Goodreads: A Social Network Analysis

Giulia Ferro
g.ferro10@studenti.unipi.it
Student ID: 597681

Roberto Caldari
r.caldari@studenti.unipi.it
Student ID: 649249

ABSTRACT

The Social Network Analysis can be a very insightful tool to understand how to promote a product in the market. The aim of this project is to understand the role of the so-called "book-influencers" in the book market, the structure of the communities that they belong to and the influence that they have. The network for the analysis collects data from one of the most used social platform which focus on books called Goodreads.
[1]

KEYWORDS

Social Network Analysis, Market analysis

ACM Reference Format:
Giulia Ferro and Roberto Caldari. 2019. Goodreads: A Social Network Analysis . In *Social Network Analysis '22*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Goodreads is a book-centric review social network that launched in January 2007 and was acquired by Amazon in 2013. It's an app for readers which can share book recommendations and save the progress with the books they are reading, with the goal of helping people find and share books they love. The readers have also the possibility to create the personalized wish lists and bookshelves to save books they would like to read, have already read or would like to buy.
Here is possible to identify new books by searching for specific titles or authors, get curated recommendations from Goodreads based on the books read and rated, or browse lists of new releases or themed lists voted on by users. Moreover, every book page presents some links to Amazon website or Ibis to buy the product we are interested in.

1.1 How does it works?

Once you sign up for a Goodreads account, you'll be prompted to review books you've already read and add them to your bookshelf so that Goodreads can provide curated recommendations of what to read next.
Each book on Goodreads has a dedicated page that shows the overall rating users have given a book, a summary, author information, and reviews from the Goodreads community. Here, you can ask a question, write your review, and like or comment on other reviews.

1.2 Goodreads: web scraping

The data is collected through web scraping due to the fact that Goodreads has disabled the developer key required to use the API in 2020. The code used to implement this step is in the Data scraping notebook.

2 OUR PROJECT

The main goal of the project is to analyze the network to understand its structure and properties. In this kind of social is interesting to understand the connections between people (followers and friends), which allow us to learn about the interactions, the common interests and needs of the users. This process enables us to figure out how the publishing houses and Amazon can use the analysis to promote a book and to understand the market dynamics.

3 DATA COLLECTION

From the data scraping step we collected two different files. The first one called nodes.csv collects only the nodes information. The result is a dataset with the following attributes:

(1) Index which represents the node id;
(2) href that is part of the link to the user's Goodreads page;
(3) books.

The second file created is called Links.csv and it collects all the links of the graph. In this case we have only two attributes which are two nodes id if there is a friendship connection between the users.

[1] **Project Repositories**
Data Collection: https://github.com/sna-unipi/data-collection
Analytical Tasks: https://github.com/sna-unipi/analytical-tasks
Report: https://github.com/sna-unipi/project-report

SNA '22, 2021/22, University of Pisa, Italy

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$0.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

3.1 Selected Data Sources

As we have said before, we started to collect data directly from Goodreads page using web scraping.
Goodreads already has a page with the list of the most followed users from which the data scraping task started.

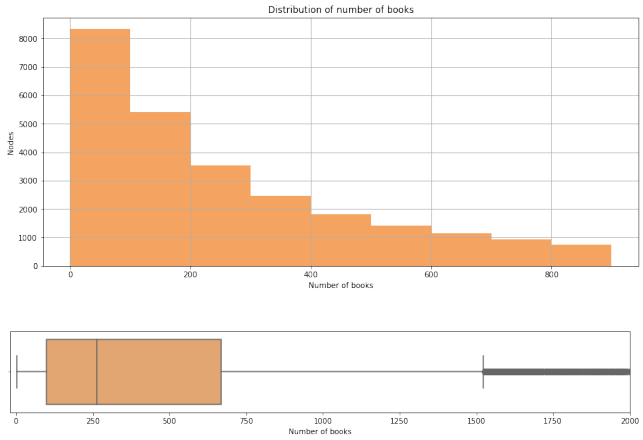
3.1.1 Crawling Methodology and Assumptions. As first step we decided to create a dictionary of the 50 most followed people of the Goodreads site in Italy. Between this users is made a selection of 37 readers by deleting the profiles of writers that are in the list just because they have a lot of followers but don't really use actively the platform. This 37 nodes will be very important because we will focus on them during our analysis.
This first list of nodes is then extended by adding all the 37 users' friends and their number of books. We decided to take the friends instead of the followers to build a smaller network and also because the friendship link seemed more tight than the followers one, which suggest that this users are more active and invested in the book platform. The following image show the first rows of the file. The information about net number of friends and the number of followers are available only for the 37 most followed user, so they aren't use during the analysis.

