisicamente vincolati (come ferrovie o strade), mentre altri tipi di trasporti (come il bike-sharing) sono liberi da tali restrizioni.

## 1.2 Helsinki City Bikes Dataset

Il dataset su cui abbiamo scelto di basare l'intero progetto è stato l'Helsinki City Bike Dataset (<https://www.kaggle.com/datasets/geometrein/helsinki-city-bikes>); quest'ultimo contiene una raccolta di tutti i viaggi fatti dai cittadini della capitale finlandese usando il servizio di biciclette pubblico.

Ogni viaggio è caratterizzato da una serie di informazioni generali quali:

- **Departure:** giorno (dd-mm-yyyy) e orario (hh:mm:ss) di partenza
- **Return:** giorno (dd-mm-yyyy) e orario (hh:mm:ss) di arrivo
- **Departure\_id:** identificatore numerico della stazione di partenza
- **Departure\_name:** nome della stazione di partenza
- **Return\_id:** identificatore numerico della stazione di arrivo

- **Return\_name**: nome della stazione di arrivo
- **Distance (m)**: distanza in metri percorsa durante il viaggio
- **Duration (sec.)**: durata in secondi del viaggio
- **Avg\_speed (km/h)**: velocità media sostenuta
- **Departure\_latitude**: latitudine del punto di partenza
- **Departure\_longitude**: longitudine del punto di partenza
- **Return\_latitude**: latitudine del punto di arrivo
- **Return\_longitude**: longitudine del punto di arrivo
- **Air temperature (degC)**: temperatura dell'aria misurata durante in viaggio in gradi Celsius.

Il bike-sharing, a differenza di altri servizi di trasporto, è di sua natura più flessibile, anche perché dipende dalle esigenze di chi usufruisce del servizio. Perciò, gli archi nella rete non sono predeterminati, ma vengono generati dagli utenti. La struttura della rete prende forma dal flusso stesso di chi utilizza il servizio e, di conseguenza, si possono trovare particolari pattern a partire da viaggi ripetuti nel tempo. In aggiunta, l'aumento o la diminuzione dell'uso del servizio possono essere elementi chiave per avere maggiori informazioni su alcune aree, sia dal punto di vista geografico che sociale.

## 2. Analisi della rete

### 2.1 Pre-processing

Dopo aver scaricato e importato il dataset nel nostro ambiente di sviluppo, è stato subito chiaro che operare su di esso sarebbe stato complicato computazionalmente (una grandezza di circa 2 GB). Per cui, come prima operazione, si è proceduto con la riduzione della dimensione del dataset, selezionando solamente i dati riguardanti l'anno 2020 ed escludendo quelli precedenti; questa semplificazione ha permesso di avere un numero minore di dati, passando da più di 12 milioni di record ad appena 3 milioni.

In seguito, sono stati convertiti i campi **departure** e **return** da formato String a formato Datetime, più comodo da utilizzare. Fatto ciò, abbiamo proceduto alle operazioni di data cleaning, considerando solo i record contenenti valori non troppo distanti dalle medie. Dopo aver eliminato i dati contenenti valori nulli, sono stati applicati alcuni filtri agli attributi più rilevanti ([Figura 1](#)).

```
df = df[
    (50 < df['distance (m)']) & (df['distance (m)'] < 30000) &
    (180 < df['duration (sec.)']) & (df['duration (sec.)'] < 43200) &
    (-20 < df['Air temperature (degC)']) & (df['Air temperature (degC)'] < 40)
]
```

*Figura 1: Filtraggio dei dati*

Abbiamo scelto di considerare i viaggi caratterizzati da una lunghezza compresa fra 50 metri (togliendo i viaggi meno significativi) e 30 chilometri (togliendo i viaggi troppo lunghi). Successivamente, abbiamo anche scartato i record caratterizzati da una durata al di fuori del range che va dai 3 minuti (viaggi molto brevi) alle 12 ore (viaggi troppo lunghi) e quelli in cui apparivano temperature fuori scala sopra i 40 gradi o sotto i - 20.

### 2.2 Descrizione del grafo e metriche

Terminate le operazioni di trasformazione, usando il metodo `from_pandas_edgelist` della libreria NetworkX, è stata creata una struttura a grafo indiretto a partire dal dataframe prodotto durante gli step precedenti.

Esplorando le informazioni del grafo appena generato, si è visto che quest'ultimo era composto da **347 nodi** e **31.693 archi** ([Figura 2](#)).

```
graph = nx.Graph()
G = nx.from_pandas_edgelist(df, source="departure_id",
                           target = "return_id",
                           edge_attr="distance (m)",
                           create_using=graph)
print(G)
```

Figura 2: Creazione del grafo indiretto

A tal punto, si sono considerate le caratteristiche strutturali della rete, così da capire quali fossero le proprietà specifiche; in particolare:

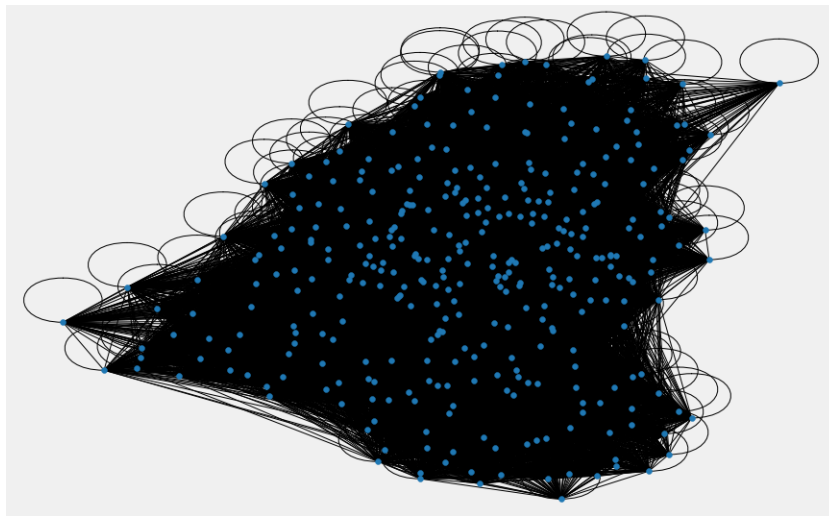
- **Densità:** calcolata come la probabilità che una qualsiasi coppia di nodi sia adiacente, cioè misura quanto i nodi sono collegati.
- **Eccentricità:** per un nodo, è il maggiore shortest path possibile tra quel nodo e tutti gli altri.
- **Diametro:** indica il maggiore shortest path tra una coppia di nodi del grafo; quindi, è il valore massimo di eccentricità possibile di un nodo.
- **Raggio:** è il minimo valore di eccentricità di un nodo.
- **Periferia:** è l'insieme dei nodi che hanno la loro eccentricità pari al diametro del grafo.
- **Centro:** è l'insieme dei nodi che hanno la loro eccentricità uguale al raggio del grafo.
- **Average Clustering Coefficient:** è la media del coefficiente di clustering locale di ogni nodo del grafo, che misura quanto i suoi vicini tendono a formare una *clique*; perciò, indica quanto i nodi tendono ad essere connessi fra loro.

La densità si assesta attorno a 0.5, per cui il grafo risulta essere molto denso. Riguardo alle eccentricità, queste assumono due soli valori possibili: 2 o 3; di conseguenza il diametro corrisponde proprio a 3, mentre il raggio a 2. Ciò testimonia che molti nodi sono direttamente connessi e, infatti, ci sono 34 stazioni che rappresentano la periferia del grafo, mentre i restanti nodi corrispondono al centro. Per quanto riguarda il coefficiente medio di clustering, il valore è circa 0.75, perciò 3/4 delle stazioni, che sono collegate con una stazione, sono anche collegate fra loro.

