Deree
School of Graduate and Professional Education

Module	**ITC 6001 – INTRODUCTION TO BIG DATA**		
Term	**FALL 2023**		
Assessment	**PROJECT**	**Weight**	**50%**
Duration			
Deliverables	1. *Report in TurnitIn* 2. *Code in Blackboard* 3. *Code in GitHub* 4. *An oral examination/ presentation of your work*		
Method of Submission	*TurinitIn and Blackboard*		
Deadline:	*Last Week of the course*		

The rules of academic ethics apply when taking this assessment, including the requirement that you produce work ***without improper or unauthorized assistance*** *from anyone.*

General Instructions

Your project involves a series of experiments, observations coming out of the experiments, and drawing conclusions. Essentially you will collect data (or they will be provided by the instructor), then a programming language will be used (you are encouraged to use python, if you intend to use anything else you need to inform the instuctor) along with the appropriate libraries to process the data. Tables, diagrammes, and data visualizations are essential for presenting your findings.

Deliverables: a) code in blackboard/ and in GitHub, along with instructions for running it b) a report that will present your findings, i.e. the experiments, the observations and conclusion c) an oral examination that includes a 15 minute presentation.

Team: 2- persons

Deliverables: a) code in blackboard in Python, along with instructions for running it b) a report of 3000±500 words that will present your findings, and will be submitted at Turnit-in. The report must be self-contained, that is all experiments performed and all conclusions should be reported. If you need to exceed the word limit, use an appendix. c) an oral presentation d) code in GitHub
Team size: three persons

Grading Peer Marking:

		Person being rated		
		Person-1	**Person-2**	**Person-3**
Person doing the rating	**Person-1**	1.25	1	0.75
	Person-2	1.10	1.10	0.80
	Person-3	1	1	1
Average Rate		**1.12**	**1.03**	**0.85**
Individual score (project grade: 80%)		**89.6**	**82.4**	**68**

Grading: peer-review

Teams consist of two or three persons. Group members will be asked to rate the relative contribution of themselves and the other group member(s). The ratings provided by each member must add up to the number of persons the group consists of (see example above). These ratings will be considered in the final grading of the project for each individual.

Example: In a group consisting of three members, each member provides a rating of all group members. As this is a three-member group, the ratings provided by each member add up to 3.00. A rating of 1.00 means that the person in question did exactly as much as expected of him/her. A rating that is less than

1.00 means that the person in question did less than expected, whereas a rating that is greater than 1.00 means that this person's contribution was greater than expected.

Coding:

You can use python, and related libraries, e.g. Json, csv, Pandas, numpy, a library for displaying data, and databases should they be useful. No other framework may be used.

Q1: Understanding the data-Exploration – 20%

Data description

Obtain the last.fm data set[1] , which contains social networking, tagging, and music artist listening information from a set of 2K users from Last.fm online music system.

Briefly describe the original data set. Display a frequency plot of the listening frequency of artists by users. The y-axis should contain the viewing frequency, and the x-axis the artists. Do likewise for the frequency of tags per user.

Outlier detection I

In a data set the elements that exhibit a strange behavior are considered as outliers. This must be quantified. One way is to use the z-score[2], and call outlier whatever is higher than a threshold (e.g., 3). Detect the outliers among the artists, tag, users, in terms of how many times they have been listened to (for artists), used (for tags), or (total listening time for users).

Do some research and find a second way to obtain outliers. Discuss the differences between the two measures.

Q2: Similar Users

Find similarities-25%

1. You will need to find the similarity between all of pairs users, based on the artists they have heard and the 'weight' parameter. As measure of similarity, you can use the cosine measure.
    a. Store the results in a csv file, entitled: *'user-pairs-similarity.data'*
2. Find the k-nearest neighbours for each user (based on similarity defined as above)
    a. Store the results as a JSON file, entitled: '*neighbors-k-users.data'*. *The <key would be the userID>* and the *values* would be the *ids* of the neighbors.
    b. Do a for k=3, k=10

[1] https://grouplens.org/datasets/hetrec-2011/

[2] https://medium.com/towards-data-science/a-brief-overview-of-outlier-detection-techniques-1e0b2c19e561

Q3: Dynamics of Listening and Tagging-20%

The tagging of artists by the users has a time stamp. Split the data into meaningful intervals, e.g. monthly or trimester.

Discover and display the activity per interval:

a. The number of users, tags, and artists per interval.
b. The top 5 (in terms of frequency of appearance): artists & tags per interval

Q4: Comparing prolific user detect methods-15%

a. Is there a correlation between the number of artists are listened to and the number of friends a user has?
b. Is there a correlation between the total listening time of a user and the number of friends he/she has?

Q5: Presentation 10%

A presentation to summarize: the data set that was used, any type of preprocessing, the methodology that was applied, results and conclusion. All team members need to participate in the presentation.

Q6: Report Quality 10%

The report should be self-contained. It should clearly describe the work done and should be split into meaningful sections. The use of Tables, and diagrammes enhances the readability.