Network Analysis Of

Team Members

Surya Rajendran (675544714)

Ashwin Narayanan (654325491)

Vigneshwaran Giri Velumani (670664553)

Contents

INTRODUCTION AND CONTEXT:

GitHub has evolved from coding repository to a social network of developers for different types of development projects. There are 37 million users and 100 million repositories. One of interesting of GitHub apart from code repository is the follow developer and creators across the globe. This feature adds the network nature and structure for the GitHub.

OBJECTIVE

Objective of the project is to study the networks and collaboration in GitHub for different types of User. Upon a simple search we can find a variety of technologies and roles of people involved with GitHub.

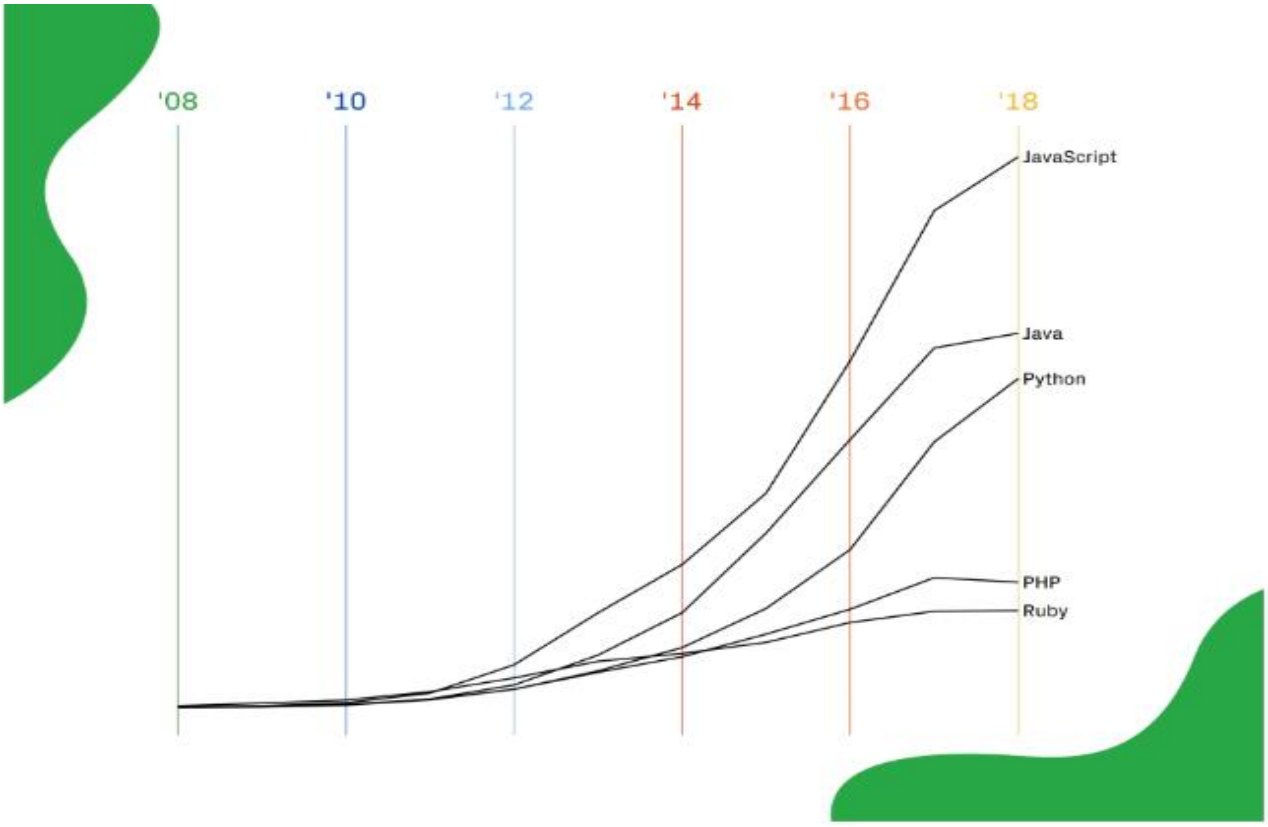


Figure 1

We see that the JavaScript and python used by web developers and machine learning developers respectively to be the highest. This conveys a subtle information that GitHub is predominantly used by Web developers and Machine learning Developers.

Our project goal is to study Web and Machine learning developer network individually and combined.

RESEARCH QUESTION

By analysing different networks, we aim to understand and answer following question.

(1) Characteristics of Machine learning and Web developer Network.
(2) Most popular Developer in both networks.
(3) Collaboration among the developers in the networks.
(4) Understand Why and how collaboration can be improved.

DATA DESCRIPTION

Data was obtained from Stanford SNAP website [1].

Dataset statistics	
Directed	No.
Node features	Yes.
Edge features	No.
Node labels	Yes. Binary-labeled.
Temporal	No.
Nodes	37,700
Edges	289,003
Density	0.001
Transitvity	0.013

Figure 2

- The data represents a large social network of GitHub developers which was obtained from a public API in June 2019.
- Nodes represent developers who have followed at least 10 repositories and Edges are the mutual followers among them.
- Nodes information is collected based on email address, employer, location and phone number.
- Along with the network data there are information available on the role of the developers flagged as follows

  0 – Web developer

1 – Machine learning Developers

id	name	ml_target
0	Eiryyy	0
1	shawflying	0
2	JpMCarrilho	1
3	SuhwanCha	0
4	sunilangadi2	1
5	j6montoya	0
6	sfate	0
7	amituuush	0
8	mauroherlei	0

ml_target is a flag representing information on the role of developer.

Figure 3

DATA PREPARATION

The Data includes the edges between nodes with no information on the role of developers. Below mentioned are the screenshots of two different datasets containing different useful information.

id_1	id_2
0	23977
1	34526
1	2370
1	14683
1	29982
1	21142
1	20363
1	23830
1	34035
6067	19720
6067	20183

id	name	ml_target
0	Eiryyy	0
1	shawflying	0
2	JpMCarrilho	1
3	SuhwanCha	0
4	sunilangadi2	1
5	j6montoya	0
6	sfate	0
7	amituuush	0
8	mauroherlei	0

Figure 4 and 5

Using joins in R we extract the role information from the target dataset and added it to the edges data. After which the data was separated into three different networks

Below mentioned flowcharts explain the process involved in preparing the data for different networks