## 3. Data visualization

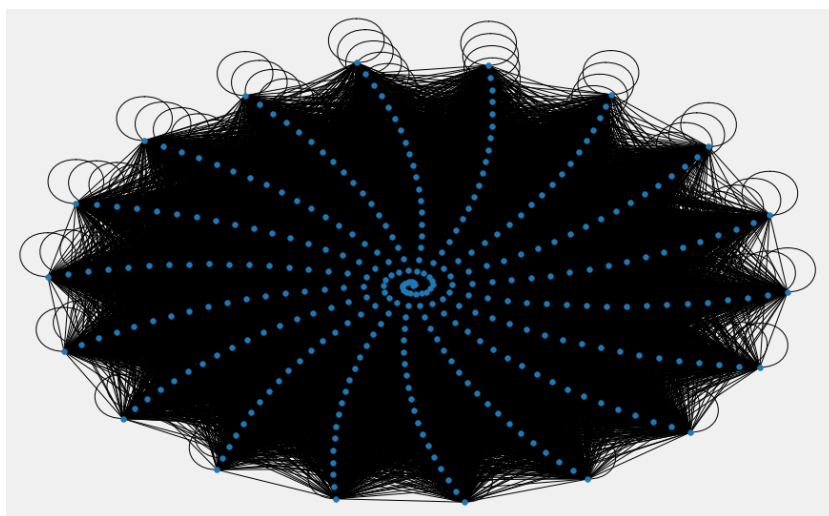
### 3.1 Visualizzazione standard

Per avere una visualizzazione grafica della rete abbiamo sfruttato alcuni layout messi a disposizione dalla libreria NetworkX. Perciò, per primo è stato considerato lo Spring Layout, per capire come fosse organizzata la rete; come si evince in [Figura 3](#), c'è un'area centrale densamente popolata, mentre nella parte più esterna i nodi si dispongono in maniera caotica. Dunque, da ciò non si rilevano peculiarità o eventuali gruppi significativi.



*Figura 3: Spring layout della rete*

Un'altra visualizzazione è data dal Spiral Layout: in [Figura 4](#), si nota che al centro ci sono nodi molto vicini tra loro, mentre sui rami della spirale i nodi sono più distanti; tuttavia, nonostante l'allontanamento dal centro, gli archi rimangono fitti. Questo conferma la densità della rete, però, non si riescono ad estrarre ulteriori informazioni.

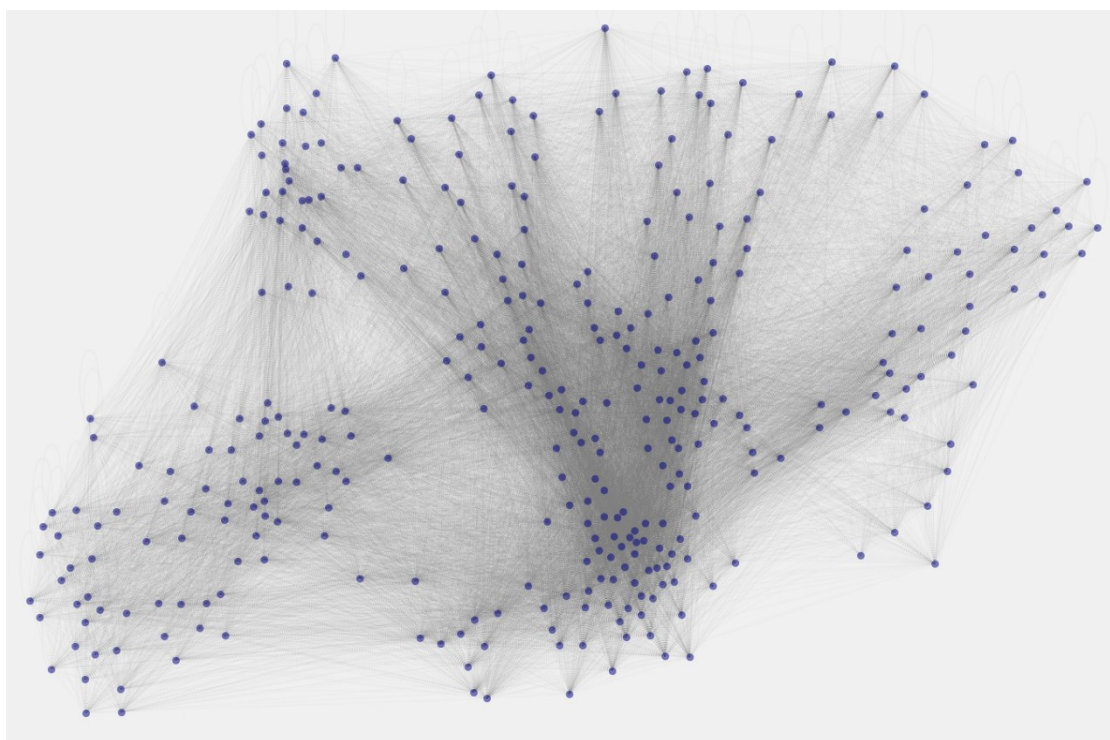


*Figura 4: Spiral layout della rete*

### 3.2 Visualizzazione geografica e mappa

Al contrario di quanto accade con molti tipi di rete, i layout forniti dalla libreria NetworkX spesso non sono utili nel contesto geografico.

Dato che le stazioni sono fisicamente legate ad una certa posizione nel territorio di Helsinki, si possono visualizzare i nodi secondo la distribuzione geografica, mappandoli con le corrispondenti longitudini/latitudini. Infatti, tramite un dizionario (nome: posizione), viene associato ogni singolo nodo del grafo con la relativa coordinata, secondo una certa scala al fine di non rendere i nodi eccessivamente sparsi. In questo modo, è possibile avere una panoramica più precisa delle stazioni di bike-sharing e – conseguentemente – estrapolare informazioni aggiuntive. In [Figura 5](#), si può notare che ci sono aree con densità di archi differenti, le quali testimoniano la presenza di una zona centrale e di zone periferiche.

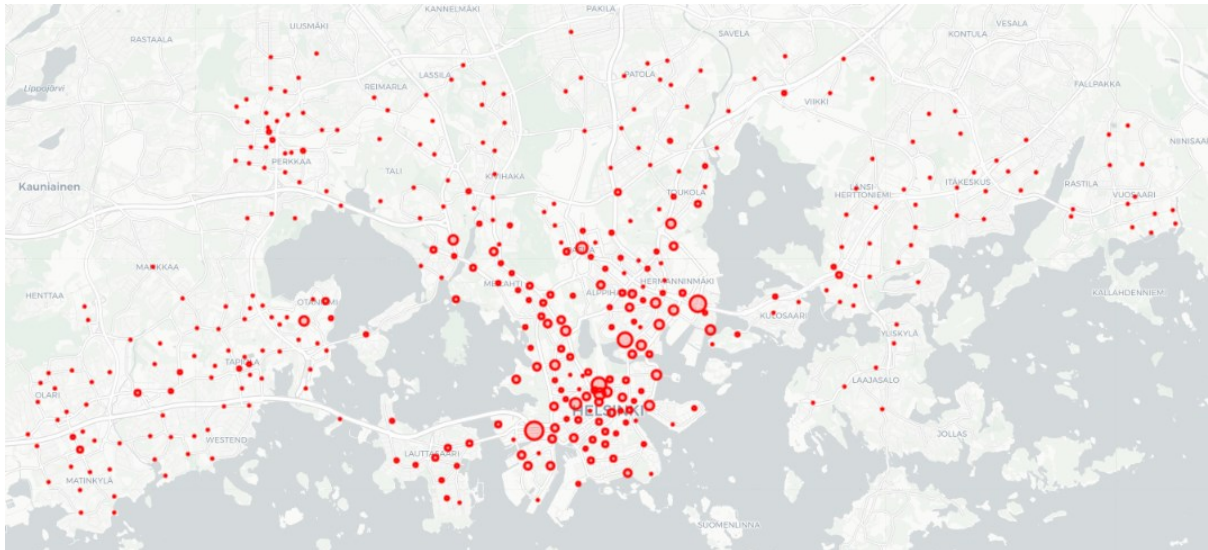


*Figura 5: Visualizzazione geografica della rete*

Per una visualizzazione ancora più precisa, si può rappresentare la rete su una mappa in modo da rilevare altre caratteristiche peculiari. Nella pianta di Helsinki ([Figura 6](#)), una stazione viene raffigurata tramite un cerchio, la cui dimensione dipende dalla sua presenza nel dataset. Come in precedenza, si individuano aree con cerchi più grandi, sinonimo di un transito maggiore in corrispondenza di quelle stazioni, e altre zone con cerchi più piccoli.