	index	href	name	books	friends	followers
1	1	/user/show/5347823...	Luca Ambrosino	1524	1457	13777
2	2	/user/show/1892253...	Ilenia Zodiaco	994	5001	10436
3	3	/user/show/4107786...	Matteo Fumagalli	2258	5001	5962
4	4	/user/show/5000946...	Julie Demar	556	5000	3586
5	5	/user/show/1457965...	Mattia Ravasi	445	931	2990
6	6	/user/show/5312173-...	Simona B	1525	437	2864
7	7	/user/show/4196681-...	Violet wells	1330	2555	2580
8	8	/user/show/5969042...	Guglielmo Scilla	197	1568	2164
9	9	/user/show/1279034...	Fiore Manni	1153	856	1674
10	10	/user/show/4740832...	Tiffany Miss.Fiction	1341	2450	1658
11	11	/user/show/1309559...	Labibliotecadidaphne	135	4285	1503
12	12	/user/show/3759782...	Maura Gancitano	780	352	1436
13	13	/user/show/1049827...	Cesare Cantelli	201	2335	1402
14	14	/user/show/5196970...	Francesca Crescentini	493	105	1377
15	15	/user/show/1049022...	Orsodimondo	2484	2927	1352
16	16	/user/show/1330667...	Silvia	1632	756	1312
17	17	/user/show/8622274...	booksnpenguins (des...	1119	2087	1266
18	18	/user/show/4413189...	Mirko Smith	877	2166	1007
19	19	/user/show/3402625...	Valentina Ghetti	725	2929	982
20	20	/user/show/6873893...	Andrea Beffiori	398	2119	925
21	21	/user/show/3588363...	Chiara Santamaria	486	711	891
22	22	/user/show/3271066...	Emma	1875	1676	860
23	23	/user/show/4121866...	Beatrice in Bookland	856	706	848
24	24	/user/show/1207041...	Anna Premoli	102	136	823
25	25	/user/show/4238317...	Chantal	717	2610	764
26	26	/user/show/11895448...	Gaia Lapasini	352	1661	761
27	27	/user/show/1204794...	Michele Monteleone	901	425	737
28	28	/user/show/4431719-...	Gianfranco Mancini	11317	620	698
29	29	/user/show/3209403...	Vanessa	741	90	650
30	30	/user/show/4314145...	Giada--ReadingCanS...	211	1748	621
31	31	/user/show/5937341...	Gaetano Pagano	125	283	602
32	32	/user/show/3805200...	Marco Cantoni	1042	917	588
33	33	/user/show/1372167...	sos.anna	70	511	587
34	34	/user/show/5416581...	Nerily	363	820	585

The second csv file with the links is created by passing all the nodes' friends and adding a tuple (node1,node2) if node2 is a friend of node1 and if it is present in the nodes original list. This step was very time consuming, so we checked all the friends for the 37 most followed user (in this way every node has at least degree 1) but set a limit for the others. The minimum number books has to be at least 100 to check the node's friend to be sure that the reader is an active user of the Goodreads account.

4 NETWORK CHARACTERIZATION

The only information that we have to characterize our network is the number of books for every user. This value represents the number of books that a reader has added in all the bookshelves or wish lists. This means that the value takes into account books already read, that the reader would like to read or would like to buy. In this way, the value can be useful to understand not only the kind of users (its level of activity), but also how easily is for them to add books in their wish lists, an important information that will be used later on in the parameters setting of some of the algorithms. In the following graph we can see the distribution of this value.

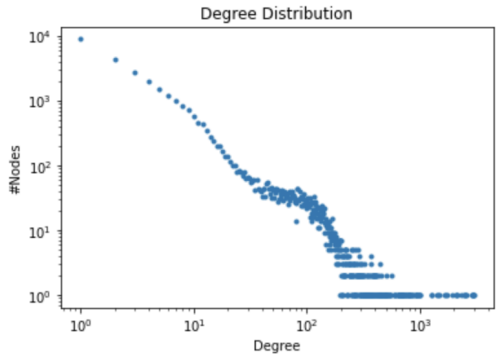


4.1 Network Analysis : Graph Analysis

The graph analysis it's an important initial step to better understand network features. In the following list we have some basic information:

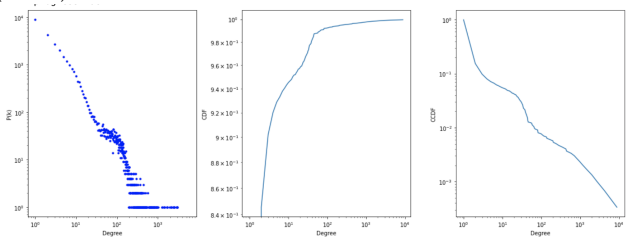
- Number of nodes: 31394
- Number of edges: 332118
- Average degree: 21.158
- Average clustering coefficient: 0.244
- Diameter: 6
- Density: 0.00067397

The graph is indirect because every link represents the friendship relation on the social network and it is composed by a unique connected component. The the graph is dense and has a high clustering coefficient as expected from a real graph. If we select only the most followed users, we have an average degree of 1453 which is due to the fact that these nodes are the hubs of our network. Focusing on the degree of the network, we can see the degree distribution of the nodes:



The distribution shows that the majority of the nodes has few links, while there are few nodes, the hubs, that have a very high degree.

In the graph below we can see the comparison between the degree distribution of the network, the Cumulative distribution Function (CDF) and the Complementary cumulative distribution Functions (CCDF).



4.2 Comparison with Erdos Renyi model

To better understand the network, we built a ER model using as parameters the same number of nodes and as probability the density of the real graph. The following table shows the differences between the two resulting models.