DATA PREPARATION PROCESS CHART

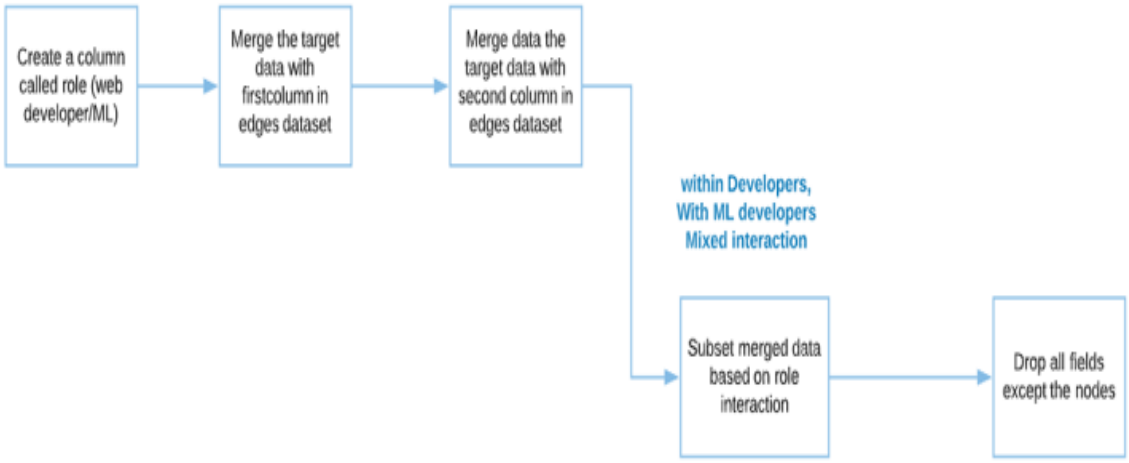


Figure 6

THREE TYPES OF NETWORK

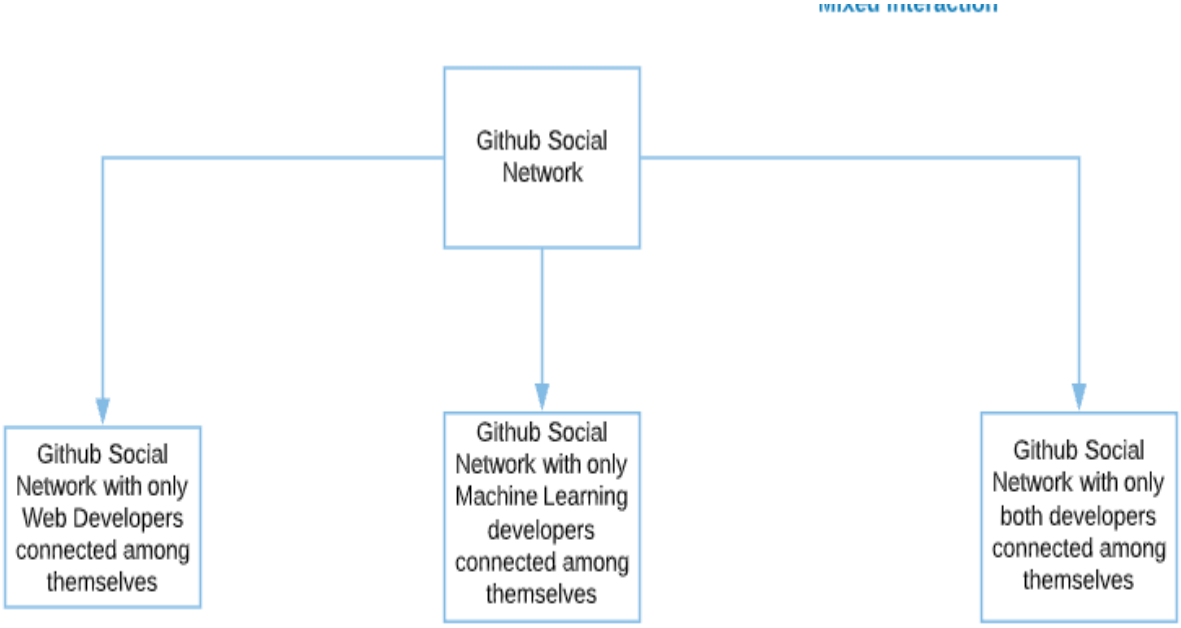


Figure 7

NODE DATA:

From the edge information for each network, distinct number of nodes where extracted for three different networks.

WEB DEVELOPER NETWORK

As with higher number of web application available in market we also see that the web developer network to be the biggest among all the networks.

Using I-graph library in R, its centrality was studied. Below table represents the summary of the centrality of the web developer network.

- The network is simple graph without any loops or edges connecting same nodes

Measures	Value
Nodes	27676
Edges	224623
Strength	16.23233
Degree	16.23233
Closeness	0.03839923
Node Betweenness	28937.97
Edge Betweenness	14468.5
Transitivity	0.01410486
Reciprocity	1
Embeddedness	0.2963884
Diameter	9
Average path length	3.094751

Figure 8

INFERENCE

SNO	CENTRALITY MEASURE	COMMENT
1	Nodes and Edges	Has the highest number of edges and nodes among all the roles in GITHUB
2	Strength and Degree	On an average a Node has 16 connection. 31890 had above 8000 connections.
3	Reciprocity	As the network undirected it has the reciprocity of 1
4	Embeddedness	We see that the Embeddedness is low. There are many structural holes. So, there are many local bridges connecting locally connected networks
5	Closeness	We see the closeness to be very less. It is less than 0.1 It signifies that the network is not closely connected. The distance of the nodes from centre is high
6	Betweenness	Betweenness value supports the low value of the closeness. We see that are lots of nodes in between and the entire network can be grouped into different communities
7	Diameter and Average path	Diameter and average path length value is really high that we see the people are connected not very closely. Lot of people don't follow a lot of people. Also, there are less mutual followers among them.

Table 1

VISUALIZATION

This section focuses on Visualization networks in detail. For community detection walk trap and label propagation was used. Fruchterman Reingold algorithm [3] was used for network layout

CHALLENGE

As the network size is big, challenge was to visualise the network to communicate the inference clearly. To improve on this the network size is reduced by following methods.

- Random sampling of Nodes
- Choosing a subset of Nodes

Rather than choosing random set of nodes, a subset of node was chosen, and a random set of edges where considered.

METHOD USED

Pointing back to hypothesis, the intent of the project is to reason the problems with collaboration for this reason, nodes with higher degree are not visualised. By visualising [2] nodes with lesser degree, we can understand why and how less connected nodes are distributed.

Nodes with Degree 10

Network of nodes with degree less than or equal to 10 [2].

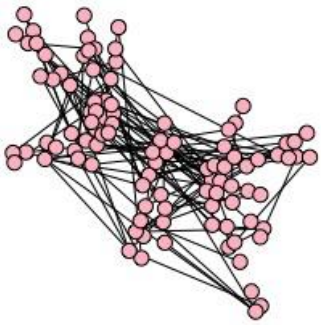


Figure 9

Community Detection

Using walk trap 14 communities where identified [4]

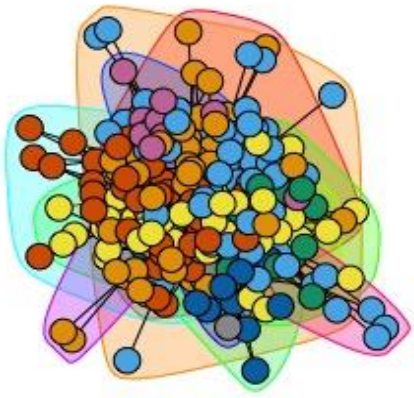


Figure 10

Community Detection for Node with highest Connection [4]

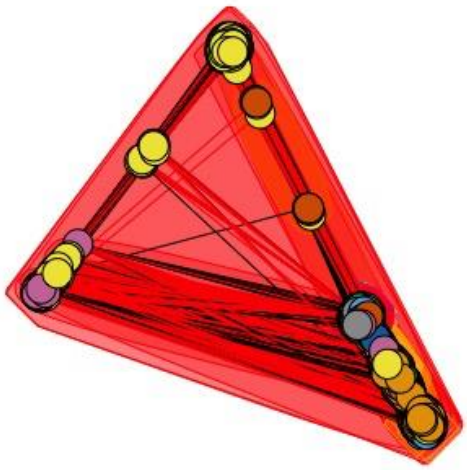


Figure 11

Nodes with Degree less than 4

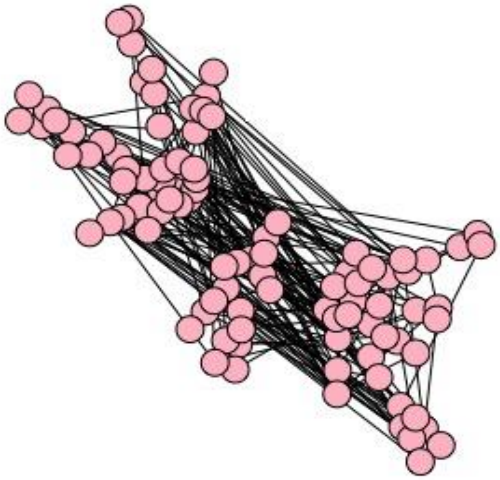


Figure 12

Network for nodes with degree 120

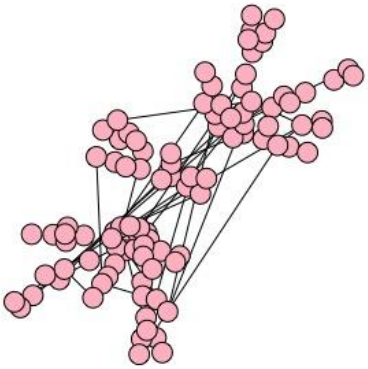


Figure 13

Community detection for Nodes with degree 120 [5].