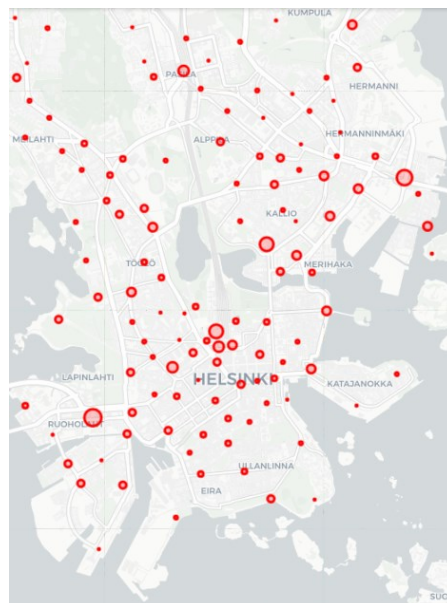
Infatti, la componente centrale, in cui ci sono nodi fortemente connessi tra loro, corrisponde al centro cittadino di Helsinki, mentre le altre aree rappresentano la periferia della città.





*Figura 6: Stazioni sulla mappa*

Come ci si potrebbe aspettare, nel centro di Helsinki (Figura 7) c'è una grande attività dal punto di vista della mobilità (in questo caso bike-sharing), poiché qui si concentrano le attività economiche, sociali e culturali.



*Figura 7: Centro di Helsinki*

Tuttavia, il centro di Helsinki non corrisponde obbligatoriamente al centro della rete, per cui è necessario analizzarla in maniera più approfondita. Infatti, le reti più complesse sono altamente eterogenee, per cui spesso alcune parti forniscono informazioni maggiori.

## 4. Analisi delle centralità

Le metriche di centralità nell'ambito della social network analysis servono a fornirci informazioni riguardo l'importanza di ogni nodo, data la sua posizione all'interno della rete. Possiamo distinguere svariate tipologie di centralità, ed ognuna di esse si riferisce a proprietà differenti. Nel contesto dei trasporti, una regione urbana in cui si concentrano i flussi maggiori di utenti potrebbe essere considerata centrale, però spesso il concetto di centralità cambia da contesto a contesto e/o con il mutare delle condizioni.

Nella nostra analisi sono state esaminate le seguenti: *Degree centrality*, *Betweenness centrality*, *Eigenvector centrality* e *Closeness centrality*.

### 4.1 Degree Centrality

La degree centrality rappresenta la misura di centralità più semplice, in quanto il suo valore indica il numero di nodi adiacenti al nodo che stiamo considerando. Nonostante sia una misura molto semplice da calcolare non ci dà informazioni riguardanti l'importanza del nodo legata al peso delle sue connessioni. Nell'ambito del bike-sharing, questa misura si riferisce al numero di stazioni verso cui gli utenti hanno viaggiato, partendo da una particolare stazione.

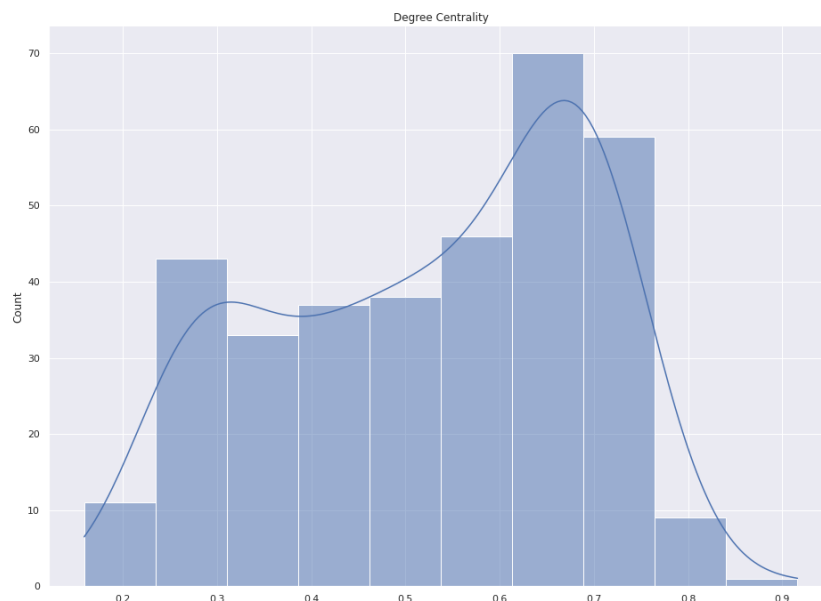
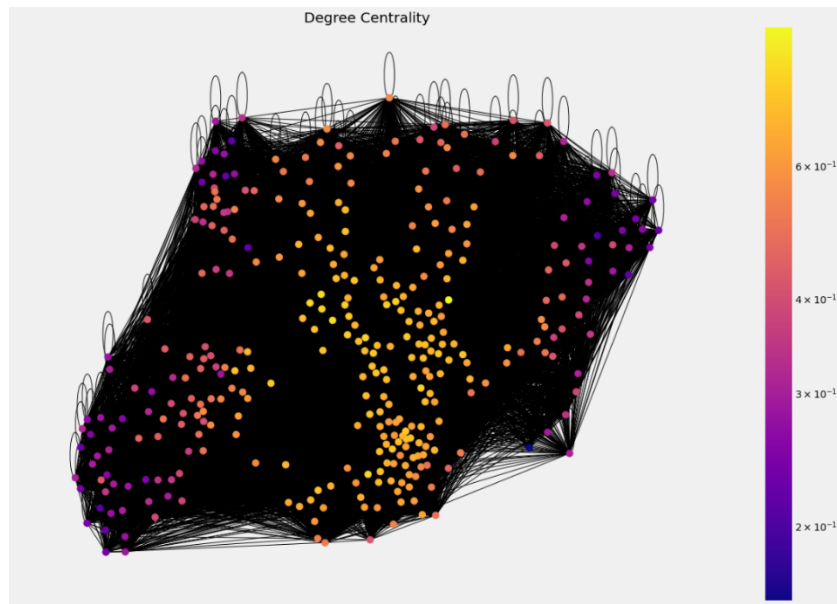


Figura 8: Distribuzione della degree centrality

Come è possibile evincere dalla [Figura 8](#), la distribuzione segue un andamento quasi gaussiano; questo è dovuto al fatto che le stazioni nel centro città hanno un numero alto di connessioni con stazioni vicine, mentre man mano che ci si sposta in periferia, le stazioni tendono ad avere un numero sempre minore di adiacenze.



