International Journal of Research in Engineering, Science and Management
Volume 5, Issue 2, February 2022
https://www.ijresm.com | ISSN (Online): 2581-5792

Link Prediction in Social Networking using Machine Learning

Lakshitaa Sehgal[1*], Sannidhi Sri Sai Hanuma[2], Paruchuri Ratan Chowdary[3], Aaditya Singh[4], Vishwas Gowdihalli Mahalingappa[5], Manikanta Ranganath[6]

***Abstract*: Social Networks specifical consciousness on building on social relations among customers who share common pastimes, background, real-lifestyles connections, and so on. People may additionally not need to maximize their social influence. For example, business page owners on Instagram want to influence as many human beings as viable for or their business benefits. However, the network is evolving in time, new customers are joining, adding pals, new connections between antique users, and so forth. Based on the contemporary community we need so one can expect the upcoming adjustments in the network and make tips accordingly. We have a photograph of Facebook's social network at the time (say 't') and based on it, we want to expect the destiny possible links. In this paper, we can be sharing our technique using machine learning to solve this example observe.**

***Keywords*: Link prediction, social networking.**

1. Literature review

The Algorithm of Link Prediction on Social Network, Liyan Dong, Yongli Li, Han Yin, Huang Le, and Mao Rui.

At gift, most link prediction algorithms are based totally at the similarity among two entities. Social network topology records are social network topology records are one of the principal resources to lay out the similarity function between entities. But the existing hyperlink prediction algorithms do now not practice the community topology information sufficiently. For lack of traditional hyperlink prediction algorithms, we advise improved algorithms: CNGF set of rules primarily based on local statistics and KatzGF algorithm based on global statistics network. For the disorder of the stationary of social community, we also provide the link prediction set of rules primarily based on nodes more than one attributes information. Finally, we established these algorithms on the DBLP facts set, and the experimental outcomes display that the overall performance of the stepped forward algorithm is advanced to that of the conventional link prediction algorithm.

Link prediction in social network based totally on local data and attributes of nodes, Yingying Liang, Lan Huang and Zhe Wang.

Link prediction is crucial to each research region and practical package. To make full use of the statistics of the network, we proposed a new approach to predict hyperlinks inside the social community. Firstly, we extracted topological records and attributes of nodes within the social community. Secondly, we integrated them into function vectors. Finally, we used XGB classifier to expect hyperlinks using feature vectors. Through increasing information supply, experiments on a co-authorship network advocate that our approach can enhance the accuracy of link prediction notably.

Link prediction in complicated networks: A nearby naïve Bayes model, Zhen Liu, Qian-Ming Zhang, Linyuan Lü and Tao Zhou.

The not unusual neighbor based technique is straightforward yet powerful to are expecting missing hyperlinks, which count on that nodes are much more likely to be linked if they have extra common pals. In the conventional technique, each not unusual neighbor of two nodes contributes equally to the connection likelihood. In this letter, we argue that unique not unusual neighbors may play exceptional roles and for that reason contributes differently, and propose a nearby naïve Bayes version. Extensive experiments had been carried out on 9 actual networks. Compared with the conventional method, the present approach can offer extra correct predictions.

Research of Dynamic Link Prediction Method Based on Link Importance, Li Yuhua, Xiao Hailing, Li Dongcai, and Li Ruixuan.

Current processes of link prediction based totally on graph topology forget about some social homes(semantics) of entities, even as different methods based totally on type without considering the time thing within the position of technology of hyperlink. Accounting to these troubles cited, a link significance based dynamic hyperlink prediction method is proposed primarily based on the medical studies' co-authorship community. Modification on classical topology characteristics and semantic similarity is applied based totally on metrics named hyperlink importance. Dynamics is taken into account to mirror the time element's influence on the technology of link. We use category technology for final prediction. Based on those proposed strategies, experiments are implemented at the DBLP dataset. The test outcomes display that link significance metric and time element improve the prediction accuracy compared with modern strategies.

Link Prediction within the Pinterest Network, Poorna Kumar, Amelia Lemionet, Viswajith Venugopal.