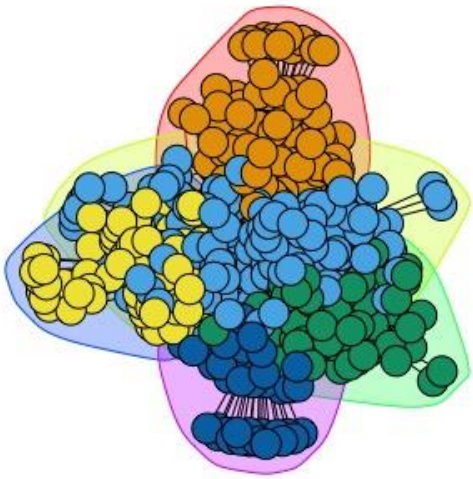


Figure 14

INFERENCE

By visualising network with degree 5, 10, 120 we see that their structure closely resembles each other. We relate this phenomenon with closeness value for entire network which was really low 0.03. Which signifies that the developer network has significant features.

- All nodes follow most popular nodes shown by a single community for 31890 nodes.
- There are nodes which are not closely connected to centre of network but exist as a distant community.

MACHINE LEARNING NETWORK

Among all the three networks, Machine Learning Developers Network is the smallest among all the networks. The following table gives the list of all centrality measures of this network. Just like the Web Developer Network, this network is also a simple, undirected, unconnected network without any loops or edges connecting the same nodes.

Measure	Value
Nodes	7431
Edges	19684
Strength	5.297806
Degree	5.297806
Closeness	0.00271144
Node Betweenness	11885.84
Edge Betweenness	5761.27
Transitivity	0.03381751
Reciprocity	1
Embeddedness	0.5375045
Diameter	13
Average path length	4.52152

Figure 15

INFERENCE

SNO	CENTRALITY MEASURE	COMMENT
1	Nodes and Edges	Has the lowest number of edges and nodes among all the roles in our Github Community
2	Strength and Degree	On an average a Node has 5 connection. The highest degree that any node had was 43. and 72% of the nodes had less than or equal to 4 neighbours (degree<=4)
3	Reciprocity	As the network undirected it has the reciprocity of 1
4	Embeddedness	We see that the Embeddedness is highest in this network. So, it indicates that the remaining 30% of the nodes consists of the structural holes of the network which serves

		as local bridges connecting other components in the network.
5	Closeness	We see the closeness to be very less than the web developer network. Its value 0.002 signifies that the network is very closely connected. So, any information passed by the developers in this network will quickly reach to their neighbours.
6	Betweenness	Betweenness value supports the low value of the closeness. Its value is the lowest compared to others, and its value is less than half of web developer network. This signifies that there are lesser communities in this network compared to others.
7	Diameter and Average path	Diameter and average path length value is higher than the web dev network which indicates that people are connected less close than the web dev network. Lot of people don't follow a lot of people. Also, there are less mutual followers among them.
8	Transitivity	This network has highest transitivity than the other network. This makes sense because the network is one half of the size of the web dev and hence its transitivity is three times of the other. Higher transitivity indicates that more ML developers are mutually connected and the knowledge they share with themselves have more in common.

Table 2

VISUALIZATION

This section focuses on visualisation networks in detail. The following concepts are covered in visualization:

- Scatterplots of various network measures like Betweenness Vs Clustering, Degree Vs Clustering, etc.
- Log-Log Degree distribution plot
- All the 3-community detection FastGreedy, Walktrap and label propagation algorithms which are compatible with undirected networks have been plotted
- 5 considerably big communities in each algorithm were visualized

CHALLENGE

Despite this ML Network was the smallest, it was challenging to visualize the top biggest communities together since the biggest communities classified by all 3 algorithms had networks with at least 1500 – 2000 vertices.

So, we have tried to visualize 5 big communities which are 5th to 10th biggest communities each having a network with less than 150 vertices.

RELATION BETWEEN DIFFERENT NETWORK MEASURES

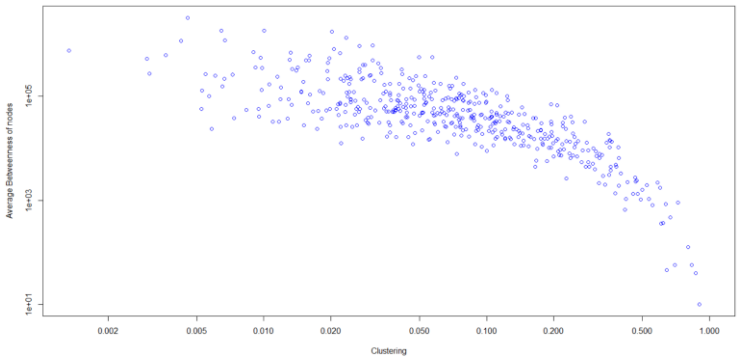


Figure 16

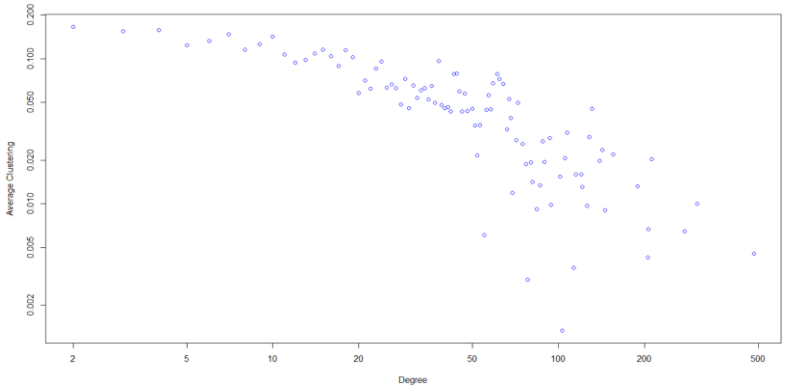


Figure 17

Betweenness Vs Clustering:

It can be understood that nodes which have fewer clustering values have the highest betweenness because those people who are not tightly connected to any cluster serve as the structural hole and act as the main node for information flow through them. It can be seen from the above plot that there are some 6 nodes in the extreme left which are the structural holes of the ML network. Moreover, the plot also proves this inverse relation between the average betweenness, and the clustering coefficient values that nodes which have fewer clustering coefficients have higher average betweenness and the relationship is linear with a negative slope.

Average Clustering Vs Degree:

We can interpret from this graph that the 7 nodes in the extreme right have highest degree but less average clustering. These are the nodes which act as the structural holes in this graph. This plot also proves the inverse relation between the clustering and the degree since the nodes which have highest clustering coefficient are so embedded in the network that they are left out from the rest of the community and tied to only one community. The relation is linear with a negative slope.

