# LINK PREDICTION ON SOCIAL MEDIA

A

MAJOR PROJECT REPORT

Submitted by

|  |  |  |
| --- | --- | --- |
| **UMANG TIWARI** | **UDIT JAIN** | **HIMANI SHEORAN** |
| **Enrollment No:** | **Enrollment No:** | **Enrollment No:** |
| **10314802718** | **10214802718** | **04514802718** |

BACHELOR OF TECHNOLOGY IN

**COMPUTER SCIENCE ANDENGINEERING**

Under the Guidance of

**Mrs. SAVITA SHARMA**

**Assistant Professor**



Department of Computer Science and Engineering

Maharaja Agrasen Institute ofTechnology,

PSP area, Sector – 22, Rohini, New Delhi – 110085 (Affiliated to Guru Gobind Singh Indraprastha, New Delhi)

(DEC 2022)

# MAHARAJA AGRASEN INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering



## CERTIFICATE

This is to Certified that this MAJOR project report “LINK PREDICTION ON SOCIAL

MEDIA” is submitted by “UMANG TIWARI ENROLLMENT NO: 10314802718, UDIT JAIN ENROLLMENT NO: 10214802718, HIMANI SHEORAN ENROLLMENT NO: 04514802718” who carried out the project work under my supervision.

I approve this MAJOR project for submission.

Prof. Namita Gupta Ms. SAVITA SHARMA

Designation (HoD, CSE) Assistant Professor

## ABSTRACT

Currently with the rapid development, online social network has been a part of people’s life. A lot of sociology, biology, and information systems can use the network to describe, in which nodes represent individual and edges represent the relationships between individuals or the interaction between individuals. Therefore, the study of complex networks has been the important branch of many scientific fields. Link prediction is an important task in link mining. Link prediction is to predict whether there will be links between two nodes based on the attribute information and the observed existing link information. Link prediction not only can be used in the field of social network but can also be applied in other fields. As in bioinformatics, link prediction can be used to discover interactions between proteins; in the field of electronic commerce, link prediction can be used to create the recommendation system; and in the security field, link prediction can help to find the hidden terrorist criminal gangs. Link prediction is closely related to many areas. Therefore, in recent years there is a lot of correlation algorithms proposed to solve the problem of link prediction.

Social networks are a popular way to interpret the interaction among the people. They can be

Visualized as graphs, where a vertex corresponds to a person and edges represent the connection between them. Understanding the dynamics that drive the evolution of social networks is a complex problem due to a large number of variable parameters. But, a comparatively easier problem is to understand the association between two specific nodes.

For the given source node and destination node we have to predict whether there is any probability of connecting between them.

Then the question is what will be target data (Labelled data) and what the training data are.

Because in Machine Learning we need training data(X) and target data(y). And here we have

Only two columns i.e. source\_node and destination node. To handle this problem we need to create new features. So that it will be helpful for the model. Also we need to create a label (0/1) attribute as the target column, 0 = Not connected and 1 = Connected. When all these are done like adding features and labels then we can say that it is converted to a Machine Learning problem.

## ACKNOWLEDGEMENT

It gives me immense pleasure to express my deepest sense of gratitude and sincere thanks to my respected guide Mrs. SAVITA SHARMA, Assistant Professor, Computer Science and Engineering MAIT Delhi, for their valuable guidance, encouragement and help for completing this work. Their useful suggestions for this whole work and co-operative behavior are sincerely acknowledged.

I am also grateful to my teacher Mrs. SAVITA SHARMA for her constant support and guidance.

I also wish to express my indebtedness to my parents as well as my family member whose blessings and support always helped me to face the challenges ahead.

Place: Delhi Student Name with Roll no.:

Umang tiwari, 10314802718

Udit Jain, 10214802718

Himani Sheoran, 04154802718

Date:

## TABLE OF CONTENTS

|  |  |
| --- | --- |
| **CANDIDATE’S DECLARATION** | **2** |
| **ABSTRACT** | **3** |
| **ACKNOWLEDGEMENT** | **4** |
| **TABLE OF CONTENTS** | **5** |
| **LIST OF FIGURES** | **6** |

|  |  |  |
| --- | --- | --- |
| **1.** | **INTRODCTION** | **7-12** |
| 1.1 | Introduction | 7 |
| 1.2 | Why is machine learning? | 7 |
| 1.3 | Evolution of ml | 8 |
| 1.4 | Application of machine learning | 8 |
| 1.5 | Methods of machine learning | 10 |
| **2.** | **ABOUT THE PROJECT** | **13-17** |
| 2.1 | Social network analysis | 14 |
| **3.** | **LITERATURE REVIEW** | **17-24** |
| 3.1 | Node neighbor analysis | 17 |
| 3.2 | Path based algorithms | 19 |
| 3.3 | Meta approaches | 22 |
| 3.4 | Linear algebraic method | 24 |
| **4.** | **RESEARCH AND APPROACES** | **25-30** |
| **5.** | **PROJECT CODE** | **31-34** |
| **6.** | **RESULTS** | **35-37** |
| **7.** | **CONCLUSION AND FUTURE SCOPE** | **38** |
| **8.** | **REFERENCES** | **39** |

|  |  |  |
| --- | --- | --- |
| **LIST OF FIGURES** | | |
| Figure1.1 | Machine Learning Introduction | 8 |
| Figure 1.2 | Application of Machine  Learning | 10 |
| Figure 1.3 | Machine Learning  Classification | 10 |
| Figure 1.4 | Supervised Learning | 11 |
| Figure 1.5 | Unsupervised Learning | 12 |
| Figure 2.1 | Social Network | 13 |
| Figure 2.2 | Community Detection in Social Networks | 15 |
| Figure 2.3 | Link Prediction in Social Networks | 16 |

**CHAPTER 1:**

**INTRODUCTION**

**INTRODUCTION**

Machine learning is a subfield of artificial intelligence (AI). The goal of machine learning generally is to understand the structure of data and fit that data into models that can be understood and utilized by people.

Although machine learning is a field within computer science, it differs from traditional computational approaches. In traditional computing, algorithms are sets of explicitly programmed instructions used by computers to calculate or problem solve. Machine learning algorithms instead allow for computers to train on data inputs and use statistical analysis in order to output values that fall within a specific range. Because of this, machine learning facilitates computers in building models from sample data in order to automate decision-making processes based on data inputs.

**WHY IS MACHINE LEARNING?**

Resurging interest in machine learning is due to the same factors that have made data mining and Bayesian analysis more popular than ever. Things like growing volumes and varieties of available data, computational processing that is cheaper and more powerful, and affordable data storage.

All of these things mean it's possible to quickly and automatically produce models that can analyse bigger, more complex data and deliver faster, more accurate results – even on a very large scale. And by building precise models, an organization has a better chance of identifying profitable opportunities – or avoiding unknown risks. The nearly limitless quantity of available data, affordable data storage, and the growth of less expensive and more powerful processing has propelled the growth of ML. Now many industries are developing more robust models capable of analyzing bigger and more complex data while delivering faster, more accurate results on vast scales. ML tools enable organizations to more quickly identify profitable opportunities and potential risks.

