

**** Late submission attracts appropriate penalty**

Title: Do an analysis of the social graph as per direction provided by extracting the graph dataset : 20 Marks (will be scaled down to 10)

Goal: Goal of this is to detect and analyse an *implicit social network*.

Team :

- 1. It is a group activity by MAXIMUM THREE members.**
- 2. THE TEAM HAS TO BE THE SAME TEAM FOR ASSIGNMENT1, ASSIGNMENT 2 and PROJECT.**
- 3. You need to mail a python notebook clearly indicating your team members' names and SRN(s).**

Required background:

- a. Google Drive -> code_used_in_class
- b. ISA1 syllabus, assignment 01 and class material about network modeling

Instruction:

- (a) **Dataset :** You will attempt to do a social network analysis by extracting the social graph (implicit) from one of the English text available at Project Gutenberg website. The novels or texts **will be assigned to teams** so that teams get to work on multiple texts.
- (b) **Quickly read up about the novel/text** from net and **make a list of all the characters and their aliases, if required.**
- (c) **Scan through the papers (paper 1, paper 2 and paper 3)** provided and write code for deriving adjacency matrix based on co-occurrence. You may need to do some elementary text processing for this. Try out several window size for getting a better graph. Once the adjacency matrix is prepared, you can redo the "assignment 01" with some extra as mentioned below. **Come up with an undirected social graph.**
- (d) **Provide a visualization that is somewhat good :** You are advised to try out Gephi visualization in fact. <https://gephi.org/users/>
- (e) **Provide the followings in the python notebook:**
 - a. What are the number of nodes and edges ?
 - b. Do a centrality analysis (degree, closeness, betweenness, eigenvector) of all the nodes
 - c. Calculate Pearson Correlation coefficient between each pair of centrality (vectors)
 - d. Find the diameter of the graph
 - e. Find the network density
 - f. For the top 5 nodes (in terms of degree centrality), draw the ego network and list their alters
 - g. Draw a degree distribution histogram and a log-log plot
 - h. Find the average path distance
 - i. Find the average clustering coefficient
 - j. Detect community using Louvain and Clique percolation methods
 - k. Make some models (random graph, small world and preferential attachment) using some of the parameters above.

(f) **Provide the analysis in submitted python notebook trying to compare your findings with the novel plot that you are aware of :**

- Analyse centrality (contrast degree, closeness, betweenness, Eigenvector) and “centrality correlation” plots. Make comments about characters (any 4 and use your judgement)
- Make a comment about the community/components detected in the context of the plot
- Do some analysis by comparing with some of the parameters you obtained from the artificial graph models (random, small world or preferential attachment) of comparable size.
- Any other interesting observation ?

Marking Guidelines

| Marking scheme | Timeliness (5 marks) | Functionality (5 marks) | Analysis (5 marks) | Visualization (5 marks) |
|----------------|------------------------|----------------------------|---------------------|--------------------------|
| Max (5) | on time | working code without issue | Well done | Well done |
| Mid (3) | on time + 3 days | working with minor issues | Somewhat done | Somewhat done |
| Min (2) | delay more than a week | working with major issues | Very poorly done | Very poorly done |

List of texts for this assignment

| | |
|--|---|
| Iliad by Homer | https://www.gutenberg.org/ebooks/6130 |
| Oliver Twist by Charles Dickens | https://www.gutenberg.org/ebooks/730 |
| Alice’s Adventure in wonderland by Lewis Carrol | https://www.gutenberg.org/ebooks/11 |
| Sense and sensibility by Jane Austen | https://www.gutenberg.org/ebooks/161 |
| The adventure of Tom Sawyer by Mark Twain | https://www.gutenberg.org/ebooks/74 |
| War and peace by Leo Tolstoy | https://www.gutenberg.org/ebooks/2600 |
| You can also choose your own text provided it is not already very common and available in Github | |