College of Business, Technology, and Engineering
MSc Big Data Analytics (2020/2021)
Module 55-706555: Programming Concepts and Practice

Coursework 2 (40%)

1. Introduction

As described in the descriptor, the module will be assessed in two phases of assignments, corresponding to coursework 1 and coursework 2, in line with the module's learning outcomes. The first assignment is linked with the second assignment. This means you cannot execute the tasks in the second assignment without those in the first assignment. Both assignments are individual pieces of work, and your submission consists of implemented modules (containing classes and objects) and a mini report.

In this assignment, specifically, you will continue with the implementations in assignment 1 but this time using object oriented programming concepts and Python data science libraries such as numpy, pandas, matplotlib and scikit-learn.

This assignment assesses the module’s learning outcomes (LO) as follows:

- In the first assignment, you focused on designing and implementing python functions and modules for loading and extracting data from datasets and for computing artist similarity and music track similarity, using an appropriate data structure, string manipulation, iteration, selection, etc.(LO1) based on procedural programming concepts(LO2).
- In this second assignment, you will continue with the implementation of your similarity metrics for **collaborative/content-based recommendation systems.** You will implement all the previous modules and functions using object oriented programming concepts and Python data science libraries in the recommendation system domain (LO3).

2. Learning Outcomes (LOs)

I. Select appropriate programming techniques and data structures to develop effective software implementations of relatively complex systems using an appropriate programming language such as Python, Java or C#.

II. Apply relevant program design strategies to the implementation of software applications using that programming language.

III. Design and implement well-engineered, domain specific software using that programming language

3. Getting Started and General Specifications

The following tasks are to be performed in this assignment:

I. **Implementing modules in assignment 1 using Object Oriented Concepts**
In this task, you will **MODIFY** and **IMPLEMENT** the first ***(load_dataset_module)***, and second ***(similarity_module)*** modules of assignment 1 using OOP. This means implementing your solution of assignment 1 as classes and methods.

II. **Ranking of artists and music**
In this task, you will implement additional methods to perform the following:
1). The first method should return the first n most similar artists to the target artist.
2). The second method should return the first n most similar music to the target music.
3). The third method will return the first n most similar music to the target artist.
Note: *The target artist is the artist to be provided with a set of recommendations.*

III. You will implement a method that generates ***n*** recommendations for any target artist based on their similarity scores.

IV. You will evaluate the accuracy of your similarity metrics. Which of the metrics produces the highest recommendation accuracy? Plot the graph of the recommendation accuracies for your similarity metrics.

V. **Mini Report**

You will write a well-structured report not more than 5 pages, summarising your implementation decisions, justifications and your class diagrams expressing the relationships among objects of your application. Figures representing the architecture of your system as well as flow diagram of the functionality of your application. Also include a section with your reflection of experience implementing this application.

4. The Dataset

The dataset for this assignment is the same dataset used in the first assignment. Therefore, details of the dataset will not be repeated in this document. If you have any doubts, please check the first assignment for details of the dataset.

5. Pay attention to the following requirements

This assignment is an individual piece of work, and your submission must be in the form of modules (.py files) or Jupyter Notebook file. We should be able to open and run your modules on a standard campus computer.

a) You will submit a mini- report. The report should provide justifications for your analysis of the solution, design decisions especially on the object oriented design. It should explain the relationships between the classes and objects. A good report should be based on evidence with critical analysis of the implemented system. Even if your application does not work correctly, you should still submit the mini report explaining what you have done, what works and what has not worked.

b) Any evidence of collusion/plagiarism will be penalised if appropriate! If there is some doubt about the authenticity of a particular piece of work, then the person submitting it will be expected to defend such work, including reasons for the programming decisions taken. You must document with references any use of libraries or existing code in your mini report.

c) This assignment is linked to assignment 1. This means that assignment 2 is a continuation of assignment 1.

d) Appropriate use of variable names for clearer understanding is desirable

e) Adequate commenting of your codes for easier understanding during grading is also desirable.

f) **Note that creativity will be rewarded** for a well-implemented system that goes the extra-mile to achieving the required functionality and doing something more creative above the given specifications. Such creative, additional functionality should be justified in a separate section in your mini report, check the assessment grid for details. An example could be an implementation of a user-friendly GUI for your application.