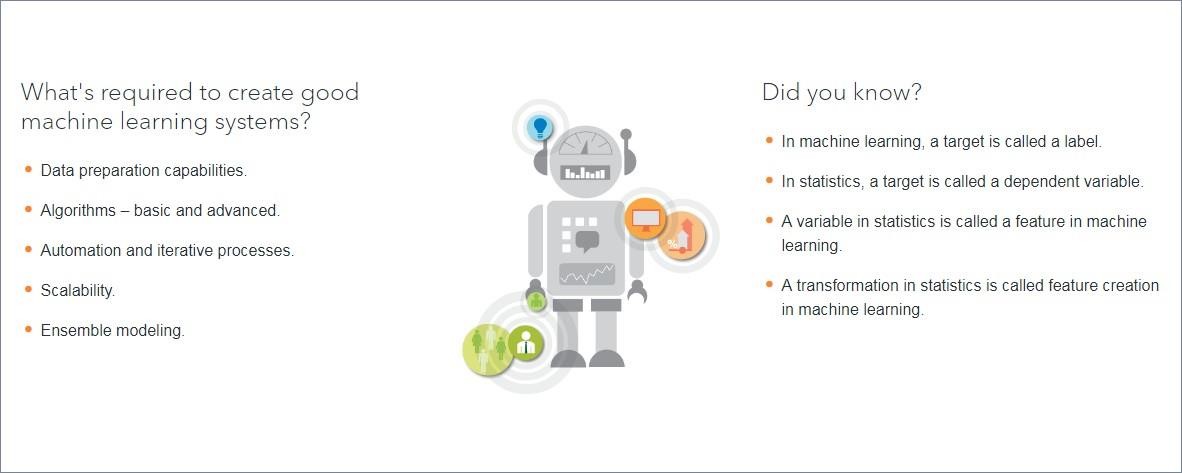


Figure 1.1 Machine Leaning Introduction

**EVOLUTION OF MACHINE LEARNING**

Because of new computing technologies, machine learning today is not like machine learning of the past. It was born from pattern recognition and the theory that computers can learn without being programmed to perform specific tasks; researchers interested in artificial intelligence wanted to see if computers could learn from data. The iterative aspect of machine learning is important because as models are exposed to new data, they are able to independently adapt. They learn from previous computations to produce reliable, repeatable decisions and results. It’s a science that’s not new – but one that has gained fresh momentum.

While many machine learning algorithms have been around for a long time, the ability to automatically apply complex mathematical calculations to big data – over and over, faster and faster – is a recent development.

**APPLICATION OF MACHINE LEARNING**

The value of machine learning technology has been recognized by companies across several industries that deal with huge volumes of data. By leveraging insights obtained from this data, companies are able work in an efficient manner to control costs as well as get an edge over their competitors. This is how some sectors / domains are implementing machine learning –

**FINANCIAL SERVICES**

Companies in the financial sector are able to identify key insights in financial data as well as prevent any occurrences of financial fraud, with the help of machine learning technology. The technology is also used to identify opportunities for investments and trade. Usage of cyber surveillance helps in identifying those individuals or institutions which are prone to financial risk, and take necessary actions in time to prevent fraud.

**MARKETING AND SALES**

Companies are using machine learning technology to analyse the purchase history of their customers and make personalized product recommendations for their next purchase. This ability to capture, analyse, and use customer data to provide a personalized shopping

experience is the future of sales and marketing.

**HEALTHCARE**

With the advent of wearable sensors and devices that use data to access health of a patient in real time, ML is becoming a fast- growing trend in healthcare. Sensors in wearable provide real-time patient information, such as overall health condition, heartbeat, blood pressure and other vital parameters. Doctors and medical experts can use this information to analyse the health condition of an individual, draw a pattern from the patient history, and predict the occurrence of any ailments in the future.

**TRANSPORTATION**

Efficiency and accuracy are key to profitability within this sector; so is the ability to predict and mitigate potential problems. ML’s data analysis and modeling functions dovetail perfectly with businesses within the delivery, public transportation, and freight transport sectors. ML uses algorithms to find factors that positively and negatively impact a supply chain’s success, making machine learning a critical component within supply chain management.

**GOVERNMENT**

Systems that use machine learning enable government officials to use data to predict potential future scenarios and adapt to rapidly changing situations. ML can help to improve cybersecurity and cyber intelligence, support counterterrorism efforts, optimize operational preparedness, logistics management, and predictive maintenance, and reduce failure rates. This recent article highlights 10 more applications for machine learning within the healthcare industry.

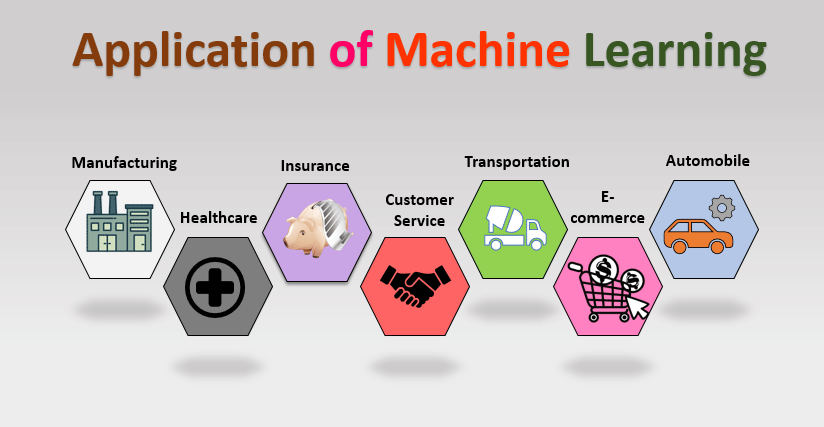


Figure 1.2 Application Of Machine Learning

**METHODS OF MACHINE LEARNING**

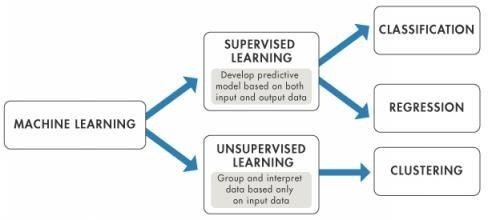


Figure 1.3 Machine Learning Classification

**SUPERVISED LEARNING**

These algorithms are trained using labeled examples, in different scenarios, as an input where the desired outcome is already known. An equipment, for instance, could have data points such as "F" and "R" where "F" represents "failed" and "R" represents "runs". A learning algorithm will receive a set of input instructions along with the corresponding accurate outcomes. The learning algorithm will then compare the actual outcome with the accurate outcome and flag an error, if there is any discrepancy. Using different methods, such as regression, classification, gradient boosting, and prediction, supervised learning uses different patterns to proactively predict the values of a label on extra unlabeled data. This method is commonly used in areas where historical data is used to predict events that are likely to occur in the future. For instance, anticipate when a credit card transaction is likely to be fraudulent or predict which insurance customers are likely to file their claims.

