THE ACHIEVERS

Siddharth S Desai, Rushikesh Jadhav, Allen C George

1. PROBLEM DESCRIPTION:

The Dataset of a Twitter Network in the form of adjacency list is given. The Dataset has 20,000 rows. The first column of the data consists of the **“from”** nodes, and the subsequent columns consists of **“to”** nodes.

The Task is to learn from the given training data, and check whether the link really exist or is a false or fabricated edge.

Submission format is a csv file with 3 columns, from, to and predicted value of the link.

2. DATASET PREPARATION:

a. Dataset reading:

- The dataset was so huge, pandas data frame was unable to load the dataset. It was taking time. The old dataset took 7-8 minutes to load in pandas data frame. We were getting memory error.
- It was nearly impossible to do task on it. So, we came up with the solution of splitting whole dataset into smaller chunks and then working on each dataset separately, later combine them to get the final cleaned dataset.
- So, we created 4 different CSVs of size 5000 rows each. Created 4 different edge lists out of it in the format of from and to nodes, and later combined all 4 to get combined_data.csv, which had **12903552** rows and **2** columns.
- This dataset was easy to operate upon.

b. Fabricating False Edges:

- The combined_data had ground truth as 1 so we needed some false edges.
- We found unique nodes from both the columns. And randomly choose one node from each of them, and checked if that combination is already present in data or not.
- If it’s not present then we added that node to the dataset, creating 1,00,000 false edges.
- We randomly choose 1,00,000 true edges from combined_data file, and appended it with false edges to create new training data with 2,00,000 rows (1,00,000 true and 1,00,000 false).

c. Featurization:

The data, didn’t had any features, so we selected 3 features to work on the data, at the initial stage. The Indegree, Outdegree, and Jaccard Coefficient.

- **Indegree [1]:** For a directed Graph, indegree is the number of links approaching a particular node.
- **Outdegree [1]**: For a directed Graph, Outdegree is the number of links leaving a particular node. To calculate the Indegree and Outdegree, the networkx library was used.
- **Jaccard Coefficient [1]:** It is the measure of similarity between two nodes.

The Feature Jaccard Coefficient was calculated. Using the networkx library, gives the inbuilt functionality to calculate the Jaccard.

3. APPROACH 1:

- We took 3 features to work upon, Indegree, Outdegree and Jaccard Coefficient, along with Random Forests.

4. APPROACH 2:

- In 2nd approach, the dataset was changed, some points were added. So started working with that new dataset.
- When that data was changed to Data Frame containing 2 columns [from, to], the No. of rows came out to be **24003361.**
- Sampled 1 lac +ve and 1 lac -ve samples from the dataset.
- Here we calculated indegree and outdegree for both **from** node and **to** node, along with Jaccard coefficient. So, I had 5 features in hand, [Src_Indegree, Src_Outdegree, Des_Indegree, Des_Outdegree and Jaccard Coefficient]
- The Accuracy dropped from 0.80 to 0.70.
- The Reason behind the same can be, the Overfitting of the data because of less variance in columns of Des_Indegree and Des_Outdegree,
- I dropped both the columns.

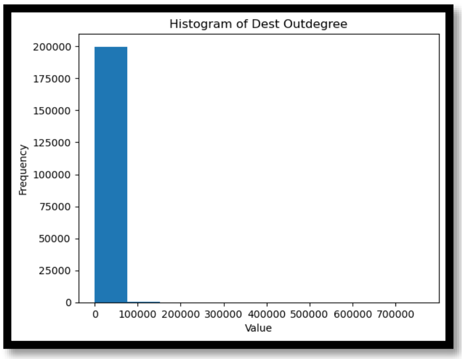

Fig 1.1 Destination Outdegree Plot

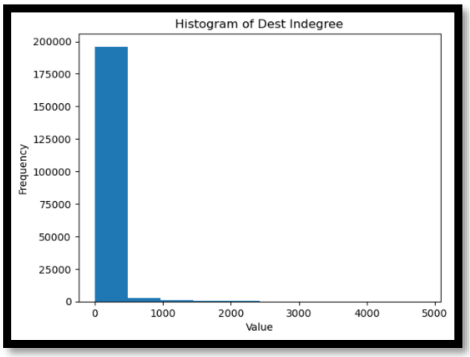

Fig 1.2 Destination Indegree Plot

It can be seen that there is no variance in this column most values are zero, similarly for Destination Indegree the distribution was maximum zeros.