6. Submission Process

Your assignment should be submitted electronically through the module's Blackboard site as a single ZIP file that contains all your source codes and mini report. Check your upload to ensure you have submitted the correct files successfully as any issues will not be considered after the deadline. Provide an explanation in your mini report on how to execute your application.

7. Submission Deadline

Friday, 22nd January 2021 by 2.59pm.

8. Assessment Criteria

This assignment will be assessed mainly by **code testing/inspection, and through a video demonstration** of the submitted codes using the data files accompanying this brief. **You should submit a video demo of maximum of ten minutes, demonstrating how your solution meets the assessment criteria.** The coursework will be assessed against the Learning Outcomes (LOs) using a set of assessment criteria. This set of assessment criteria allows assessing how successful you have met the LOs. In order to ensure consistent application of the relevant criteria, the assessment criteria are summarised in the following assessment matrix and grid. This is an indicator of how the marks will scale across each category of the learning outcomes it covers.

Assignment	**Assessment Criteria**	**Marks**	**Learning Outcomes**		
			LO1	LO2	LO3
Assignment 2	Definition and application of basic OO programming (classes and objects) concepts (/15)	15%	x	x	X
	Clear understanding definition and use of method, method parameters and arguments (/10)	10%	x	x	X
	Use of python libraries (/10)	10%			X
	Quality and usefulness of the submitted mini report/video demo (/5)	5%			X

9. Assessment Rubric

Fail (<50%)	Pass (50-59)	Merit (60-69)	Distinction (70% +)
Definition and application of basic OO (classes, objects, inheritance, etc) programming concepts (15%**)**			
No evidence of understanding of basic object oriented programming concepts such as classes, methods and objects. Application crashes, etc. No submission	Some evidence of understanding and application of OOP but only partial understanding of appropriate ones to develop programming solutions. Some issues such as naming conventions for objects, etc.	Very good understanding of OOP concepts, for developing working solution. Some minor issues of correctness.	Exceptional and creative exploration of OOP concepts, very clear identification of the most appropriate approach with justifications.
Clear understanding definition and use of method, method parameters and arguments . (10%**)**			
No evidence of understanding and use of parameter and argument passing. Nothing is submitted.	Evidence of clear and consistent understanding of the method definition and application. Evidence of practical solution.	Very good and appropriate definition of methods, parameters and argument passing. Very good understanding of the relationship between classes methods and attributes	Exceptional understanding and creative use of the class methods of programming solutions.
Use of python libraries (10%**)**			
No evidence of the use of Python libraries such as numpy or pandas. Not able to apply	Clear and good evidence of the use and application of python libraries with some correct and expected outputs. But some minor issues with outputs.	Very good understanding and good implementation using python libraries to implement some of the functionality of the system with justifications and correct	Excellent under tanding and implementation of relevant python libraries, such as numPy , pandas,

appropriate python libraries. No submission		outputs. Program executes and produces expected	matplotib and scikit-learn with outstanding results. excellent user interaction through GUI, etc.
Quality and usefulness of the submitted mini report & Video demo (5%)			
Unclear structure, poor report and presentation. Inaccurate information. Poor use of language. Little or incorrect referencing.	Clear, well structured, concise and accurate presentation. Sources are correctly referenced. Minor issues of language, etc.	Very good use of language and style. Clear evidence of professional practice and presentation.	Exceptional report and presentation of work done. Evidence of exceptional understanding of the developed system. Report provides very good insight into the developed systems.

****

All work must be your own*. If evidence of collusion/copying is found, then* **such collusion will be *penalised, severely if appropriate!*** *If there is some doubt about the authenticity of a particular piece of work, then the person submitting it will be expected to give a detailed explanation of such work, including reasons for the programming decisions taken.*

*** ******