# CP 318: Link Prediction in a Graph Network

## Project 1 Report: Team PATH FINDERS

**AAYUSHI KOCHAR, HARINI VEMURI, PRAVEEN PRASAD HANDIGOL**

### Brief about Problem Statement

Social networks like Twitter have become a significant part of modern communication, with users following one another to stay updated with their activities. In this study, we explore the task of predicting the probability of a link between users in a Twitter network using machine learning models. We are provided with a dataset containing historical information about 20,000 source nodes and their respective destination nodes. Our goal is to leverage this data to train various machine learning models and then apply them to predict link probabilities for a test dataset of 2,000 source and destination nodes.

### Dataset Description:

The dataset comprises 20,000 source nodes (user IDs) and their corresponding destination nodes (users followed by the source). Each record in the dataset represents a potential link between a source and a destination node. The dataset includes information about user interactions, followers, and other features that can be used to predict the probability of a link.

### Methodology:

Task 1: Sampling: We created a sample set with 20000 datapoints. Source ID is selected randomly with replacement form source column of dataset. For 10000 true predictions, receiver nodes are randomly selected from respective source row. For 10000 false predictions, receiver nodes are randomly selected from unique receiver node of entire data.

Task 2: Feature Extraction: To train the model on the train dataset, some features are needed. Features extracted based on the given dataset are:

- **Source Follower:** This feature represents the number of unique IDs that follow the Source node.
- **Source Following:** It denotes the count of unique IDs that the Source node follows.
- **Receiver Follower:** This feature quantifies the number of unique IDs that follow the Receiver node.
- **Receiver Following:** Like the Source Following feature, this attribute signifies the count of unique IDs followed by the Receiver node.
- **Transitive Friends ("Likelihood of Connection"):** potential connection between two nodes relies on the existence of common connections or mutual friends in the network.

$$transitive - friends(u, v) = || \Gamma out(u) \bigcap \Gamma in(v) ||$$

- **Jaccard Coefficient:** The Jaccard coefficient is a metric for assessing the similarity between two sets. In the context of this feature, it measures the similarity between the sets of connections of the Source and Receiver nodes. A higher Jaccard coefficient indicates a greater likelihood of a connection between the two nodes.

$$JaccardCoefficient = |\Gamma(va) \cap \Gamma(vb)|/|\Gamma(va) \cup \Gamma(vb)|$$

- **Total Friends:** This feature is computed as the sum of the number of friends associated with both the Source and Receiver nodes.

- **Common Friends IN:** No. of common outbound friends from source and receiver nodes.

$$Common - friend\sin(u,v) = |\Gamma in(v) \cap \Gamma in(u)|$$

- **Common Friends OUT:** No of common outbound friends from source and receiver nodes.

$$Common - friendsout(u,v) = |\Gamma out(v) \cap \Gamma out(u)|$$

- **Cosine:** capture the similarity between nodes based on their common neighbors.

$$Co\sin e = |\Gamma(va) \cap \Gamma(vb)|/(|\Gamma(va)|)(|\Gamma(vb)|)$$

- **Preferential attachment score**: measures the likelihood of forming a connection between two nodes in a network based on the number of existing connections each node has.

$$Preferential\ attachment\ score\ (u,v) = |\Gamma(v)|.|\Gamma out(u)|$$

Task 3: Training different Machine Learning models on the training data based on the features extracted. Different models used are: AdaBoost, Artificial Neural Network, Ensemble, K-Nearest Neighbor, Logistic Regression, Naïve Bayes, Random Forest, Support vector Machine and XGBoost.

To assess the importance of individual features in a dataset, multiple models are executed, with the features "Source Follower," "Source Following," "Receiver Follower," and "Receiver Following" held constant for comparative analysis.