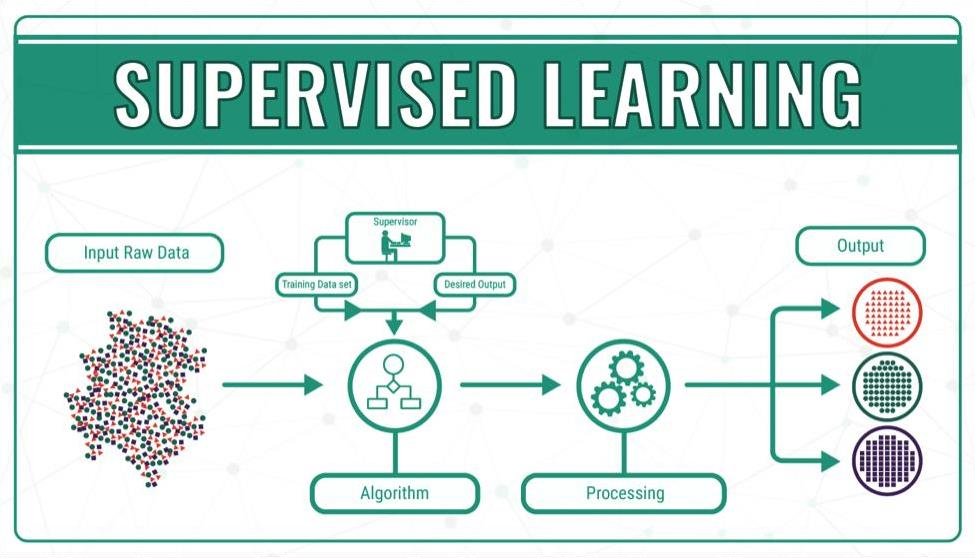


Figure 1.4 Supervised Learning

**UNSUPERVISED LEARNING**

This method of ML finds its application in areas were data has no historical labels. Here, the system will not be provided with the "right answer" and the algorithm should identify what is being shown. The main aim here is to analyse the data and identify a pattern and structure within the available data set. Transactional data serves as a good source of data set for unsupervised learning. For instance, this type of learning identifies customer segments with similar attributes and then lets the business to treat them similarly in marketing campaigns. Similarly, it can also identify attributes that differentiate customer segments from one another. Either ways, it is about identifying a similar structure in the available data set. Besides, these algorithms can also identify outliers in the available data sets.

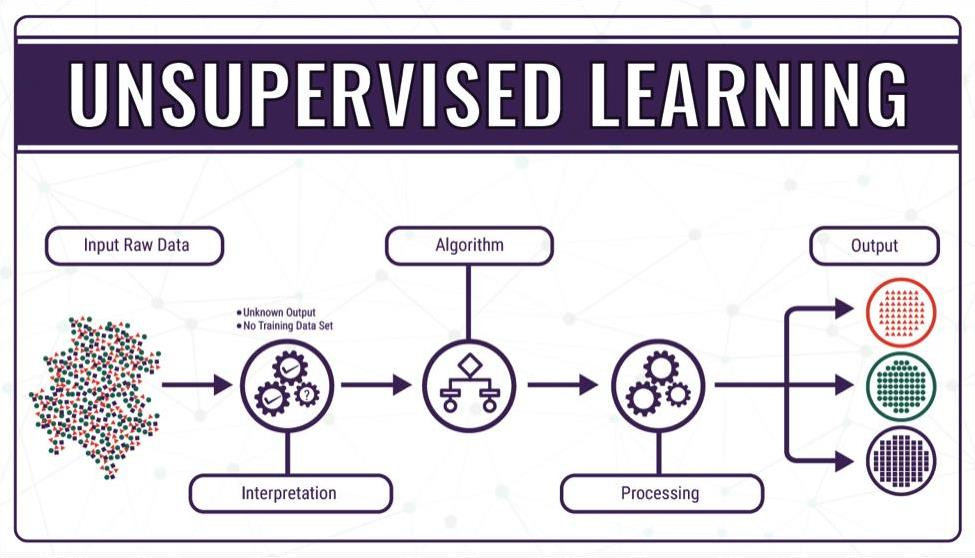


Figure 1.5 Unsupervised Learning

**PROBLEM STATEMENT**

Parkinson's disease occurs when nerve cells, or neurons, in an area of the brain that controls movement become impaired and/or die. Normally, these neurons produce an important brain chemical known as dopamine. When the neurons die or become impaired, they produce less dopamine, which causes the movement problems of Parkinson's.Besides motor symptoms, the person may see, hear, or experience things that are not real (hallucinations), or believe things that are not true (delusions).

**CHAPTER 2:**

**INTRODUCTION TO SOCIAL NETWORKS**

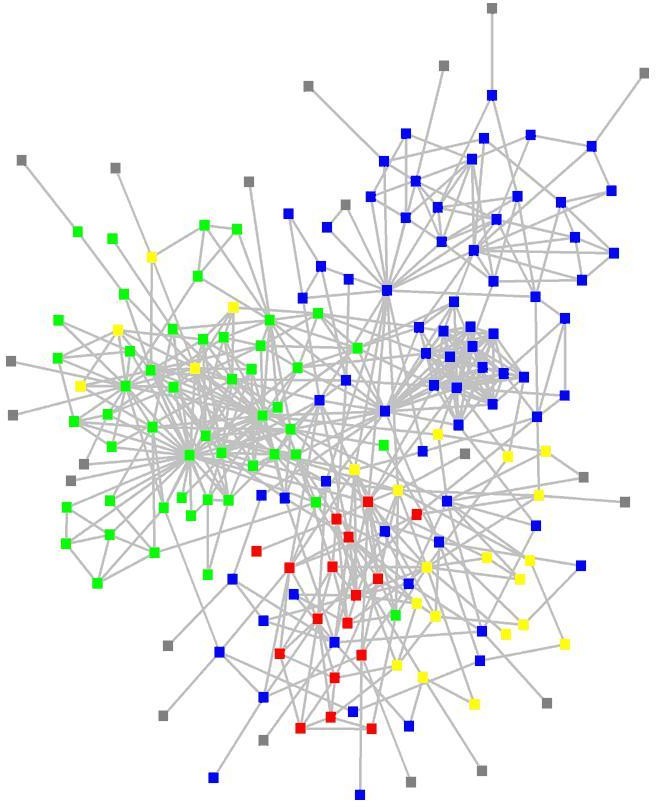
A Social Structure consists of nodes(Individuals or Organizations) and nodes are connected by different types of relationships. A set of social actors or nodes(such as individuals or organizations) and a set of the dyadic ties between these nodes constitute a social network. For example scientists in a discipline, employees in a large company, business leaders can be thought as nodes in a network and co- authors of a paper, working on a project, serve together on board can be thought as edges respectively. The idea behind Social Networks is to create opportunities to develop friendships, share information and promote business in a network. OSN like Facebook and Twitter have become important part of daily life of millions of people. The enormous growth and dynamics of these networks has led to several researches that examine the network properties i.e. structural and behavioral properties of large scale social networks.

Figure 2.1: Social Network

**Social Network Analysis**

Social network analysis(SNA) is in depth analysis of social networks. SNA is the mapping and measuring of relationships, links and owes between nodes(people, groups, organizations, computers) and many other connected entities which pro-

vides some knowledge and information. The vertices or nodes in the network are the people and groups while the links show relationships or owes between the nodes. We can do visual and a mathematical analysis of human relationships through SNA that helps us to make sense out of the social network, to and the complex structure of social networks, to understand the evolution of social net- works, network dynamics and to discover complex communication patterns and characteristic features of the network.

**Tasks Of Social Network Analysis**

Social networks are dynamic by nature. They change very quickly over a specific interval. Continuously new relationships establish between nodes and many old relationships break. These relational changes(when people become friends through common friends), characteristics of the nodes, characteristics of pairs of actors or link weights and random unexplained events in sequences the graph characteristics. The key tasks of SNA include different measures to rank nodes(or edges), Link prediction problem, Inferring social networks from social events, Viral marketing, Community detection, Design of incentives in networks, Determining implicit social hierarchy, Network formation, Spars cation of social networks(with purpose). There are many measures to rank nodes like degree centrality, closeness central-ity, clustering coe cient, betweenness centrality, Katz centrality and Eigen vector centrality. Link prediction is predicting the links that does not exist or exist, but not known and have probability to occur in the near future. Viral marketing deals with exploiting social connectivity patterns of users to propagate the awareness of product. Community detection involves graph partitioning based on activities over the social network and determining the dense sub graphs in a social network. In designing the incentives, only the person who answers the query is rewarded, with no reward for the intermediaries. Since individuals are often rational and intelligent, they may not participate in answering the queries unless some kind of incentives are provided. SNA has many applications like informa- tion sharing, Information sharing, Understand the spread of diseases, Marketing in e-Commerce and e-Business, determine the in entail entities, build e active social and political campaign, Predict future events, tracking terrorists and location based crowdsourcing.