	Real Graph	ER Graph
Number of nodes	31394	31394
Number of edges	332118	332095
Density	0.000673	0.000673
Average degree	21.158	21.101
Average clustering	0.244	0.000676

The main diversity is regarding the average clustering coefficient, the ER model has low clustering coefficient with respect to real graphs. This is due to the fact in the ER configuration the degree distribution follows a normal distribution, that can be seen later on the graphs comparison.
Another interesting aspect is the regime analysis of the ER graph. The computations show that the probability p which is 0.000673 is higher than pc (lnN/N) which is 0.0003298, at the same time the average degree is higher than ln(N). This means that the regime of the network is connected and the graph is dense. This can be due to the initial criterion of the data selection, we expected a dense graph because we start the data scraping from the most followed readers and their friends circle whose are surely part of a dense community.

4.3 Comparison with Barabasi Albert model

Another comparison is made with the BA model. The parameters setting applies the following rules:

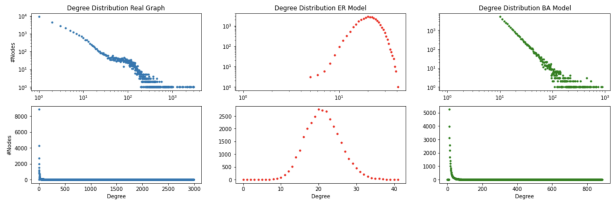
- Same number of nodes of the real graph
- The number of edges (m) that will link the new node to the existing nodes, following the preferential attachment rule, is set to 10. This value is calculated with the ratio between the number of edges and the number of nodes of the real network.

	Real Graph	BA Graph
Number of nodes	31394	31394
Number of edges	332118	313840
Density	0.000673	0.0.000636
Average degree	21.158	19.9936
Average clustering	0.244	0.00463

In this case we can see that the average degree is slightly lower in the BA model than in the ER configuration due to the presence of a lot of nodes with low degree, which means user with few friends, due to the preferential attachment rule.
Another aspect of the BA model studied is the degree of the biggest hub which is 1147 while in the real graph is higher and equal to 2987. The fraction of edges incident to the largest hub is 0.0365 for the BA graph and 0.0951 for the real graph.

4.4 Degree distribution Comparison: Real Network, Erdos Renyi, Barabasi Albert

All the information about the three model discussed can be better understood in the following graphs that show the different degree distributions of the configurations (using the log scale in the first row).



In the plot, we can see the normal distribution which characterizes the ER model in the red graph. Moreover, the similar trend of the degree distribution of the real graph (first one) and the BA model (last one) due to the presence of hubs.

5 TASK 1: COMMUNITY DISCOVERY

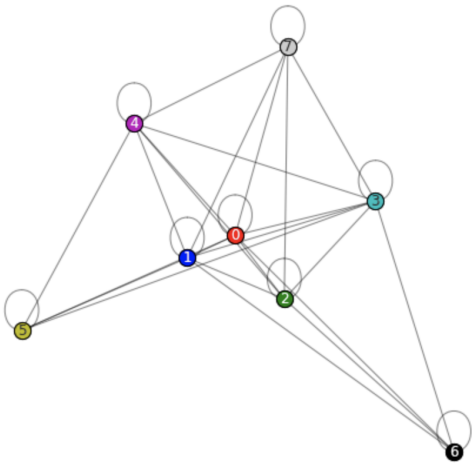
For the community discovery task, we decided to test three algorithms studied during the lectures of the year and the last one from the same library.
The algorithms tested are:

(1) Label Propagation
(2) K-Clique
(3) Angel
(4) SCAN (Structural Clustering Algorithm for Networks)

Every community configuration is compared with one another with internal and external evaluation to check the best performance.

5.1 Label Propagation

The Label Propagation algorithm is the first one tested and it gives us not overlapping communities. The graph below is a representation of the top 8 communities detected and their links.



In the following table we collected the information about the parameter setting and the results:

Algorithm	**Label Propagation**
Number of communities	105
Coverage	100%
Newman Girvan modularity	0.000673
Modularity density	0.11
Erdos Renyi Modularity	0.42

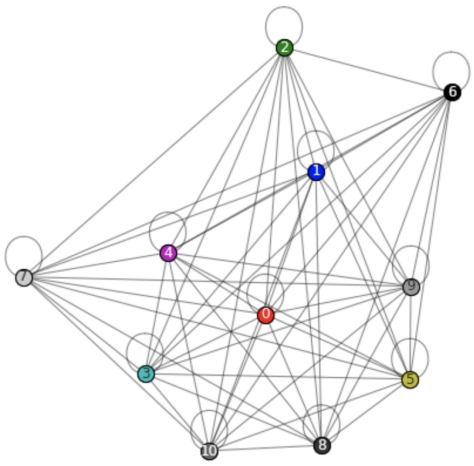
The biggest 8 communities have the following characteristics:

Community	C1	C2	C3	C4
Number of nodes	14501	9277	3369	1485
Average Internal degree	12.12	45.65	4.31	2.14
Internal edge density	0.00083	0.004922	0.00128	0.00144
Hub dominance	0.1971	0.2435	0.7004	0.996

From the table, the second community has the highest internal edge density and average internal degree. The value of the hub dominance grows when the number of nodes in the community diminish, in fact is more likely that the node with the highest degree in a small community is connected to the majority of the other nodes. In bigger communities is difficult to find very high values of hub dominance.
Focusing on the 37 hubs we have that 19 of them are in the first community, 9 of them in the second, another two share the third community and each of the remaining 7 are put in different groups. If we would like to reach the majority of the nodes we should use as strategic nodes the hubs that belong to this 4 biggest communities, in this way we could reach the 91.20% of the network.

