# COMP90051: PROJECT1 REPORT

**TEAM: 2B || !2B**
Atheena Skathidharan (924244, ashakthidha)
Jaye Philip Heffernan (639038, jHeffernan)
Navnita Nandakumar (921834, nnandakumar)

---

## Introduction:

Link prediction has been a spurred research topic over the years. With the growth of social media, there has been a need to predict new links or relationships, identify fake relations, missing edges and incomplete networks. The project aims to predict the probabilities of existences of relationship between two Twitter users based on existing trained data. A partial crawl of Twitter data is used to train the system, with each row identified as the source to sink relation. The first element in the row represents the source and all the remaining nodes identify the sink nodes which the source user follows. The test data consists of 2000 edges represented by the source and sink nodes, for which the system predicts the probability of existence and this is submitted to the Kaggle in-class competition.

## Edge Probability Prediction System:

The final edge prediction system submitted to Kaggle is a well-trained, researched and improved version from the initial submissions. This prediction system is trained using the training dataset made available. A total of 60000 data points from the dataset are used for the purpose of training the system. For the purpose of training and to avoid over-fitting of data, the training set was split for training and validation purpose. A randomly shuffled combination of these prior knowledge data points and erratically generated edges are used to create the sets. The training set had 50000 data points against a validation set of 500 points. An extensive set of features extracted with a good predictive model is the key to a well-developed system. The system is pipelined with three main steps – feature extraction, preprocessing, and classification.

## I. Feature Extraction

The edge predictions are computed based on the postulate that similar nodes are more likely to be connected. In order to identify a possibility of an edge between two users, it is important to understand and answer the real-world scenarios of how likely is a user expected to connect to another and with who are the probabilities more [1]. The system utilizes 55 features per edge and the polynomial interaction between them, among the source and destination nodes of the edge in order to compute the probability of a connection. These features are mainly based on different concepts identified and listed below.

### Inbound and Outbound Degrees

The degree of inbound and outbound edges from a node is a representation of how many connections a user has, either following or being followed. This is an important feature of the final system as helps identify all the neighbors of a user node. The number of inbound and outbound relations of the source and destination nodes of the edge is calculated and the polynomial interaction between is computed. The number of followers of the source and of sink users and the users that the source and sink nodes follow are important features that help predict the possibility of connectivity. The interactions between these degrees have added a considerable amount of accuracy in the prediction.

### Similarity Scores

Edge prediction methods are highly vulnerable to the similarity of the users or the neighbour of these users under consideration. Users are more likely to follow or be followed by similar people as they follow or are being followed. For this reason, the system computes the Jaccard similarity scores among different combinations of these nodes. The Jaccard index helps add value to the intersection vs. the union of compared nodes. The four combinations of nodes used for Jaccard similarity in this system are

1. Similarity between the followers of the source and sink nodes
2. Similarity between the followees of the source and sink nodes
3. Similarity between the followers and followees of the source node

4. Similarity between the followers and followed of the sink node

**Subgraph Similarity**

A feature extracted to predict the probability of existence of an edge is that if a user A follow user B, then A should be similar to B's follower and B should be similar to other followees of A. The similarity scores for random samples of A and B were taken. For similarity, the Jaccard and cosine metrics was used and this generated 'n' (5 in the final system) scores for each metric. These scores were manipulated by computing the mean, standard deviation and variance. Addition of this feature helped to increase score considerably and helped jump from a 0.7 to a 0.85 in Kaggle.

**Follows Back**

The chances of a false edge reduce when the sink node follows back the source user. This is another crucial feature that the system incorporates to predict probability of existence.

**Friends of Users**

The system identifies who many of friends of the source user follows the sink and the vice versa. The more number of friends in either direction would mean that there is greater chance of connection between the nodes.

## II. Pre-processing

The initial step of feature extraction outputs training and validation matrix based on the features discussed, over which the model learns. These matrices are then pre-processed to complete the missing values and standardised and scaled to unit variance. These matrices are then processed to generate the polynomial and interaction features among the extracted features. The polynomial feature interaction helped create a feature matrix with all polynomial combinations of the features adding to the higher predictability of the edge existence. This feature was introduced at a later stage of the system design and showed it incredibility and effectiveness.

## III. Classification

The Multi-Layer Perceptron (MLP) is the classifier model used for the edge prediction system. The initial system was based on LR model. With the increase in training data points used and feature space dimensions, the system performed better with MLP over the LR model. The system adopted the MLP classifier with hidden layers count in proportion to the features used, as it has proved to be a better and accurate approach in such cases [2]. Logistic Regression can be considered to be a single layer neural network, while the MLP classifier is a more sophisticated network system that uses logistics activation functions which can learn more complex and non-linear functions.