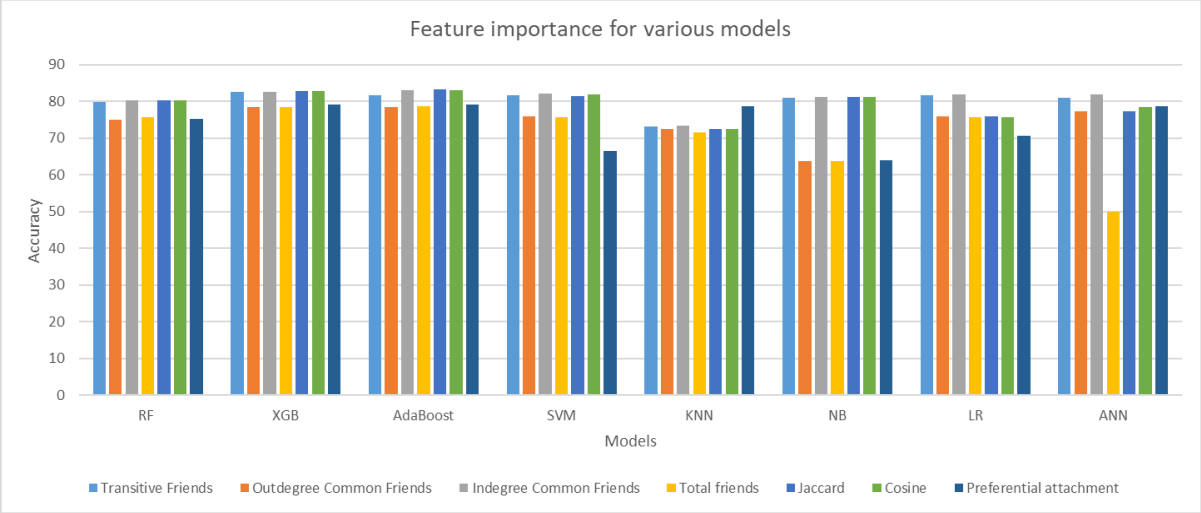


*Figure 1: Model vs accuracy plot to understand the importance of various features.*

## Results and Discussion:

- On running different models with various features, the **Jaccard Coefficient and Transitive Friends** are top performing, followed by Cosine and Indegree common friends. Jaccard measures similarity between two sets of nodes, and for the given dataset, where common neighbors between two nodes indicate a potential link, Jaccard is found to be performing well.
- Plotting Receiver Operating Characteristic Curve for different models helps to identify top model: **Artificial Neural Network (ANN)**, followed by XGBoost, AdaBoost, Random Forest and Logistic Regression.
- Scaling data before running the model improves accuracy by ensuring that all features contribute equally to the learning process.
- With the increase in training dataset size, the models could generalize better to unseen examples and hence the accuracy improved with larger training dataset.

