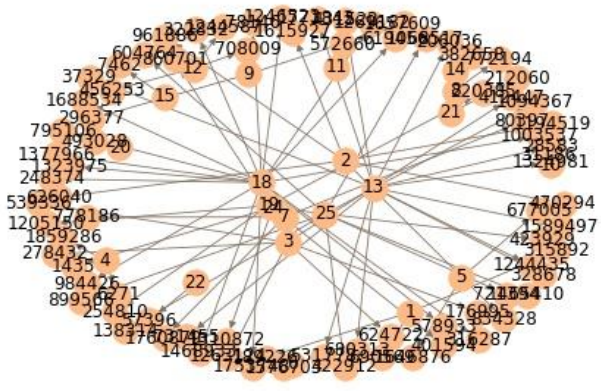
Facebook Recruiting Challenge: Approach to Solution

Introduction

The problem statement for this challenge required to predict the missing links to recommend users given a directed graph. It is a link prediction problem where we have to predict the links with high probability. As per the requirements Machine learning, Supervised approach is used to find the best possible solution. The idea of the solution is adopted from the reference papers acknowledged bellow.

Data Analyses

First data is thoroughly analyzed to get a complete idea of the type of people or individuals in data I.e how much people a person is following or is followed by or is not following or not followed by anyone or are weakly connected to each other etc. This is done with the help of graph nodes, indegree and outdegree values. A sample graph of the data is as follows:



Split data and Feature Engineering

After thorough analysis we split the data into test and train chunks using 80:20 split ratio. I first generate the bad edges so that our data is not biased just towards good edges and thus allow the model to perform better. Next I do some feature engineering where we try to analyze the data further for both followers and followees, I.e calculate its jacquard distance, cosine score, Katz score, preferability score, hits score, page rank, Adar index, compute the shortest path, weakly connected nodes, SVD of adjacency matrix constructed form the graph. All these measures are transformed into features which are next used.

Model Training

Next I train two models over it one is the random forest classifier and the other one is xgboost. First remove , ['source_node', 'destination_node','indicator_link'], these features because we no more need them. Incase of Random Forest generate results over an ensemble of random forests with some estimator values and plot a graph of the results. Next generate results over an ensemble of depth values and plot the results. F1 score for Random Forest is:

```
+--------------+--------------------+-------------------+
|        Model           |      Train_f1_Score      |      Test_f1_score      |
+--------------+--------------------+-------------------+
```

```
|     RandomForest     |     0.9950792126740279     |     0.891837767654761     |
+--------------+--------------------+------------------+
```

Next train the xgboost model by feeding in all the features and it will find the best features that perfectly represent the data. To get the get the best params, I carried out 4 experiments and then trained the model. The f1 score for XGBoost is:

```
+---------+---------------+--------------------+
|       Model       |    Train_f1_Score    |          Test_f1_score              |
+---------+---------------+--------------------+
|   XGBoost   |                 1.0                    |   0.8892053039067367   |
+---------+---------------+--------------------+
```

Conclusion

As per the f1 score confusion matrices and the roc curves of both the models, RandomForest performed better. XGBoost as most likely overfitted the data that is why, performed poor on test data as compared to RandomForest.

References:

Data: https://www.kaggle.com/c/FacebookRecruiting

Problem approach:

https://www.cs.cornell.edu/home/kleinber/link-pred.pdf

https://www3.nd.edu/~dial/publications/lichtenwalter2010new.pdf

https://www.youtube.com/watch?v=2M77Hgy17cg

https://kaggle2.blob.core.windows.net/forum-message-attachments/2594/supervised_link_prediction.pdf