5.2 K-Clique

The second algorithm tested is K-clique which focuses on the structural aspect of the network. In this case the communities overlap between each other and the top 11 are showed in the following graph where we can see that the communities are more linked with each other with respect to the one of the Label Propagation algorithm, but are also way smaller.



The information about the algorithm configuration and results are collected in the table:

Algorithm	**K-Clique**
Number of communities	186
Coverage	4%
Newman Girvan modularity	-0.01
Erdos Renyi Modularity	0.12

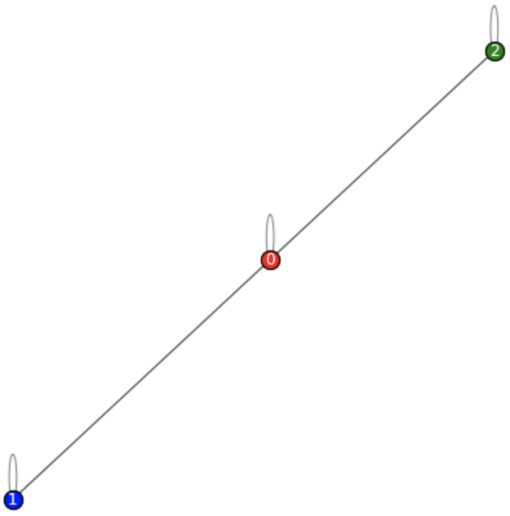
We can see that the performances are worst then the Label propagation algorithm, especially if we focus on the coverage.
If we analyze the biggest communities we have the following results:

Community	C1	C2	C3	C4
Number of nodes	716	471	20	15
Average Internal degree	84.00	57.04	14.7	12.4
Internal edge density	0.1174	0.121	0.774	0.88
Hub dominance	0.597	0.676	1.0	1.0

Given the coverage of the communities detected, the number of nodes is as expected very low, which is also why we have such high hub dominance in the third and fourth communities where the hub is connected to every other nodes in the community.
If we focus on the hubs of the network, 14 of the 37 are left out, and the majority of the other belong to multiple community. This bad performance is due to the high value of k selected as model parameter which represent the size of the smallest clique. We couldn't make this value lower due to computability reasons.

5.3 Angel Algorithm

The third algorithm tested is Angel because it is implemented by following the idea that "communities as sets of nodes grouped together by the propagation of the same property, action or information" which can be insightful in our specific case of study. Even in this case we have communities that overlap with each others, but in this case they are only 3 as can be seen in the graph.



Here there are the information about the results and the model configuration:

Algorithm	**Angel**
Number of communities	3
Coverage	59.50%
Newman Girvan modularity	-0.16
Erdos Renyi Modularity	0.39

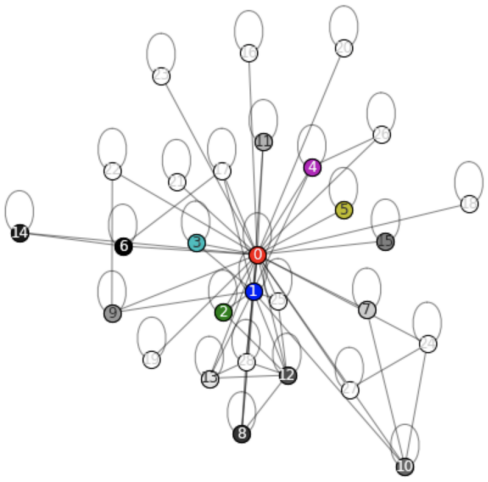
In this case we have a higher coverage of the network, the parameters chosen are a threshold of 0.30 and the minimum community size is 10. The properties of the three communities are listed in the following table.

Community	C1	C2	C3
Number of nodes	18623	16	10
Average Internal degree	33.6	1.87	2.2
Internal edge density	0.0018	0.125	0.2444
Hub dominance	0.1386	1.0	1.0

In this case, the majority of nodes are covered in the first community which also has the highest average internal degree. In this community are inserted all the 37 hubs of the network, only one of the hub belongs to all the communities.

5.4 SCAN (Structural Clustering Algorithm for Networks)

The last algorithm is called Structural clustering Algorithm (SCAN) and it was chosen because it should be able to identify densely-connected clusters, hub vertices and outliers in the graph.
Together with the Label Propagation algorithm, this model identifies communities that don't overlap.



Algorithm	**SCAN Algorithm**
Number of communities	326
Coverage	9.17%
Newman Girvan modularity	-0.01
Erdos Renyi Modularity	0.03

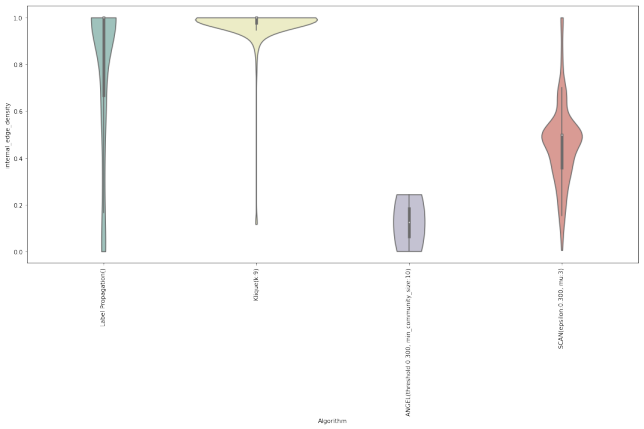
For this model the parameter setting was: 0.3 for the minimum threshold to assigning cluster membership (epsilon) and 3 for the minimum number of neighbors with a structural similarity, that exceeds the threshold epsilon (mu). Here the coverage is again low, but even by changing the value of the parameters mu and epsilon, the number of communities grows a lot which means that their sizes remain very small.

