# CP 318: Data Science for Smart City Applications
# Project 1: Link Prediction in a Graph Network

Team: Network Ninzas
Authors: Alok Mishra (21401), Sushil Jangra (21242), Sital Kumar Sahu (21764)
Submitted to: Prof Punit Rathore, Assistant Professor (IISc)

## 1. Introduction & Problem Statement

In our daily lives, we encounter various pairwise relationships, from friendships among individuals to computer communication links and similarities between cars. These relationships form networks, with nodes representing entities and edges denoting connections. However, real-world network data often has missing edges due to errors or constraints, impacting analysis and predictions. This project tackles the challenge of predicting missing edges in a Twitter network, where nodes are Twitter users and edges signify user-following relationships. Our goal is to find the probability whether a set of test edges truly exists in the Twitter network or is fabricated.

## 2. Data Description

The dataset is structured as an edge list, containing information about the relationships between nodes. Specifically, each row in the dataset represents an edge between a source node and one or more destination nodes.

- The first column of the dataset contains unique identifiers representing Twitter users, often referred to as "source nodes." In our dataset, we have approximately **20,000** distinct source nodes, each corresponding to an individual Twitter user.
- The subsequent columns in the dataset represent the destination nodes connected to the corresponding source node. These destination nodes signify Twitter users who are being followed by the respective source node. Since each source node can follow multiple destination nodes, the number of destination nodes in each row can vary.
- It includes a total of approximately **1.5 million unique nodes**, indicating the presence of numerous Twitter users within the network. Moreover, the dataset comprises approximately **30 million edges**, reflecting the connections between these nodes.

## 3. Methodology

Our approach to link prediction in the Twitter network involves the following key steps:

- **Data Preprocessing**: We begin by reading the training data, provided in the form of an adjacency edge list. This data is used to create a directed graph representation of the Twitter network. We then select a subset of source nodes for modeling and create a balanced set of positive (genuine) and negative (fabricated) samples for training.
- **Data Sampling**:We commenced by collecting a balanced dataset consisting of 30,000 samples, equally divided into 15,000 positive samples (representing existing edges) and 15,000 negative samples (representing non-existent edges). This balance was maintained to ensure that our model does not become biased towards either class.A third column was added representing classes as 0 or 1 for negative and positive samples respectively.
- **Feature Engineering**: To capture meaningful insights from the Twitter network, we design a set of features for the edges. These features include common neighbors, Jaccard coefficient, Adamic/Adar coefficient, resource allocation index, Sørensen index, total followers, friends measure, in-degree, out-degree, transitive friends, and edge rank.

- **Model Training**: The above described features were used to train various models.To explore various classification techniques, we employed the following models: Logistic Regression, Naive Bayes, Random Forest, AdaBoost, and Artificial Neural Network (ANN). Each model offered unique advantages and challenges.

| Serial No. | Feature Name | Description |
|---|---|---|
| 1 | Common Neighbors | Number of common neighbors between two nodes in a graph. |
| 2 | Jaccard Coefficient | A similarity measure based on the size of the intersection divided by the size of the union of two sets. |
| 3 | Adamic/Adar Coefficient | A measure of similarity between nodes in a network based on shared neighbors. |
| 4 | Resource Allocation Index | A measure of similarity between nodes in a network based on the allocation of resources. |
| 5 | Sørensen Index | Similarity coefficient based on the size of the intersection of two sets. |
| 6 | Total Followers | Total number of followers of a Twitter user. |
| 7 | Friends Measure | A metric indicating the level of friendship or connection between two users in a social network. |
| 8 | In-Degree | Number of incoming edges to a node in a directed graph. |
| 9 | Out-Degree | Number of outgoing edges from a node in a directed graph. |
| 10 | Transitive Friends | Number of friends who have friends in common with a user. |
| 11 | Edge Rank | A measure indicating the importance or significance of an edge in a network. |

Table 1: List of Features and Descriptions

## 4. Selection and Improving the Model

Various performance evaluation measures were taken to study the performance of a model.Different models were also compared using AUC score. The key steps are outlined below:

1. **Performance Evaluation:** To assess the effectiveness of above described models, we constructed confusion matrices and calculated the F1 score for validation. The F1 score provides a balanced measure of precision and recall.We then tested each of these models using AUC score for test data.

