

Primo progetto di Social Computing

Anno Accademico 2020-2021

Bazzetto Luca, matricola: 144760
Campagnolo Agustin, matricola: 142294
Piovesana Edoardo, matricola: 142332
Verardo Thomas, matricola: 143735

Svolgimento dei punti

Punto 1

Per scaricare gli utenti “followers” dei cinque account richiesti, ovvero @mizzaro, @damiano10, @Miccighel-, @eglu81 e @KevinRoitero, sono stati trovati inizialmente gli ID di questi account. Successivamente, anche gli ID dei loro *followers*. Questi dati sono stati serializzati in un file (*followers_nomeutente*) per ciascun utente.

Per gli utenti “*friends*” è stato eseguito lo stesso procedimento (*friends_nomeutente*).

Punto 2

Followers dei followers

Per ogni utente sono stati estratti 5 utenti followers in modo casuale dal rispettivo file. Per fare ciò è stata usata la libreria “*random*”.

Successivamente, per ciascuno di questi, sono stati scaricati ulteriori 10 utenti followers e tutti serializzati nel file *followersOfFollowers*.

Punto 3

Following dei following

Per ogni utente sono stati estratti 5 utenti following a caso dal rispettivo file *json*, per farlo è stata usata la libreria “*random*”.

Successivamente, per ciascuno di questi, sono stati scaricati ulteriori 10 utenti following nel file *friendsOfFriends*.

Punto 4

Come prima cosa, tutti gli utenti precedentemente scaricati sono stati messi in un’unica lista. Per ottenere questo risultato è stata utilizzata la funzione *mergeFiles(file_names)*.

Dopodiché sono stati eliminati tutti i duplicati per non impiegare maggior tempo nello scaricamento dei dettagli di tutti gli utenti (*deleteDuplicates(allItems)*).

Per ogni utente recuperato sono stati scaricati i dettagli del profilo utilizzando *api.get_user* e salvati in nel dataframe *usersDetails.csv*.

I dettagli scaricati sono i seguenti: *id*, *name*, *screen_name*, *location*, *followers_count* e *friends_count*.

Punto 5

Gli utenti della lista sono stati confrontati con i 5 account iniziali per stabilire il tipo di relazione esistente. con la funzione *friendship(sourceid, targetid)* che, dati due utenti, restituisce la relazione fra questi.

I possibili tipi di relazione sono *Both*, ovvero gli utenti si seguono a vicenda, *SourceToTarget*, ovvero

un utente (denominato “source”) segue un altro utente (“target”) ma non viceversa. Infine, c’è la relazione *TargetToSource*, che come si può intuire, è il contrario della precedente.

Successivamente, sono stati eliminati gli account che non avevano alcuna relazione con i 5 account iniziali, in modo tale da avere un grafo connesso.

Utilizzando il dataframe con i dettagli degli utenti (*usersDetailsForGraph.csv*) e il dataframe delle relazioni (*sourceTarget.csv*) sono stati creati i grafi.

Inizialmente è stato deciso di costruire due grafi con gli stessi nodi e stessi archi: uno orientato e uno non orientato. Questo per poter esaminare al meglio il grafo. Ad entrambi i grafi, son stati aggiunti i membri del gruppo del progetto come attributi del grafo.

Durante la costruzione del grafo è stato inserito l’*id* di ciascun utente per identificare il nodo, assieme agli attributi, ovvero *name*, *screen_name*, *location*, *followers_count* e *friends_count*.

Punto 6

La visualizzazione interattiva del grafo è stata resa possibile utilizzando il codice fornito a lezione ed è possibile vederla nel seguente [link](#).

Punto 7

Per verificare se il grafo connesso è stata utilizzata la funzione “*is_connected*”, la quale ha ritornato il valore “*true*” che sta a significare che il grafo è connesso, infatti tutti i nodi del grafo sono collegati da almeno un arco.

Inoltre, con la funzione “*is_bipartite*” è stato verificato che il grafo non è bipartito. Infatti, l’insieme dei nodi del grafo non può essere partizionato in due sottoinsiemi X e Y tali che ciascun arco del grafo ha una delle due estremità in X e l’altra in Y.

Punto 8

Le misure del centro, del diametro e del raggio possono essere calcolate solamente se il grafo è connesso, quindi dopo averlo verificato nel punto 7, sono risultate le seguenti misure:

- Centro: [19659370, 132646210, 3036907250], ovvero [@eglu81, @damiano10, @KevinRoitero];
- Diametro: 4 archi;
- Raggio: 2 archi.

Punto 9

Betweenness centrality

Misura quanto un nodo è importante nel connettere altri nodi:. Si deduce dal grafo [qui presente](#) che i nodi più importanti sono i nodi dei 5 utenti iniziali.

Closeness centrality

Questa misura indica quali sono i nodi centrali che possono raggiungere velocemente gli altri nodi. Da [questa figura](#) si nota infatti, osservando la scala di colori, quali sono i nodi con Closeness Centrality più alta.

Degree centrality

Misura un valore che più è alto, più un nodo è centrale nel grafo. Infatti, in [questo grafo](#), si nota che i nodi non colorati di blu, sono quelli più centrali.