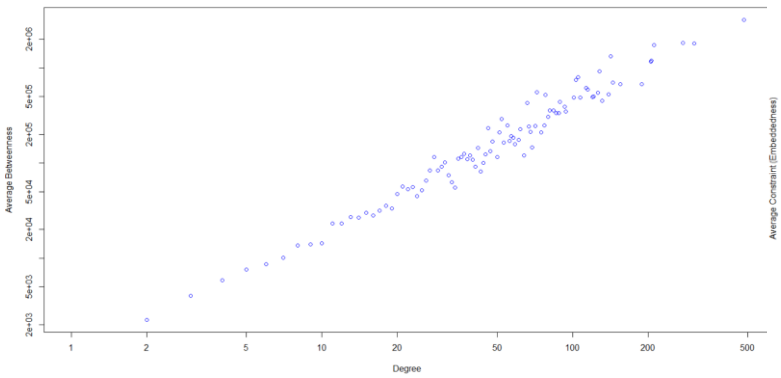


Figure 18

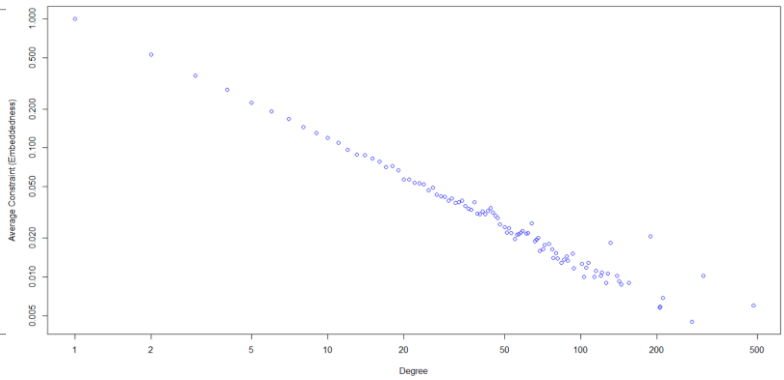


Figure 19

Average Betweenness Vs Degree:

From the graph on the left, we can see that the rightmost top 5 nodes have the highest betweenness as well as the highest degree. So, this indicates that these nodes are not only connected to most of the neighbours but also serve as the important people in the middle for information exchange in the network. So, if we remove them, sharing of information will be hindered. Also, this plot proves the direct relation between the Degree and Betweenness with a positive slope.

Embeddedness Vs Degree:

From the graph on the right, we can see that the rightmost bottom 4 nodes have the highest degree and lowest embeddedness, indicating they are the biggest structural holes in the network who are responsible for diffusing novel ideas in several communities. This plot also proves the indirect relation between embeddedness and Degree with a negative slope.

Log-Log Degree distribution

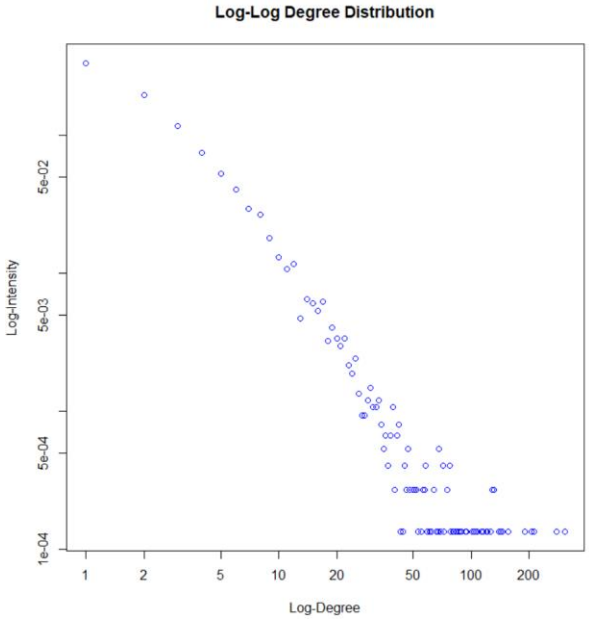


Figure 20

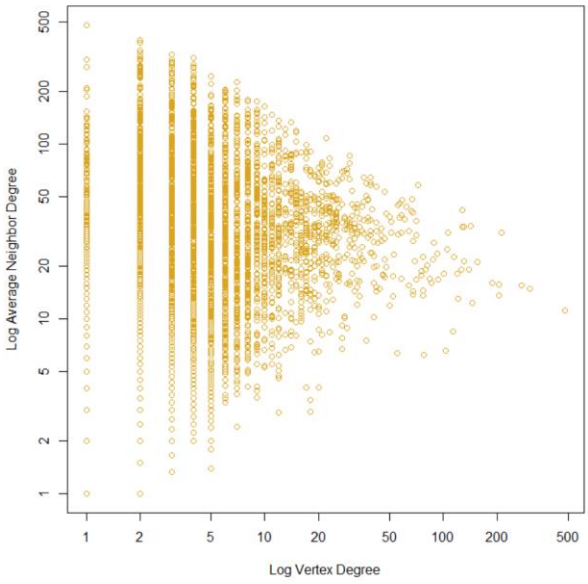


Figure 21

From Fig 20. it is evident that there is a nearly a linear decay in the log intensity as a function of log-degree. It also signifies that the distribution is scale free and follows a power law degree distribution. With that being said, the “Rich Get Richer” phenomenon [6] looks prominent in

this network, which is ML developers having lesser connections tend to connect with the ML developers having highest number of connections.

Next, we tried to plot Avg. Neighbour Degree Vs Log Vertex Degree in Fig.21, which showed that lower degree nodes tend to get attached with other nodes which have both lower and higher degrees. But we can also see that the nodes with higher degree tend to form connections with others only if they have a particular range of node in their network. So, we can assume that there is some preferential attachment happening in this ML network. This can be relatable because when ML developers gain some years of experience, they tend to get connected with people having more expertise than them to learn from them, as well as connect with less knowledge people to teach or help them.

COMMUNITY DETECTION:

We tried 3 different community detection algorithms, the Fast Greedy, Walktrap and the label propagation algorithms. Let's look at one by one.

Fast Greedy Algorithm

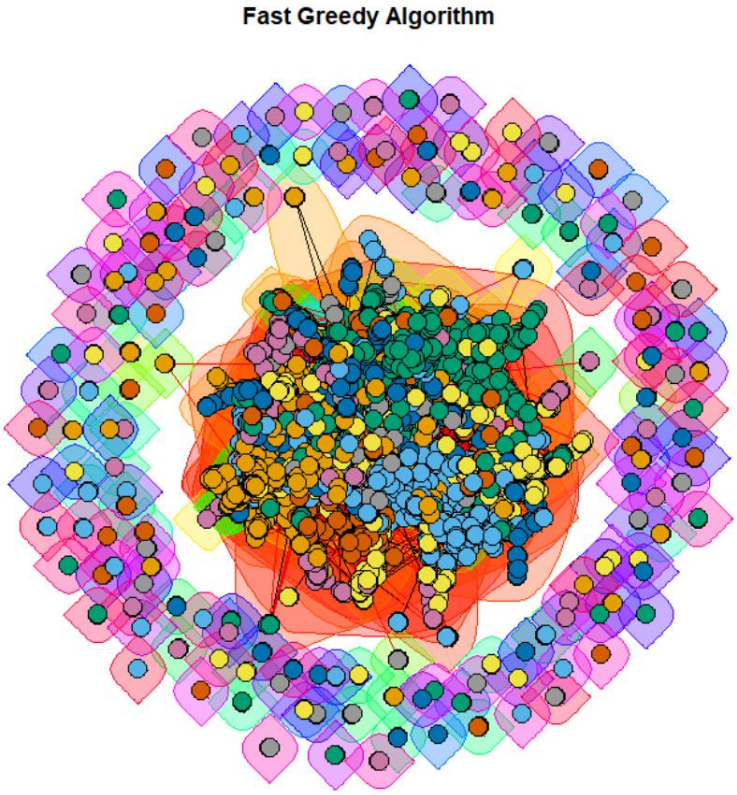


Figure.22

The Fast-Greedy Algorithm detected 305 communities in the ML Network. The plot of the whole network has been shown above in Figure.

