Problem statement:

Given a directed social graph, have to predict missing links to recommend users (Link Prediction in graph)

Data Overview

Taken data from facebook's recruiting challenge on kaggle
https://www.kaggle.com/c/FacebookRecruiting

data contains two columns source and destination each edge in graph - Data columns (total 2 columns):

- source_node int64

- destination_node int64

Mapping the problem into supervised learning problem:

- Generated training samples of good and bad links from given directed graph and for each link got some features like no of followers, is he followed back, page rank, katz score, adar index, some svd features of adj matrix, some weight features etc. and trained ml model based on these features to predict link.

Business objectives and constraints:

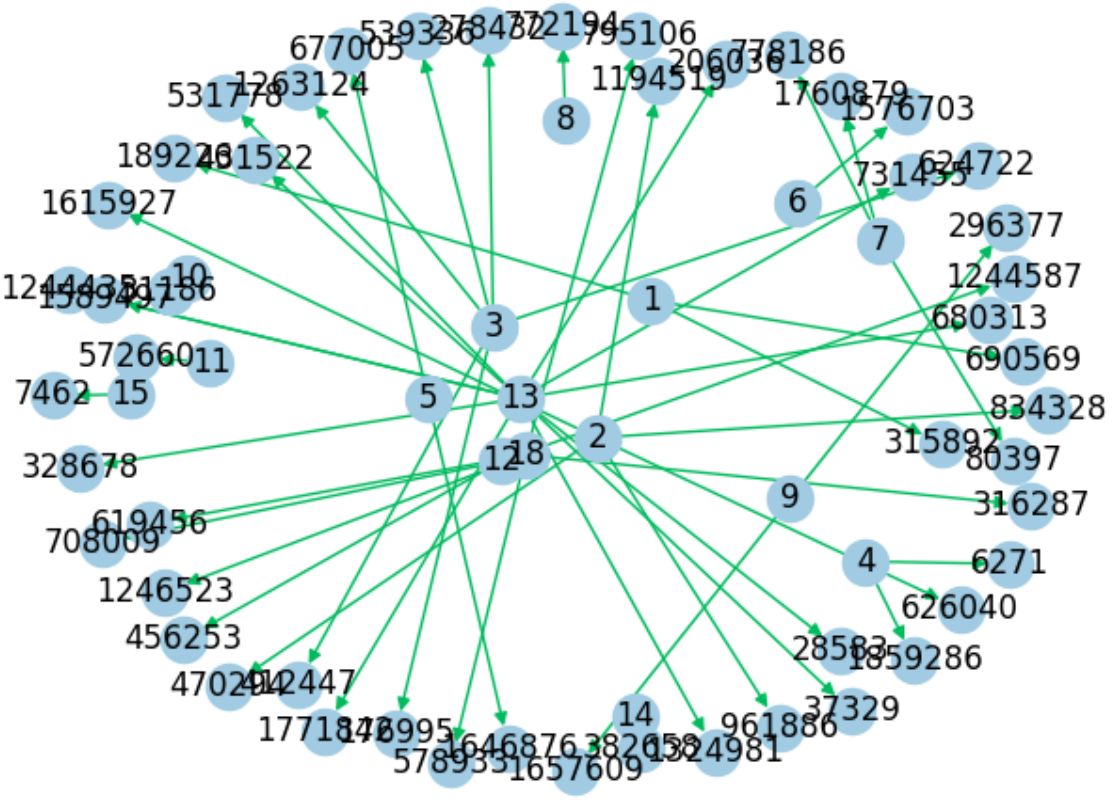
- No low-latency requirement.
- Probability of prediction is useful to recommend highest probability links

Performance metric for supervised learning:

- Both precision and recall is important so F1 score is good choice
- Confusion matrix

Approach to the problem statement

The first step would be data reading and Exploratory Data Analysis . We start by creating nodes, and join them by edges. Then the graph would look like:



Now identifying unique profiles and followers. We can see the number of followers for each person, and from basic Exploratory Data Analysis, after that it is seen that 99% of all the data have 40 followers. After that, inspect the number of people each person is following. Further on we can inspect the number of people each person is following, and the minimum and the maximum number of followers or following. Then we are going to generate the missing edges in the graph. After that we split the training and test data.

Jaccard distance, $j = \frac{X \cap Y}{X \cup Y}$

Cosine distance, $c = \frac{|X \cap Y|}{|X|.|Y|}$

Jaccard distance and cosine distance assist us to understand how likely two nodes are to be connected. We further work on PageRank, which computes a ranking of the nodes in the graph based on the structure of the incoming links.

The final step is building a prediction model and improving upon its accuracy. The final accuracy obtained using XGBoost for training dataset is 0.99, while for testing dataset, the accuracy is 0.91.