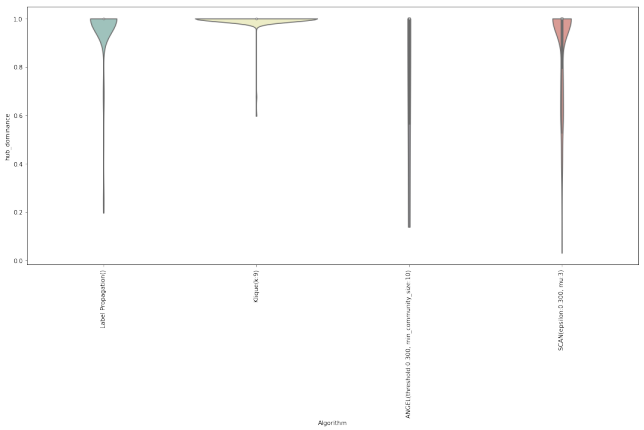
Community	C1	C2	C3	C4
Number of nodes	421	183	172	90
Average Internal degree	2.55	39.19	33.36	26.57
Internal edge density	0.0061	0.2153	0.1951	0.298
Hub dominance	0.0309	0.6923	0.46199	0.6629

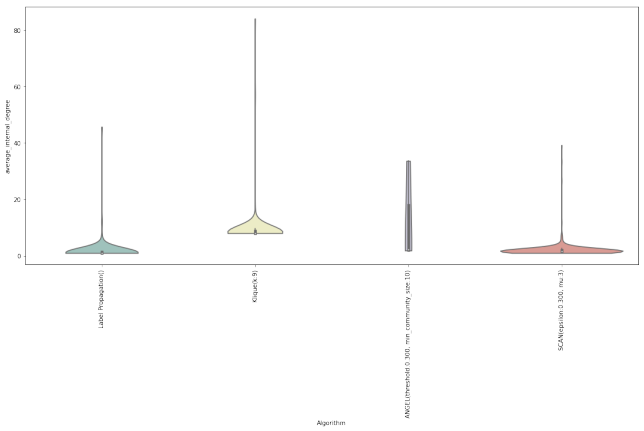
Given the high dimension of the first community, it is also the less dense with respect to the others. In this algorithm the majority of the hubs are not assigned to a community, only 5 of them belong to the same one.

5.5 Evaluation

Focusing on the different results that we've obtained, we decided to do a comparison between the communities found. The following three graphs show the difference between the characteristics of the algorithm focusing on: internal edge density, hub dominance and average internal degree.







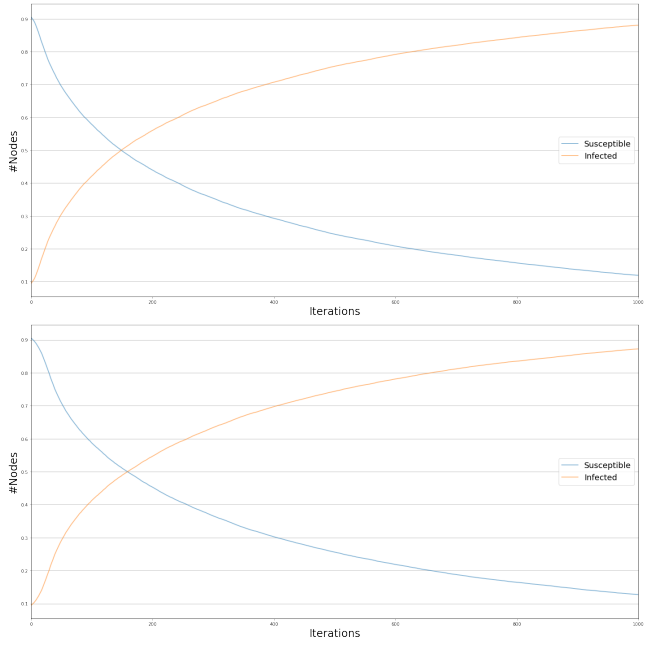
For all the algorithms, the majority of the communities has low average internal degree, the highest values can be found for the K-clique algorithm which is understandable given the fact that the algorithm start by searching clique in the network. In fact, the community found in this context are also the one with the highest internal edge density. The main problem of the K-clique algorithm (in this context) remains the low coverage of the graph, the communities found are a lot and very small.
The Angel algorithm give as result three very imbalance communities, the smaller ones have few nodes, an the biggest one include almost the 60% of the network.
The best performance, also in terms of modularity measures is given by the Label Propagation Algorithm. The information given us by this model will be used also to enhance some results later on in the analysis.

