Report (Project Two)

Student Name: Danita Anubhuti Prakash
Student ID: s4745511
Student Email: d.prakash@uqconnect.edu.au

Introduction

The project's main goal is to design and implement an effective algorithm to predict social links. Based on the given training and validation sets of the co-author network, a program is generated to rank the unlabeled edges in the test set. For each pair of nodes in the the data given, this program computes a proximity score which is used for link prediction. The provided co-author network has 5,240 nodes, 11,696 edges. The edges of the whole co-author network are then split into three parts – training set, validation set and test set. The 10,100 pairs of nodes are ranked according to proximity score in descending order using a machine learning model and top 100 nodes are stored as output into a text file. Python language is used to generate the codes using packages like numpy, pandas, networkx and sklearn. The program implementation is discussed in more detail in the nest section.

Task

I started with importing different libraries and loading the files into different dataframes. Once the data was loaded I performed different algorithms to find the proximity scores for each pair of nodes. I selected four different similarity scores - Adamic Adar, Jaccard Similarity, Preferential Attachment and Total Common Neighbours for each pair of nodes given. For each of the mentioned scores a separate method is called, namely **adamic_adar_index**, , **calculate_jaccard_similarity**, **calculate_total_neighbors_in_common** and **calculate_total_neighbors_in_common.**

These scores were calculated by constructing an undirected graph with 5,240 nodes and 11,696 edges using **construct_graph** method. As all the pairs in the training data contain edges, new pairs with no edges were sampled using the method - **find_negative_edges**. A total of 22992 nodes were sampled which is 2 times the co-author nodes. This ensures that the model is built with no bias between the two groups. The pairs with edges were labeled as 1 while the others as 0. The similarity scores and edges are found on the training and validation sets. Next, I built a machine learning model on the training set to classify the Edges based on the similarity scores using Binary Logistic Regression. This model works well for values being strictly 0 or 1 [1]. I choose this model as it an classification method and is helpful to find the probability of an event like having an edge or not [2].

In the method **build_model**, the training data is divided into features containing the scores and target containing the Edge for each pair. Columns which contain the node names were deleted as they are not needed for model building. The the method – **evaluate_model**, is used for prediction on 3 datasets – validation positive, validation negative and the combined data of positive and negative edges from validation set. The below picture shows the accuracy of each of dataset.

After testing the model agaist the validation set, the predictions are done on the test set. In same manner, the proximity scores are decided. Through the method – **predict_top100_edges,** the probability of each pair of nodes having an edge is calculated. According to this probability, the nodes are arranged in decending order. The top100 predicted links are then stored in a text file.

From the results shown below, it is clear that the model beforms better on the negative data compared to positive data, this suggests that that the model predicts better on negative data. The accuracy for all the three models is good with positive. For the combined model, the accuracy is 98% which means that model performs well with high proportion of correct predictions. For the positive set, the model has relativly small amount of false negative and classifies the most of the positive data correctly. The negative validation set performs bteer than the positive with a total accuracy of 98%.

At the end, after the model predictions are performed on the test set, a print statement is given specifying that the file containing the top 100 edges is saved.

```
The results from combined validation set
Confusion Matrix is:
 [[9878  122]
 [   7   93]]
Accuracy is:
 0.9872277227722772

The results from positive validation set
Confusion Matrix is:
 [[ 0  0]
 [ 7 93]]
Accuracy is:
 0.93

The results from negative validation set
Confusion Matrix is:
 [[9878  122]
 [   0    0]]
Accuracy is:
 0.9878

Successfully saved the top-100 edges in file
```

Summary

The program helped to understand how future links can be decided based on different proximity scores. Different algorithms were used to get the required scores such as Jaccard similarity, Common Neighbours, Preferential attachment and Adamic Adar Similarity. With the help of these score, a machine learning model was prepare which helped to classify the data. Here, binary logistic regression is applied to build the model on training data and it's predictions are tested agaist the validation and test data. The following are the functions used:

1. **construct_graph**
2. **adamic_adar_index**
3. **calculate_total_neighbors_in_common**
4. **calculate_jaccard_similarity**
5. **calculate_preferential_attachment**
6. **find_negative_edges**
7. **build_model**
8. **evaluate _model**
9. **predict_top100_edges**
10. **main**

Here, the **main** is used to start the execution of the codes. The accuracy obtained on the validation sets helps to understand the the model behaviour. At the end, the top – 100 nodes from test set are saved into a file.

Reference

[1] Swaminathan, S. (2019, January 18). *Logistic regression - detailed overview*. Medium. https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc

[2] GeeksforGeeks. (2023, January 10). *Advantages and disadvantages of logistic regression*. GeeksforGeeks. https://www.geeksforgeeks.org/advantages-and-disadvantages-of-logistic-regression/