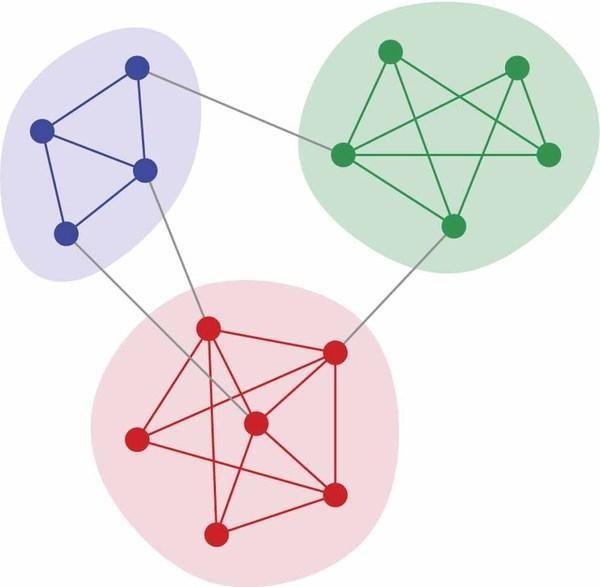


Figure 2.2: Community Detection in Social Networks

**Link Prediction Problem**

Different kind of links or edges between the nodes exist in a social network. For example, social contacts, phone-calls or hyper-references. On analysis of social networks, there can be many information about the linkage between the nodes that are not discovered or unknown at a given point of time. Link Prediction is the problem of predicting links that either dont yet exist at the given time t or exist, but unknown up to this time. Given a picture of a social network(nodes and links) at time t, we need to predict accurately the links that will be added to the network during the interval from time t to a given future time t+1. In effect, the link prediction problem concentrates on to what extent can the evolution of a social network be modelled by using intrinsic features of the network itself? Let

us consider a co-authorship network among researchers, for example, there are di erent reasons, outside to the network, why two researchers who have never written a paper together will do so in the next few years. Or, when one of the researchers changes institutions, they may come geographically very close. Such interactions are be hard to predict. But by studying the network characteristics, we can predict the possible links that are going to form. Our objective is to make this intuitive notion very exact, and to understand which measures of proximity in a graph lead to accurate predictions.

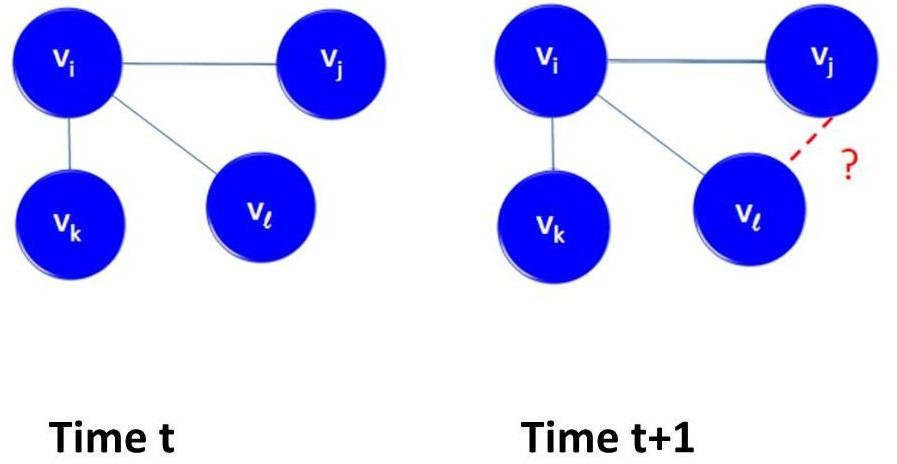


Figure2.3: Link Prediction in Social Networks

The link prediction problem is also deals with the problem of getting missing links from a known network, in a number of elds. It involves prediction of addi- tional links that are not directly visible currently, are likely to exist in a network based on observable data. It considers a static picture of the network, rather than taking network evolution and network dynamics. It also considers speci c prop-erties of the nodes in the network, rather than computing the power of prediction methods that focuses on the graph structure.

**Application of Link Prediction**

Apart from its role as a basic question in social network formation, the link pre- diction problem could be related to a number of interesting applications of so-cial networks. It is found that a large and medium organisation like a company can bene t from the involvement within the social network among its employees. These bene ts to supplement the organisation hierarchy de ned by the organiza-tion. E ective and e cient methods for link prediction could be used to analyse and study such a social network, and suggest interactions that have not yet been utilized within the organization, more likely to form. Link Prediction has a great role in security research, largely inspired by the problem of controlling terrorist networks and predicting their future involvement. In bioinformatics, e cient link prediction techniques can be used to predict interactions between proteins. In e-commerce it helps in building the recommendation systems so that helps in viral marketing and e ective product awareness.

**CHAPTER:4**

**LITERATURE REVIEW**

Given a social network G(V; E) in which an edge represents some kind of interactions between its vertices on nodes at a given time t. Suppose we have

a snapshot of a social network at a given time. We choose four times t0 < t00 < t1 < t01, and give our algorithm to predict links that are likely to be formed in the near future from the network G[t0; t00]. That results in predicting new links, not present in G[t0; t00], that are expected to appear in the network G[t1; t01]. We refer to [t0; t00] as the training interval and [t1; t01] as the test interval [1].

The most basic approach for similarity between any pair of nodes is by taking the length of their shortest path in graph. We rank pairs of nodes in descending order of score(x; y), where score(x; y) is the negative of the shortest path length between x and y. We take a snapshot of a social network as training set and predict the interactions among the nodes of training set that are likely to occur in near future.The algorithms are classi ed as belows [2].

**NODE NEIGHBORHOOD ALGORITHMS**

Node neighborhood meaning the nodes directly connected to the two given nodes. It is simple technique which traverse only paths of length 2. For any node A it check the neighbors of neighbour of A and computes their similarity with A. It considers only local features of a network, focusing mainly on the nodes structure(i.e. based on the number of common friends that two users share).

**Common Neighbors**

The Common Neighbors method provide a measure for similarity by calculating the intersection of the sets of neighbors of the nodes to predict future linkage. The Common Neighbors(CN) is de ned as follows

CN(x; y) := (x) (y)

This measurement is based on the idea that two nodes a and b have an increased probability to connect if they have a shared neighbor c. With a growing number of shared neighbors this probability grows even higher.

The weighted Common Neighbors(CN w) is de ned as follows where w(x; y) is the number of interactions between the nodes x and y.

**Jaccard coe cient**

Jaccards coe cient measures number of the features(neighbors) that are shared between two nodes commensurate to all features that either one of the nodes has. Jaccards coe cient is a normalized variation of Common Neighbors [? ]R7). The Jacard coe cient is de ned as follows

J(x; y) := (x) \ (y)

(x) [ (y)

This is the Common Neighbors measurement normalized by the union of the node neighborhoods.

**Adamic/Adar**

It is a measurement that compares how many attributes two nodes have in com- mon. They rate items that are unique to a few users more heavily than items shared amongst a huge group of users. This measurement can easily be adjusted in the context of node neighborhood by looking at shared neighbors as an at- tribute. Therefore the sum over the shared neighbors inverse of the logarithms of their neighborhoods is proposed [3].

The Adamic/Adar is de ned as follows

X 1

AA(x; y) :=

logj (z)j

z (x)\ (y)

The weighted Adamic/Adar (AAw) is de ned as follows where w(x; y) is the number of interactions between the nodes x and y [4].