6 TASK 2: EPIDEMICS ANALYSIS

The study proceeds with the Epidemic analysis. The next algorithms are chosen for the following reasons. One of our interest is how a publishing house or Amazon should promote a book to reach the maximum number of readers. If a user adds a book in one of their bookshelf, all their friends and followers see the post. In this way, the epidemics models seem more suitable if we want to understand to which node a publishing house should send a book to reach the maximum number of users in the simplest way possible. In this case we just want to raise interest in a specific product by promoting it, than to influence the opinion on that book.
The models tested to implement this idea are: SI, SIS, SIR, Threshold model. We left out from this report the SIS model because it doesn't really apply in our context (even thought we have the result of it), so in the following section we will focus only on the other models. Each algorithm is tested on different nodes selections: on a random subsample of infected node, on a selected subsample of nodes that includes the node with the highest degree (node 4) and its neighbors, on the ER model and on the BA model.

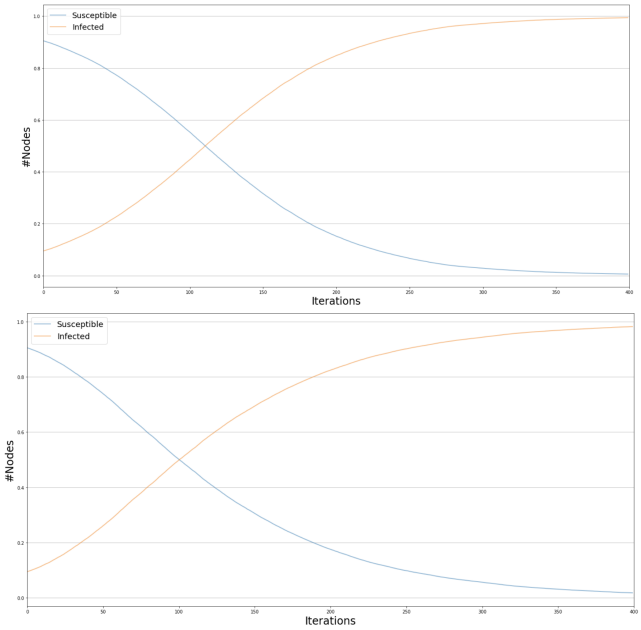
6.1 SI model

In the SI model each individual has contacts with randomly chosen others individuals per unit time, this parameter is set equal to 0.001. The initial random fraction of infected nodes is equal to the 9.5% of the network. This last value is chosen because it is the same number of nodes that we obtain by the selection of the node with the highest degree and its neighbors, in total 2987.



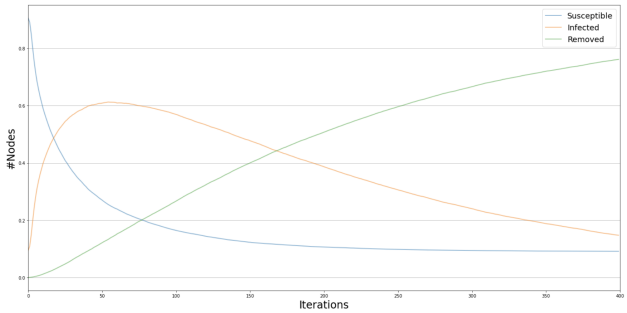
The first graph shows the random configuration, the second the

hub configuration. The model give us more or less the same results, the difference is that if we apply this concept on our case of study, in the second case we can chose to send the promotional book only to the hub (node 4) in a way that all its neighbors (that include other hubs) will be automatically reach, with the same performance of the random configuration.
The same parameters are tested also on the ER model (first following graph) and in the BA model (second graph) which present similar performances. This configurations present a faster epidemic development due to their structures.

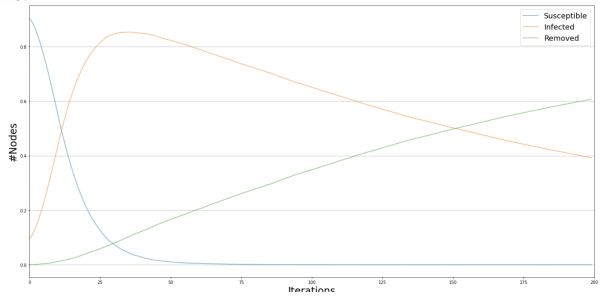


6.2 SIR Model

The SIR model can add a step in our analysis. We can consider the removed set as nodes that, after have seen the book (being infected), decide to add it to their library. In this way Amazon can also build a custom suggestion for the user that expressed that interest.
In this case each individual has still beta contacts with randomly chosen others individuals per unit time, for which is kept the value 0.01. Moreover, each infected individual has gamma probability of being remove after being infected, this parameter is set to be 0.005, which means that the probability of saving the book is smaller than the probability of seeing it.
Even in this case we tested the same configuration as for the SI model and the performance very similar, that's why we just show the model with the selected infected set with the hub.



Like before, we've also checked the performance of the ER and BA model (in the graph) and like before the still present similar performance and need less iterations than the real graph to allow the epidemic spreading, the 50% of the node are infected after 200 iterations.