*Figura 9: Degree centrality della rete*

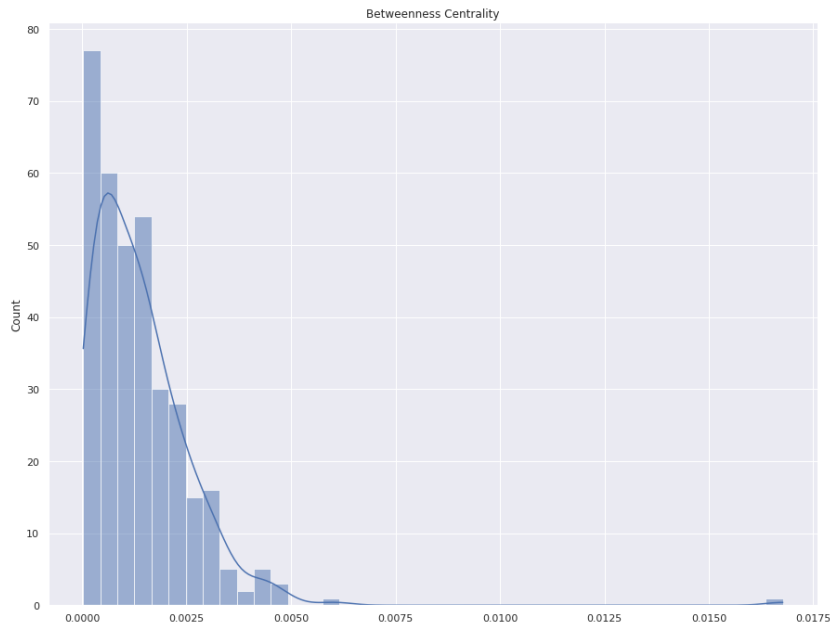
Dopo aver localizzato in Figura 9 la zona con degree centrality più alta, si possono analizzare in dettaglio i nodi del grafo. Infatti, si può notare che le dieci stazioni con degree centrality più alta sono:

- Haukilahdenkatu: 0.9161849710982658
- Paciuksenkaari: 0.815028901734104
- Itämerentori: 0.8092485549132947
- Laajalahden aukio: 0.8092485549132947
- Huopalahdentie: 0.8092485549132947
- Munkkiniemen aukio: 0.7890173410404624
- Ympyrätalo: 0.7803468208092486
- Töölöntulli: 0.7803468208092486
- Tilkanvierto: 0.7745664739884393
- Paciuksenkatu: 0.7687861271676301

Poiché la stazione di Haukilahdenkatu si trova in prima posizione, si può asserire che un gran numero di cittadini, provenienti anche da diversi distretti, eseguono percorsi regolari da e per questa stazione. Ciò potrebbe indicare la presenza di un punto nevralgico o di particolare interesse.

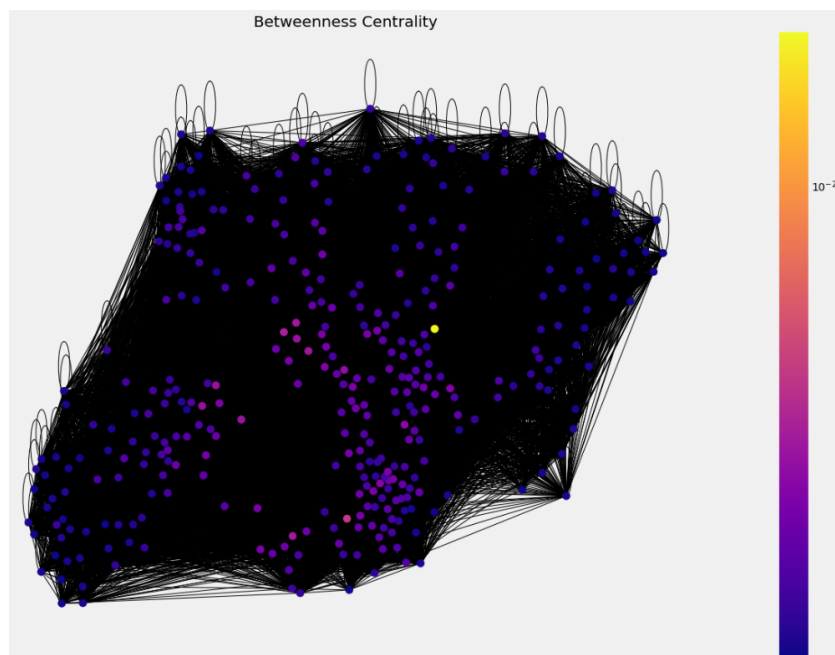
## 4.2 Betweenness Centrality

Questa seconda tipologia di centralità ha lo scopo di misurare quanto un nodo funge da punto di passaggio obbligato fra due porzioni della rete. Dato un nodo  $x$ , viene calcolato lo shortest path per ogni coppia di nodi del grafo; fatto ciò, si conta il numero di volte che  $x$  appare in questi cammini: più sarà alto questo valore e più il nodo  $x$  rappresenterà un “ponte” fra due parti della rete.



*Figura 10: Distribuzione della betweenness centrality*

Osservando la Figura 10 riportante tale misura, si nota una concentrazione dei nodi sui valori più bassi. Questo fenomeno è riconducibile alla natura stessa della rete, la quale, avendo 31.693 archi e soli 347 nodi, risulta estremamente interconnessa e - di conseguenza - è difficile trovare dei nodi con valori di betweenness alti.

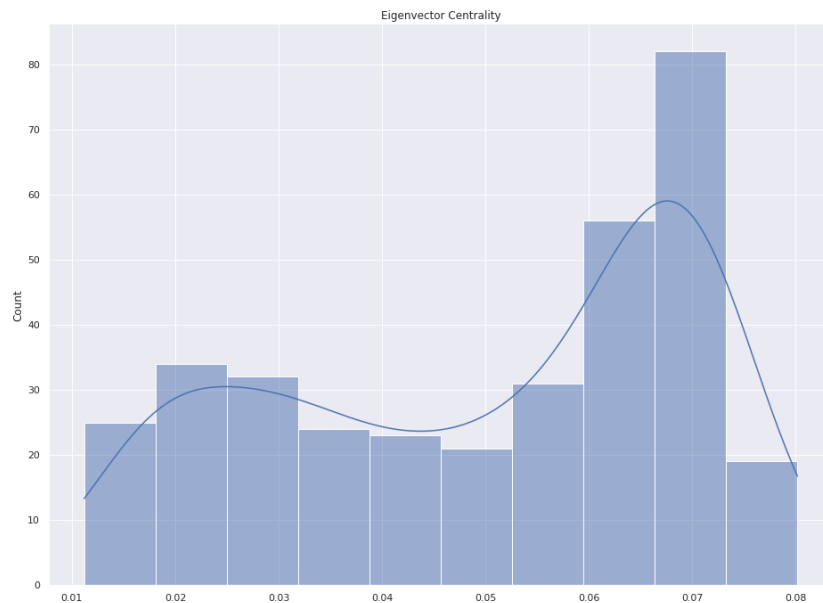


*Figura 11: Betweenness centrality della rete*

Dalla Figura 11 si può notare un solo nodo con una betweenness centrality di una certa entità, corrispondente alla stazione di Haukilahdenkatu (0.01677); ciò indica che, anche se venisse a mancare questa stazione, comunque si troverebbero percorsi alternativi senza causare grandi disagi.

### 4.3 Eigenvector Centrality

Questa misura di centralità ci fornisce un valore di influenza del nodo all'interno della rete, basato sull'importanza di quest'ultimo rispetto ai nodi che lo circondano. La misura è calcolata seguendo un processo iterativo durante il quale connessioni con nodi aventi un alto coefficiente di eigenvector centrality alzano il coefficiente stesso del nodo di riferimento; al contrario, adiacenze con nodi a basso coefficiente forniscono dei contributi minori.