2. **Optimizing ANN:** Among the models, we observed that the Artificial Neural Network (ANN) yielded most promising results. We construct a neural network model using TensorFlow and Keras. The model consists of multiple layers, including dense layers with activation functions. Binary cross-entropy is used as the loss function, and accuracy is monitored during training.To further enhance the accuracy of ANN, we implemented several measures. We employed early stopping to prevent overfitting and reduce training time. Additionally, we utilized the "ReduceLROnPlateau" technique to dynamically modify the learning rate during training, helping the model converge faster.

3. **Activation Function Modification:** Another modification that contributed to our model's success was the change in the activation function. We shifted from the Rectified Linear Unit (ReLU) activation function to the Leaky ReLU. This adjustment allowed our model to capture more nuanced patterns in the data, enhancing its ability to make accurate predictions.

## 5. More about Model Evaluation

We utilized 20 percent of the sampled data for model validation, where we obtained a confusion matrix and several evaluation metrics. An example of this is presented below.

| Confusion Matrix | | |
|---|---|---|
| Actual | Predicted | |
| | 0 | 1 |
| 0 | 2901 | 51 |
| 1 | 103 | 2945 |

Table 2: Confusion Matrix

| Classification Report | | | | |
|---|---|---|---|---|
| Class | Precision | Recall | F1-Score | Support |
| 0 | 0.97 | 0.98 | 0.97 | 2952 |
| 1 | 0.98 | 0.97 | 0.97 | 3048 |
| Accuracy | 0.97 (6000 total samples) | | | |
| Macro Avg | 0.97 | | | |
| Weighted Avg | 0.97 | | | |

Table 3: Classification Report

## 6. Results and Discussion

As depicted in the following graph, we can observe the training and validation loss across epochs. This graph illustrates the diminishing loss and convergence as epochs progress. Subsequently, we utilize the trained model to predict the existence of a link between two IDs in the test data.

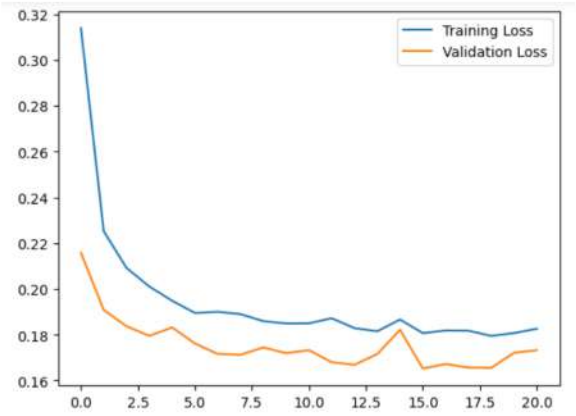


Figure 1: Training and Validation Loss Over Epochs

## 7. Novelty

Novel approaches were employed to improve the model's performance, as elaborated upon below:

1. **Intelligent Sampling:** A noteworthy aspect of our approach is the use of intelligent sampling. We incorporated sources of test data into the training set, enabling our model to gain insights from the test data distribution. This intelligent sampling strategy helped improve the model's predictive capabilities.

2. **Feature Standardization:** We employed feature standardization to achieve uniformity, improve model performance and generalization, ensure numerical stability, and maintain consistent model performance across diverse features. This process resulted in the development of more resilient and efficient neural networks. Additionally, standardization facilitated quicker convergence during model training.

## 8. Conclusion

In conclusion, our approach to link prediction in the Twitter network involved comprehensive feature engineering and the use of a neural network model. The choice of features and the neural network architecture were motivated by the need to capture the intricacies of the network structure and user interactions. Through this project, we gained insights into the complexities of real-world network data and the importance of feature engineering. The project provided a valuable opportunity to apply machine learning techniques to a practical problem and participate in a Kaggle competition. It challenged us to think critically and make informed decisions to improve link prediction accuracy in the Twitter social network.