Link prediction is a traditional problem in networks, of superb sensible relevance – it may suggest to customers what

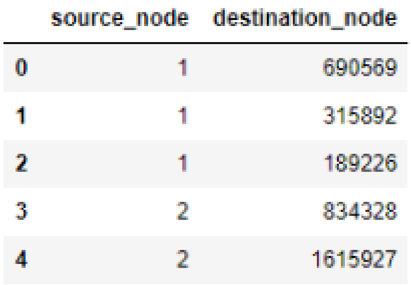
[*] Corresponding author: lakshitaasehgal29@gmail.com

items they should purchase on an e-commerce web page, who they have to comply with within a social network, and many others. In our response paper, we summarize and critique important papers managing this mission, which includes, which talks approximately the way to use community structures to compute link prediction ratings, which formulates hyperlink prediction as supervised gaining knowledge of hassle and, which proposes a novel Supervised Random Walk approach to leverage the strength of both node/aspect attributes and network structure. We critique our readings and brainstorm future studies instructions.

2. Proposed Work

Understanding the Data:

After the data gets downloaded, it reads something like this;



	source_node	destination_node
0	1	690569
1	1	315892
2	1	189226
3	2	834328
4	2	1615927

We check the data for any missing rows/ duplicates. From the data, we can infer that it is a directed graph wherein we're supplied with nodes; a source and a destination node. We tried to visualize the given community the use of the NetworkX python library. NetworkX is a device for developing, manipulating and carrying out examine of the structure of complex networks. Characterstics of the given records:

Number of nodes: 1862220
Number of edges: 9437519
Average in degree: 5.0679
Average out degree: 5.0679

So, the total quantity of particular nodes on this network are 1862220 and the whole quantity of edges inside the community are 9437519. Note that we're provided with edges/hyperlinks and now not nodes.

Therefore, the total wide variety of viable edges/hyperlinks/connections in this community may be 1862220 C 2. Out of those, we're provided with the handiest 9437519. The final 1862220 C 2–9437519 edges do not exist within the community.

Well, the friend recommendation can be defined as a binary type hassle that takes a fixed of capabilities from the customers and maps them to '1' if there exists a hyperlink between a pair and '0' otherwise.

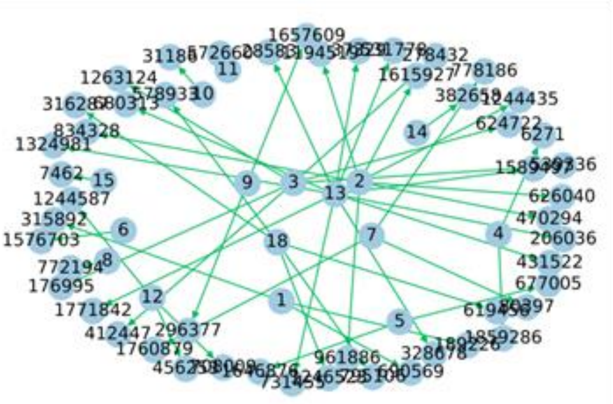
We will create an indicator variable for the hyperlink to be '1' if there is a link among two nodes and 'zero' if there's no link between the 2 nodes. However, in our records, we're furnished with all of the hyperlinks within the given network.

So, the remaining 1862220 C 2–9437519 combinations of nodes do not have any links among them. Although, we can not take this complete ultimate mixture due to the fact this gives us entirely biased facts. So, we can be randomly sampling 9437519 from it to get a piece of balanced information. Therefore, the very last length of the statistics can be 9437519 x 2 = 18875038. So, our final data will have 9437519 rows with indicator variable '1' and 9437519 rows with indicator variable '0'.

3. Data Visualization

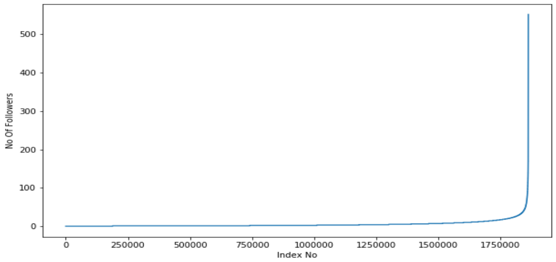
We took a sample of 50 data just to view the network using the networkX tool.

