Assessment

Emerging Technologies, Winter 21/22

Due: last commit on or before December 19th, 2021

These are the instructions for the assessment of Emerging Technologies in Winter 2021/2022. The assessment is worth 100% of the marks for the module. Please read the *Using git for assessments* [1] document on the Moodle page which applies here. As always, you must also follow the code of student conduct and the policy on plagiarism [2].

Instructions

The purpose of this assessment is to ensure that you have achieved the learning outcomes of the module while also providing you with sample work to show prospective employers. The overall assessment is split into three components, as follows. The percentages beside each bullet point indicate the overall weighting of that item in your overall mark. However, the examiners' overall impression of your submission may override the individual weightings where deemed appropriate.

GitHub Repository

Create a GitHub repository containing two Jupyter notebooks – these are described further down. The repository should contain or demonstrate the following:

10% A clear and informative `README.md` explaining what is in the repository, how to run the notebooks, and why the repository exists.

10% A `Dockerfile` and/or a `docker-compose.yml` file that enables someone to quickly run your notebooks with minimal configuration. You should also include any other required files, such as a `requirements.txt` file, data files, and image files.

Scikit-Learn Jupyter Notebook

Include a Jupyter notebook called `scikit-learn.ipynb` that contains the following.

10% A clear and concise overview of the `scikit-learn` Python library [3].

20% Demonstrations of interesting `scikit-learn` functionality. You may choose these yourself, based on what is covered in class or otherwise. Note that the demonstrations are at your discretion – you may choose to have an overall spread of examples across the library or pick a particular part that you find interesting.

10% Appropriate plots and other visualisations to enhance your notebook for viewers.

Quantum Computing Jupyter Notebook

Include a Jupyter notebook called `quantum-deutsch.ipynb` that contains the following.

15% A clear and concise comparison of quantum computing and classical computing.

25% An explanation of Deutsch algorithm [4] and code simulating it using `qiskit` [5].

More information about marking

The following four components will be used to inform the marking of each component in your submission. It is important that your submission provides direct evidence of each of each component. For instance, your commit history should demonstrate and provide evidence that you had a pragmatic attitude to completing the assessment. Likewise, your submission should have references in it to demonstrate that you considered the literature and the work of others.

Research

Evidence of research performed on topic; submission based on referenced literature, particularly academic literature; evidence of understanding of the documentation for any software or libraries used.

Development

Environment can be set up as described; code works without tweaking and as described; code is efficient, clean, and clear; evidence of consideration of standards and conventions appropriate to code of this kind.

Consistency

Evidence of planning and project management; pragmatic attitude to work as evidenced by well-considered commit history; commits are of a reasonable size; consideration of how commit history will be perceived by others.

Documentation

Clear documentation of how to create an environment in which any code will run, how to prepare the code for running, how to run the code including setting any options or flags, and what to expect upon running the code. Concise descriptions of code in comments and README.

References

[1] I. McLoughlin, "Using git for assessments," https://github.com/ianmcloughlin/using-git-for-assessments/.

[2] GMIT, "Quality assurance framework," https://www.gmit.ie/general/quality-assurance-framework.

[3] "scikit-learn: machine learning in python — scikit-learn 0.24.2 documentation," 2021. [Online]. Available: https://scikit-learn.org/stable/index.html

[4] "Deutsch-jozsa algorithm," 2021. [Online]. Available: https://qiskit.org/textbook/ch-algorithms/deutsch-jozsa.html

[5] "Qiskit," 2021. [Online]. Available: https://qiskit.org/