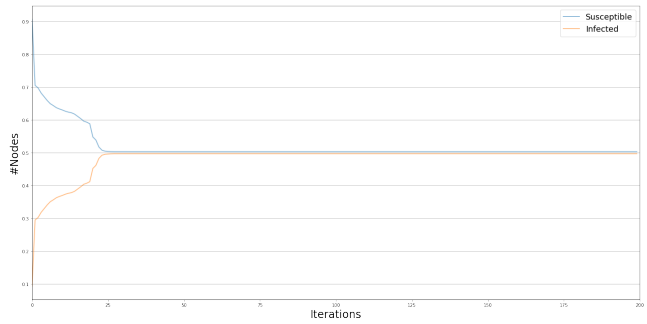
6.3 Threshold Model

For the last algorithm tested we decided to add some element to the configuration. Like already said during the network analysis, we have and addition information on the readers which is the number of books. To build the threshold model we decide to build custom thresholds for different kind of users. The readers that has a lot of saved books are usually the ones that use the platform more often and add books more easily. For this people we can set a lower threshold, which means that it is sufficient that few of their neighbors adopt a specific behaviour (in this case add the book, review it, buy it) to follow their decisions. Looking at the number of book distribution, we set the following thresholds:

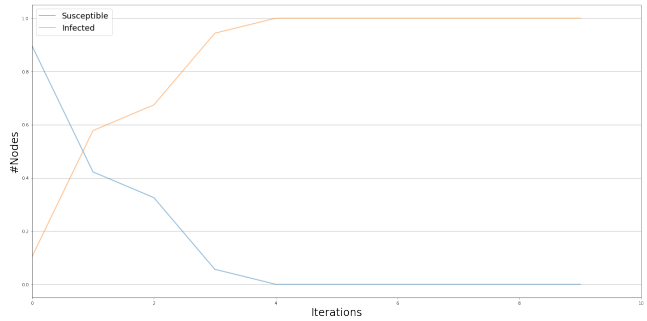
- Less than 200 books: **0.7**.
- Between 200 and 400 books: **0.6**.
- Between 400 and 600 books: **0.5**
- More than 600 books: **0.4**

The first threshold assume that the users that saved less than 200 will need that at least 70% of their neighbors adopt the same behaviour to follow them. By using the Threshold model adopting different thresholds, our goal is to understand how influenceable the nodes are based on the use that they do of the social platform. After the model configuration, it was tested on different sets of initial infected nodes. For example the node 2 is a quiet famous influencer in the platform and we can see that if we start from that node and its neighbors (that are in total the 9.27% of the network, we can reach the 50% of the graph after 25 iterations. After the 25

iterations we reach a stationary state as can be seen in the following graph.



The interesting thing is that keeping the same threshold configuration, we can reach all the network if we decide to add as initial infected set another hub node and its neighbors. This second node is not randomly chosen, but belong to the second big community identify by the Label Propagation algorithm. This means that the first hub chosen is still node 2, which belong to the biggest community, and the second hub is node 27 that belong to the second biggest community. In this way we have a initial set of infected nodes equal to the 10.41% of the network, with this nodes we can reach all the others with very few iterations as can be seen in the graph below. The difference in the fraction of node selected before and now is small, but the results are very different.



7 OPEN QUESTION

For the open question part we tried to answer to:

- **How should we promote a book to reach the majority of users?**
- **What is the probability that a user will save a book after its friends decision to save it?**
- **To which influencer should a publishing house send a book to promote it?**

7.1 SIWR analysis

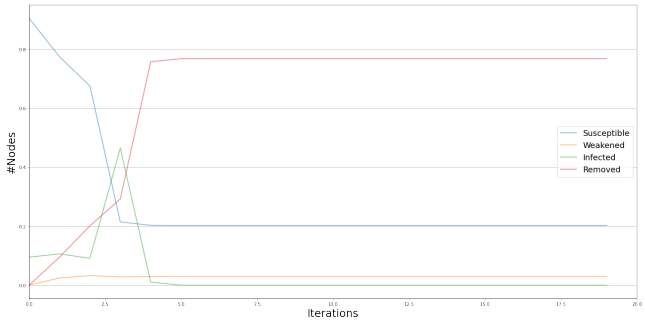
To better model the same ideas explained in the Epidemics Analysis section, we decided to implement another epidemic model called SIWR. In this model, a node is allowed to change its status from Susceptible (S) to Weakened (W) or Infected (I), then to Removed (R). These status represent the following ideas in our context:

- **Susceptible (S)**: set of nodes that has never seen the new book;
- **Weakened (W)**: set of nodes that has seen the new book a few time;
- **Infected (I)**: set of nodes that has seen the new book often;
- **Removed (R)**: set of nodes that has added the book to the library.

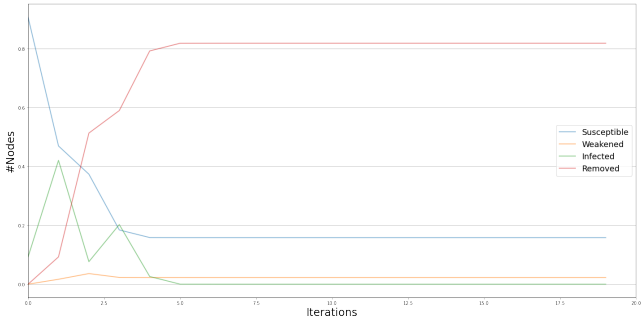
For each of this states are established different thresholds that define the passage from one status to another.