Fast Greedy – Community Key Properties:

Group	# Vertices	Vertex (With highest degree)	Degree	Name
1	1537	14954	474	rasbt
5	1487	29023	119	nd1511
3	1092	16631	95	sebastianruder
2	677	17850	35	jogendra
4	402	23589	76	antirez
14	151	910	34	holdenk
8	140	18298	12	hamelsmu
6	110	25692	10	TianxiaoHu
9	92	4184	11	nemaniarjun
7	89	30481	12	mari-linhares

Table 3

The top 5 community groups highlighted in gold are the biggest ones. Since the #Vertices are high, the next 5 communities have been visualized in the plot below. The developer **rasbt** has the highest number of neighbours 472. The top 5 groups constitute 70% of the total network as per this algorithm.

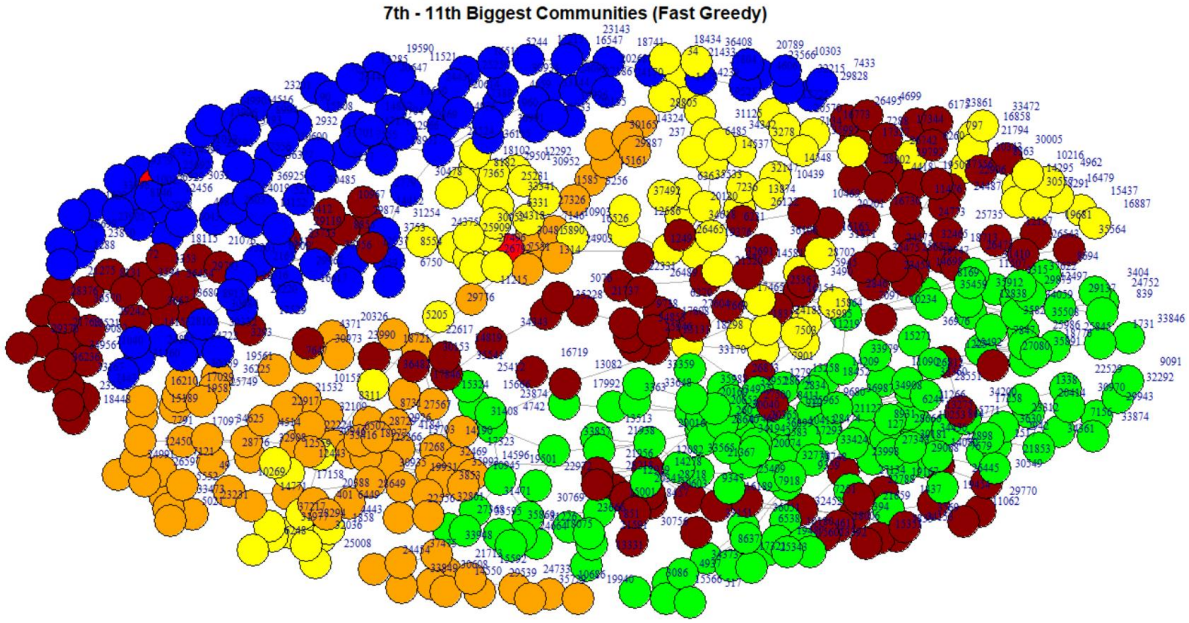


Figure.23

The above visualization consists of **6th-10th biggest communities** classified by the Fast-Greedy algorithm. We have tried visualizing the node with the highest degree of each community in red. We can see that the communities in **yellow** and **brown** are widely diffused among other communities, indicating that their group has a greater number of structural holes in their community. The communities in **blue and green** seems to be tightly knit as compared to rest of the other 3.

Walktrap Algorithm

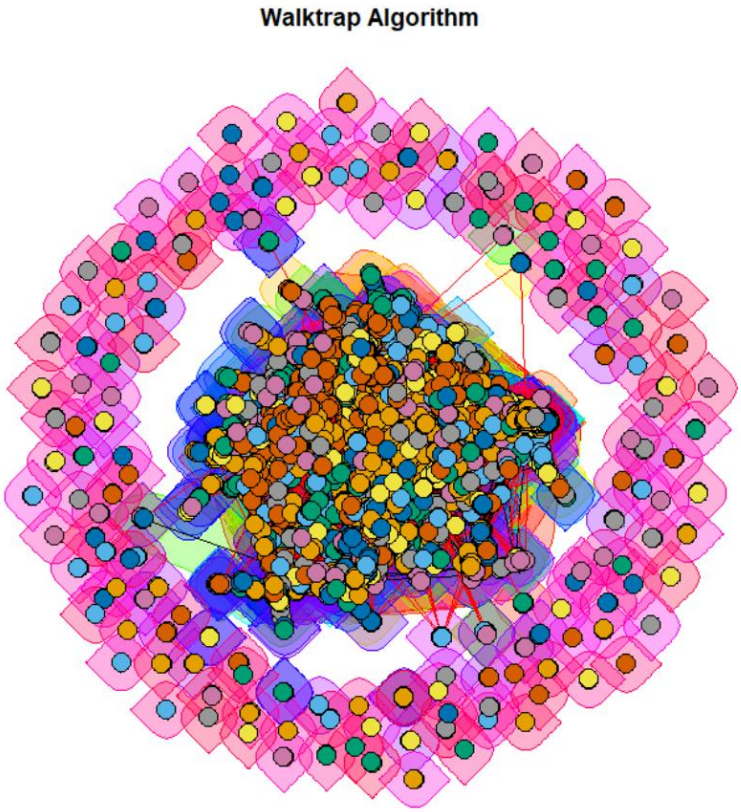


Figure.24

The Walktrap Algorithm detected 1056 communities in the ML Network. The plot of the whole network has been shown above in Figure.

Walktrap – Community Key Properties:

Group	# Vertices	Vertex (With highest degree)	Degree	Name
9	1705	14954	335	rasbt
6	1245	16631	99	sebastianruder
20	249	34603	64	dgrtwo
28	173	17792	55	hunkim
30	98	3757	26	nishnik
21	88	26918	26	chapmanb
40	77	23589	45	antirez
115	72	32566	26	hitcm
13	66	2701	13	hpdang

Table 4

The top 4 community groups highlighted in gold are the biggest ones. Since the #Vertices are high, the next 5 communities have been visualized in the plot below. The developer **rasbt** has once again been detected to be having the highest number of neighbours 335. The top 4 groups constitute 45% of the total network as per this algorithm.