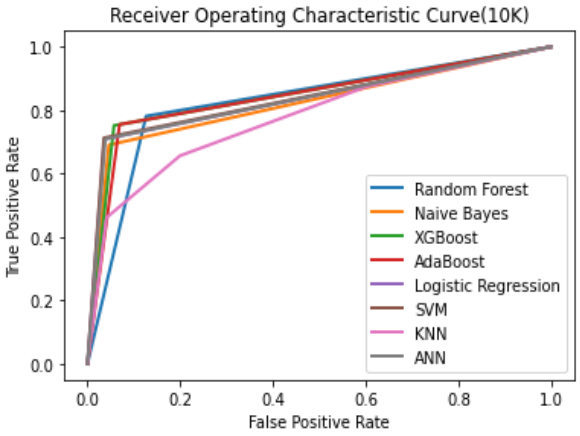


*Figure 2: ROC plot for 10k dataset*

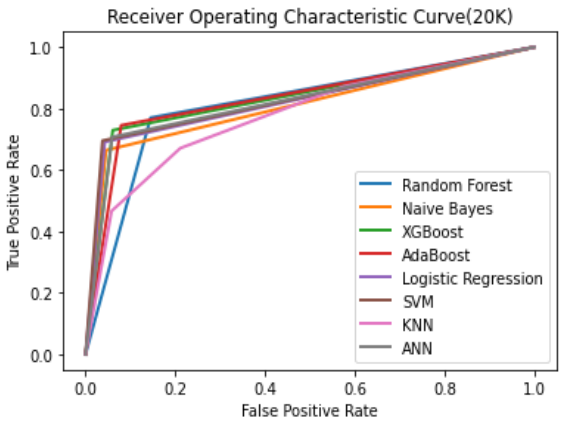


*Figure 3: ROC plot for 20k dataset*

- **XGBoost: Extreme Gradient Boosting Model**, a new model used for prediction, gave validation accuracy **83.46%** with learning rate 0.01. It works by building an ensemble of decision trees sequentially, with each other correcting the error of the previous ones. The objective function of XGBoost takes leaf node weight and tree depth, which is added to the loss function to control the complexity of the model and prevent overfitting. Also, optimization of the objective function is done using second-order Taylor series expansion, which handles both gradient and curvature information and contributes to better convergence and prediction accuracy.
- Using n/10 estimators in a Random Forest balances between reducing model variance and increasing bias, resulting in a less prone-to-overfitting model.
- Comparing different models: **Artificial Neural Network (ANN)** outperformed other models with the highest Train and Test accuracy. This could be due to ANN's ability to capture non-linear relationships and patterns in data, which is important in link prediction. While running the model, the input layer is activated by Leaky ReLU with 11 neurons, two hidden layers with PreLU activation function and 128 and 32 neurons respectively, and the output layer has sigmoid activation function. With a batch size of 32, and 30 epochs, validation gave an accuracy of **83.77%.** Various other combinations of neurons and activation functions were also tried, but the above-mentioned activation function and number of neurons gave the best performance. Leaky ReLU mitigates the "dying ReLU" problem, providing a small gradient for negative inputs, which helps prevent neuron inactivity. This results in improved gradient flow, reduced sensitivity to weight initialization, and potentially better generalization. Parametric ReLU (PReLU) allows each neuron to learn an optimal slope for negative inputs during training, again reducing the risk of dying ReLUs. PReLU can improve model flexibility, convergence, and generalization. Random Forest, AdaBoost, and XGBoost are ensemble methods that combine multiple models to improve predictive performance. Logistic Regression is simpler and interpretable but gives lower accuracy.
- The idea of ensembling all the above models was also implemented. All the probabilities predicted by various models were stacked together and median of the probabilities were taken. This produced a validation accuracy of **83%.**

## References:

**1.** Govinda K, Rajkumar Rajasekaran, V.S.R.P. Venkata Krishna Varun, Davya Vuyyuru, Battula Thirumaleshwari Devi, "Link Prediction in Social Networks using Machine Learning", 2023 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE), DOI: 10.1109/IITCEE57236.2023.10091004.

**2.** M. Fire, L. Tenenboim, O. Lesser, R. Puzis, L. Rokach and Y. Elovici, "Link Prediction in Social Networks Using Computationally Efficient Topological Features," 2011 IEEE Third International Conference on Privacy Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing, Boston, MA, USA, 2011, pp. 73-80, DOI: 10.1109/PASSAT/SocialCom.2011.20.

**3.** Prateek Joshi (2016), "A Guide to Link Prediction – How to Predict your Future Connections on Facebook", https://www.analyticsvidhya.com/blog/2020/01/link-prediction-how-to-predict-your-future-connections-on-facebook/

**4.** W. Cukierski, B. Hamner and B. Yang, "Graph-based features for supervised link prediction," The
2011 International Joint Conference on Neural Networks, San Jose, CA, USA, 2011, pp. 1237-1244, DOI: 10.1109/IJCNN.2011.6033365

**5.** Vignesh Iyer (2019), "Link Prediction in a Social Network" https://vgnshiyer.medium.com/link-prediction-in-a-social-network-df230c3d85e6

**6.** ChatGPT by OpenAI.