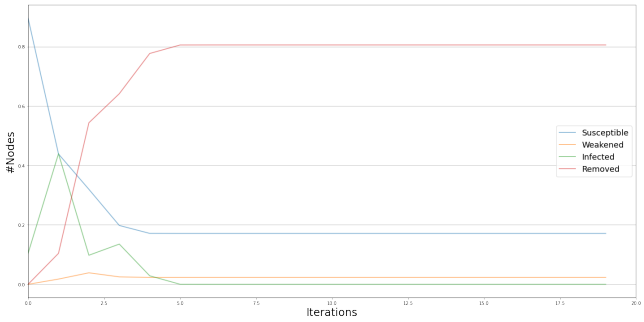
- From Susceptible to Infected: 0.002
- From Susceptible to Weakened: 0.03
- From Weakened to Infected: 0.04

The process of the algorithm is the following:
At time t a node in the state I is selected randomly and the states of all neighbors are checked one by one. If the state of a neighbor is S then this state changes either i) to I with probability kappa or ii) to W with probability mu. If the state of a neighbor is W then the state W changes to I with probability nu. We repeat the above process for all nodes in state I and then changes to R for each associated node.
The following graph show the performance of the algorithm applied on the random selection of 9.5% of the network nodes.



The model is then tested on the same initial infected node set of the Threshold model. The first graph represents the performance where initial set is given by the node 2 (hub) and its neighbors, the second shows the results with the selection of the node 2 and 27 an their neighbor. Unlike the Threshold model, here the performance are quite similar, which means that we can reach the same result even if we include another hub that belongs to a different community identified by the Label Propagation algorithm.



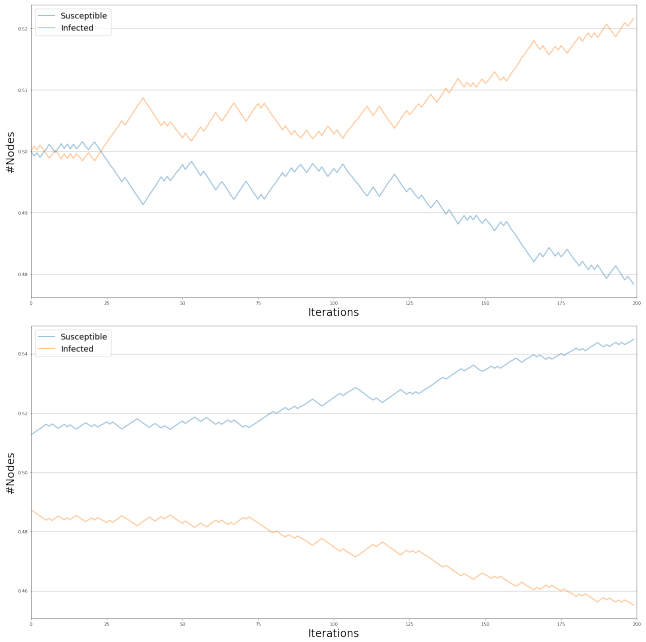


From this results we can see that we can reach more users we the pre-selected set of nodes with respect to the random selection (the blue line representing the susceptible reach lower level in the two last configurations).

7.2 Opinion Dynamics Analysis

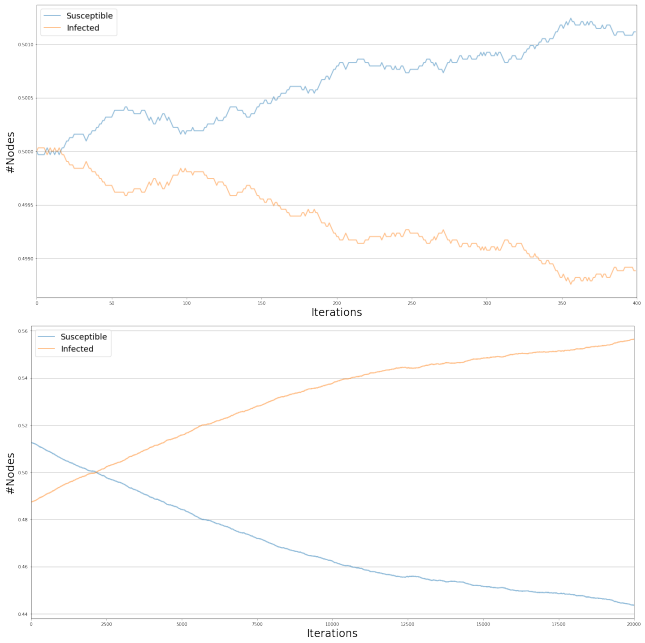
- **How opinions spread in the network?**
- **How a person's opinion changes by their friends?**

7.2.1 Majority rule model. The first algorithm tested is the Majority rule. The q parameter is set equal to 50. The first graph shows the algorithm performance on the random selection of nodes (50% of the graph). The second shows the results by the selection of 9 hubs and their neighbours, reaching in this way the 48% of the network.



In this last graph we can see that the results easily polarized without the exchanges that we see at the beginning of the first graph.

7.2.2 Voter model. The second algorithm is the Voter model. The two following graphs follow the same reasoning for the Majority rule model, the set for the selected nodes are the same.



This last graph follow the same trend that we saw with the epidemic model (especially SI).

7.2.3 Sznajd model. The last algorithm tested is Sznajd model. In this case we can see that in the second graph, in the first 300 iterations there isn't a prevailing opinion, which is very different from what happens by the implementation of the same model by starting from the selection of a random set of nodes.