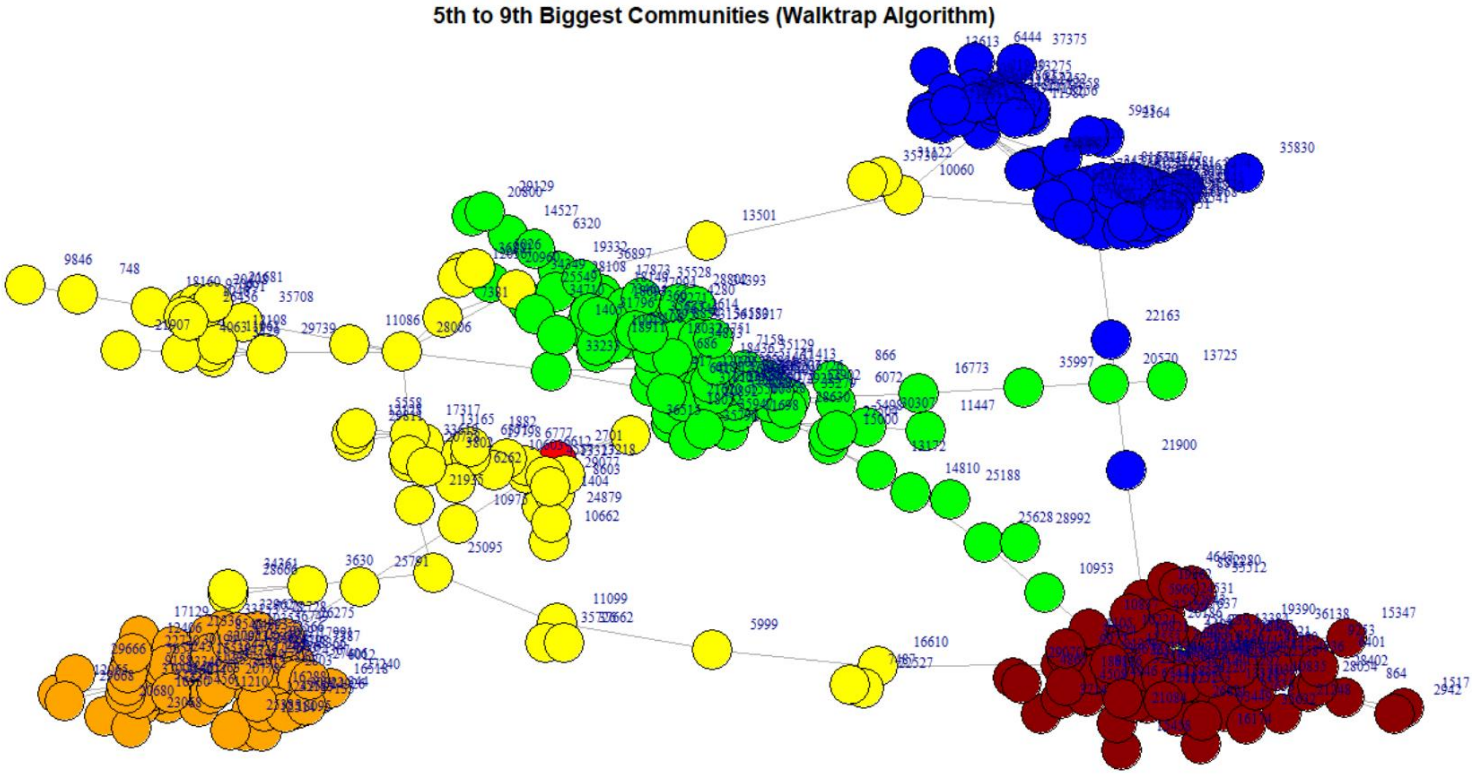


Figure.25

The above visualization consists of **5th-9th biggest communities** classified by the Walktrap algorithm. We have tried visualizing the node with the highest degree of each community in red. We can see that the community in **yellow** has widely diffused among other communities, indicating that their group has a greater number of structural holes in their community. Specific nodes that are worth to mention to serve as structural nodes are:

Structural Hole Node	Name of Developer
13501	SheikhZayed
21900	SimonLarsen
10953	myaooo
3630	Rashwan
2701	janw
28666	JasonChu1313

Since, the yellow structural node indicates that the yellow community dominates the structural node group. Another interesting thing in this plot is the brown community is so tightly knit that it does not have any structural hole node.

Label Propagation Algorithm

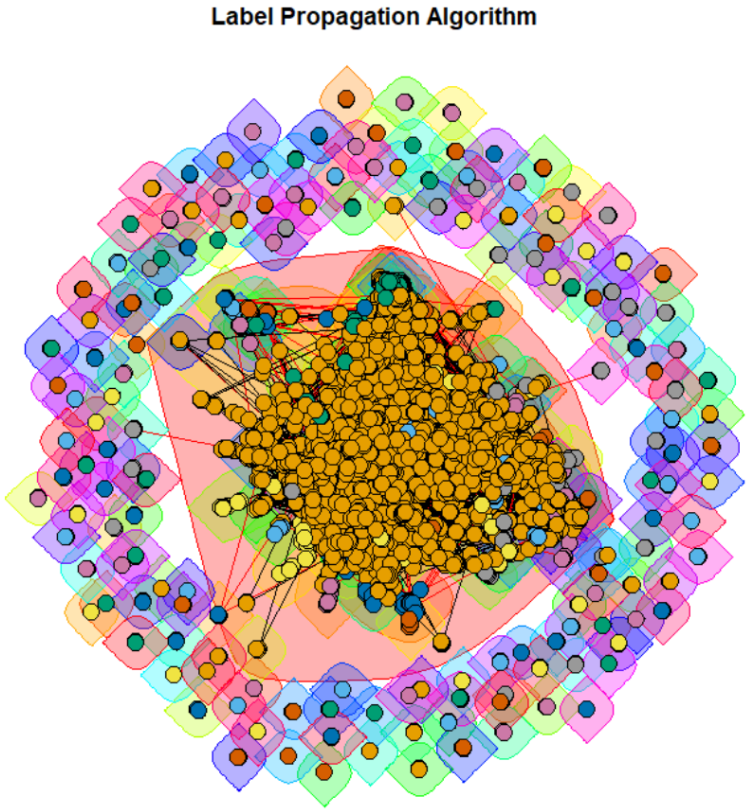


Figure.26

The Label Propagation Algorithm detected 251 communities in the ML Network. The plot of the whole network has been shown above in Figure.

Label Propagation – Community Key Properties:

Group	# Vertices	Vertex (With highest degree)	Degree	Name
1	6325	14954	474	rasbt
3	230	17128	51	IndrajeetPatil
10	53	26918	25	chapmanb
13	27	28429	8	BaronZ88
83	22	18663	16	lfarah
14	20	6875	34	jobovy
25	18	35403	13	csoni111

Table 5

The top 2 community groups highlighted in gold are the biggest ones. Since the #Vertices are high, the next 5 communities have been visualized in the plot below. The developer **rasbt** has once again been detected to be having the highest number of neighbours 474. The top 2 groups constitute 88% of the total network as per this algorithm.

