

## CS 5683: Big Data Analytics

### Assignment - 4: Graph Convolution Networks

**Total Points: 100 (15% towards final)**

**Due date: Nov. 17, 2023 at 11:59pm**

This assignment focusses on Graph Convolutional Networks which we provided the template code to understand the basics. We will be using this template program to develop new models, and compare models with multiple evaluation metrics.

**Datasets:** Use the same dataset we used for the class demo

#### Tasks:

1. **Task 1 - Deep GCN:**
  - a. **Task 1.1 – 2-layer GCN:** Edit the given demo program to add another layer of Graph Convolution for node representation learning. Ideally this model will have 2 hidden layers and one output layer. So, there will be 3 sets of weights and biases
  - b. **Tas 1.2 – 2-layer GCN with shared parameters:** This is similar to Task 1.1 to add two layers of Graph Convolutions. But we will share the weights and biases with both hidden layers. Meaning that, we will use same weights and biases for both layers. Edit the given demo program accordingly
2. **Task 2 – Evaluate models with Precision, Recall, and F1:** We have a function to calculate *Accuracy of the model*. Add program modules to find *Precision*, *Recall*, and *F1* in addition to the accuracy. Report all four measures for all three models (GCN, 2-layer GCN, and 2-layer GCN with shared parameters) in the results. Compare the models and give your reasoning for the best model
3. **Bonus Task – Custom neighborhood aggregation:** As you can notice, the GCN aggregates from all its neighbors, which can be costly at times. The idea for this task is to set a maximum number of random neighbors to aggregate information from. This can be done after our data pre-processing step by simply converting some entries of the adjacency matrix to zeroes. Evaluate this model's performance against all other models. **Note:** This has to be done for only 2-layer models and each layer should have different random samples for neighborhood aggregation.

**Things to submit:** Create one python notebook for each model and submit all notebooks

#### Grading Rubric:

1. Task-1.1: 35 points
2. Task-1.2: 35 points
3. Task-2: 20 points
4. Documentation: 10 points
5. Bonus task: 20 points