Type: DiGraph
Number of nodes: 66
Number of edges: 50
Average in degree: 0.7576
Average out degree: 0.7576



We can be aware of how all the supply nodes by default are positioned in the direction of the middle of the graph and the destination nodes are positioned outwards by using the networkX library.

Let's have a examine some distributions.

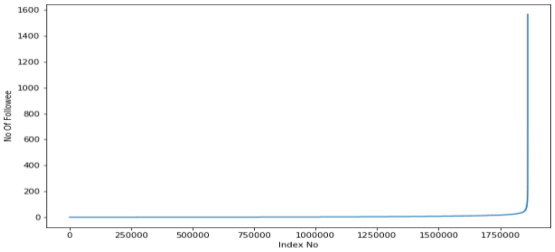


Here's the distribution of range of fans of each node in the training set as well as the range of followees of each node.

Observations:

1. Most of the users in this network have followers in the range of 40 to 50.
2. The maximum number of followers by a user is 52.

Followers' distribution:

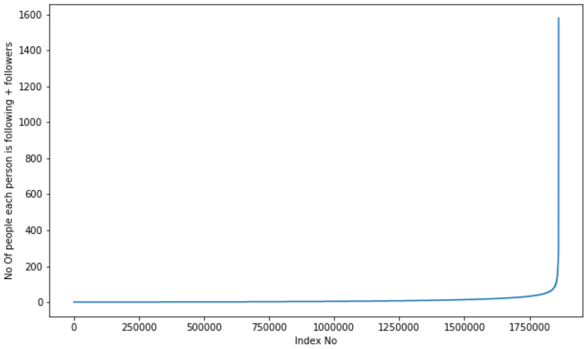


Observations:

1. The number of followees for most users fall in the range of 40–50.
2. Maximum is 1566.

We can notice that 14% of the users in our data have 0 followers and 10% of users have 0 followers.

Now we will find the number of friends the users have (people followed by the user and are followed back in return).



Observations:

1. Most values lie within the variety of 80–a hundred.

We can observe that the above distribution for fans and followers are the followers/followers except for pals. Followers of a person may or won't be followed lower back with the aid of the user. Similarly, the followers of the consumer may also or might not follow back the consumer.

Data Preparation:

Now, we can have to generate the ones 9437519 missing edges as discussed earlier (This may additionally take time).

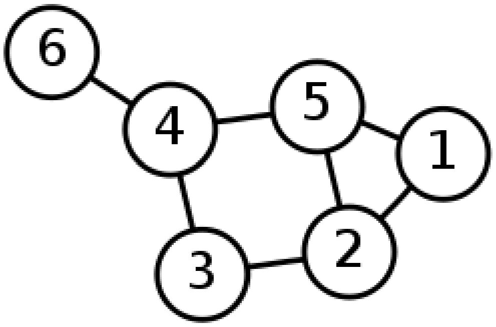
So, we can run a while loop to pick out precisely the 9437519 range of missing edges and include them into a set named 'missing edges'.

We first randomly select nodes from the facts and take a look at if the brink between them is a gift in the dictionary that we created and we take a look at if the two randomly selected nodes aren't the equal node (a node cannot be connected with itself).

Now, we test if the shortest direction between two nodes is extra than two and if sure, we upload them to our lacking set.

The intuition behind this.

Consider nodes 6, 4 and 3. The shortest course between nodes 4 and 3 is two. Now, as nodes 6 and three are each friend of four, they are most probable to be buddies with each other. Therefore, we do away with this situation so that our version may want to examine better. This isn't essential though.



Now, permit's keep the two lists as fantastic and negative records factors and acting test-educate cut up.

```
Number of nodes in the graph with edges 9437519
Number of nodes in the graph without edges 9437519
=======================================
Number of nodes in the train data graph with edges 7550015 =
7550015
Number of nodes in the train data graph without edges 7550015
=7550015
=======================================
Number of nodes in the test data graph with edges 1887504 =
1887504
Number of nodes in the test data graph without edges 1887504
= 1887504
```

As the final data has been prepared and saved in four csv files, let's try to visualize the final data using the network library and highlight some of its characteristics.

```
Type: DiGraph
Number of nodes: 1780722
Number of edges: 7550015
Average in degree:   4.2399
Average out degree:   4.2399
Name:
Type: DiGraph
Number of nodes: 1144623
Number of edges: 1887504
Average in degree:   1.6490
Average out degree:   1.6490
No. of people common in train and test -- 1063125
No. of people present in train but not present in test -- 717597
No. of people present in test but not present in train -- 81498
 % of people not there in Train but exist in Test in total Test data are 7.1200735962845405 %
```

81000 people are present in the test set but not in the train set, this is a slight cold start problem.