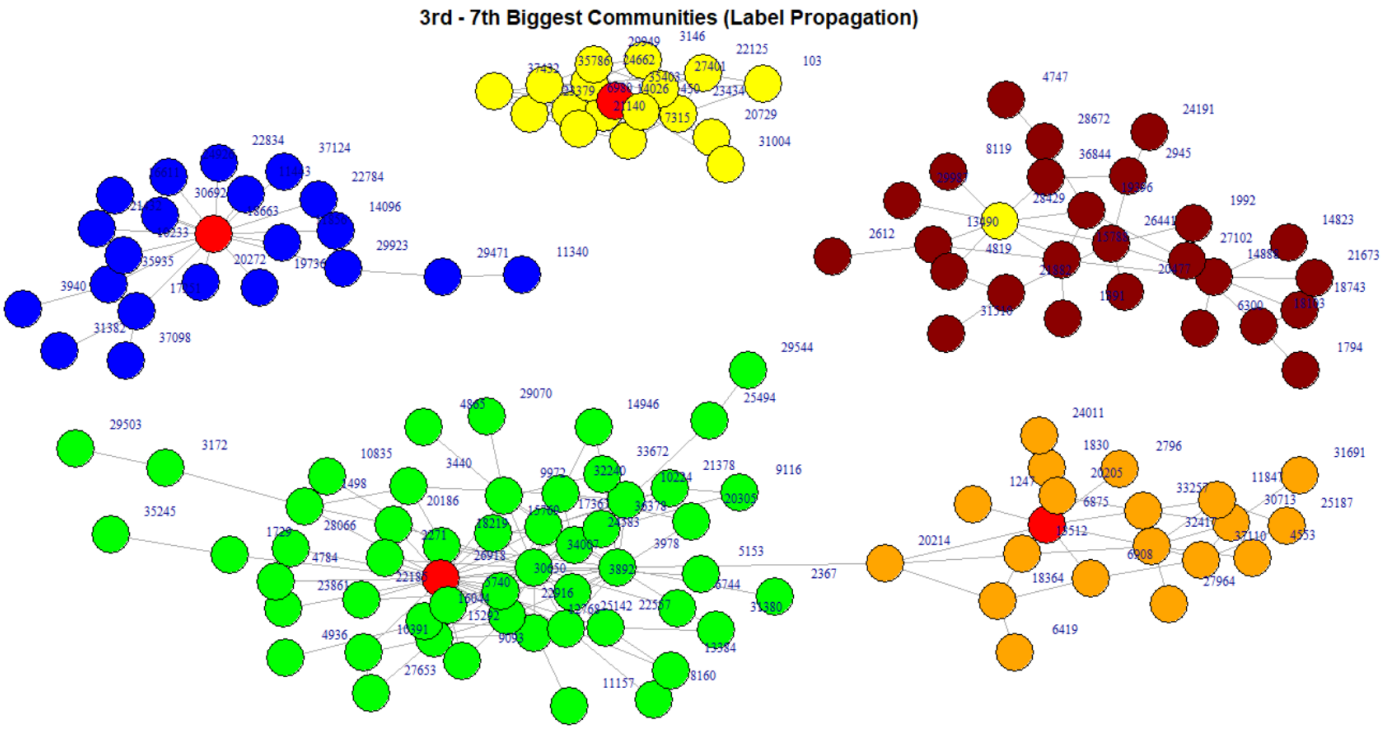


Figure.27

The above visualization consists of **3rd-7th biggest communities** classified by the Label Propagation algorithm. We have tried visualizing the node with the highest degree of each community in red. For the brown community the highest degree node has been visualized in yellow. We can see that the community in **orange** has only one structural hole – the node: **20214 – bgriffen** which happens to be the only structural node in this entire group of 5 communities.

The interesting thing in this plot is except that one orange structural hole node all the communities are so tightly knit that they are so isolated with one another.

MIXED DEVELOPER NETWORK:

Mixed developer network consists of nodes of both the web developers and the machine learning developers and the edges are the mutual connections between them.

Using I-graph library in R, its centrality measures of the Mixed network of web and machine learning developers was studied. Below table represents the summary of the centrality of the Mixed developer network.

Measure	Value
Nodes	18635
Edges	44696
Strength	4.8
Degree	4.8
Closeness	0.001512422
Node Betweenness	30908.13
Edge Betweenness	16510.75
Transitivity	0
Reciprocity	1
Embeddedness	0.57
Diameter	13
Average path length	4.55

Figure 28

INFERENCE FROM CENTRALITY MEASURES

SNO	CENTRALITY MEASURE	COMMENT
1	Nodes and Edges	The number of nodes and edges are comparatively higher which signifies the number of connections between the web and machine learning developer community.
2	Strength and Degree	The mean number of connections among the nodes is 5 and Node 27803 has highest degree
3	Reciprocity	Since there is mutual connection among the nodes the reciprocity is 1.
4	Embeddedness	Embeddedness value is found to be quiet high which signifies that there are many clusters in the mixed network.
5	Closeness	We see the closeness to be very less. It is less than 0.1 It signifies that the network is not closely connected. The network spread is huge as the distance is higher between the nodes.

6	Betweenness	Betweenness value suggests that are lots of nodes in between and the entire network can be grouped into various types community among them.
7	Diameter and Average Path Length	Diameter and average path length value is really high that says that there not many mutual connections among the developers in the mixed network which describes the low closeness value of the network.

Table 6

VISUALIZATION:

This section focuses on visualization networks in detail.

CHALLENGE:

As the network size is big, challenge was to visualize the network to communicate the inference clearly. To improve on this the network size is reduced by following methods.

- Random sampling of Nodes
- Choosing a subset of Nodes

Rather than choosing random set of nodes, a subset of node was chosen and a random set of edges where considered.

METHOD USED:

Pointing back to hypothesis, the intent of the project is to reason the problems with collaboration for this reason, nodes with higher degree are not visualized. By visualizing nodes with lesser degree, we can understand why and how less connected nodes are distributed.

Network graph of Nodes with Degree 5 [4]:

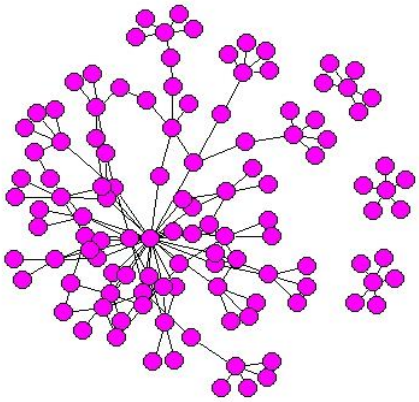


Figure 29

Community Detection [5]:

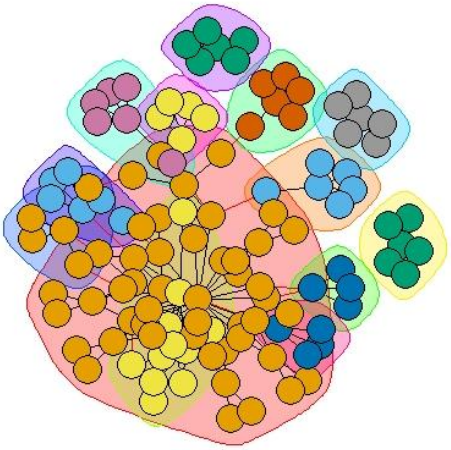


Figure 30

Network graph of Nodes with Degree 10 [4]:

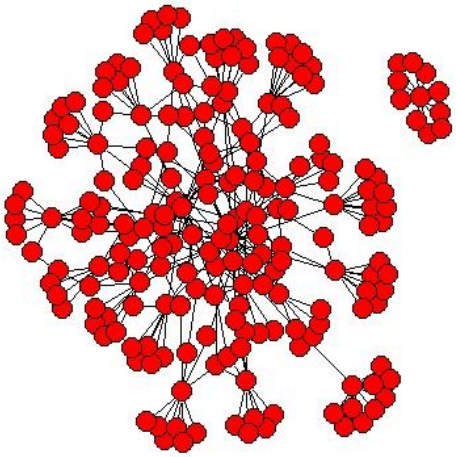


Figure 31

Community Detection [5]:

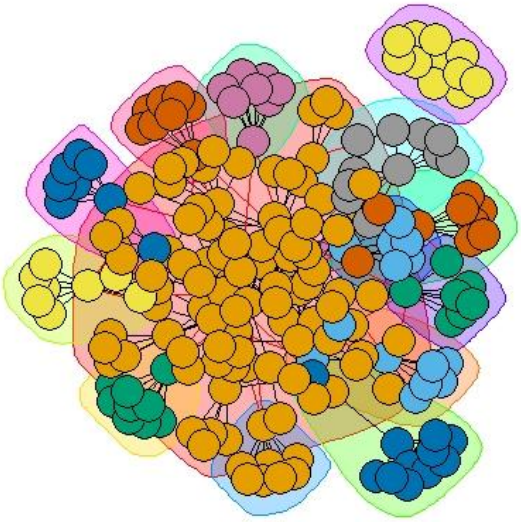


Figure 32

Network graph of Nodes with Degree 50:

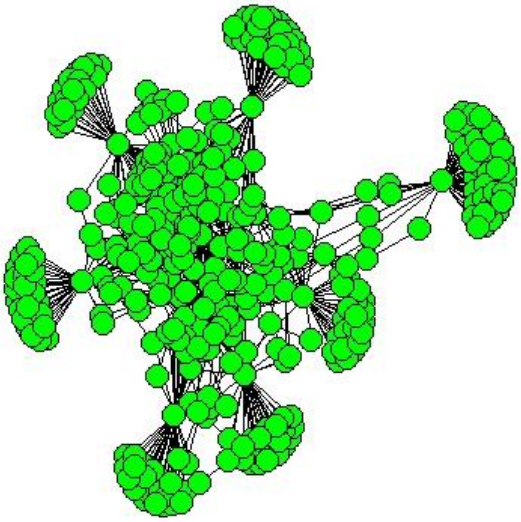


Figure 33

Community Detection [5]:

