ECS8052 Knowledge Engineering Module Assessment 2

Assignment Details

- The date for submission is as published on canvas.
- You will receive feedback within 10 working days of submission.
- **This assessment contributes 70% of the module mark.**
- This is an **individual assessment**. When submitting your assignment, you are agreeing to the following statement:

> *I certify that the submission is my own work, all sources are correctly attributed, and the contribution of any AI technologies is fully acknowledged.*

- Plagiarism is a serious offense. Please ensure you have read and understood the university policy on plagiarism. Lack of awareness of what is and is not acceptable will not be accepted as an excuse. Ensure that you use your own words even when referencing a source. Make sure libraries, tutorials, and code that you use are properly acknowledged. Indicate clearly where you have used AI assistants (including, but not limited to ChatGPT, Claude, Gemini, Copilot).
- Keep a copy of all submitted coursework, including all code and data files used during the submission.
- Remember to back your work up regularly using the School's Gitlab service or Microsoft OneDrive (available to you for free as a QUB student). Loss of work due to computer failure will not be accepted as a valid reason for late or non-submission.

Assignment Brief

In this assignment you will complete the second part of a project in which you will apply the techniques studied in the course to the *OREGANO Knowledge Graph*. This graph is designed to support *drug repurposing*. That is, It contains information about thousands of diseases, their effects, their genetic associations, and the drugs that can be used to treat them. OREGANO is fully described in a recent paper by Boudin et al] and also at the associated git repository. You should familiarise yourself with the main concepts described in the paper and on the website. *You are not expected to understand all of the biological details*.

Getting started

- You should continue to work in the environment that you created for Assignment 1.

Assignment Brief

This assignment continues from Assignment 1. You may freely use any code or results obtained in Assignment 1.

1. Select a `compound` that is involved in multiple `has_target` relations. By searching over the graph, identify three potential diseases that that compound might treat (relation `is_substance_that_treats`) that are not present in the graph. Clearly state the pathways (including nodes, node types, and relation types) by which the inferences have been made. Comment on your results, paying particular attention to which relations are involved. **[30 marks]**
2. Design a Bayesian network to model the joint distribution of `compound`, `protein`, `gene` and `disease`, using the relations in the graph to inform your design. **[30 marks]**
3. Select five diseases and for each, compute the three compounds which are *most likely* to be linked to each of those compounds. Please make sure to clearly explain how you are estimating the relevant distributions. Are your findings consistent with what is known about the drug (you may consult an LLM)? **[40 marks]**

If you use a procedure that requires a random seed, you must use your student number as the random seed. Your code should give the same results each time it runs.

Deliverables

You should submit **ONLY** the following files packaged in a single `zip` file:

1. A single Jupyter notebook containing all of the code needed to reproduce your results together with explanations of your methods and analysis and discussion of your results.
2. A `requirements.txt` or equivalent YAML file that can be used to recreate your environment.
3. Any files required to run your code that were not included in the downloaded data (for example, a RDF file generated as part of Assignment 1).
4. A five-minute video recording in which you briefly present your work.

There must not be any folders within the zip file. You must not include a copy of the data.

These files should be named, respectively:

1. `{fname}_{sname}_{student_id}_code.ipynb`
2. `{fname}_{sname}_{student_id}_requirements.txt`
3. `{fname}_{sname}_{student_id}_video.mp4`

where {`fname`} is replaced with your first name (e.g. `Iain`), {`sname`} is replaced with your surname (e.g. `Styles`) and {`student_id`} is replaced with your student ID number (e.g. `40123456`). The first name and surname should match exactly the name used on your Canvas account.

In accordance with university policy, assessed work submitted after the deadline will be penalised at the rate of 5% of the total marks available for each calendar day late up to a maximum of five calendar days, after which a mark of zero shall be awarded.

The Notebook

Your notebook should cover all aspects of the assignment. In particular it should:

- Contain all code used during the assignment, structured according to the six parts of the assignment presented **in order** with clear headings in markdown cells used to indicate the purpose of each block of code.
- Include detailed commentary on methods, results and findings, presented in markdown cells.

For each part, your solution should demonstrate your understanding of the task, of the methods used, and of the technical details of the solution. You should present your results using graphical visualisations and/or tables where appropriate, and provide your interpretation of the results. Your submission should contain

Your submission should:

- Demonstrate your understanding of the task and of the methods you have used to solve it.
- Explain any decisions or choices you have made, justifying these with reference to external sources if appropriate.
- Provide your interpretation of the results and finding.
- Describe any difficulties you encountered and explain how you overcame them.
- References any external sources you have used to support your work. This include code sourced from online repositories.

The word limit for this assignment is **2000 words**. This includes only the text in the markdown cells of the notebook. Explanation and commentary that is not presented in markdown cells (for example, as comments in code or text in figures) will not be marked. **You should still comment your code and annotate your figures appropriately.**

Penalties for exceeding the word count follow the University Guidance and are as follows:

Length	Penalty
+10%	no penalty
+>10% - 20%	10% penalty
+>20% - 30%	20% penalty
+>30% - 40%	30% penalty
+>40% - 50%	40% penalty
+>50%	maximum mark of 50%.

These deductions are applied to the mark for the assignment. Deductions will not lower the assignment grade below the passing threshold.

Code Requirements

You must include a `requirements.txt` file (noting the naming requirement listed above) that can be used to recreate your environment to ensure that your code can be run. You can create this from a terminal in which your project virtual environment is active using the command

```
conda list -e > requirements.txt
```

To test and evaluate your code, the assessors will create a new Python virtual environment for each submission and install the required dependencies using the command `conda create --name test-env -file requirements.txt`. Please ensure that this works before submission.

- Your code **MUST** assume that the data files will be in `../data/*.tsv` relative to the notebook to ensure that the notebook can be run easily.
- **DO NOT INCLUDE A COPY OF THE DATA FILES downloaded from Canvas.**

Submissions that do not meet these criteria and thus cannot be easily executed will be deemed to not run and will be subjct to a 20 mark deduction

The Video

You should submit a 5-minute video presentation in which you summarise the methods you have used and the main findings of your project. Your presentation should include one introductory slide that lists your name and student ID, and no more than three slides of content. The presentation does not

need to cover everything you have done; instead it should highlight the most important aspects of the problem, your approach, and your findings.

The production quality of the video is not important and it is sufficient to record a Teams session in which you share your screen.

Mark Scheme

Marks will be allocated as indicated in the assignment briefing.

For each part, marks will be allocated in the following proportions

Criterion	Marks available per
Demonstrating a sound understanding of the problem	3/10
Description and justification of your solution	2/10
Evaluation and analysis of results and findings	3/10
Quality, clarity, and efficiency of code	2/10

Submissions that include non-running code (after installing necessary dependencies from `requirements.txt` or equivalent `YAML` file) will receive a 20 mark deduction.