Now, let's concatenate the positive and negative edges together and save the final test and train data.

```
Number of nodes in the train data graph with edges 7550015
Number of nodes in the train data graph without edges
7550015
=======================================
Number of nodes in the test data graph with edges 1887504
```

Number of nodes in the test data graph without edges 1887504
Final shape of test and train data
Data points in test data (3775008, 2)
Shape of target variable in train (15100030,)
Shape of target variable in test (3775008,)

Feature Engineering:

Since we have no features to work with in our data, we will be generating our own features.

1) Jaccard Distance

$$J(A,B) = \frac{|\ A\ \cap B\ |}{|\ A\ \cup B\ |}$$

The Jaccard Index measures the similarity between finite sets and is defined as the size of the intersection divided by the union of the sample sets. The Jaccard distance which measures the similarity between sample sets is complimentary to the Jaccard index and is obtained by subtracting the Jaccard index by 1.

$$d_J\ (A,B) = 1 - J(A,B)$$

We will be calculating the Jaccard distance between the followers and followers of each node pair in the training set.

2) Cosine Distance

$$CosineDistance = \frac{|X \cap Y|}{SQRT(|X| \cdot |Y|)}$$

Cosine distance is a metric used to a degree how comparable the files are regardless of their size. Mathematically, it measures the cosine of the perspective among two vectors projected in a multi-dimensional space.

3) Page rank

PageRank (PR) is an algorithm used by Google Search to rank web pages of their seek engine results. PageRank became named after Larry Page, one of the founders of Google.

PageRank works by using counting the quantity and best of links to a web page (nodes in this case) to determine a rough estimate of how crucial the internet site(node) is.

Here, we need to calculate the PageRank around each node pair (supply and vacation spot) in the education setting.

For all the statistics points which might be a part of the test dataset, however, aren't within the schooling dataset we cannot have the PageRank for these statistics points. For those records factors, we can use the suggested PageRank as an imputation.

4) Shortest direction

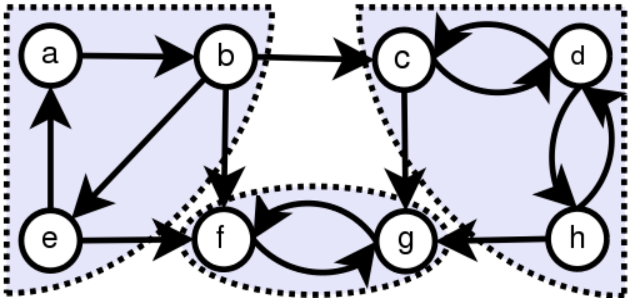
The shortest course is the route among nodes such that the sum of their weights is minimum.

We might be calculating the shortest direction among each node pair in our network.

For all the node pairs, if the nodes already have a hyperlink among them, we will first delete that hyperlink and compute the shortest direction for it. We are doing this so that our version could recognize the community better.

5) Weakly connected additives

A precise component is stated to be strongly linked if there is at least one course from any given node to any other node. To borrow an instance from Wikipedia.



In the primary factor (inclusive of a, b, c) each node is available from each other node in that component. We can cross from a →b, e →a, b →e and b →a via e.

A directed graph is weakly linked if replacing all its directed edges with undirected edges. Therefore, every strongly related thing is a weakly linked issue. However, if it isn't a strongly connected issue, then to test whether it is a weakly linked factor get rid of the instructions of the edges and spot if still there may be at least one path from any given node to some other node.