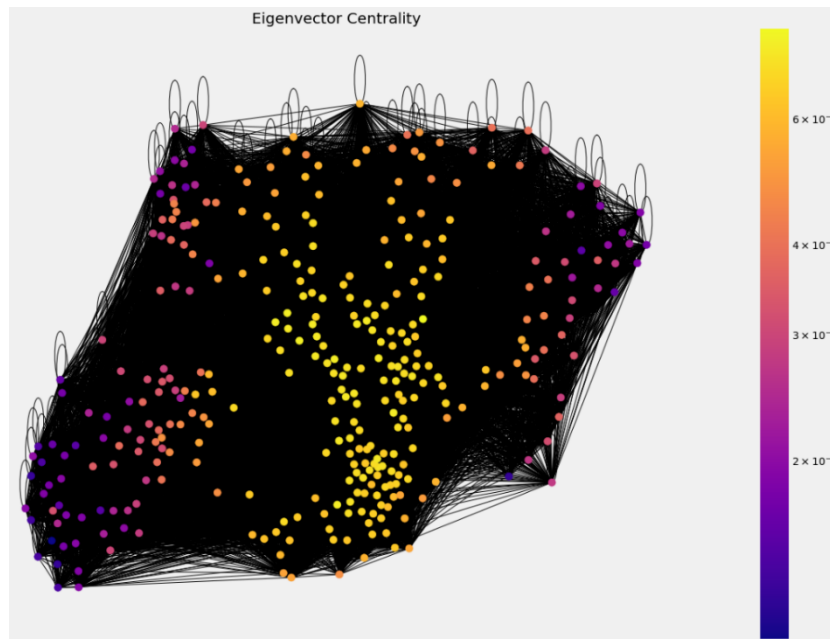
*Figura 12: Distribuzione dell'eigenvector centrality*

Nel nostro caso è possibile osservare come la distribuzione (Figura 12) di questa misura ricalchi abbastanza fedelmente l'andamento della degree centrality. Questo comportamento può essere fatto risalire, come nel caso precedente, alla maggiore densità di links nella parte interna della rete (corrispondente al centro città). D'altra parte, per lo stesso motivo si ha anche un abbassamento dei valori medi di eigenvector centrality.

Anche in Figura 13, si può notare una similarità sostanziale con la degree centrality; infatti, le dieci stazioni con eigenvector centrality più alta sono:

- Haukilahdenkatu: 0.08017456245366736
- Paciuksenkaari: 0.07660782649717626
- Huopalahdentie: 0.07640991844763213
- Laajalahden aukio: 0.07582533040235645
- Töölöntulli: 0.07577013109459187
- Itämerentori: 0.07564122953604463
- Ympyrätalo: 0.07556224884855904

- Tilkanki: 0.07537363940444101
- Pasilan asema: 0.0751848451773669
- Linnanmäki: 0.07473912538258863

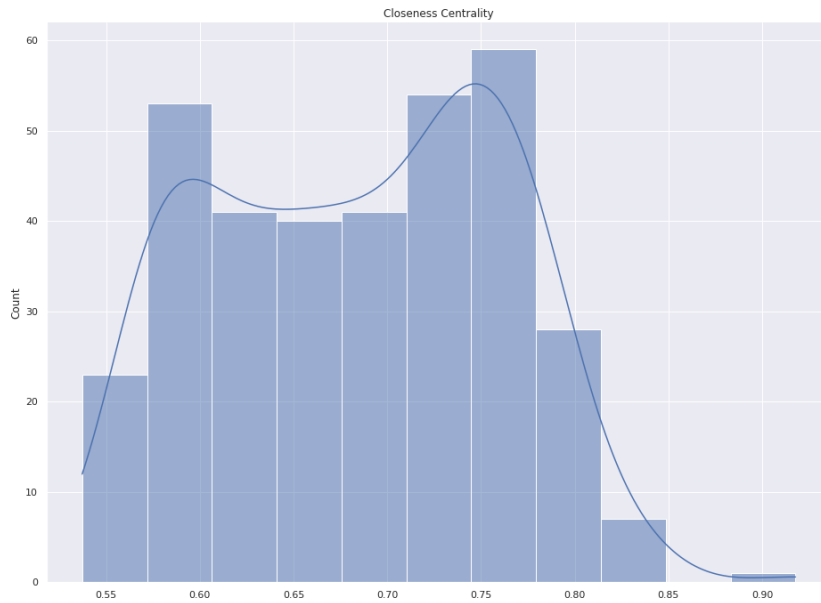


*Figura 13: Eigenvector centrality della rete*

In una rete riguardante i trasporti pubblici, l'eigenvector centrality permette di sottolineare non solo l'importanza dei singoli nodi, ma anche zone all'interno delle città che possono avere una certa rilevanza. Infatti, l'importanza di una stazione deriva soprattutto dalla relazione con gli altri nodi, piuttosto che da caratteristiche geografiche (posizione, dimensione).

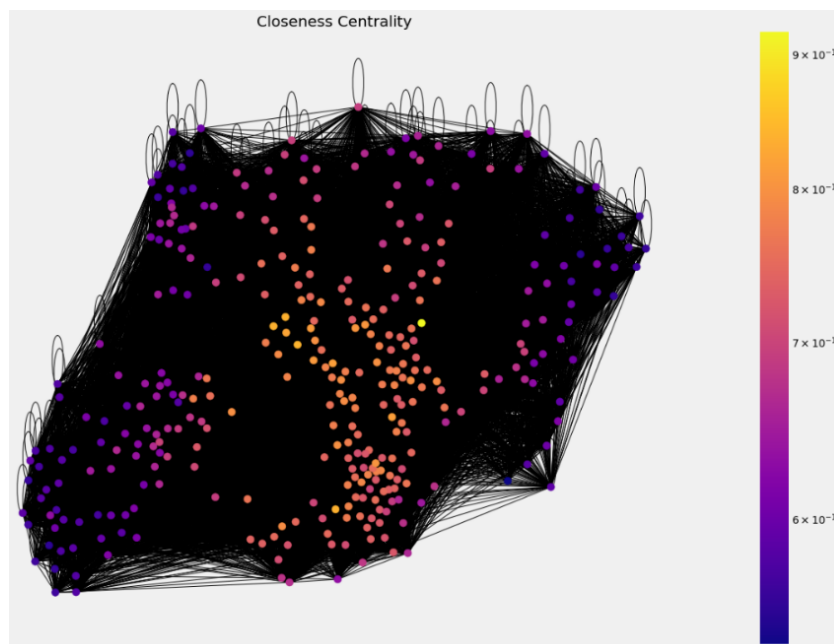
#### 4.4 Closeness Centrality

L'ultima misura di centralità da noi valutata è stata la closeness centrality. Per ogni nodo, essa fornisce un valore che indica la capacità di quel nodo di distribuire informazione nella rete. Viene misurata tramite il calcolo della distanza media del nodo da tutti gli altri nella rete. Nel nostro caso, un nodo caratterizzato da un alto valore di closeness centrality indica una stazione che potrebbe fungere da intermediaria per le altre stazioni. Dalla [Figura 14](#) si osserva come la concentrazione maggiore di punti si trovi nella zona dei valori medio-alti di closeness centrality. Questo fenomeno può essere motivato considerando che il sistema di biciclette pubbliche è stato studiato con precisione, ponendo particolare attenzione all'ottimizzazione della distribuzione geografica delle stazioni.



*Figura 14: Distribuzione della closeness centrality*

Osservando la cartina di Helsinki, infatti, si può notare come le stazioni siano posizionate in maniera decisamente omogenea su tutta l'area della città, senza lasciare zone vuote fra un punto e l'altro. Nell'ambito del bike-sharing, gli utenti potrebbero essere motivati a fare delle fermate nelle stazioni intermedie in modo da evitare pagamenti maggiori.