X w(x; z) + w(y; z)

:

**Preferential Attachment**

Preferential Attachment is based on the hypothesis that a node x will get new neighbors faster than a node y given y has less neighbors than x. So the probability that a node will form a new link varies with number of its present neighbors. The likelihood of two nodes being connected by an edge based on preferential attachment is measured by multiplying the number of their neighbors [5]. The Preferential Attachment is de ned as follows

P A(x; y) := (x): (y)

The weighted Preferential Attachment (P Aw) is de ned as follows where w(x; y) is the number of interactions between the nodes x and y:

X

P Aw(x; y) := w(x; x0): y0 " (y)w(y0; y)

x; (x)

**PATH BASED ALGORITHMS**

Some measurements of link prediction take all paths between two nodes in consid- eration. The computation of graphs that take the entire graph in consideration is by nature much more complex than node neighborhood algorithms.

Katz

A measurement that takes all paths between two nodes in consideration while rating short paths more heavily. The measurement exponentially reduce the con-tribution of a path to the measure in order to give less weightage longer paths. Therefore it uses a factor of l where l is the path length.

The Katz is de ned as follows

1

X

K(x; y) := l:jpaths<l>x;yj

l=1

where paths<l>x;y the set of all paths from source x to destination y that have the path length l.

Unweighted : paths<l>x;y = 1, if x and y have collaborated and 0 otherwise

Weighted : paths<l> is the number of times that x and y have collaborated

x;y

The can be used to control how much the length of the paths should be considered. A very small concludes to an algorithm where paths of length three or more are taken much less into account and therefor the algorithm converges node neighborhood algorithms. It has roughly cubic complexity as it requires matrix inversion [6].

**SimRank**

If two nodes are referenced by more similar objects, then the two nodes have large similarity value. Every object obviously has a similarity score of 1 to itself. Node x and node y are then similar to the degree they are joined to similar neighbors [7]. a constant with [0; 1]. The constant can be thought of as a con dence level. If you consider a situation in which a and b are both neighbors to c, than obviously the similarity of c to itself is 1, but we do not want to conclude that

s(a; b) = s(c; c) = 1. Instead we let s(a; b) = s(x; x) because we are not as con dent about the similarity of a and b as we are about s(x; x) = 1.

**Hitting Time and Commute Time**

Starting from a node x a random walk on a given graph moves iteratively over the graph while choosing the next node each step at random. The expected number of steps to get from x to y via a random walk is de ned as the Hitting Time H(x; y). A short hitting time implies node similarity and therefor a heightened chance of future linking. The commute time C(x; y) is a variant of Hitting time which is useful for undirected graphs, because the hitting time is not symmetric. Therefore it is de ned as follows:

C(x; y) := H(x; y) + H(y; x)

The commute time can have high variance, hence, prediction by this feature can be poor. If z is a node with high stationary probability far o x and y, then a random walker would probably reach the neighborhood of z. To avoid that we can use reset the random walker to x with a xed probability of .

two normalized versions Hitting Time normalized (Hn) and Commute-Time

normalized (Cn) are de ned where x is the stationary probability of x to safeguard it against vertices with a very high :

Hn(x; y) := H(x; y): y

Cn(x; y) := (H(x; y): y + H(y; x): x)

**Rooted PageRank**

Rooted PageRang is a modi cation of the Page Rank measure (which is an at- tribute of a single vertex) for link prediction. It is the amount of step from x to y with a probability of to return to x each step (and 1 to go to a random neighbor). This metric is asymmetric and can be made symmetric by summing with the counterpart where the role of x and y are reversed [8].

The rooted pagerank(RPR) between all node pairs is calculated as follows: Let D be a diagonal degree matrix de ned as:

X

D[i; i] := A[i; j]

j

And let N be the following matrix with normalized row sums.

N := D 1

Then the Rooted Pagerank can be calculated as

RP R := (1 )(I N) 1

**PropFlow and High-Performance Link Prediction**

The unsupervised PropFlow method calculates the probability that a random walker reaches node y from node x in l steps or fewer while using link weights as transition probabilities. If the algorithms revisits any node including x or if it reaches y the algorithm terminates. When compared to Rooted PageRank the algorithm is more localized and is insensitive to topologic noise far from the source node. It is faster to compute because it does not require random resets. High-Performance Link Prediction as a framework for link prediction. They distinguish between two variants:

HPLP: Does not use the existing unsupervised methods, but only simple Measures like In- and Out-Degree, Max. Flow, Shortest Paths or PropFlow

HPLP+: Uses the full feature set adding Adamic/Adar, Jaccards coe cient, Katz and Preferential Attachment [2].

**Supervised Random Walks**

Node and link attributes along with node structure information are used for pre- diction. Supervised learning strength is assigned to the edges that are likely to have new links in the future so that random walker can visit them more likely. The Strength is not set manually, but learned from the features of each edge and nodes between them.

**META APPROACHES**

Meta-Approaches alter the data before being passed to one of the algorithms mentioned above.

**Low-rank approximation**

For a lot of the mentioned algorithms there is a equivalent formulation for an

adjacency matrix M. For a large Matrix M the Matrix Mk is the rank-k matrix, what can be done e ciently by singular value decomposition [9].

Katz measure using Mk rather than 4MCommonN eighborsscoringbyinnerproductsof rows rather than M

The contains most related nodes to x under score(x; :) are de ned as Sx< >. So after calculating the score(x; y), we need to calculate the Sx< >.

U B(x; y) := jz : z (y) \ Sx< >j

X

U Bw(x; y) := score(x; z)

z (y)\S < >

x

**Clustering**

This includes improving the quality of the algorithms by a clustering procedure and after that the algorithm is applied to the modi ed sub graph. To achieve that the measure is computing score(x; y) for all edge in the original graph and only keeping the p fraction of these edges, where the score is highest [10]. After that the score algorithm is applied to the modi ed graph. Using this technique, the measurement is only applied to those nodes, in which the scoring algorithm has the most con dence in. This can be seen as a cleaning up by removing of tenuous edges [11].

**Bayesian Probabilistic Model**

There are two types of probabilistic approaches to predict links.

The rst approach extends a framework of probabilistic relational models cap- turing probabilistic interactions between attributes of related entities by modelling interactions between the attributes and the link structure itself [12]. For a proba- bility distribution over a database a template describing the relational schema for the domain and the probabilistic dependencies between attributes of the domain in form of a PRM(probabilistic relational model) is speci ed. Probability distribu-tion on the properties of the nodes and the links can be de ned. By including the links into the probabilistic model they can be used to predict other links and to help make predictions about other attributes in the model. If we look at existence uncertainty no assumptions are made about the number of links that exist they are part of the probabilistic model, but can still be used to make inferences about other attributes in the model [13].

The second approach is based on the topological features of network structures only. A probabilistic evolution model of network structure modelling probabilistic ips of existence of edges depending on a copy-and-paste mechanism of edges is presented. Based on this model a transductive learning algorithm for link pre-diction based on an assumption of the stationarity of the network is proposed. The algorithm realizes a maximum likelihood estimation procedure using expo-nentiated gradient ascent. This is based on the idea that if a node a has a strong in uence on a node b and there is an edge between a and another node c. The authors assume a high probability that a link will establish between b and c and that there is a very low probability that there will never be a node between them [14].

**LINEAR ALGEBRAIC METHOD**

It is a general method to solve the link prediction problem which works directly on the graph adjacency matrix or Laplacian matrix. The problem is reduced to a one- dimensional regression problem [9][15, 16]. They training set is reduced to its biggest connected component. The resulting set was then split into two adjacency matrices A and B, where A was the source matrix and B containing one third of all edges the target matrix. Di erent curve tting methods can be used to predict the edge sets in the test set [17].

