The Group Project of COMP30770 Programming for Big Data

Project Title: *[No more than 20 words]*

Student ID: Student Name / Student ID: Student Name / Student ID: Student Name

Code Link: This should be a publicly (or at least inside UCD) accessible link (e.g., Github, Gitlab, Google Drive) to all your code relevant to this project. This link must be valid before the end of June 2025.

Section 1. Introduction (1 page)

- Please briefly describe your dataset in 2 or 3 sentences (1.5')
- Please justify the "volume" of your dataset. (2')
  - *[Not just in xx data entries, xx GB/TB. Present your hardware and software specs and cite the execution time of some key steps in Section 3 would be even better]*
- Please justify the "variety" of your dataset. (1')
  - [*They do not have to be structured/unstructured/semi-structured datasets; it should be fine as long as you can justify that the two (or more) datasets you use have different structures/impacts towards achieving your "value".]*

Section 2. Project Objective (0.5 page)

- Please explain the "value" of your big data project. Specifically, what is the overall objective of your big data project? (1.5')

Section 3. Traditional Solution (2.5 pages)

Normally, in practice, before we develop the big data pipeline, we quickly prototype the processing logic on the same dataset (or its smaller version) first to test its feasibility and get its performance profile. The prototype should use *no parallelism* and can be any high-level programming language such as Shell, SQL, Python, Java, C++, etc. Please decompose your overall objective into several (roughly 3 to 6) small steps.

  - o [*Note that each task should be directly translated to one or a few Shell or SQL statements or small code snippets in other programming languages*.]

- Briefly introduce each step (2')
- The key code [*not all*] of SQL or Shell or other single-threaded high-level programming language solutions should be presented here. (5')
- The execution results / execution time / memory requirements should be presented here. (5')

Section 4. MapReduce Optimisation (2 pages)

Please identify 1 or 2 most time-consuming steps in your Section 3 that can be optimised by big data programming paradigms: MapReduce. You are free to use either Hadoop MapReduce *or* Spark MapReduce (**Spark Core API**, *NOT Spark SQL or Dataframe etc.*)

- Explain why they can be optimised using MapReduce and present your expectations (e.g., reduce execution time by 2). (3')
- Present MapReduce solution (3')
- Present MapReduce results. (3')
- Explain why the results match or deviate from your expectations. (3')