*Figura 15: Closeness centrality della rete*

Come visibile in [Figura 15](#) e come descritto in precedenza, un gran numero di stazioni sono connesse direttamente a Haukilahdenkatu e questo nodo è caratterizzato dalla closeness centrality più alta ( $0.9177718832891246$ ). Come prevedibile dalla struttura stessa della rete, la closeness centrality delle altre stazioni segue lo stesso andamento della degree centrality.

## 5. Analisi delle strutture

Un altro aspetto molto interessante appartenente al campo della social network analysis riguarda lo studio delle sottostrutture della rete. È possibile definire molte tipologie di sotto grafi ed ognuno di questi ci fornisce informazioni specifiche riguardanti la rete come la robustezza, la coesione o l'interconnessione. In questo progetto, ci focalizziamo soprattutto sui concetti di *Community* e di *Triade*.

### 5.1 Community detection

Una tipologia di sottostrutture interessanti all'analisi di una rete è rappresentata dalle communities. In maniera intuitiva, queste ultime possono essere definite come dei sottoinsiemi di nodi della rete, fra i quali esiste un'interconnessione più densa rispetto al resto del grafo. Riguardo al bike-sharing, l'identificazione di communities può aiutare a comprendere eventuali pattern nell'utilizzo del servizio o a focalizzarsi su elementi di natura economica (come l'adeguamento dei prezzi).

Il processo di community detection avviene attraverso la creazione di una struttura ad albero chiamata dendrogramma, nella quale le foglie rappresentano i nodi della rete e gli archi servono a connettere nodi o gruppi di nodi, di fatto andando a creare una struttura gerarchica di communities innestate (Figura 16).

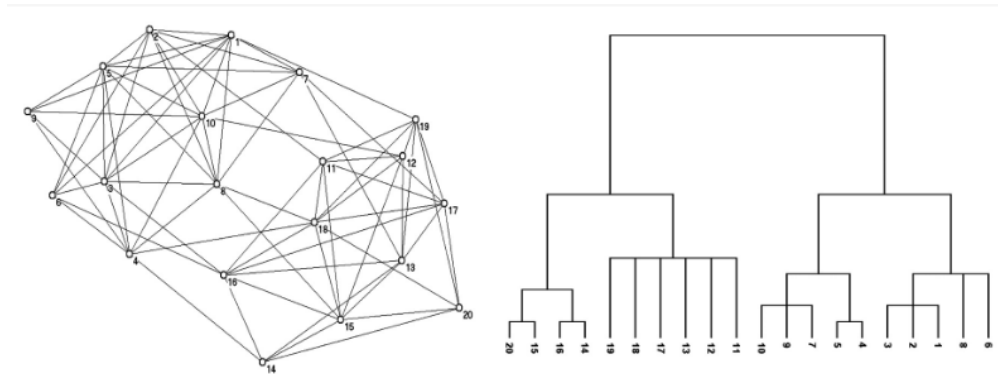


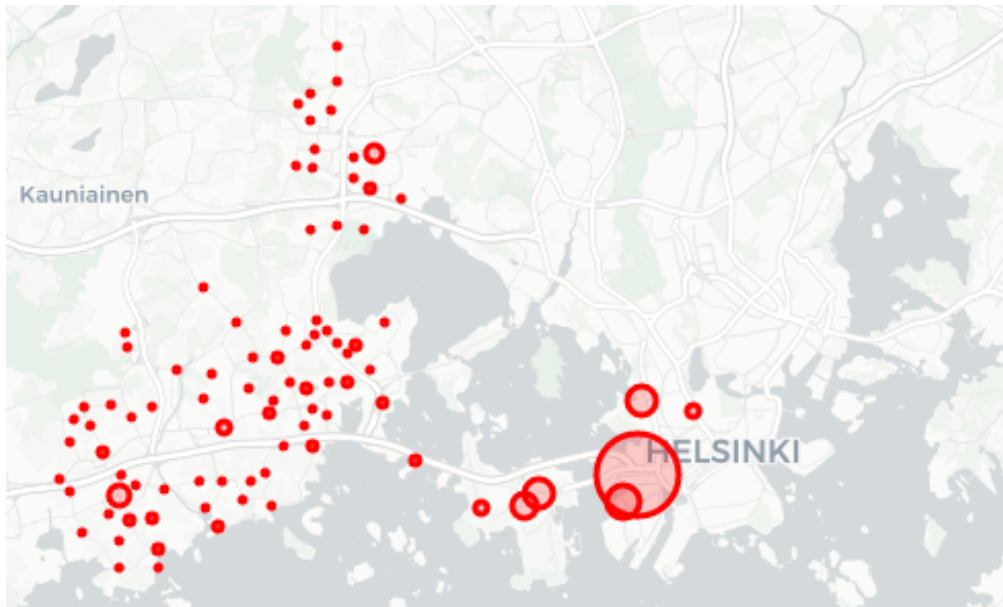
Figura 16: Dendrogramma

Applicato questo procedimento alla rete di stazioni di biciclette, come risultato abbiamo ottenuto una lista di array rappresentanti le varie comunità riscontrate. Il più grande di questi sottoinsiemi contiene 88 stazioni mentre le più piccole contano sotto i 10 elementi.

La presenza di una community così numerosa all'interno della nostra rete è un'ulteriore riprova di quanto sia ben progettata tale infrastruttura, la quale



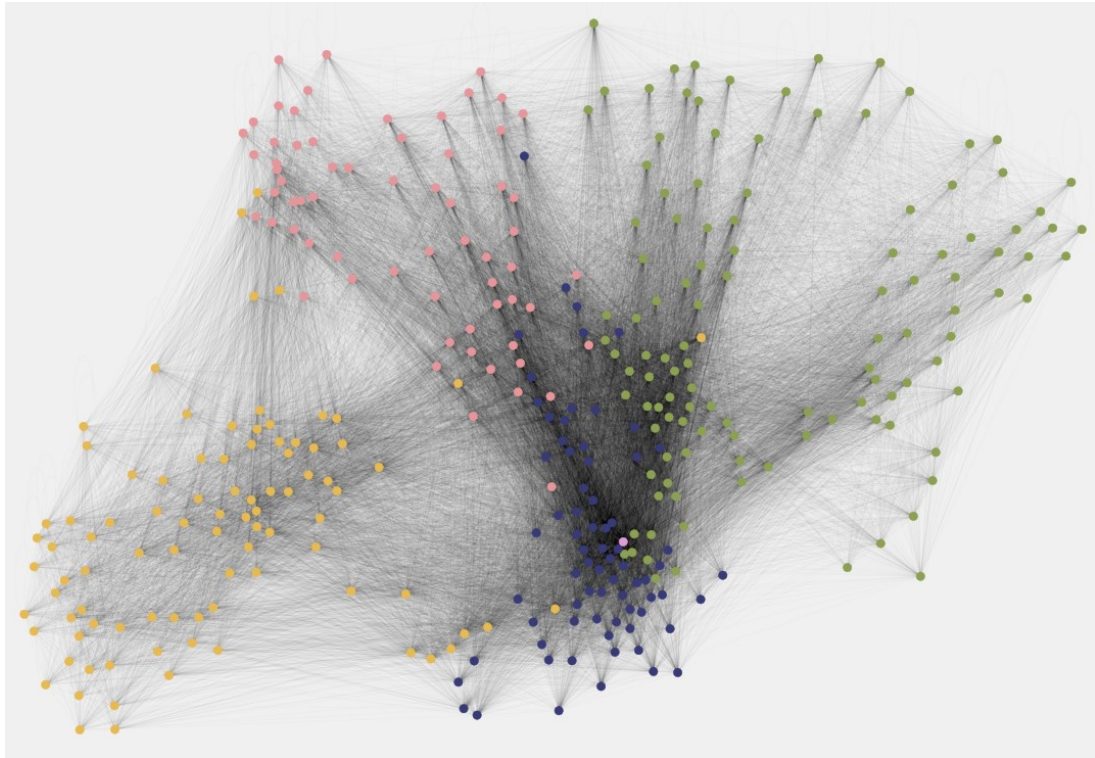
garantisce una distribuzione altamente capillare di stazioni. In [Figura 17](#), viene riportata la community descritta sulla mappa della città di Helsinki.