**CHAPTER 4:**

**RESEARCH AND APPROACH**

Let’s define a social network first before we dive into the concept of link prediction.

A social network is essentially a representation of the relationships between social entities, such as people, organizations, governments, political parties, etc.

The interactions among these entities generate unimaginable amounts of data in the form of posts, chat messages, tweets, likes, comments, shares, etc. This opens up a window of opportunities and use cases we can work on.

That brings us to **Social Network Analytics (SNA). We can define it as a combination of several activities that are performed on social media.** These activities include data collection from online social media sites and using that data to make business decisions.

The benefits of social network analytics can be highly rewarding. Here are a few key benefits:

Helps you understand your audience better

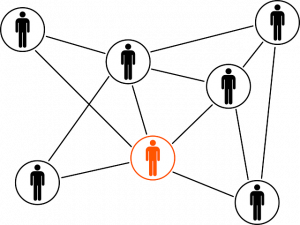
Used for customer segmentation

Used to design Recommendation Systems

Detect fake news, among other things

A Primer on Link Prediction

Link prediction is one of the most important research topics in the field of graphs and networks. **The objective of link prediction is to identify pairs of nodes that will either form a link or not in the future.**

[](https://cdn.analyticsvidhya.com/wp-content/uploads/2019/12/lp_10.png)

Link prediction has a ton of use in real-world applications. Here are some of the important use cases of link prediction:

Predict which customers are likely to buy what products on online marketplaces like Amazon. It can help in making better product recommendations

Suggest interactions or collaborations between employees in an organization

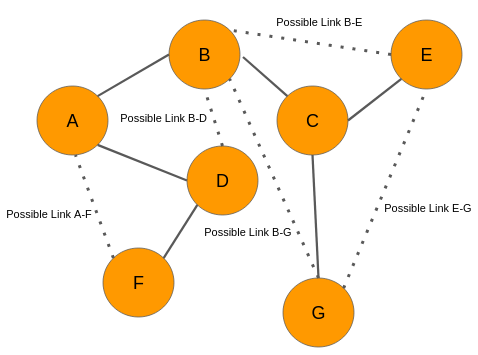
Extract vital insights from terrorist networks

In this article, we will explore a slightly different use case of link prediction – predicting links in an online social network!

Strategy to Solve a Link Prediction Problem

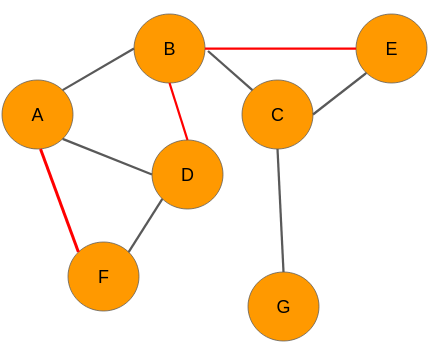
If we can somehow represent a graph in the form of a structured dataset having a set of features, then maybe we can use [machine learning](https://courses.analyticsvidhya.com/courses/applied-machine-learning-beginner-to-professional?utm_source=blog&utm_medium=link-prediction-how-to-predict-your-future-connections-on-facebook) to predict the formation of links between the unconnected node-pairs of the graph.tcou

Let’s take a dummy graph to understand this idea. Given below is a 7 node graph and the unconnected node-pairs are AF, BD, BE, BG, and EG:

[](https://cdn.analyticsvidhya.com/wp-content/uploads/2019/12/lp_1.png)

Graph at time t

Now, let’s say we analyze the data and came up with the below graph. A few new connections have been formed (links in red):

[](https://cdn.analyticsvidhya.com/wp-content/uploads/2019/12/lp_2.png)

Graph at time t+n

We need to have a set of predictor variables and a target variable to build any kind of machine learning model, right? So where are these variables? Well, we can get it from the graph itself! Let’s see how it is done.

**Our objective is to predict whether there would be a link between any 2 unconnected nodes.** From the network at time t, we can extract the following node pairs which have no links between them:

A-F

B-D

B-E

B-G

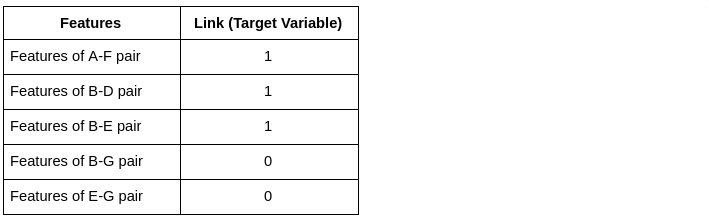
E-G

Please note that, for convenience, I have considered only those nodes that are a couple of links apart.

The next step for us is to create features for each and every pair of nodes. The good news is that there are several techniques to extract features from the nodes in a network. Let’s say we use one of those techniques and build features for each of these pairs. However, we still don’t know what the target variable is. Nothing to worry about – we can easily obtain that as well.

Look at the graph at time t+n. We can see that there are three new links in the network for the pairs A-F, B-D, and BE respectively. Therefore, we will assign each one of them a value of 1. The node pairs B-G and E-G will be assigned 0 because there are still no links between the nodes.

Hence, the data will look like this:

[](https://cdn.analyticsvidhya.com/wp-content/uploads/2019/12/lp_3.png)

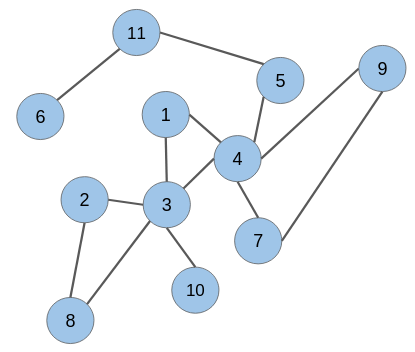
Now that we have the target variable, we can build a machine learning model using this data to perform link prediction.

So, this is how we need to use social graphs at two different instances of time to extract the target variable, i.e., the presence of a link between a node pair. Keep in mind, however, that in real-world scenarios, we will have data of the present time only.

Extract data from a Graph for Building your Model

In the section above, we were able to get labels for the target variable because we had access to the graph at time t+n. However, in real-world scenarios, we would have just one graph dataset in hand. That’s it!

Let’s say we have the below graph of a social network where the nodes are the users and the edges represent some kind of relationship:

[](https://cdn.analyticsvidhya.com/wp-content/uploads/2019/12/lp_4.png)

The candidate node pairs, which may form a link at a future time, are (**1 & 2)**, (**2 & 4)**, (**5 & 6)**, (**8 & 10)**, and so on. We have to build a model that will predict if there would be a link between these node pairs or not. This is what link prediction is all about!

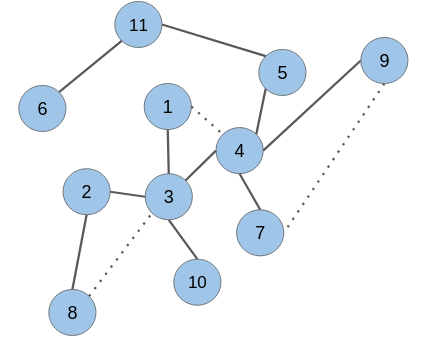
However, to build a link prediction model, we need to prepare a training dataset out of this graph. It can be done using a simple trick.

Picture this – how would this graph have looked like at some point in the past? There would be fewer edges between the nodes because connections in a social network are built gradually over time.

Hence, keeping this in mind, we can randomly hide some of the edges from the given graph and then follow the same technique as explained in the previous section to create the training dataset.