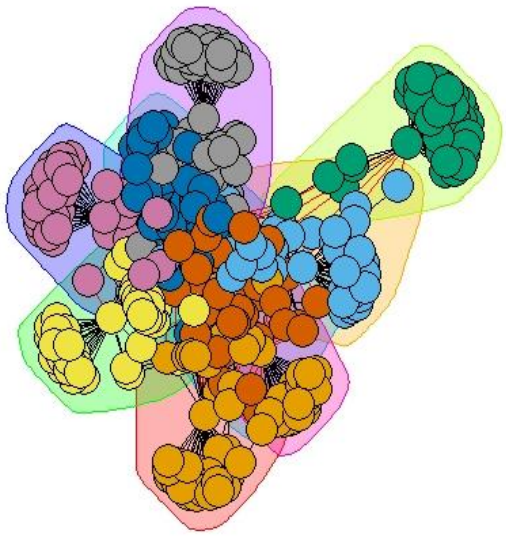


Figure 34

INFERENCE:

- By visualizing network with degree 5, 10, 120 we see that their structure closely resembles each other.
- We could see from the network and the community detection plots that there are various communities present in a single network.
- All of the community detection plots show that there is a large community many small communities within a single network.
- The small communities have interactions only among themselves and there are some local bridges in the networks which connects the smaller communities in the network with the single large community present in the network.
- The large community present in the network is seen as the hub of the network which has many connections of nodes.

CONCLUSION AND SUMMARY

Centrality Measures summary for three networks

Measure	Web Developer Network	ML Developer Network	Mixed Network
Nodes	27676	7431	18635
Edges	224623	19684	44696
Strength	16.23233	5.297806	4.8
Degree	16.23233	5.297806	4.8
Closeness	0.03839923	0.00271144	0.001512422
Node Betweenness	28937.97	11885.84	30908.13
Edge Betweenness	14468.5	5761.27	16510.75
Transitivity	0.01410486	0.03381751	0
Reciprocity	1	1	1
Embeddedness	0.2963884	0.5375045	0.57
Diameter	9	13	13
Average path length	3.094751	4.52152	4.55

Figure 35

- **Nodes, Edges –** From above table we see that Web developer network has the greatest number of connections and it is interesting to see the ML and Web developer's cross interaction is 18635 which is greater than ML developer network. Which shows that ML developers collaborate more than Web developers
- **Strength and Degree - Though** Mixed network has more edges but its degree is less than the degree within the ML Network. Which shows still a lot of developers in mixed network follow very less number of other developers in the same.
- **Closeness –** Though Web developer network is the largest we see it has better closeness then other network, but as overall network the closeness is a low value.
- **Reciprocity –** Since all networks are undirected the reciprocity is 1.
- **Transitivity –** Opportunity for triadic closure is very less in all networks. In case of mixed network, we see it is zero. Which shows there are no opportunity for triadic closure.
- **Embeddedness –** Mixed Network has the highest Embeddedness which is easily relatable to a scenario that is a developer good at machine learning development is also good at web development. So, the node could at centre for all the Web and ML developers to follow.
- **Diameter and Average Length.** Though the web development network is large we see the path length to be less. Which shows the nodes are closely connected than other network. Information reachability in Web development is faster than other two networks.

Comparing popular nodes in networks

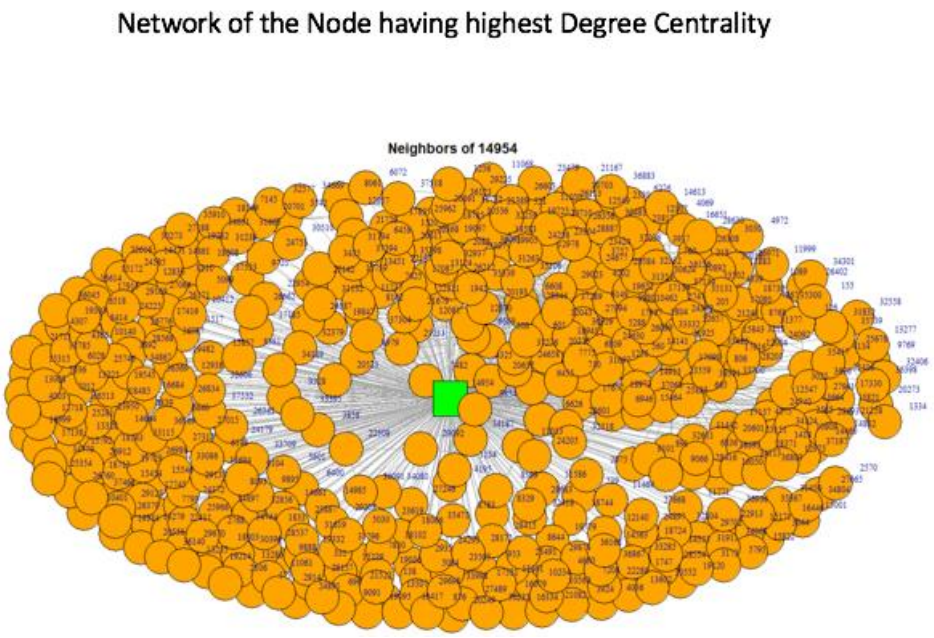
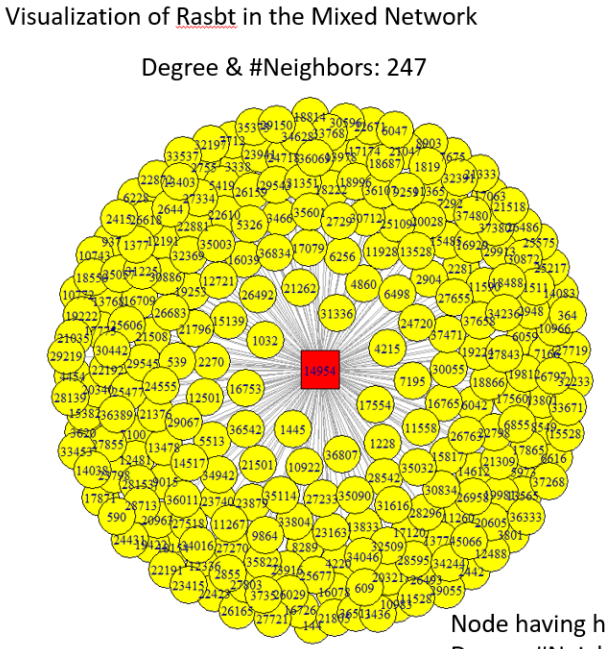


Figure 36



Node having highest degree: nfultz
Degree #Neighbors: 1620
He/she is a developer

Figure 37

NAME OF NETWORK	DEGREE
ML DEVELOPER	**482**
MIXED DEVELOPER	**247**

Table 7

Node 14954 (Rasbt) is a Machine learning developer with 482 edges but the same person has high number of connections in mixed network too. This shows the developer contributes to both the network well.

Structure of Web and Mixed Development Network

By only analysing node of less degree in Mixed and ML network we can learn two important insights

- **Rich get Richer phenomena [6]** – All the nodes follow the most popular developers.
- **Homophily** – We see that nodes with less degree forms group of isolated communities connected by local bridge to the core. This affects the collaboration of the network as people only contribute in the close-knit community except for the node which acts as the local bridge.

REFERENCES

[1] http://snap.stanford.edu/
[2] https://www.r-graph-gallery.com/247-network-chart-layouts.html
[3] https://github.com/gephi/gephi/wiki/Fruchterman-Reingold
[4] https://kateto.net/netscix2016.html
[5] https://rpubs.com/shestakoff/sna_lab5
[6] Adamic, L. A., Zhang, J., Bakshy, E., & Ackerman, M. S. (2008, April). Knowledge sharing and yahoo answers: everyone knows something. In Proceedings of the 17th international conference on World Wide Web (pp. 665-674). ACM.