*Figura 17: Community più grande*

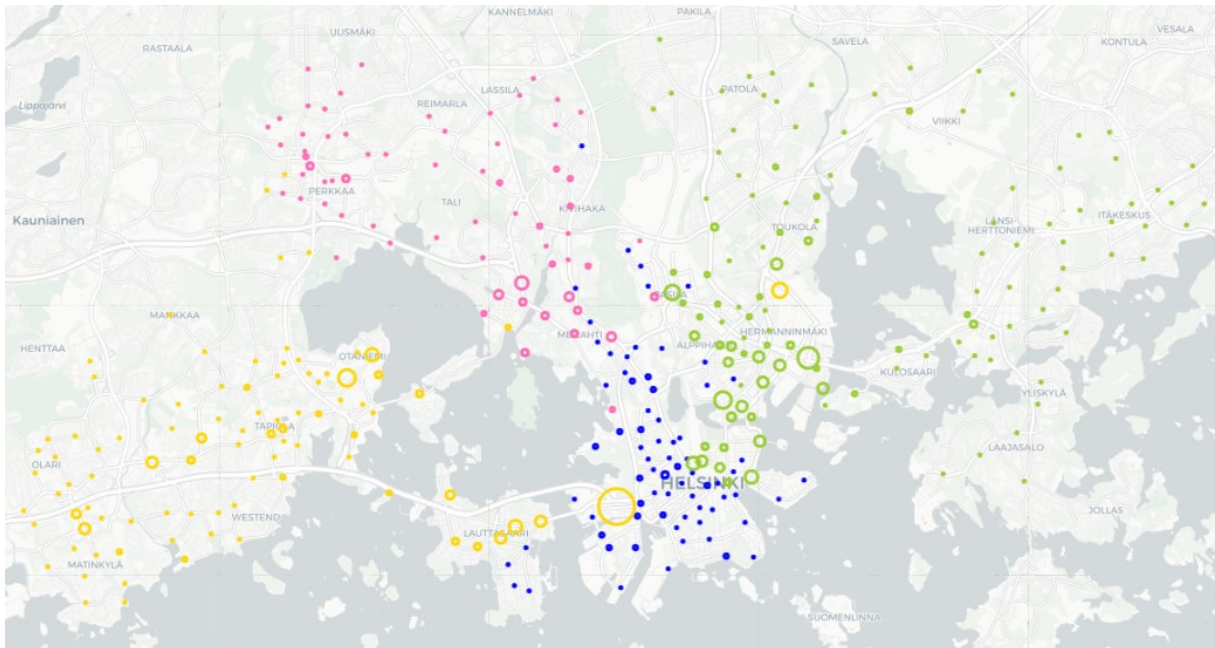
Dato che la libreria NetworkX fornisce soluzioni differenti per la ricerca di communities, è stato applicato il metodo di Louvain che risulta essere semplice e immediato. Infatti, esso è un metodo euristico che si basa su due step: nella prima fase ogni nodo rappresenta una comunità e si assegnano i nodi alle comunità limitrofe solo se si trova un guadagno positivo della modularità (si giunge al termine quando non ci sono miglioramenti del valore della modularità); nella seconda fase si costruisce una nuova rete dove ogni nodo fa parte della comunità assegnata in precedenza.

Applicando la funzione *louvain\_communities* alla rete, si ottengono quattro comunità distinte, identificate dai colori blu, verde, giallo e rosa ([Figura 18](#)). Ciò significa che l'uso della bicicletta all'interno di queste comunità è più alto di quello tra di esse. In aggiunta, si può constatare che i nodi con betweenness centrality più alta corrispondono a zone di frontiera tra differenti comunità, confermando le osservazioni precedenti.



*Figura 18: Comunità rilevate con il metodo di Louvain*

Dal punto di vista geografico ([Figura 19](#)), è interessante notare che le comunità riprendono le suddivisioni spaziali di Helsinki e che quindi la presenza del mare causa la separazione delle comunità stesse (infatti c'è un arcipelago a sud-ovest). Ciò testimonia anche la presenza di vincoli fisici nello sviluppo dell'infrastruttura di bike-sharing.



*Figura 19: Rappresentazione geografica delle comunità rilevate con il metodo di Louvain*

## 5.2 Triadi

In un grafo orientato, per triade si intende un gruppo di 3 nodi A, B e C fra i quali possono esistere svariate combinazioni di collegamenti. Il modo in cui questi tre nodi sono connessi fornisce informazioni riguardo alla capacità della rete di diffondere informazioni e riguardo al coefficiente di robustezza. Prendendo come esempio la triade chiusa (in cui tutti e 3 i nodi sono collegati), essa rappresenta la forma ultima di robustezza, in quanto se anche uno dei 3 link dovesse venire a mancare, l'informazione sarebbe comunque in grado di raggiungere tutti e 3 i punti.

In particolare, considerando un grafo diretto, è possibile definire sedici tipologie di interconnessione fra i nodi A, B e C; ognuna di queste combinazioni è caratterizzata da un codice identificativo. Nella [Figura 20](#), riportata di seguito, si lega ogni conformazione al relativo codice.

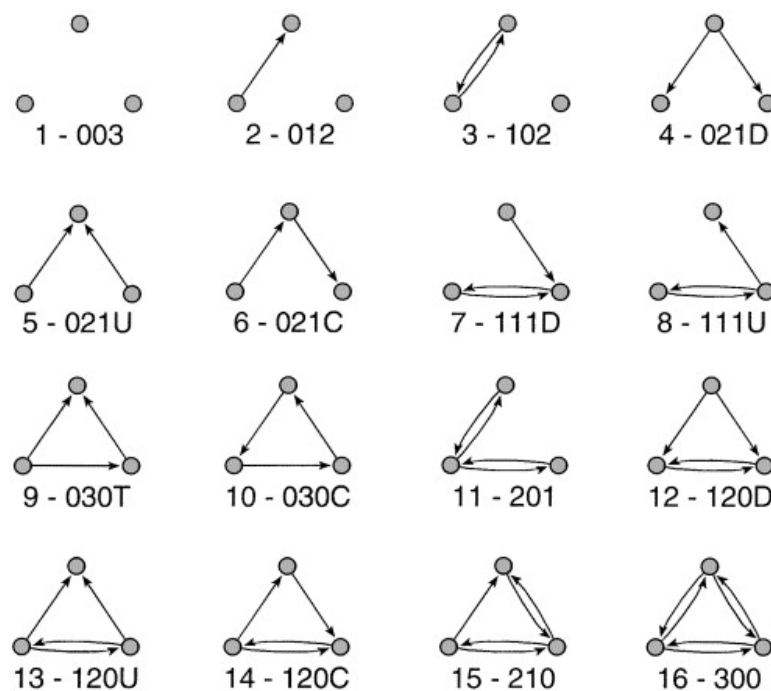


Figura 20: Possibili interconnessioni tra 3 nodi

Per valutare la numerosità di ognuna di queste triadi all'interno della rete, è stato necessario ricavare un grafo orientato. Tale operazione è svolta dal comando `from_pandas_edgelist` (già utilizzato in precedenza) all'interno del quale va specificato che gli archi sono contraddistinti da una direzione, passando all'attributo `create_using` la classe "DiGraph" ([Figura 21](#)).

```

digraph = nx.DiGraph()
G = nx.from_pandas_edgelist(df, source="departure_id",
                           target = "return_id",
                           edge_attr="distance (m)",
                           create_using=digraph)
print(G)