**Strike Off Links from the Graph**

**While removing links or edges, we should avoid removing any edge that may produce an isolated node (node without any edge) or an isolated network.** Let’s take off some of the edges from our network:

[](https://cdn.analyticsvidhya.com/wp-content/uploads/2019/12/lp_5.png)

As you can see, the edges in the node pairs (1 & 4), (7 & 9), and (3 & 8) have been removed.

Add labels to extracted data

Next, we would need to create features for all the unconnected node pairs including the ones for which we have omitted the edges. The removed edges will be labeled as ‘1’ and the unconnected node pairs as ‘0’:

|  |  |
| --- | --- |
| Features | Link (Target Variable) |
| Features of node pair 1 – 2 | 0 |
| Features of node pair 1 – 5 | 0 |
| Features of node pair 1 – 7 | 0 |
| Features of node pair 1 – 8 | 0 |
| Features of node pair 1 – 9 | 0 |
| Features of node pair 1 – 10 | 0 |
| Features of node pair 2 – 4 | 0 |
| Features of node pair 2 – 10 | 0 |
| Features of node pair 3 – 5 | 0 |
| Features of node pair 3 – 7 | 0 |
| Features of node pair 3 – 9 | 0 |
| Features of node pair 4 – 8 | 0 |
| Features of node pair 4 – 10 | 0 |
| Features of node pair 4 – 11 | 0 |
| Features of node pair 5 – 6 | 0 |
| Features of node pair 5 – 7 | 0 |
| Features of node pair 5 – 9 | 0 |
| Features of node pair 8 – 10 | 0 |
| Features of node pair 1 – 4 | 1 |
| Features of node pair 3 – 8 | 1 |
| Features of node pair 7 – 9 | 1 |

It turns out that the target variable is highly imbalanced. This is what you will encounter in real-world graphs as well. The number of unconnected node pairs would be huge.

Let’s take up a case study and solve the problem of link prediction using Python.

Case Study: Predict Future Connections between Facebook Pages

This is where we’ll apply all of the above into an awesome real-world scenario.

We will work with a graph dataset in which the nodes are Facebook pages of popular food joints and well-renowned chefs from across the globe. If any two pages (nodes) like each other, then there is an edge (link) between them.

# CHAPTER 5 :

# PROJECT CODE

## 4.1 CODE

import networkx as nx

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

plt.style.use('seaborn')

from sklearn.model\_selection import train\_test\_split,GridSearchCV,RandomizedSearchCV

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import roc\_curve, auc, roc\_auc\_score, confusion\_matrix,accuracy\_score,f1\_score

from xgboost import XGBClassifier

from time import time

from collections import Counter

import random

import os

print("Number of NUll Values in Training Data : ",sum(df\_train.isna().any(1)))

print("Number of Duplicate Values in Training Data : ",sum(df\_train.duplicated()))

nodes = G.nodes()

print("Number of Nodes : ",len(nodes))

print("Number of Edges : ",len(edge\_list))

## 

pos=nx.spring\_layout(subgraph)

plt.figure(figsize=(15,10))

nx.draw\_networkx(subgraph,pos,edge\_color='#111',font\_color='white',node\_size=400)

plt.savefig("graph\_sample.png")

print(nx.info(subgraph))

X\_train\_pos, X\_test\_pos, y\_train\_pos, y\_test\_pos  = train\_test\_split(df\_pos,np.ones(len(df\_pos)),test\_size=0.2, random\_state=9)

X\_train\_neg, X\_test\_neg, y\_train\_neg, y\_test\_neg  = train\_test\_split(df\_neg,np.zeros(len(df\_neg)),test\_size=0.2, random\_state=9)

print("Number of nodes in the Train data graph with edges", X\_train\_pos.shape[0]," : ",y\_train\_pos.shape[0])

print("Number of nodes in the Train data graph without edges", X\_train\_neg.shape[0]," : ", y\_train\_neg.shape[0])

print("Number of nodes in the Test data graph with edges", X\_test\_pos.shape[0]," : ",y\_test\_pos.shape[0])

print("Number of nodes in the Test data graph without edges", X\_test\_neg.shape[0]," : ",y\_test\_neg.shape[0])

#mapping jaccrd followers to train and test data

df\_final\_train['jaccard\_followers'] = df\_final\_train.apply(lambda row:jaccard\_for\_followers(row['source\_node'],row['destination\_node']),axis=1)

df\_final\_test['jaccard\_followers'] = df\_final\_test.apply(lambda row:jaccard\_for\_followers(row['source\_node'],row['destination\_node']),axis=1)

#mapping jaccrd followees to train and test data

df\_final\_train['jaccard\_followees'] = df\_final\_train.apply(lambda row:jaccard\_for\_followees(row['source\_node'],row['destination\_node']),axis=1)

df\_final\_test['jaccard\_followees'] = df\_final\_test.apply(lambda row:jaccard\_for\_followees(row['source\_node'],row['destination\_node']),axis=1)

#mapping jaccrd followers to train and test data

df\_final\_train['cosine\_followers'] = df\_final\_train.apply(lambda row:cosine\_for\_followers(row['source\_node'],row['destination\_node']),axis=1)

df\_final\_test['cosine\_followers'] = df\_final\_test.apply(lambda row:cosine\_for\_followers(row['source\_node'],row['destination\_node']),axis=1)

#mapping jaccrd followees to train and test data

df\_final\_train['cosine\_followees'] = df\_final\_train.apply(lambda row:cosine\_for\_followees(row['source\_node'],row['destination\_node']),axis=1)

df\_final\_test['cosine\_followees'] = df\_final\_test.apply(lambda row:cosine\_for\_followees(row['source\_node'],row['destination\_node']),axis=1)

df\_final\_train.drop(['source\_node', 'destination\_node','indicator\_link'],axis=1,inplace=True)

df\_final\_test.drop(['source\_node', 'destination\_node','indicator\_link'],axis=1,inplace=True)

tuned\_params = {'max\_depth': [1, 2, 3, 4, 5], 'learning\_rate': [0.01, 0.05, 0.1], 'n\_estimators': [100, 200, 300, 400, 500], 'reg\_lambda': [0.001, 0.1, 1.0, 10.0, 100.0]}

model = RandomizedSearchCV(XGBClassifier(), tuned\_params, n\_iter=15, scoring = 'roc\_auc', n\_jobs=-1)

model.fit(df\_final\_train,y\_train) # actual data and actual prediction

print('Train f1 score',f1\_score(y\_train,y\_train\_pred))

print('Test f1 score',f1\_score(y\_test,y\_test\_pred))

def plot\_confusion\_matrix(test\_y, predict\_y):

    C = confusion\_matrix(test\_y, predict\_y)

    A =(((C.T)/(C.sum(axis=1))).T)

    B =(C/C.sum(axis=0))

    plt.figure(figsize=(20,4))

    labels = [0,1]

    # representing A in heatmap format

    cmap=sns.light\_palette("blue")

    plt.subplot(1, 3, 1)

    sns.heatmap(C, annot=True, cmap=cmap, fmt=".3f", xticklabels=labels, yticklabels=labels)

    plt.xlabel('Predicted Class')

    plt.ylabel('Original Class')

    plt.title("Confusion matrix")

    plt.subplot(1, 3, 2)

    sns.heatmap(B, annot=True, cmap=cmap, fmt=".3f", xticklabels=labels, yticklabels=labels)

    plt.xlabel('Predicted Class')

    plt.ylabel('Original Class')

    plt.title("Precision matrix")

    plt.subplot(1, 3, 3)