5. APPROACH 3:

- Now I again had 3 features, Src_Indegree, Src_Outdegree and Jaccard.
- I added few more features, they were cosine similarity, preferential attachment score [3], using networkx library.
- Again, gave a submission but accuracy didn't make any significant difference.
- The calculation of Cosine and preferential attachment score must have gone wrong somewhere. I later figured out that, I created undirected graph of the data, rather than creating a directed graph.
- Now next I created a directed graph and calculated the features again.

6. FURTHER APPROACHES:

Calculated features using directed graph, but no significant improvement was seen in the featurization. I didn't submit any submission on Kaggle for this, because when I tested it on my end, I didn't saw any major change in it.

Then I tried adding features without using network library. The dataset was so huge which networkx may not be able to handle and plot the graph for the same. So, I used Data frame of from and to nodes to calculate few new features like friends measure, opposite direction friends and same community [3].

The max values in the feature of Friends_Measure is zero, hence it won't be able to add much weightage to the accuracy.

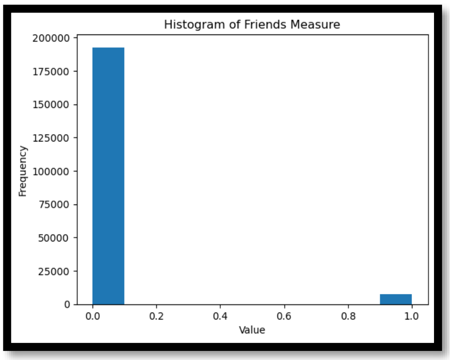

Fig 2.1 Distribution of Friends_Measure

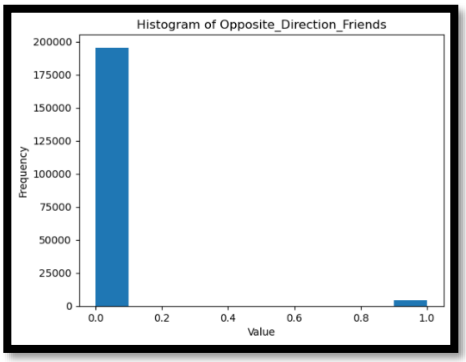

Fig 2.2 Distribution of Opposite_dirn Friends

So got accuracy between 70-75%, in all my submissions.

In opposite direction friends, the distribution didn't showed much variation, which may have overfitted my model, giving bad accuracy

7. ACCURACY:

After submitting the prediction with 3 features [Src_Indegree, Src_Outdegree, and Jaccard], the accuracy we got was 0.80854, the best until now.

MODEL USED	FEATURES	ACCURACY
Random Forest	Indegree, Outdegree, Jaccard Coeff	**0.808054**
Random Forest	Indegree, Outdegree, Jaccard Coeff, cosine similarity	**0.73724**
Random Forest	Indegree, Outdegree, Jaccard Coeff, Cosine Similarity, Preferential Attachment Score	**0.70704**
Single Decision Tree	Indegree, Outdegree, Jaccard Coeff, cosine similarity, Preferential Attachment Score	**0.70685**

8. MODELS USED:

- The first model used was Random Forests, criteria used was Gini Index, max_Depth was chosen as 14.
- The second model which we tried was with a single decision tree, but it gave bad accuracy, because it uses single decision tree, which may overfit the data, and give bad accuracy with test data.
- So, we sticked with random forests, all the submissions were done with random forests only.

BIGGEST MISTAKE AND LEARNING:

In one of my submissions, I got the accuracy of around 49%, it was because the columns of train set and test set were not in order. i.e., the order of headers of both train and test were different. So, my model calculated weights with some order which was present in train set, but the order changed in test set, which resulted in an error.

This is the mistake due to negligency.

9. CITATIONS:

[1] Fire, Michael & Tenenboim, Lena & Lesser, Ofrit & Puzis, Rami & Rokach, Lior & Elovici, Yuval. (2011). Link Prediction in Social Networks Using Computationally Efficient Topological Features. 73-80. 10.1109/PASSAT/SocialCom.2011.20.

[2] Chat GPT

[3] 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.