6) Adar Index

Adamic/Adar measures is described as inverted sum of levels of commonplace associates for given two vertices.

$$A(x,y) = \sum_{u \in N(x) \cap N(y)} \frac{1}{\log|N(u)|}$$

This metric measures the closeness of two nodes based on their shared neighbors. A value of 0 indicates that two nodes are not close, while higher values indicate nodes are close.

7) Follow lower back

This is a simple function to find out if a node follows the lower back after being accompanied by using a node.

8) Katz Centrality

Katz centrality of a node is a degree of centrality in a community. Unlike traditional centrality measures which recollect most effective the shortest direction among a pair of actors, Katz centrality measures affect by means of taking into consideration the total quantity of walks among a couple of actors.

It is similar to Google's PageRank.

9) HITS

Hyper-hyperlink induced topic seek (HITS) identifies good government and hubs for a subject by using assigning two numbers to a node: an expert and a hub weight. Authorities estimate the node price primarily based on the incoming hyperlinks. Hubs estimate the node value based on outgoing links.

We will create a pattern of 500000 factors for training and a pattern of 50000 for trying out. We might be computing the above features for our sampled information.

We may even compute the quantity of fans, followers for both supply and destination and inter followers and followers between them.

Model Building:

Now, that we've mapped this trouble right into a binary elegance classification hassle and prepared a few capabilities for our information, we will proceed with schooling our gadget mastering model on the usage of the Random Forest Classifier.

We will be storing our target variable (indicator_link) in a separate variable for both test and train facts so that we can later examine the real and expected effects to evaluate our model.

4. Results and Discussion

Now that we have trained our model, we have to take a look at our model.

F1 score:

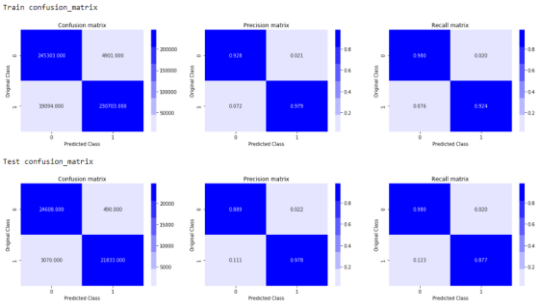
The F1 rating may be interpreted as a weighted average of the precision and take into account, in which an F1 score reaches its satisfactory cost at 1 and worst rating at zero.

Confusion Matrix:

A confusion matrix is frequently used to describe the overall performance of a category model. Is a summary of prediction outcomes on category trouble.

The variety of correct and wrong predictions are summarized with remember values and damaged down utilizing every magnificence.

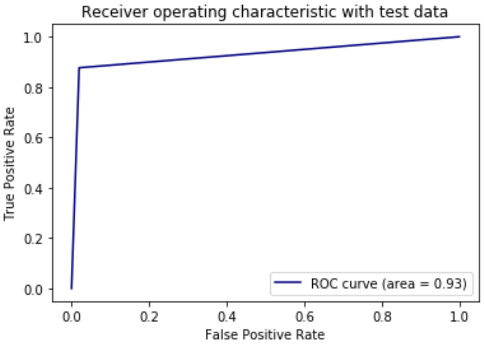
We will be writing a script to compute the confusion matrix, precision matrix, remember matrix for our version and also let us visualize the results in a good way to be smooth to apprehend.



Observations:

1. From the test results, 490 of the points we falsely classified as 'negative' where as they were actually 'positive' points.
2. Similarly, 3070 were wrongly classified as 'positive'.

ROC curve:



ROC curve is another way for measuring the performance of a classification model. Higher the area under the curve (AUC), better the model is at predicting 0s as 0s and 1s as 1s.

The ROC curve is plotted with TPR against the FPR where TPR is on y-axis and FPR is on the x-axis. The objective is to maximize area under the curve (AUC).

5. Conclusion

This paper presented an overview on link prediction in social networking using Machine Learning.

References

[1] David Liben-Nowell and Jon Kleinberg. 2003. The link prediction problem for social networks. In Proceedings of the twelfth international conference on Information and knowledge management (CIKM '03). Association for Computing Machinery, New York, NY, USA, 556–559.

[2] Guillaume Le Floch, "Link Prediction in Large-Scale Networks," August 2018.