    # representing B in heatmap format

    sns.heatmap(A, annot=True, cmap=cmap, fmt=".3f", xticklabels=labels, yticklabels=labels)

    plt.xlabel('Predicted Class')

    plt.ylabel('Original Class')

    plt.title("Recall matrix")

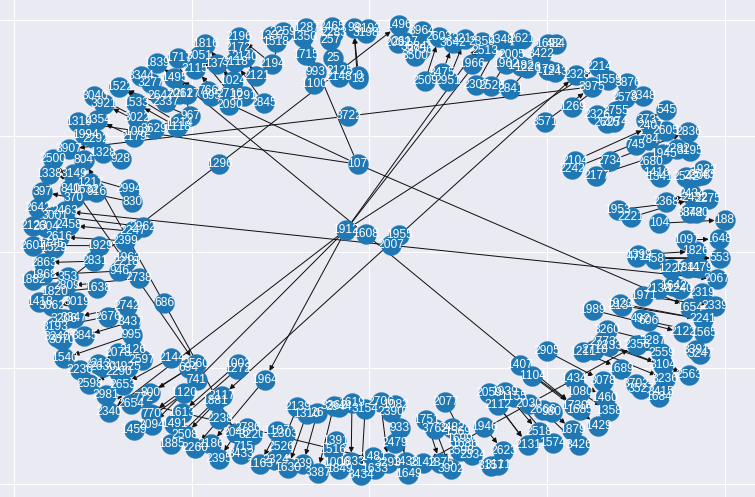
    plt.show()

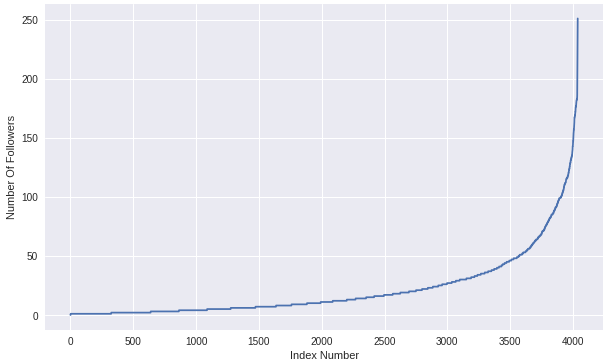
# 

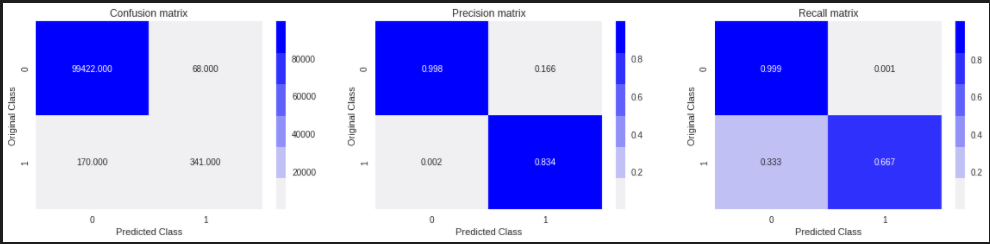
# 

# CHAPTER 6:

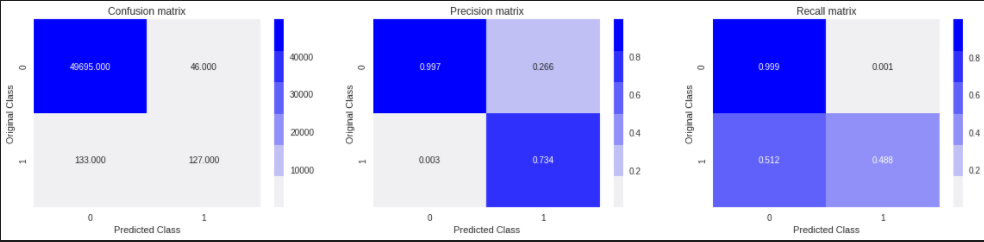
# RESULTS

****

****

****

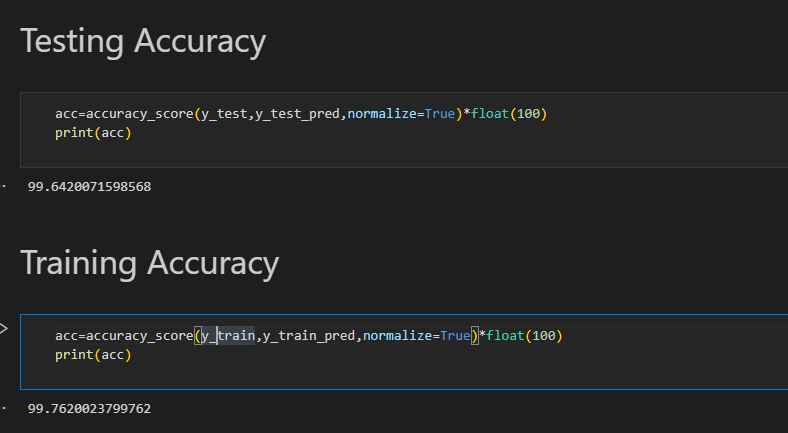
**Train confusion matrix**

****

**Test confusion matrix**

****

**ROC CURVE**



**The given ensembled model (XG BOOST) has achieved an accuracy of 99.6%.**

**CHAPTER 7 :**

**CONCLUSION & FUTURE SCOPE**

As in bioinformatics, link prediction can be used to discover interactions between proteins; in the field of electronic commerce, link prediction can be used to create the recommendation system; and in the security field, link prediction can help to find the hidden terrorist criminal gangs. Link prediction is closely related to many areas. Therefore, in recent years there is a lot of correlation algorithms proposed to solve the problem of link prediction.

The given ensembled model will help in achieving the above objectives efficiently.

**REFERENCES**

* Liyan Dong,1,2 Yongli Li,3 Han Yin,1,2 Huang Le,1,2 and Mao Rui1, The Algorithm of Link Prediction on Social Network, Volume 2013 |Article ID 125123
* [Mahdi Jalili](https://royalsocietypublishing.org/doi/10.1098/rsos.160863), [Yasin Orouskhani](https://royalsocietypublishing.org/doi/10.1098/rsos.160863), [Milad Asgari](https://royalsocietypublishing.org/doi/10.1098/rsos.160863), [Nazanin Alipourfard](https://royalsocietypublishing.org/doi/10.1098/rsos.160863) and [Matjaž Perc](https://royalsocietypublishing.org/doi/10.1098/rsos.160863" \o "Matjaž Perc),

Link prediction in multiplex online social networks, Published:01 February 2017 <https://doi.org/10.1098/rsos.160863>

* [Journal of the American Society for Information Science and Technology](https://asistdl.onlinelibrary.wiley.com/journal/15322890),The link-prediction problem for social networks,[David Liben-Nowell](https://asistdl.onlinelibrary.wiley.com/action/doSearch?ContribAuthorRaw=Liben-Nowell%2C+David),[Jon Kleinberg](https://asistdl.onlinelibrary.wiley.com/action/doSearch?ContribAuthorRaw=Kleinberg%2C+Jon), First published: 26 March 2007, <https://doi.org/10.1002/asi.20591>

# New perspectives and methods in link prediction,Ryan N. Lichtenwalter, Jake T. Lussier, [KDD '10: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining](https://dl.acm.org/doi/proceedings/10.1145/1835804)July 2010 Pages 243-252 <https://doi.org/10.1145/1835804.1835837>

# Link Prediction using Supervised Learning ∗ Mohammad Al Hasan, Vineet Chaoji, Saeed Salem, and Mohammed Zaki Rensselaer Polytechnic Institute, Troy, New York 12180 {alhasan, chaojv, salems, zaki}@cs.rpi.edu