Tasks 2020

Machine Learning and Statistics

Due: last commit on or before December 18$^{\text{th}}$, 2020

These are the instructions for the tasks assessment for Machine Learning and Statistics in 2020. The assessment is worth 50% of the marks for the module. Please read the *Using git for assessments* [3] 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

Four tasks will be listed here at different times during the semester. You should complete all tasks in a single jupyter notebook. This, along with relevant files like a README, should be in a single git repository synced with a hosting provider like GitHub [1]. That URL should then be submitted using the link on the Moodle page.

1. **October 5$^{\text{th}}$, 2020:** Write a Python function called `sqrt2` that prints to the screen the square root of 2 to 100 decimal places. Your code should not depend on any module from the standard library or otherwise. You should research the tass first and include references and a description of your algorithm in the notebook.

Marking scheme

The following marking scheme will be used to mark your submission out of 100%, which will then be scaled to 50%. The examiner's overall impression of the assignment may influence marks in each individual component. It is important that your submission provides direct evidence of each of the items listed in each category. 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.

25%	**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.
25%	**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.
25%	**Consistency**	Evidence of planning and project management; pragmatic attitude to work as evidenced by well-considered commit history; commits are of a reasonable size; evidence of consideration.
25%	**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, what to expect upon running the code. Concise descriptions of any code in comments and README.

References

[1] Inc. GitHub. Github.
    https://github.com/.

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

[3] Ian McLoughlin. Using git for assessments.
    https://github.com/ianmcloughlin/using-git-for-assessments/.