The optimal system was derived by various trial and error mechanisms and combination of features and techniques. This system was built on simple concepts of Python dictionaries and needed some hand-coded functions unlike the existing packages of NetworkX. Another approach using the NetworkX package was also implemented and compared before this final system was identified optimal among them. The alternative system, methodologies and reasons of selections used through the development stages of this implementation are also discussed in the following sections of the report.

## Alternative Approaches:

An alternative system was implemented using the NetworkX package. This represented the users as nodes and the relationship as edges of a graph structure. This approach identified and adopted seven features, namely - Numbers of users, the source is following, number of users that are followers of sink, if there exists any mutual friend in between of the two user nodes, if the source has more followers than following or if the sink has more followers than following, similarity metrics using forward and backward Jaccard index. A random forest model was trained using 20000 random instances of data, among which half were from prior knowledge and the rest were randomly generated edges. Though the approach predicted a good f-score of 0.8015, it was computationally intensive and required nearly ten hours of execution. The computational load of NetworkX should be the prime factor for the high accomplishment time.

On the other hand, the selected final approach was simpler and faster to work. Keeping in mind time constraint of project deadline and the feasibility to quickly improve and modify, this approach was selected to be worked and optimized even though it required hand coded function implementation.

**Error Analysis:**

The final system was deduced based on a continuous development approach. The system was introduced as a simple predictive classifier with very few features of source and sink followers and followees. The initial few developments included adding in features like similarity scores of Jaccard and cosine. Addition of multiple features helped to increase the f-score from our original of 0.49 to 0.70. These features were trained on a trifling subset of training data with 200 data points due to computation time involved and increased complexity. With a moderately scoring stable system developed, we now concentrated on increasing the interaction between the features and also increased the number of data points used. A plot representing the train and test mean scores against the fold size was used to identify and measure the impact of the training data on the learning of the model. Figure 1 is one such plot during the course of development and indicated an increasing error score in the validation set with increase in fold size when LR classification was used. This initiated the thought process to try a deep learning classification instead of a simple LR. The validation set is considered as impersonate of the test set. The graph acts as a good indicator of mean accuracy of the classification. Figure 2 clearly indicates a tremendous decrease and stabilization of the error score with increased number of data points and is the plot from the final system. With 60000 points used to train, the classification is predicted to score a low error rate and better trained.

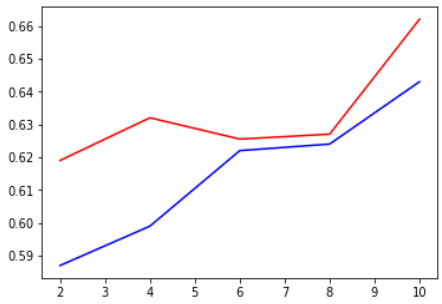


Figure 1: LR method

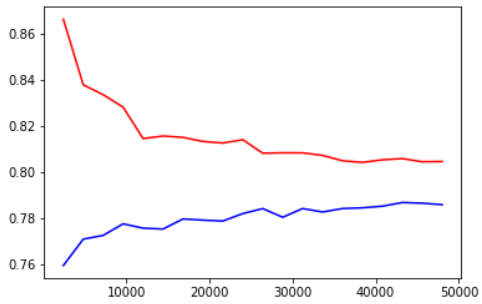


Figure 2: MLP classifier

The MLP classifier proved to be a far better approach and allowed a good approximate solution for the large dataset and supported the demanding complexity of number of features and its interactions. An important upgrade planned was to train the MLP classifer using the entire data set available, but is regarded as improvement due to time constraint.

**Conclusion:**

The final implemented system and the alternative approach has been implemented using different machine learning and predictive model techniques taught through the semester. The ability to identify and reason the elect based on various constraints involved, is an important learning from the project. The performance of the final system is quite promising and has been placed well in the competition. We have identified scope of improvements in both the designs and a possibility of higher correlation scores with the use of the entire dataset and larger number of features, however, was not implemented owing to the time constraint.

**References:**

[1] Cukierski, W., Hamner, B., & Yang, B. (2011, July). Graph-based features for supervised link prediction. In Neural Networks (IJCNN), The 2011 International Joint Conference on (pp. 1237-1244). IEEE.

[2] https://sebastianraschka.com/faq/docs/logisticregr-neuralnet.html