In-degree centrality

In-degree centrality misura il numero di archi entranti in un nodo. Come si può notare da [questa figura](#), i nodi con più “followers” sono i 5 account iniziali.

Out-degree centrality

Out-degree centrality stima il numero di archi uscenti da un nodo. Nel grafo [qui presente](#) si può notare infatti che i nodi non colorati di più sono i nodi che hanno più “following”, cioè che seguono più account.

Page Rank

Il Page Rank è un algoritmo che misura l’“importanza” di un nodo. Più un nodo è collegato da altri nodi importanti più è importante. Come in [questo caso](#) i nodi più importanti sono i 5 nodi iniziali.

HITS

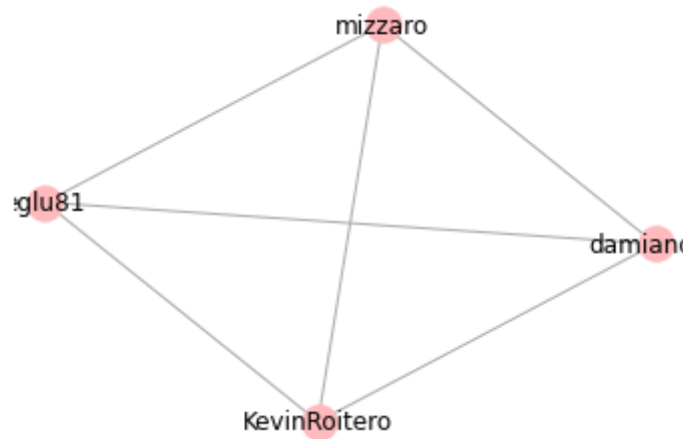
Con l’algoritmo di HITS, noto anche come l’algoritmo di Kleinberg, vengono calcolati due score differenti: Hub e Autorità. Questi due score determinano l’“importanza” di un nodo. In [questo grafo](#) si nota come varia lo score Hub, mentre in [questo](#) come varia lo score Autorità.

Punto 10

Il sottografo del nodo @KevinRoitero si trova nel seguente [link](#).

Cricca massima

Quanto gli amici di una persona sono amici fra di loro:



Dimensione della cricca massima: 5.

Punto 11

Siccome il grafo è connesso, è stato possibile calcolare la copertura minima degli archi, la quale mostra le relazioni minime tra i nodi del grafo. Successivamente, è stato disegnato il grafo con *pyvis* per ottenere una visualizzazione interattiva. Da questa visualizzazione è stato dedotto che il grafo ha meno archi perché anche se un nodo è collegato da più archi (relazioni), ne viene rappresentato solamente uno.

Punto 12

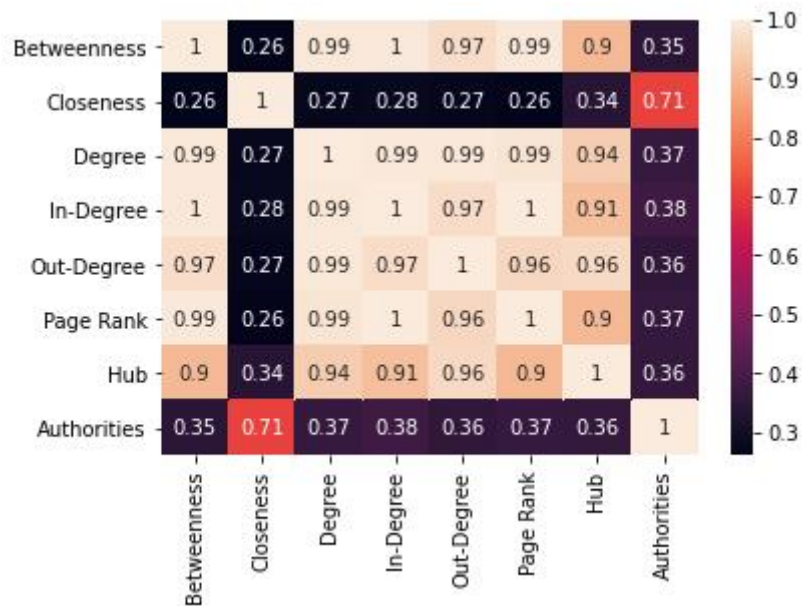
Coefficienti del piccolo mondo:

- Coefficiente omega: 0,000778;
- Coefficiente sigma: 0,982465.

Punto 13

La seguente tabella rappresenta la correlazione lineare di Pearson Rho. Si può notare che la variabile *Closeness* è la variabile che correla meno con tutte le altre variabili. Questo è dovuto al fatto che le relazioni trovate sono state create a partire dai 5 account iniziali.

La variabile *Betweenness* invece correla con quasi tutte le variabili perché i 5 account iniziali sono i cardini del grafo.



La tabella sottostante rappresenta la correlazione di Kendall Tau. Più il valore è alto, più due variabili hanno rank simile, di conseguenza, più il valore è vicino allo 0, più il rank fra queste due variabili è diverso.

Un esempio fra due variabili che correlano tra loro è *Hub* e *Out-Degree*.