```

Figura 21: Creazione del grafo diretto

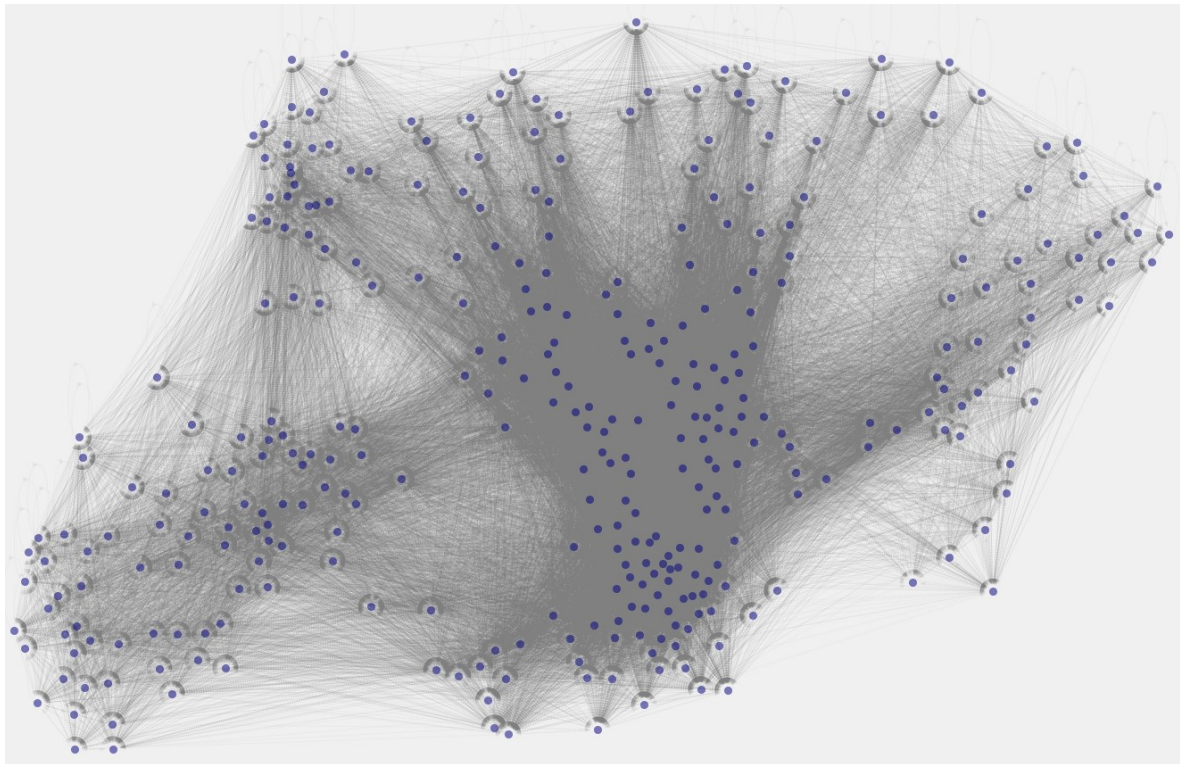


Figura 22: Grafo diretto che rappresenta la rete

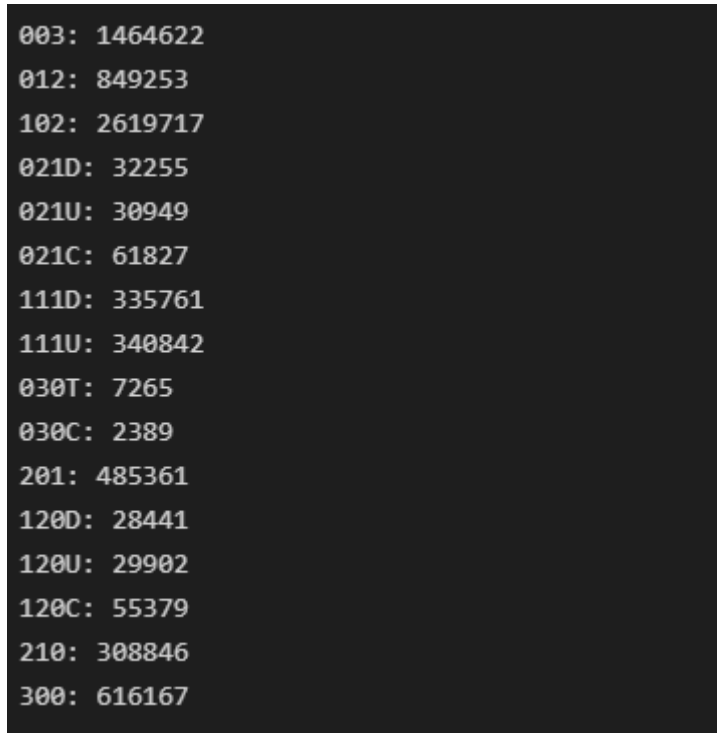
Ottenuto il nostro grafo diretto (Figura 22), per contare tutte le occorrenze di triadi è stato sfruttato il metodo *triadic\_census* della libreria NetworkX (Figura 23). Come risultato si ottiene un dizionario che lega ad ogni codice un valore intero, rappresentante la numerosità riscontrata nella rete della triade in questione.

```

triadic_census = nx.triadic_census(G)
for key, value in triadic_census.items():
    print(f"{key}: {value}")
✓ 19.8s

```

Figura 23: Comando per ottenere le triadi



003:	1464622
012:	849253
102:	2619717
021D:	32255
021U:	30949
021C:	61827
111D:	335761
111U:	340842
030T:	7265
030C:	2389
201:	485361
120D:	28441
120U:	29902
120C:	55379
210:	308846
300:	616167

*Figura 24: Triadi presenti nella rete*

Osservando i risultati (Figura 24) è impossibile non notare l'alto numero complessivo di triadi trovate; questi valori sono così alti a causa del gran numero di archi presenti nella nostra rete. In particolare, è interessante osservare la forte presenza di triadi complete (61 mila istanze della triade "300", cioè completamente chiusa), che permette di ribadire la forte interconnessione e la robustezza che caratterizzano questa rete di stazioni.

## 6. Conclusioni

Tramite l'uso della libreria NetworkX è stato possibile analizzare una rete complessa, come quella descritta dal servizio di bike-sharing nella città di Helsinki. All'inizio è stato eseguito uno studio del dataset in modo da capire la grandezza della rete e la numerosità delle stazioni. Una volta che è stata rappresentata la rete su un grafo indiretto, sono state calcolate le metriche principali, al fine di avere una panoramica più precisa. Proprio per questo motivo, è stata utilizzata una visualizzazione geografica e la corrispondenza dei nodi sulla mappa di Helsinki ha permesso di rilevare peculiarità, precedentemente non individuate. A tal punto, si è proceduto con lo studio delle centralità e sono state considerate le stazioni più importanti. In conclusione, è stata eseguita l'analisi delle strutture, con un'attenzione particolare sulla community detection e sulle triadi: nel primo caso sono stati utilizzati due diversi metodi messi a disposizione dalla libreria di Python, trovando varie comunità e le relative corrispondenze sulla mappa; invece, riguardo al secondo caso, a partire dallo stesso dataset è stato ricavato un grafo diretto e sono state studiate le triadi rilevate in esso.