INFS7450 Social Media Analytics

Project 1 – Fast Computation of User Centrality Measures

Semester 1, 2025

Marks:	15 marks (15%)
Submission Due:	16 Apr 2025 16:00 (Brisbane Time)
Deliverables:	See deliverables part
How to submit:	Electronic submission via Blackboard

Goal: This project aims to implement a number of efficient algorithms to compute various centrality measures for user nodes such as PageRank and Betweenness. Students are required to finish this project individually.

Dataset: In this project, you will be working with publicly available Facebook social network data. The Facebook data has been anonymized by replacing the Facebook-internal ids for each user with a new value. The data contains 4039 nodes, and 88234 edges in total. Each line of the data represents an undirected link starting from one node to another.

The dataset is available from UQ blackboard. See /Assessment/INFS7450 Project One.

Tasks:
1. Calculate the Betweenness Centrality for nodes in the Facebook dataset. (**8 marks**)
   Overview: write code to load the Facebook social network data and construct an undirected and unweighted graph. Based on the constructed graph, you are required to write a program to calculate the betweenness centralities for the graph vertices.
   Input: The provided Facebook social network data.
   Output: The top-10 nodes with the highest betweenness centralities.

2. Calculate PageRank Centrality for nodes in the Facebook dataset. (**7 marks**)
   Overview: write code to load the Facebook social network data and construct an undirected and unweighted graph. Based on the constructed graph, you are required to write a program to calculate the PageRank (with α=0.85,β=0.15) centralities for the graph vertices.
   Input: The provided Facebook social network data.
   Output: The top-10 nodes with the highest PageRank centralities.
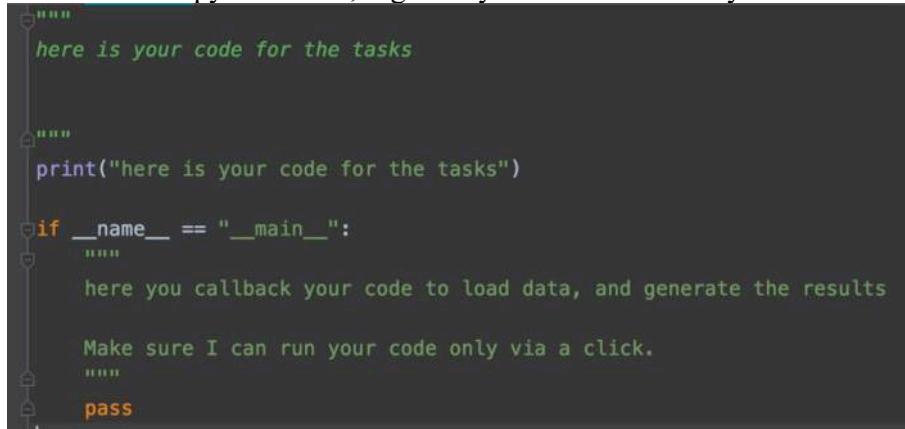
Requirements:
1. You may use third-party libraries, such as NetworkX to read, load and manipulate the Facebook network dataset. **However, you must write your own code to implement the function of node centrality calculation rather than using the third-part or built-in functions.** (You can use any functions in NetworkX other than the functions for centrality calculation.)
2. You can refer to the codes provided in the tutorial, but are not allowed to directly reuse or copy them.
3. You are not allowed to look at the code of any other student. **All submitted codes and reports will be subject to electronic plagiarism.**
4. **Failure to properly cite AI contributions is a form of academic misconduct.**

Programming Languages:

Python and NetworkX are recommended. However, you have your own choices of preferred programming languages including, but not limited to, Python, MATLAB, Java, C, C++, etc.

Deliverables (!!VERY IMPORTANT):

1. A report (.pdf). See the given appendix for an example template.
   Submit your report in the PDF format, not in the word format!
2. A source code file. For python users, organize your code in this way:

Figure 1. the example of your submitted source code

> **If you are using other programming languages, please organize your code in one file, and let me know how to run your code only via a simple click.**
>
> **Only submit the source code in one file, don't submit the data file.**

3. A result text file. The file must contain two lines of results. The first line is the top-10 nodes with the highest betweenness centralities. The second line is the top-10 nodes with the highest PageRank centralities. (each node id in each line should be separated by a space.) See the following picture as an example.

Figure 2. the example format of the results

> **Don't change the format! Your result file will be checked and marked via a script. If you change the format, your submitted results will not be accepted. Keep in mind the first line is the results of task 1 and the second line is the results of task 2. Don't mix the order.**
> **Don't use other delimiters, only use a space.**
> **Only report the top-10 node ids, don't fill in with node centrality scores!**

4. **<u>Name all the submitted files by your student ID</u>**. For example, 41234567.py for the source code, 41234567.txt for your submitted results, and 41234567.pdf for your report.
5. Submit one archive file with your student number as the file name (e.g.12345678.zip) with all the files mentioned above.

> **For example:**
> **A student (with id 12345678) can submit his project as follows:**
> **12345678.zip/**
> **-------12345678.py**
> **-------12345678.txt**
> **-------12345678.pdf**

> **Any submitted project which doesn't follow the above guidelines will be desk rejected without marking, which means you will get zero marks for the corresponding parts.**

Marking criteria (Total marks: 15):
- Task 1: 8 marks = 3 marks (code) + 3 marks (results) + 2 marks (report)
- Task 2: 7 marks = 2 marks (code) + 3 marks (results) + 2 marks (report)

- Your results should be reproducible and your codes should be readable. If your codes cannot be executed or generate the results as reported, the corresponding marks for the code and results will be deducted.
- We will evaluate your submitted results by calculating the Jaccard Similarity between the submitted results and the ground truth. That means your mark for each task will be calculated by:

Result Mark = Jaccard Similarity (Submitted Results, Ground Truth) * 3

Kaggle:
We will be deploying our projects on the Kaggle platform. Students are encouraged to upload their results to Kaggle and compete with other classmates.

You can join the Kaggle competition via the below links:
Betweenness Centrality: https://www.kaggle.com/t/7202e772475f470fb7fcd1478cf57f78
PageRank Centrality: https://www.kaggle.com/t/949a83bcef134ea5b8b82a28e9a9c862

> **Please note that the scores on this platform will not be considered in the final grading. The final score will still be determined by the results you upload to Blackboard.**